# Content

1. **About This Guide.** .............................................................................................................. 20  
   1.1 About the Documentation Set. ....................................................................................... 20  
   1.2 Who Should Read This Guide? ....................................................................................... 21  

2. **Getting Started.** ............................................................................................................ 22  
   2.1 What is SAP BusinessObjects Design Studio? .......................................................... 22  
   2.2 Basic Concepts. ............................................................................................................ 23  
   2.3 Launching SAP BusinessObjects Design Studio. ....................................................... 24  
   2.4 Starting with the Welcome Page. ................................................................................. 25  
   2.5 Working with the Design Tool. .................................................................................... 26  
   - Components View. ........................................................................................................ 26  
   - Outline View. ................................................................................................................ 28  
   - Properties View. ............................................................................................................ 28  
   - Error Log View. ............................................................................................................. 29  
   - Problems View. ............................................................................................................. 29  
   - Layout Editor. ................................................................................................................ 29  
   - Maintaining Settings in the Design Tool. ...................................................................... 29  
   - Storage of Applications and Images. .......................................................................... 33  

3. **Creating New Analysis Applications.** .......................................................................... 35  
   3.1 Adding Components to an Application. ...................................................................... 36  
   - Specifying the Properties of a Component. .................................................................. 37  
   - Deleting Components in an Application. ..................................................................... 37  
   - Selecting Multiple Components in the Editor. ............................................................. 38  

4. **Creating Analysis Applications for Mobile Devices.** .................................................. 39  

5. **Creating the Layout of an Analysis Application.** ........................................................ 40  
   5.1 Changing the Layout Properties and Docking Behavior. .......................................... 40  
   5.2 Using Container Components. .................................................................................... 42  

6. **Working with the Outline View.** .................................................................................. 43  
   6.1 Using the Context Menu in the Outline View. ............................................................. 44  
   6.2 Using the Context Menu (Technical Component). ...................................................... 48  

7. **Adding a Data Source.** .................................................................................................. 50  
   7.1 Selecting a Connection. ............................................................................................... 51  
   7.2 Selecting a Data Source. ............................................................................................... 53  
   - Creating a Query. ........................................................................................................... 55
About the Member Selector .....................................................56
Filtering Data in the Query Panel .............................................65
7.3 Universe Access in Design Studio .........................................70
7.4 Selecting Data from a Data Source for Charts and SDK Extensions ...........................................................................72
7.5 Deleting a Data Source ..........................................................74
7.6 Inserting a Copied Data source from SAP BusinessObjects Analysis, Edition for Microsoft Office ...........................................................................74

8 Working with the Initial View Dialog Box for Data Source Aliases ..............................................................76

9 Assigning a Data Source to a Component ................................................................................................80
10 Binding the Properties of Standard Components to Data Sources ...............................................................80
10.1 Displaying Cell Values ............................................................80
10.2 Configuring a List Box for Filtering Data .....................................82
10.3 Configuring an SDK Component to Use Multiple Data Sources ...........................................................................82

11 Saving an Application .............................................................84

12 Working with Templates ..........................................................85
12.1 Using the Ad-Hoc Analysis Template .......................................87
12.2 Using the Data Discovery and Visualization Template .................88
12.3 Using the Generic Analysis Template .........................................93
12.4 Using the KPI Dashboard and KPI Details Templates ...................95
12.5 Using the Online Composition Template ...................................96
12.6 Using the Planning Template ...................................................98

13 Executing an Application ..........................................................100

14 Executing an Application on a Mobile Device ................................101

15 Recording Applications for Offline Use ......................................102

16 Printing an Analysis Application to a Browser ................................104

17 Exporting to PDF .....................................................................106
17.1 Enabling Export to PDF in Analysis Applications.....................107
17.2 WYSIWYG Export to PDF ......................................................107
17.3 Application Export to PDF .....................................................109
17.4 Scripting WYSIWYG Export to PDF .........................................108
17.5 Scripting Application Export to PDF .........................................110

18 Exporting to SAP Lumira Desktop ..............................................112

19 Changing an Analysis Application ............................................113
19.1 Saving an Application Using a Different Name .........................113
19.2 Maximum Number of Steps Back or Resetting within an Application ...........................................................................114
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.3</td>
<td>Configuring Additional Chart Types.</td>
<td>182</td>
</tr>
<tr>
<td>26.4</td>
<td>Conditional Formatting.</td>
<td>184</td>
</tr>
<tr>
<td>27</td>
<td>Working with Geo Maps.</td>
<td>186</td>
</tr>
<tr>
<td>27.1</td>
<td>Geo Map Settings.</td>
<td>187</td>
</tr>
<tr>
<td>27.2</td>
<td>Configuring Initial Geo Map Settings.</td>
<td>187</td>
</tr>
<tr>
<td>27.3</td>
<td>Configuring Geo Map Layers.</td>
<td>188</td>
</tr>
<tr>
<td></td>
<td>Configuring a Shapes Layer.</td>
<td>189</td>
</tr>
<tr>
<td></td>
<td>Configuring a Points Layer.</td>
<td>190</td>
</tr>
<tr>
<td></td>
<td>Configuring a Charts Layer.</td>
<td>191</td>
</tr>
<tr>
<td>27.4</td>
<td>Interacting with Geo Maps.</td>
<td>192</td>
</tr>
<tr>
<td>28</td>
<td>Working with Real-Time Dashboards.</td>
<td>194</td>
</tr>
<tr>
<td>28.1</td>
<td>How to Create Real-Time Dashboards with Streaming Data Sources.</td>
<td>194</td>
</tr>
<tr>
<td>28.2</td>
<td>How to Create Real-Time Dashboards with a Single Pull-Based Data Source.</td>
<td>196</td>
</tr>
<tr>
<td>28.3</td>
<td>How to Create Real-Time Dashboards with Multiple Pull-Based Data Sources.</td>
<td>197</td>
</tr>
<tr>
<td>28.4</td>
<td>Streaming Data Source Additional Properties.</td>
<td>198</td>
</tr>
<tr>
<td>28.5</td>
<td>Time Based Line Chart.</td>
<td>200</td>
</tr>
<tr>
<td>29</td>
<td>Advanced Design Tasks.</td>
<td>203</td>
</tr>
<tr>
<td>29.1</td>
<td>Scripting for User Interaction Enablement.</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>Using the Script Editor.</td>
<td>210</td>
</tr>
<tr>
<td></td>
<td>Using the Statement Wizard.</td>
<td>214</td>
</tr>
<tr>
<td></td>
<td>Selecting Members of a Dimension.</td>
<td>215</td>
</tr>
<tr>
<td></td>
<td>Script Processing Flow in Applications.</td>
<td>217</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>29.2</td>
<td>Working with Global Scripts Objects and Global Script Functions.</td>
<td>227</td>
</tr>
<tr>
<td>29.3</td>
<td>Enabling Text Translation in Analysis Applications.</td>
<td>229</td>
</tr>
<tr>
<td></td>
<td>Using Scripts for Translatable Dynamic Texts.</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>Working With Translatable Texts in Analysis Applications.</td>
<td>231</td>
</tr>
<tr>
<td>29.4</td>
<td>Deploying SDK Extensions.</td>
<td>232</td>
</tr>
<tr>
<td></td>
<td>Installing Design Studio SDK Extensions to SAP BusinessObjects Design Studio.</td>
<td>232</td>
</tr>
<tr>
<td></td>
<td>Installing CVOM Chart Extensions for SAP BusinessObjects Design Studio.</td>
<td>233</td>
</tr>
<tr>
<td></td>
<td>Removing Extensions from SAP BusinessObjects Design Studio.</td>
<td>234</td>
</tr>
<tr>
<td>29.5</td>
<td>Creating a Generic Analysis Template for RRI Jump Targets.</td>
<td>235</td>
</tr>
<tr>
<td>29.6</td>
<td>Exporting Analysis Applications.</td>
<td>236</td>
</tr>
<tr>
<td>29.7</td>
<td>Importing Analysis Applications.</td>
<td>238</td>
</tr>
<tr>
<td>29.8</td>
<td>Exporting Applications As Templates.</td>
<td>240</td>
</tr>
<tr>
<td>29.9</td>
<td>Using the Backend Connection Component for Data Source Browsing.</td>
<td>241</td>
</tr>
<tr>
<td>29.10</td>
<td>Using Processing Groups for Parallel Query Execution.</td>
<td>244</td>
</tr>
<tr>
<td>29.11</td>
<td>Unmerging Prompts (Variables).</td>
<td>246</td>
</tr>
<tr>
<td>30</td>
<td>Troubleshooting.</td>
<td>249</td>
</tr>
<tr>
<td>30.1</td>
<td>Auto-Recovering Applications.</td>
<td>249</td>
</tr>
<tr>
<td>30.2</td>
<td>Using the Script Problems View.</td>
<td>249</td>
</tr>
<tr>
<td>30.3</td>
<td>Working with Large eval() Statements.</td>
<td>250</td>
</tr>
<tr>
<td>30.4</td>
<td>Setting Network Connections If Logon Problems Occur.</td>
<td>250</td>
</tr>
<tr>
<td>30.5</td>
<td>Activating Runtime Traces.</td>
<td>251</td>
</tr>
<tr>
<td>30.6</td>
<td>Activating SAP JCo Traces.</td>
<td>252</td>
</tr>
<tr>
<td>30.7</td>
<td>Viewing And Collecting Statistics Data At Runtime.</td>
<td>253</td>
</tr>
<tr>
<td>30.8</td>
<td>Collecting Support Information.</td>
<td>254</td>
</tr>
<tr>
<td>30.9</td>
<td>Managing Logs in the Design Tool.</td>
<td>254</td>
</tr>
<tr>
<td>30.10</td>
<td>Best Practices.</td>
<td>255</td>
</tr>
<tr>
<td></td>
<td>Increasing Performance of Analysis Applications with Multiple Data Sources.</td>
<td>255</td>
</tr>
<tr>
<td>31</td>
<td>User Interface Reference.</td>
<td>257</td>
</tr>
<tr>
<td>31.1</td>
<td>Properties of the Application.</td>
<td>257</td>
</tr>
<tr>
<td>31.2</td>
<td>Properties of the Data Source Alias.</td>
<td>263</td>
</tr>
<tr>
<td>31.3</td>
<td>General Properties for All Components.</td>
<td>265</td>
</tr>
<tr>
<td>31.4</td>
<td>Display Properties for All Components.</td>
<td>266</td>
</tr>
<tr>
<td>31.5</td>
<td>Layout Properties for All Components.</td>
<td>267</td>
</tr>
<tr>
<td>31.6</td>
<td>Analytic Components.</td>
<td>268</td>
</tr>
<tr>
<td></td>
<td>Chart.</td>
<td>268</td>
</tr>
<tr>
<td></td>
<td>Crosstab.</td>
<td>282</td>
</tr>
<tr>
<td></td>
<td>Dimension Filter.</td>
<td>291</td>
</tr>
<tr>
<td></td>
<td>Filter Line.</td>
<td>293</td>
</tr>
</tbody>
</table>
Filter Panel .................................................. 294
Geo Map ...................................................... 297
Navigation Panel ......................................... 299

31.7 Basic Components .................................. 300
Button .......................................................... 300
Chart Type Picker ........................................... 302
Checkbox ....................................................... 303
Checkbox Group .............................................. 303
Date Field ..................................................... 304
Dropdown Box ............................................... 305
Formatted Text View ................................... 306
Fragment Gallery .......................................... 308
Image .......................................................... 309
Input Field .................................................... 311
List Box ......................................................... 311
Radio Button Group ....................................... 312
Text ............................................................. 314

31.8 Container Components ............................. 315
Grid Layout .................................................. 315
Pagebook ...................................................... 315
Panel ........................................................... 317
Popup ........................................................... 319
Tabstrip ......................................................... 319
Split Cell ....................................................... 320

31.9 Custom Components ................................ 321
Timer ........................................................... 321

32 API Reference ........................................... 323
32.1 Application (Application) ......................... 330
Alert (alert) ................................................... 330
Export (export) .............................................. 330
Create Error Message (createErrorMessage) .... 333
Create Info Message (createInfoMessage) ......... 333
Create Warning Message (createWarningMessage) 334
Do Background Processing (doBackgroundProcessing) 334
Get Info (getInfo) .......................................... 334
Get Resource String (getResourceString) .......... 335
Get Tick Count (getTickCount) ....................... 335
Get User Agent (getUserAgent) ....................... 336
Load Data Sources (loadDataSources) ............. 336
Log (log) ....................................................... 337
Open New Window (openNewWindow) ............... 337
Get All Bookmarks By Folder (getAllBookmarksByFolder) ............................................. 362
Get BookmarkInfo (getBookmarkInfo) ........................................................................... 362
Get Bookmark Url (getBookmarkUrl) ............................................................................ 363
Load Bookmark (loadBookmark) .................................................................................... 363
Save Bookmark (saveBookmark) ................................................................................... 364
Share Bookmark (shareBookmark) ................................................................................ 365

32.8 Backend Connection ......................................................................................... 365

32.9 Button ........................................................................................................ 365

Get Text (getText) ....................................................................................................... 365
Get Tooltip (getTooltip) ............................................................................................. 366
Is Enabled (isEnabled) .................................................................................................. 366
On Click (onClick) ....................................................................................................... 367
Set Enabled (setEnabled) ............................................................................................. 367
Set Text (setText) ....................................................................................................... 367
Set Tooltip (setTooltip) ............................................................................................. 368

32.10 Chart ........................................................................................................... 368

Get Chart Type (getChartType) .................................................................................... 368
Clear Selection (clearSelection) ................................................................................... 369
Set Axis Scaling (setAxisScaling) ............................................................................... 369
Remove Axis Scaling (removeAxisScaling) ................................................................. 370
Get Axis Scaling Max (getAxisScalingMax) ............................................................... 370
Get Axis Scaling Min (getAxisScalingMin) ................................................................. 370
Get Legend Position (getLegendPosition) ................................................................. 371
Get Selected Member (getSelectedMember) .............................................................. 371
Get Selected Members (getSelectedMembers) ............................................................ 372
Get Style (getStyle) .................................................................................................... 372
Is Visible (isVisible) .................................................................................................. 373
On Select (onSelect) .................................................................................................. 373
Set Chart Type (setChartType) .................................................................................... 373
Set Legend Position (setLegendPosition) ................................................................. 374
Set Style .................................................................................................................... 374
Set Visible (setVisible) ............................................................................................. 375
Show Data Labels (showDataLabels) ......................................................................... 375
Show Scaling Factors (showScalingFactors) ............................................................. 376
Show Totals (showTotals) .......................................................................................... 376
Swap Axes (swapAxes) .............................................................................................. 377
Set Data Selection (setDataSelection) ....................................................................... 377

32.11 Checkbox ...................................................................................................... 378

Get Text (getText) ..................................................................................................... 378
Get Tooltip (getTooltip) ............................................................................................ 378
Is Checked (isChecked) ............................................................................................... 379
<table>
<thead>
<tr>
<th>Method</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is Enabled (isEnabled)</td>
<td>379</td>
</tr>
<tr>
<td>Is Visible (isVisible)</td>
<td>379</td>
</tr>
<tr>
<td>On Click (onClick)</td>
<td>380</td>
</tr>
<tr>
<td>Set Checked (setChecked)</td>
<td>380</td>
</tr>
<tr>
<td>Set Enabled (setEnabled)</td>
<td>381</td>
</tr>
<tr>
<td>Set Text (setText)</td>
<td>381</td>
</tr>
<tr>
<td>Set Tooltip (setTooltip)</td>
<td>382</td>
</tr>
<tr>
<td>Set Visible (setVisible)</td>
<td>382</td>
</tr>
<tr>
<td>32.12 Checkbox Group</td>
<td>383</td>
</tr>
<tr>
<td>Add Item (addItem)</td>
<td>383</td>
</tr>
<tr>
<td>Get Selected Texts (getSelectedTexts)</td>
<td>383</td>
</tr>
<tr>
<td>Get Selected Values (getSelectedValues)</td>
<td>384</td>
</tr>
<tr>
<td>Is Enabled (isEnabled)</td>
<td>384</td>
</tr>
<tr>
<td>Remove All Items (removeAllItems)</td>
<td>385</td>
</tr>
<tr>
<td>Remove Item (removeItem)</td>
<td>385</td>
</tr>
<tr>
<td>Set Enabled (setEnabled)</td>
<td>385</td>
</tr>
<tr>
<td>Set Items (setItems)</td>
<td>386</td>
</tr>
<tr>
<td>Set Selected Values (setSelectedValues)</td>
<td>387</td>
</tr>
<tr>
<td>Sort (sort)</td>
<td>387</td>
</tr>
<tr>
<td>32.13 Constants</td>
<td>388</td>
</tr>
<tr>
<td>Axis</td>
<td>388</td>
</tr>
<tr>
<td>ClickArea</td>
<td>388</td>
</tr>
<tr>
<td>DataSourceBrowseType</td>
<td>389</td>
</tr>
<tr>
<td>DataSourceConnectionType</td>
<td>389</td>
</tr>
<tr>
<td>DataSourceExportType</td>
<td>390</td>
</tr>
<tr>
<td>DataSourceType</td>
<td>390</td>
</tr>
<tr>
<td>Layout</td>
<td>390</td>
</tr>
<tr>
<td>LegendPosition</td>
<td>391</td>
</tr>
<tr>
<td>MemberDisplay</td>
<td>391</td>
</tr>
<tr>
<td>MemberPresentation</td>
<td>392</td>
</tr>
<tr>
<td>NegativeNumberDisplay</td>
<td>393</td>
</tr>
<tr>
<td>Scaling</td>
<td>393</td>
</tr>
<tr>
<td>TotalsDisplay</td>
<td>395</td>
</tr>
<tr>
<td>TotalsPosition</td>
<td>395</td>
</tr>
<tr>
<td>UnitsAndScalingFactorsDisplay</td>
<td>396</td>
</tr>
<tr>
<td>ZeroDisplay</td>
<td>396</td>
</tr>
<tr>
<td>32.14 Component</td>
<td>397</td>
</tr>
<tr>
<td>Get Bottom Margin (getBottomMargin)</td>
<td>397</td>
</tr>
<tr>
<td>Get CSS Class (getCSSClass)</td>
<td>398</td>
</tr>
<tr>
<td>Get Height (getHeight)</td>
<td>398</td>
</tr>
<tr>
<td>Get Left Margin (getLeftMargin)</td>
<td>398</td>
</tr>
</tbody>
</table>
Get Right Margin (getRightMargin) ........................................... 399
Get Top Margin (getTopMargin) ................................................ 399
Get Width (getWidth) .............................................................. 399
Hide Loading State (hideLoadingState) ......................................... 400
Is Visible (isVisible) .............................................................. 400
Set Bottom Margin (setBottomMargin) ........................................ 401
Set CSS Class (setCSSClass) ................................................... 401
Set Height (setHeight) ............................................................. 402
Set Left Margin (setLeftMargin) ................................................ 402
Set Right Margin (setRightMargin) ............................................ 403
Set Top Margin (setTopMargin) ................................................ 403
Set Visible (setVisible) ........................................................... 404
Set Width (setWidth) .............................................................. 404
Show Loading State (showLoadingState) ..................................... 405

32.15 Connection ................................................................. 405
Get Children (getChildren) ...................................................... 405
Get Connections (getConnections) .............................................. 406
Get Root Folders (getRootFolders) ............................................ 406
Get Selected Data Source (getSelectedDataSource) .................... 407
Get System (getSystem) ......................................................... 407
Search Data Sources (searchDataSources) ............................... 408
Set System (setSystem) .......................................................... 408
Show Data Source Browser (showDataSourceBrowser) ............... 409

32.16 Context Menu (ContextMenu) ............................................ 409
Get Click Area (getClickArea) ................................................ 409
Get Context (getContext) .......................................................... 410

32.17 Convert ................................................................. 410
Float to String (floatToString) .................................................. 410
Float to String Using Locale (floatToStringUsingLocale) ............. 411
IndexOf (indexOf) ................................................................. 412
Replace All (replaceAll) .......................................................... 413
StringLength (stringLength) .................................................... 413
String to Float (stringToFloat) ................................................. 414
String to Float Using Locale (stringToFloatUsingLocale) ............ 415
String to Int (stringToInt) ...................................................... 415
Substring (substring) ............................................................. 416
URL Encode (urlEncode) ......................................................... 417

32.18 Crosstab ............................................................... 417
Get Selected Member (getSelectedMember) ................................ 417
Get Selected Members (getSelectedMembers) ............................ 418
Get Selection (getSelection) .................................................... 418
Is Column Resizing Enabled (isColumnResizingEnabled). ........................................... 419
Is Conditional Formatting Visible (isConditionalFormattingVisible). ......................... 419
Is Hierarchy Navigation Enabled (isHierarchyNavigationEnabled). .......................... 420
Is Sorting Enabled (isSortingEnabled). ................................................................. 420
On Select (onSelect). ......................................................................................... 420
Remove Selection (removeSelection). .............................................................. 421
Reset All Column Widths (resetAllColumnWidths). ................................................. 421
Reset Column Width (resetColumnWidth). ....................................................... 422
Set Column Resizing Enabled (setColumnResizingEnabled). ................................. 422
Set Column Width (setColumnWidth). ........................................................................ 423
Set Conditional Formatting Visible (setConditionalFormattingVisible). .................... 423
Set Default Column Width (setDefaultColumnWidth). ............................................ 424
Set Sorting Enabled (setSortingEnabled). ............................................................ 425
Set Units and Scaling Factors Display (setUnitsAndScalingFactorsDisplay). ............ 425

32.19 DataBoundComponent. .................................................................................. 426
   Get Data Source (getDataSource). ................................................................. 426
   Set Data Source (setDataSource). ............................................................. 426

32.20 Data Cell (dataCell). ....................................................................................... 427

32.21 Data Source Alias (DataSourceAlias). .......................................................... 428
   Activate Hierarchy (activateHierarchy). ......................................................... 428
   Assign Data Source (assignDataSource). ...................................................... 428
   Assign Hierarchy (assignHierarchy). .......................................................... 429
   Clear All Filters (clearAllFilters). ............................................................... 429
   Clear Filter (clearFilter). ............................................................................ 430
   Collapse Node (collapseNode). ................................................................. 430
   Configure Input Readiness (configureInputReadiness). .................................. 431
   Copy Filters (copyFilters). ........................................................................... 431
   Deactivate Hierarchy (deactivateHierarchy). ................................................. 432
   Expand Node (expandNode). ........................................................................ 432
   Export (export). ........................................................................................... 433
   Get Assigned Hierarchy (getAssignedHierarchy). ......................................... 434
   Get Conditional Formats (getConditionalFormats). ....................................... 434
   Get Conditional Format Name (getConditionalFormatName). ......................... 435
   Get Conditional Format Value (getConditionalFormatValue). ......................... 435
   Get Conditional Format Value Ext (getConditionalFormatValueExt). ................. 436
   Get Data (getData). ....................................................................................... 436
   Get Data as String (getDataAsString). .......................................................... 437
   Get Data as String Ext (getDataAsStringExt). .............................................. 439
   Get Decimal Places (getDecimalPlaces). ....................................................... 440
   Get Dimension Text (getDimensionText). ....................................................... 440
Set Filter (setFilter) .................................................. 461
Set Filter Ext (setFilterExt) ........................................... 462
Set Measure Filter Active (setMeasureFilterActive) ............. 463
Set Member Display (setMemberDisplay) .......................... 464
Set Negative Number Display (setNegativeNumberDisplay) ... 464
Set Scaling Factor (setScalingFactor) ............................... 465
Set Totals Display (setTotalsDisplay) .............................. 465
Set Totals Position (setTotalsPosition) ............................. 466
Set Variable Value (setVariableValue) .............................. 466
Set Variable Value Ext (setVariableValueExt) ...................... 468
Set Zero Display (setZeroDisplay) .................................. 468
Sort by Attribute (sortByAttribute) ................................. 469
Sort by Hierarchy (sortByHierarchy) ................................. 469
Sort by Measure (sortByMeasure) .................................... 470
Sort By Member (sortByMember) ...................................... 470
Swap Dimensions (swapDimensions) ................................. 471
Unassign Hierarchy (unassignHierarchy) .................. 472

32.22 Data Source Info (dataSourceInfo) .......................... 472
32.23 DataSourceConnection ........................................... 474
32.24 DataSourceDescriptor ............................................ 475
32.25 DataSourceName .................................................. 475
32.26 Date Field ............................................................ 476
   Get Date (getDate) .................................................. 476
   Is Enabled (isEnabled) ............................................. 476
   On Select (onSelect) .............................................. 476
   Set Date (setDate) .................................................. 477
   Set Enabled (setEnabled) .......................................... 477
32.27 Dimension ........................................................... 478
32.28 Dimension Filter .................................................... 479
   Cancel (cancel) ..................................................... 479
   Get Dimension Key (getDimensionKey) ......................... 479
   Get Dimension Name (getDimensionName) ...................... 479
   Set Dimension (setDimension) .................................. 480
   Show Filter Dialog (showFilterDialog) ......................... 480
   Submit (submit) .................................................... 481
32.29 Dropdown Box, List Box, Radio Button Group ................. 481
   Add Item (addItem) ................................................. 481
   Get Selected Text (getSelectedText) ........................... 482
   Get Selected Texts (getSelectedTexts) ......................... 482
   Get Selected Value (getSelectedValue) ......................... 483
   Get Selected Values (getSelectedValues) ....................... 483
Dropdown Box: Get Tooltip (getTooltip). ................................................. 484
Is Enabled (isEnabled). .............................................................................. 484
On select (onSelect). ................................................................................. 484
Remove All Items (removeAllItems). .................................................... 485
Remove Item (removeItem). ...................................................................... 485
Set Enabled (setEnabled). .......................................................................... 486
Set Items (setItems). ................................................................................. 486
Set Selected Value (setSelectedValue). .................................................. 487
Set Selected Values (setSelectedValues). ................................................ 487
Dropdown Box: Set Tooltip (setTooltip). ................................................. 488
Sort (sort). ............................................................................................... 488

32.30 Fragment Gallery ............................................................................... 489
Add Item (addItem). .................................................................................. 489
Add Items (addItems). .............................................................................. 489

32.31 Filter Panel .......................................................................................... 490
Cancel (cancel). ....................................................................................... 490
Set Dimensions (setDimensions). ........................................................... 490
Submit (submit). ...................................................................................... 491

32.32 Formatted Text View ........................................................................ 491
Get HTML Text (getHTMLText). .............................................................. 491
Set HTML Text (setHTMLText). .............................................................. 492

32.33 FragmentBookmarkInfo .................................................................... 492

32.34 Geo Map .............................................................................................. 492
Center Map (centerMap). .......................................................................... 492
Get Copyright Text (getCopyrightText). .................................................. 493
Get Selected Layer (getSelectedLayer). .................................................. 493
Is Layer Visible (isLayerVisible). ............................................................. 494
Set Copyright Text (setCopyrightText). ................................................... 494
Set Layer Visible (setLayerVisible). ....................................................... 494
Set Map Url (setMapUrl). ........................................................................ 495
Get Selected Member (getSelectedMember). ........................................... 495

32.35 Hierarchy ............................................................................................. 496

32.36 Image ................................................................................................... 496
Get Click Image (getClickImage). ............................................................. 496
Get Hover Image (getHoverImage). ........................................................ 497
Get Image (getImage). .............................................................................. 497
Get Opacity (getOpacity). ....................................................................... 498
Get Tooltip (getTooltip). ........................................................................ 498
On Click (onClick). ................................................................................... 498
Set Click Image (setClickImage). ........................................................... 499
Set Hover Image (setHoverImage). ........................................................ 499
32.37 Input Field.

- Set Image (setImage).
- Set Opacity (setOpacity).
- Set Tooltip (setTooltip).
- Get Value (getValue).
- Get Tooltip (getTooltip).
- IsEditable (isEditable).
- IsEnabled (isEnabled).
- Set Editable (setEditable).
- Set Enabled (setEnabled).
- Set Tooltip (setTooltip).
- SetValue (setValue).

32.38 KeyValueCollection.

32.39 JSON.

- ForEach (forEach).

32.40 Math.

- Abs (abs).
- Acos (acos).
- Asin (asin).
- Atan (atan).
- Atan2 (atan2).
- Ceil (ceil).
- Cos (cos).
- Exp (exp).
- Floor (floor).
- Log (log).
- Max (max).
- Min (min).
- Pow (pow).
- Random (random).
- Round (round).
- Sin (sin).
- Sqrt (sqrt).
- Tan (tan).

32.41 Member.

32.42 Navigation Panel.

- SetDimensions (setDimensions).

32.43 Pagebook.

- GetPageCount (getPageCount).
- GetSelectedPage (getSelectedPage).
- GetSelectedPageIndex (getSelectedPageIndex).
Get Selected Page by Name (getSelectedPageByName) ................................................. 519
On Select (onSelect) ...................................................................................................... 519
Set Selected Page by Name (setSelectedPageByName) ............................................. 520
Set Selected Page Index (setSelectedPageIndex) ..................................................... 520

32.44 Panel. ..................................................................................................................... 521
On Click (onClick) ......................................................................................................... 521

32.45 PDF. ....................................................................................................................... 521
Export Application (exportApplication) ................................................................. 521
Export Panel Screen (exportPanelScreen) ............................................................. 522
Export Application Screen (exportApplicationScreen) ........................................ 522

32.46 Planning. ............................................................................................................... 523
Client Reset (clientReset) .......................................................................................... 523
Has Client Changes (hasClientChanges) ................................................................. 523
Has Unsaved Changes (hasUnsavedChanges) ......................................................... 524
Recalculate (recalculate) .......................................................................................... 524
Reset (reset) .............................................................................................................. 524
Save (save) ................................................................................................................ 525

32.47 PlanningFunction. .............................................................................................. 525
Clear All Filters (clearAllFilters) ............................................................................... 525
Clear Filter (clearFilter) ............................................................................................. 526
Copy Filters (copyFilters) .......................................................................................... 526
Execute (execute) ....................................................................................................... 527
Get Dimension Text (getDimensionText) ................................................................. 527
Get Dimensions (getDimensions) .............................................................................. 528
Get Filter Ext (getFilterExt) ...................................................................................... 528
Get Filter Text (getFilterText) ................................................................................... 528
Get Member List (getMemberList) ............................................................................ 529
Set Filter (setFilter) ................................................................................................... 530
Set Filter Ext (setFilterExt) ........................................................................................ 530

32.48 PlanningObjectWithVariables. ........................................................................... 531
Copy Variable Value (copyVariableValue) ............................................................... 531
Get Variable Value Ext (getVariableValueExt) ...................................................... 532
Set Variable Value Text (setVariableValueText) .................................................... 532
Set Variable Value (setVariableValue) ................................................................. 533
Set Variable Value Ext (setVariableValueExt) ....................................................... 534
Set Variable Value Range (setVariableValueRange) ........................................... 535

32.49 PlanningSequence. ............................................................................................. 535
Execute (execute) ....................................................................................................... 536

32.50 Popup. ................................................................................................................... 536
Hide (hide) .................................................................................................................... 536
Is Showing (isShowing) .............................................................................................. 536
32.51 SdkDataSource. .............................................................. 537
32.52 SearchExpression. ...................................................... 538
32.53 SingelMemberFilter. ..................................................... 538
32.54 Splitcell Container. ...................................................... 538
   Is Display Mode (isDisplayMode). .................................... 538
   On Delete (onDelete). ................................................... 538
   On Drop (onDrop). ....................................................... 539
   Set Display Mode (setDisplayMode). ................................ 539
32.55 State. ........................................................................ 540
   Back One Step (backOneStep). ........................................ 540
   Back To Start (backToStart). ........................................... 540
   Is Back To Start Available (isBackToStartAvailable). ....... 541
   Is Back One Step Available (isBackOneStepAvailable). .... 541
   Set Personalization (setPersonalization). ......................... 542
   Delete Personalization (deletePersonalization). ............... 542
32.56 String. ....................................................................... 543
   Index Of (indexOf). .......................................................... 543
   Length (length). ............................................................ 543
   Split (split). .................................................................. 543
   Substring (substring). ..................................................... 544
32.57 Tab. ............................................................................ 545
   Get Text (getText). .......................................................... 545
   Is Enabled (isEnabled). .................................................... 545
   Set Enabled (setEnabled). ................................................ 545
   Set Text (setText). .......................................................... 546
32.58 Tabstrip. ................................................................. 546
   Get Selected Tab (getSelectedTabIndex). ......................... 546
   Get Selected Tab Index (getSelectedTabIndex). ............... 547
   Get Tab. ....................................................................... 547
   On Select (onSelect). ..................................................... 548
   Set Selected Tab (setSelectedTab). ................................ 548
   Set Selected Tab Index. ................................................... 548
32.59 Text. ........................................................................... 549
   Get Text (getText). .......................................................... 549
   Get Tooltip (getTooltip). ................................................ 549
   Set Text (setText). .......................................................... 550
   Set Tooltip (setTooltip). ................................................ 550
32.60 Variable. ................................................................. 551

33 Working with the Local Mode of the Design Tool. ..................... 552
33.1 Launching SAP BusinessObjects Design Studio. ...................... 552
33.2 Creating New Analysis Applications .................................................. 553
33.3 Maintaining Settings in the Design Tool in Local Mode. ..................... 554
33.4 Storage of Applications and Images. .................................................. 558
33.5 Selecting a Connection. ................................................................. 559
   Defining Connections to BI Backend Systems. .................................. 560
33.6 Saving an Application Using a Different Name. ............................... 561
33.7 Executing an Application. ............................................................. 562
33.8 Executing an Application on a Mobile Device. ................................. 562
33.9 Collecting Support Information in Local Mode. ............................... 563
33.10 Coordinating the Translation of Translatable Texts in Local Mode. ....... 564
33.11 Configuring the Report-Report Interface for Analysis Applications in Local Mode. .......................................................... 565

34 Terminology Essentials. .................................................................... 567
1 About This Guide

1.1 About the Documentation Set

The documentation set for SAP BusinessObjects Design Studio comprises the following guides and online help products:

Tip

The guides and tutorials are regularly updated and enhanced. Make sure that you have the latest version by checking the SAP Help Portal on a regular basis.

Administrator Guide

The Administrator Guide contains detailed information, which users need in order to install, configure and manage SAP BusinessObjects Design Studio. The guide is available on the SAP Help Portal at http://help.sap.com/boad.

Application Designer Guide

The Application Designer Guide contains the conceptual information, procedures and reference material, which application designers need in order to create analysis applications. The guide is available on the SAP Help Portal at http://help.sap.com/boad.

End User Guide

The End User Guide contains procedures and background information for users working with analysis applications. This guide is available on the SAP Help Portal at http://help.sap.com/boad.

Developer Guide

The Developer Guide: Design Studio SDK contains procedures, reference material and background information, which enables developers to create 3rd-party components with the Design Studio SDK and enhance the analysis applications with custom components. The guide is available on the SAP Help Portal at http://help.sap.com/boad.
What's New Guide

The What's New Guide lists new and modified features in SAP BusinessObjects Design Studio that have been implemented since the previous release. The guide is available on the SAP Help Portal at http://help.sap.com/boad.

Online Help

The Online Help contains the same information as the Application Designer Guide and the Developer Guide: Design Studio SDK, plus an additional chart properties guide called Charts: Additional Information. Choose Help Help Contents to open the guides in the design tool. You can also set the online help as a view within the design tool. Choose View Help.

1.2 Who Should Read This Guide?

This guide is intended for application designers and users interested in building analysis applications using SAP BusinessObjects Analysis Design Studio.
2 Getting Started

2.1 What is SAP BusinessObjects Design Studio?

SAP BusinessObjects Design Studio enables application designers to create analysis applications and dashboards - based on BW, SAP HANA and universe data sources - for browsers and mobile devices (iPads, for example). It is the product of choice when full support for BW and HANA data models and engine capabilities is required. The product offers a design tool that allows you to create applications easily and intuitively without the need for native HTML and iPad UI programming skills.

SAP BusinessObjects Design Studio can be used locally and integrated in the following platforms:

- SAP BusinessObjects Business Intelligence (BI platform)
- SAP NetWeaver
- SAP HANA

Note

In local mode, you can create demo analysis applications for evaluating SAP BusinessObjects Design Studio with users at customer locations.

Besides analysis applications, application designers can also create planning applications that support both manual and automated data entry and changes to data. The application user can enter the planning data manually in the crosstab (in cells or rows) or use planning functions and planning sequences (also known as planning objects) in SAP NetWeaver BW Integrated Planning to enter data automatically. For planning data, you have to use a BW backend system as the planning system.

A Real-Time package, also available with Design Studio, allows application designers to create visualizations with streaming data (push based), and to create visualizations, which have a near Real-Time connection to SAP HANA or SAP BW (pull based). Real-Time dashboards apply in the area of operational systems, where data may only be relevant as it occurs and where it may not be necessary to persist the data - for example, utility companies monitoring smart buildings and the financial capital markets.

In addition to the standard palette of components in Design Studio, which are used to visualize data and enable user interaction, you can develop 3rd party components with the Design Studio SDK and enhance your analysis applications with custom components. Besides SDK components that visualize data from a data source, you can also create SDK components that act as data sources for SDK components themselves. This enables SDK components to access a broad range of data sources such as local files, Web services, or new types of backend system. You can store and provide access to the analysis applications containing the 3rd party components on any of the supported platforms.

Related Information

Working with the Local Mode of the Design Tool [page 552]
Creating Planning Applications [page 120]
2.2 Basic Concepts

Entities

The design tool of SAP BusinessObjects Design Studio enables application designers to create and edit applications. These applications are saved in an XML file format. Applications generally consist of user interface components (such as charts, crosstabs, buttons) and data source aliases.

Data source and data source alias

A data source alias represents an instance of a data source (for example, a BW query, or a SAP HANA data source) at runtime of the application. A single application can contain multiple instances of one data source. Every instance, for example, can apply its own filters on the same data source and thus represents its own subset of data. You can see the list of components and data source aliases used in an application in the outline view of the design tool.

Data binding

To visualize data from a data source alias in a crosstab or chart, define a data binding in the design tool for these components. The data binding is simply a reference to the data source alias which provides the data. When the data of a data source alias changes (for example, a filter is applied at runtime), the system automatically updates all components that have a data binding to that data source alias. You can also see the data binding relationships between components and data source aliases in the outline view.

User interaction

Typically you create applications that not only visualize data, but which also provide possibilities for the application user to interact with the data; for example, changing filters, selecting drill-down dimensions or branching into a details view for a selected set of aggregated data. To enable user interaction with the data, add the relevant UI components to the application. For example, you could provide a row of buttons above a crosstab and chart component to filter the visualized data by different criteria.

Eventing and script API

UI components provide a set of specific events that the application user can execute on the component. For example, every button provides an on click event. You can see a list of all available events of the component in the properties view of the design tool. Here, you can open the script editor and specify which actions should take place when the application user triggers the event. The script language is a subset of JavaScript and allows a
sequence of script API method calls to be defined. The script API provides access at runtime to the application itself as well as to its components and data source aliases. Therefore, the application designer has flexible control of the application behavior by using event scripts.

Setting properties in design time and run time

Besides the events, components also offer a variety of other properties in the properties view of the design tool. Some properties are common for all or almost all components: the component name identifies the component within the application, for example, and the layout properties define the size and position of the component within the application. Other properties are specific and depend on the component type. In all cases, the property values displayed in the properties view define the initial state of the components at application startup time. At application runtime, you can enable modification of properties by executing event scripts, as almost all component properties are accessible through the script API.

2.3 Launching SAP BusinessObjects Design Studio

Procedure

1. To launch SAP BusinessObjects Analysis Design Studio, choose Start > All Programs > SAP Business Intelligence > SAP BusinessObjects Design Studio > Design Studio.

2. Depending on the mode you log on (BI Platform, SAP NetWeaver, SAP HANA or local), perform the following steps.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| BI Platform     | 1. In the Logon to SAP BusinessObjects BI Platform dialog box, type in your user name for the BI Platform.  
  2. Type in your password for the BI Platform.  
  3. When you launch the design studio for the first time, type in the host and port into the Web Service URL. For further information, contact your system administration.  
  4. Select the authentication.  
  5. Click OK.                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Note            | Once you have entered the host and port in the Web Service URL, this information is stored by the system and you do not have to type it again.                                                                                                                                                                                                                                                                                                                                                                                      |
| SAP NetWeaver   | 1. In the Logon to SAP NetWeaver dialog box. When you launch the design studio for the first time, select the BW system from the dropdown box.                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Note            | You can only select a system from the dropdown box, if you have added the system in your SAP Logon dialog box.                                                                                                                                                                                                                                                                                                                                                                              |
|                 | 2. Type in the client for the system.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
### Option Description

3. Type in your user for the system.
4. Type in your password for the system.
5. Type in the language you prefer.
6. Click **OK**.

**Note**

Once you have entered the system and and, this information is stored by the system and you do not have to type it again.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| **SAP HANA** | 1. In the *Logon to SAP HANA* dialog box, type in your user name for the SAP HANA system.  
2. Type in your password for the SAP HANA system.  
3. When you launch the design studio for the first time, type in the XSEngine URL which consist of  
   `http://<server>:<port>` . This URL describes the connection to SAP HANA.  
   For further information, contact your system administration. |
| **Local** | The design studio is launched immediately, and you do not have to enter any user credentials. |

### Results

The design tool is launched with the welcome page.

**Note**

If you want to switch form a platform mode to the local mode (for example, from the BI platform to the local mode) during logon, click **Skip** on the logon dialog box of the BI platform.

### Related Information

[Starting with the Welcome Page](#)

### 2.4 Starting with the Welcome Page

When you start SAP BusinessObjects Design Studio for the very first time, the welcome page is displayed. The welcome page consists of different sections described below:

- **Getting Started**: Enables you to learn how to use the design tool of SAP BusinessObjects Design Studio by providing access to tutorials. These are videos and How To documents that guide you through essential and basic tasks and concepts of the solution and help you to familiarize yourself with the design tool and its features.
- **Create New**: Enables you to create a new analysis application. You can select different templates that are optimized for desktop Web browser applications or mobile applications. You can choose between blank...
templates or predefined templates which correspond to various design and business needs. If you choose a predefined template, the system automatically creates a copy of this template. You can change your copy according to your needs.

- Recently Used Analysis Applications: Enables you to open recently-used applications.
- Useful Links: Enables you to find further useful information about the solution, for example, SAP Community Network and SAP Help Portal.

**Note**
When you want the welcome page to be displayed every time the design tool is started, select the *Always show the Welcome page on startup* checkbox.

On the right bottom corner of the welcome page you will find buttons for social media. Clicking a button launches the Web browser and opens the relevant social media website.

**Related Information**

Creating New Analysis Applications [page 553]

### 2.5 Working with the Design Tool

Before you start creating applications, take a moment to familiarize yourself with the interface. The design tool includes a layout editor, the Components, Outline and Properties views, a menu and a toolbar. You can show or hide the views or move them to other screen positions.

**Related Information**

Components View [page 26]
Outline View [page 28]
Properties View [page 28]
Layout Editor [page 29]
Managing Logs in the Design Tool [page 254]
Maintaining Settings in the Design Tool [page 29]
Storage of Applications and Images [page 558]
Activating Runtime Traces [page 251]

### 2.5.1 Components View

The Components view contains all components that you can use for creating analysis applications. Components are user interface elements that you can drag and drop into the layout editor and thus create the content of the
application. Once you have inserted a component in your application, you can change its layout and behavior by editing its properties in the Properties view. Most of the components offer a set of specific events that the application user can execute on the component. As components have different functions, they are grouped in different folders:

**Analytic components**

You use analytic components like crosstabs and charts to visualize your data. After you have dragged and dropped a crosstab or a chart into the layout editor, the component initially displays dummy data. As soon as you assign a data source to the crosstab or chart, it displays the data of the data source. This group contains the following components:

- chart
- crosstab
  The crosstab displays multi-dimensional data in a table with analytic functions.

**Basic components**

The Basic Components folder contains a variety of components with different functions. Components like Dropdown Box, Radio Button and Checkbox are used to select or filter data and thus enable user interaction at runtime. Components like Button and Image are used to enable navigation, whereas components like Text and Date Field are used to enhance the design and layout of the application. This group contains the following components:

- button
- checkbox
- date field
- dropdown box
- image
- input field
- list box
- radio button group
- text

**Container components**

Container components are used to group and structure the content of an application. They can also be used to optimize an application for a mobile device and thus enable specific motion gestures in mobile applications. This group contains the following components:

- grid layout
- pagebook
- tabstrip
2.5.2 Outline View

The Outline view gives a hierarchical overview of the currently active application (the application whose editor is in focus). The overview contains all data sources and all UI components.

There are several ways of using drag and drop in the outline view:

- You can drag and drop components from the components view onto another container UI element in the outline view.

  ➤ Remember
  The Layout node also serves as a container.

  If a container component (tabstrip, pagebook, grid layout) only supports certain child types, drag and drop might not be allowed. For example, you cannot drop anything onto a grid layout node or a tabstrip node. Instead you need to drop onto grid cells and tabs.

- You can move/drag and drop components within the outline view (within the same container to change the order of UI components, or between different containers). For example, drag a button from a grid cell to a tab on a tabstrip. You can even drag a tab from one tabstrip to another tabstrip. Grid cells, however, are treated in a special way and cannot be dragged at all.

- You can assign data source aliases to data-binding sensitive components like crosstabs or charts by using drag and drop.

You can use various functions in the context menu of the respective elements (components or data sources).

Related Information

Working with the Outline View [page 43]
Using the Context Menu in the Outline View [page 44]

2.5.3 Properties View

The Properties view is the standard Eclipse properties view. It displays the properties of the currently selected object (component or data source), which can be a tree node in the outline view or a component in the editor.

➤ Note
Some components do not expose properties, for example the cells of a grid layout.

When you click on the Value column of a property, a cell editor specific to the property is displayed. The cell editor might be a text input field, a dropdown box, or contain a button that opens a dialog.

If you select multiple components (currently only possible in the outline view), a property sheet is displayed. Only those properties that are common to all selected components are displayed. These properties can now be edited for all selected components simultaneously.
Restriction
Only simple properties support this multi-editing function.

2.5.4 Error Log View

The Error Log view can be used to indicate general errors like system problems and messages that will be displayed when using some special methods for script validation.

Related Information

Error Analysis in Scripting [page 212]

2.5.5 Problems View

The Problems view can be used to display errors in scripting during design time and after saving the application. To resolve and examine the problems double-click the errors listed in the view.

2.5.6 Layout Editor

For each application, an editor is opened in the editor area. The editor contains an SWT browser control. Under Windows, Internet Explorer is embedded into this SWT browser control. With other operating systems, a browser such as Mozilla or Safari is embedded into the SWT browser control.

There is a two-way interaction between the browser control and the various parts of the design tool:

- Dragging and dropping from the components view to the editor area: This creates new components.
- Dragging and dropping a data source alias from the outline view onto a data-binding aware component in the editor: This assigns the data source alias to the target component.
- Selecting a component in the outline view: This updates the selection in the editor and the reverse is also true.
- Moving, resizing or deleting components in the editor: This updates the outline and properties view.

2.5.7 Maintaining Settings in the Design Tool

You can define settings for the design tool in the Preferences dialog box. To start the Preferences dialog box, choose Tools > Preferences > Application Design.
Change the default values as required and choose Apply to make the entered values effective. If you want to reset the default values, choose Restore Defaults. To close the Preferences dialog box, choose OK.

The following settings are available:

**General**

You can switch the startup mode. In the default setting, the design tool starts in local mode after installation. If you want to use SAP BusinessObjects Design Studio integrated into one of the supported platforms, select the required platform:

- **SAP BusinessObjects BI Platform**
- **SAP HANA**
- **SAP NetWeaver**

After restarting the design tool, you need to log on to the selected platform.

*Note*

In the Logon to <selected platform> dialog box, you can still decide to log on locally by clicking Skip, even if the startup mode is set to one of the supported platforms. The design tool starts in local mode.

In the Undo History Size field, you can change the default value 50 and enter a number between 20 and 999. This value defines how often application designers can undo their changes when working in the design tool.

**Embedded Web Server**

You can define the network port for the embedded Web server of the design tool. Enter a number between 1024 and 65535. When an application is executed, the port number can be seen in the URL. If the port is set to 5555 for example, the URL is http://localhost:5555/aad/web.do?APPLICATION=MYAPP. If you leave the default value 0, the system automatically assigns a network port.

*Tip*

If users create bookmarks for their applications, the network port has to be set to a fixed number. Auto-assigning the network port does not guarantee that the bookmarks will work properly.

**Application Preview**

The language of message texts and tooltips is defined by the language setting in the BI launch pad or the Web browser. The format for numbers, dates and times varies according to the defined language setting. To preview the analysis applications in the required language and formatting, application designers can switch between the preferred viewing locale of the BI launch pad (BI Platform User Settings radio button) or the Web browser language (Web Browser radio button).
Application Recovery

In the default setting, the system automatically saves unsaved applications every minute. You can configure the auto-save time interval as required. Select the Save application recovery information every <1> minute checkbox and enter the required number for the auto-save time interval.

Tip
The design tool needs to be restarted for the new interval to become active.

There is a background job that searches for unsaved applications during the specified interval. If unsaved applications are found, the system extracts their XML code and stores it under <user home directory> \Analysis-workspace\.metadata\.plugins\com.sap.ip.bi.zen. The content of this file is encrypted using Eclipse secure store technology.

Note
The auto-save function does not replace saving an application. Saving an application or closing it and answering the Save changes? question with either Yes or No will delete the XML from the autosaves file.

In the event of a system crash, the auto-saved XML persists. When restarting the design tool and opening the affected application again, the designer is informed that an auto-saved version of the application exists.

- If the designer decides to restore the auto-saved version, the system takes the stored XML, saves the application and opens the application in the layout editor. The auto-saved XML is removed.
- If the designer decides to discard the auto-saved version, the auto-saved XML is removed as well.

Member Selection

Application designers need to pick single members of a dimension when using statements like setFilter for a dropdown box, for example. They can pick the members in the content assistance of the Script Editor dialog box or in the Select Member dialog box. The Select Member dialog box can list a small or large number of members, depending on the maximum threshold number of members. You can set the default threshold in the Preferences dialog box.

- You can define the maximum number of members that are displayed in the Select Member dialog box. Enter the required number in the Maximum number of members to fetch from backend in dialog checkbox. The default threshold is 1000.
- You can define the maximum number of members that are displayed in the content assistance of the Script Editor dialog box. Enter the required number in the Maximum number of members to fetch from backend in content assistance checkbox. The default threshold is 20. If the number of available members exceeds this threshold number, the content assistance does not list single members. Instead it offers the Select Member... entry, which opens the Select Member dialog box.

Tip
Designers can access the content assistance in the script editor by pressing CTRL + SPACE.
You can also decide whether the system should display warnings in the script editor whenever designers manually enter non-existent values. To activate the warnings, select the Display warnings for manually entered invalid values checkbox.

Prompt Handling

In the Prompts dialog box, application designers and application users set values for prompts. For SAP NetWeaver BW data sources, prompts are defined as variables. In SAP HANA, prompts are defined as input parameters or variables. When working with data sources with defined variables, application designers might be prompted to set the required values before continuing their work in the design tool (if there are mandatory variables without default values, or if variables have invalid default values, for example). SAP BusinessObjects Design Studio stores all valid prompt values of each analysis application in the user’s cache file in <user home directory>/Analysis-workspace\metadata\plugins\com.sap.ip.bi.zen\cache. This provides application designers with a smooth workflow in the design tool. If this was not the case, the Prompts dialog would appear when designers reload an analysis application or modify the initial state of the data source in the Initial View dialog box, for example.

In the Preferences dialog box, you can specify whether the Prompts dialog box should appear when an analysis application is executed locally:

- If you want to simulate how an application user opens the analysis application, leave the checkbox deselected (default setting). When you execute an analysis application locally, the Prompts dialog box appears and you can set the prompt values in the same way an end user would do.
- If you want to use the prompt values from the cache file, select the Use cached prompt values for local execution checkbox. The Prompts dialog box does not appear and application designers can test their analysis applications quickly.

**Note**

If the Prompts dialog box still appears even if this setting is activated, check if the Force Prompts On Startup property of the analysis application is set to true. This property always forces the Prompts dialog box to appear, regardless of whether the Use cached prompt values for local execution checkbox is selected.

If you want to clear the prompt values of an analysis application in the cache file, press Clear Prompt Value Cache... You can select the required analysis application(s) for this cache deletion. When you reload the application, the Prompts dialog box will appear and you can set new values, for example.

Application Templates

SAP BusinessObjects Design Studio includes a set of templates that offer designers an easy way to get started. When creating new applications, designers can choose between different templates that are optimized for desktop Web browser applications or mobile applications in the New Application dialog box. For more information, see “Creating a new analysis application” in the Application Designer Guide: Designing Analysis Applications under Help in the design tool.
Application designers can also create analysis applications and provide them as templates for other application designers. Under |Tools| Preferences |Application Design| Application Templates, you can add the path to the folder where these templates are stored. Here you can also define your own template categories that indicate the target device types recommended for a specific template. The template categories are then listed in the New Application dialog box |Application| New..., and also in the Export Application As Template dialog box |Application| Export as Template... . For more information, see “Exporting Applications As Templates” in the Application Designer Guide: Designing Analysis Applications under |Help| Help Contents in the design tool.

Backend Connections

Under |Tools| Preferences |Application Design| Backend Connections, you can reload new connections or universes in the design tool. This function allows application designers to continue their work in the design tool without closing and opening it again.

Network Connections

Under |Tools| Preferences |Application Design| Network Connections, you can specify the proxy settings to be used when opening connections. For more information, see “Network Connection Preferences” in the online help. You can access this chapter by pressing F1 or the Help button in the Preferences dialog box.

Support Settings

Under |Tools| Preferences |Application Design| Support Settings, you can specify the amount of information stored in a log file and activate functions to record traces or collect statistics data.

Related Information

Setting Network Connections If Logon Problems Occur [page 250]
Activating Runtime Traces [page 251]
Activating SAP JCo Traces [page 252]
Viewing And Collecting Statistics Data At Runtime [page 253]

2.5.8 Storage of Applications and Images

To work efficiently with the design tool on the BI platform, you need to know where your applications are stored and where to store the images and icons that you want to insert in your applications.
Analysis applications, images and icons

You store the analysis application files in folders on the BI platform. You can also store images and icons you use in your analysis applications on the BI platform.

You can also use images and icons in your analysis applications from the Internet or intranet.

Tip

We recommend that you use one folder for all images and icons. This makes it easier to transport analysis applications and the referred images or icons between different BI platform systems.

Analysis applications history

The analysis applications history in the Application menu is persisted in your cache file <home directory>\.sap\com.sap.ip.bi\cache.

Data source history

The data source history in the Add Data Source dialog box is persisted in <home directory>\.sap\com.sap.ip.bi\cache.
3 Creating New Analysis Applications

Context

When creating new analysis applications, you can select different templates that are optimized for desktop Web browser applications or mobile applications. You can choose between blank templates or predefined templates that correspond to various design and business needs. When you choose a predefined template, the system automatically creates a copy of this template. You can change the copy according to your needs.

Procedure

1. Click [Application New...] in the menu of the design tool or click Create Analysis Application on the Welcome page. The New Application dialog box is displayed.
2. On the New Application dialog box, perform the following steps:
   a. In the Folder box, browse for the folder on the BI platform, where you want to save your applications to. Make sure that you have the relevant access rights for the folder. If you encounter problems, contact your system administrator.
   b. In the Name box, type a unique name for the application.
   c. In the Description box, type a meaningful, easy-to-understand description for the application.
   d. In the dropdown box on the right of Target Device specify whether you want to create a Desktop Browser application or an iPad or iPhone application.
      If you need to change the type of your application (desktop browser or iPad or iPhone application) later on, you can do this by editing the application property Theme.
3. Click Next to go to the template selection screen.
4. On the template selection screen, choose one of the following options:
   ○ Choose Blank if you want to create an empty application.
   ○ Choose one of the predefined templates if you want to create an application with specific contents for your business needs.
5. Click Finish. The Editor is now ready for editing.

Results

You have created a new application. Now you can define the content of your application by adding components and data sources to the blank template or by adjusting the predefined template according to your needs.
Related Information

Working with Templates [page 85]

3.1 Adding Components to an Application

Prerequisites

You have opened a new or existing application.

Context

You add components to an application to make the data visible, enable user interaction and create the layout of the application. Usually you use crosstabs or charts to visualize the data. Buttons or dropdown boxes enable the application user to interact with the data.

Procedure

In the Components view, click a component:

- Drag and drop the component of your choice into the editor area.
- Drag and drop the component of your choice into the Layout folder of the Outline view.
- Enter a text (no wildcards) in the filter text box at the top of the Components view, if you want to filter for a certain component. The component groups (for example, Analytic Components or Basic Components) are automatically hidden if no match was found inside the group, or expanded if they are currently collapsed and contain a match with the filtering string.

Results

You have created the general user interface of your application. You can now specify and change the properties of the components you have added to the application.

Related Information

Specifying the Properties of a Component [page 37]
3.1.1 Specifying the Properties of a Component

Context

Once you have dragged and dropped one or more components into the editor area, you can adjust the layout and behavior of the components by changing their properties.

Procedure

1. Choose the component you want to adjust:
   - Click on a component in the editor area.
   - Click on a component in the Layout folder of the Outline view.
   The properties of the component are now ready for editing in the Properties view.
2. In the Properties view, click on the property you want to change.
3. Enter the corresponding property value on the right side.

There are several ways to set the value of the property (depending on the property type):

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric entry</td>
<td>Enter a number and click Enter. This number represents either a numeric pixel description (such as for the properties Top Margin, Left Margin, Bottom Margin, Right Margin, Width, Height), or an absolute number (such as for the properties Grid Row and Grid Column).</td>
</tr>
<tr>
<td>string entry</td>
<td>Enter a text and click Enter. Properties of this type include Caption and Tooltip.</td>
</tr>
<tr>
<td>Boolean choice</td>
<td>Click the dropdown box in the Value column of the relevant property. Click false or true and then Enter. Properties of this type include Enabled and Style.</td>
</tr>
<tr>
<td>interaction using</td>
<td>With more complex properties you open and work in special dialogs. Click the Value column of the relevant property. The value field is now ready for editing. Click on the button next to the field. A dialog box opens. You can now edit the items for the dropdown-box component, add a data source or use the script editor for creating interactive components.</td>
</tr>
<tr>
<td>dialog boxes</td>
<td></td>
</tr>
</tbody>
</table>

3.1.2 Deleting Components in an Application

Prerequisites

You are in the editing mode of an application.
Context

You want to change an application by removing existing components.

Procedure

You can either:

○ Click the component you want to delete in the editor and press [Del] on your keyboard.
○ Right-click the component you want to delete in the Layout folder of the Outline view and click Delete.

Results

You have deleted the selected component from your application.

3.1.3 Selecting Multiple Components in the Editor

When working on your application in the editor, you can select (and move) multiple components by holding the CTRL key while clicking.

When selecting multiple components in the editor, take the following hints into account:

● You can move multiple components once they are selected.
● You can only move components that are in the same container.
● It is not possible to resize the components once they are selected.
● When changing a container, the position of the mouse is used to specify the container which all selected components are moved to.
● You can select every component in the application but you can only move components that are in the same container.
● Container components do not change to show multiple selected components (unlike single selection).
● You can only move container components into or out of container components as described above (and not by using the arrow keys).
4 Creating Analysis Applications for Mobile Devices

When creating mobile analysis applications for mobile devices, note the following SAP recommendations:

- Keep the applications simple and do not use too many components.
- Do not use crosstab components for mobile applications.
- Only design mobile applications in portrait format. Landscape orientation is not supported when running the application in the SAP BusinessObjects Mobile application.

**Note**

- When using an iPhone, only create applications based on the predefined templates for iPhones (do not use the blank template for iPhones). In the New Application dialog box, select iPhone as the target device. In the next step, select a predefined iPhone template (for example, iPhone Template 1).
- When using an Android phone or tablet, there are no predefined templates. In the New Application dialog box, leave the default value as the target device. Within the Display properties of the application, select one of the following mobile themes listed:
  - mobile black
  - blue crystal
  - mobile

For more information about Android version support, see the SAP Product Availability Matrix (PAM).
5 Creating the Layout of an Analysis Application

You create the layout of an application by inserting components in the editor, changing the properties of the components and arranging the components within an application (by using container components). Most of the work is performed in the layout editor of the design tool. The editor provides a What-You-See-Is-What-You-Get display. This enables you to view the working application during the creation process and get immediate feedback. Some of the steps you perform in the layout editor can also be performed in the Outline and Properties view of the design tool.

The following aspects are relevant during the layout creation process:

- changing the (layout) properties
- working with the layout structure of the Outline view
- using container components

5.1 Changing the Layout Properties and Docking Behavior

You create the layout of an application by inserting components in the editor and changing the layout properties of the components. You can change the layout properties of components either in the Properties view (where you can edit all properties) by entering the values manually or in the layout editor by dragging the borders of a component. The following figure shows a button component in the editor:

The red rectangle shows the size of the component. It contains eight red handles for resizing. In this figure you see two (out of a possible four) docking visualizers that show the distance between the component and the top border and left border of the application. Docking visualizers are displayed as an arrow with a number if the component is bound to a border (number is displayed by the relevant property). A docking visualizer displayed as an empty circle means that the component is not bound to a border (“auto” is displayed by the relevant property).
number and the empty circle are interactive. Clicking the number changes the display to “auto” and clicking on the empty circle changes the display to a number that represents the distance to the container border.

When changing the layout properties, keep the following points in mind:

There are three properties that specify the position and size of a component in an application:

- **width**: distance to left margin and distance to right margin along the horizontal axis
- **height**: distance to top margin and distance to bottom margin along the vertical axis

To enable dynamic/relative resizing of the components with changing Web browser windows, one of these three properties of each axis is always set to **auto**. It is not possible to set all three properties to **auto**.

### Example

**Crosstab with fixed margins**

- Top margin = 60
- Left margin = 120
- Bottom margin = 116
- Right margin = 455

The values for width and height are set to **auto**. This means that the distances between the margins of the crosstab and the margins of the application are fixed, while the height and width of the crosstab are variable. The height and width of the application vary according to different screen or window sizes. In this case the height and width of the crosstab vary accordingly.

### Example

**Crosstab with fixed width and height**

If you set the height and width of the crosstab to fixed values, one property of each axis is set to **auto**.

- Top margin = 60
- Left margin = 120
- Bottom margin = **auto**
- Right margin = **auto**
- Width = 600
- Height = 400

In this case, the left margin and the width of the crosstab are fixed, while the third property on the horizontal axis (the right margin) is variable. The top margin and the height of the crosstab are fixed, while the third property on the vertical axis (the bottom margin) is variable. With different screen or window sizes, the height and width of the application vary. In this case, the bottom margin and the right margin vary accordingly.

### Related Information

- [Working with the Outline View](page 43)
5.2 Using Container Components

You can use container components to arrange and organize the structure and layout of your application. After inserting a container component (tabstrip, pagebook, grid layout) in the editor, you can place analytic components, basic components and other container components into the (first) container component.
6 Working with the Outline View

In the Outline view of the design studio, you see all available components and elements of an application like data sources and components. They are listed in a hierarchical folder structure with each folder representing one type of application element. You can use the filter text box to type in the names of application elements or parts of these names. As soon as you have typed in a filter string, the box is marked in orange bold and the findings in the structure are displayed in bold as well. In addition, you can use the various context menu functions to create your application (copy and paste, for example). Currently the following folders are available:

- In the Layout folder, you will find all components used in the application. You can easily change the order and position of the components within the folder or within container components.
- The Data Source folder lists all data sources used in the application. You can assign a data source to a component by simply dragging and dropping it to the component in the editor or in the layout folder.
- In the Planning Objects folder, you can add planning functions and sequences of SAP NetWeaver BW Integrated Planning, which enable you to enter data automatically.
- In the Technical Components folder, you will find the following elements:
  - backend connection
    The Backend Connection enables the application user to select a generic data source at runtime by means of a data source selection dialog box. You can either use the predefined data source selection dialog box for runtime and configure it in the properties of the Backend Connection, or you can use the API to create your own user interface for browsing data sources. Before you can use the Backend Connection, you have to assign a system at design time. This can be done in the Property view of the design tool or using the API.
  - context menu
    The technical component Context Menu enables the application user to navigate and analyze data at runtime. At design time, the context menu is automatically added to every new application, but can be removed if not required. Each application can have only one context menu.
  - global scripts object
    The Global Scripts Objects is a technical component type, which provides a grouping of global script functions. On each global scripts object, you can create any number of script functions. Each script function has a configurable return type and can have any number of typed input parameters.
  - pdf
    The PDF is a technical component type, which enables the export to PDF functionality. Depending on the scripting methods applied, you can export to PDF a WYSIWYG version of your application or panel within your application. You can also export to PDF a report style version of your application, with one component per page.
  - text pool
    The Text Pool enables text translation in analysis applications. In addition to the texts of the data from the back-end system (SAP NetWeaver BW), analysis applications can contain translatable texts, like labels on buttons or messages, which are created by you, the application designer. If you want to provide your analysis applications in different languages, you need to make your application translatable by adding a Text Pool component to your application. The system collects all translation-relevant texts that you enter as property values in the Properties view in the Text Pool component and saves them for translation.
6.1 Using the Context Menu in the Outline View

The context menu for the respective elements in the Outline view offers you a variety of functions that help you create your application and work with it efficiently. The scope of functions depends on the element (data source or component) in the structure.

Context menu functions for data sources

For single data sources in the Outline view, you have the following functions:

- **copy**
  Use **Copy** to copy the selected data source.

- **paste**
  Use **Paste** to paste a copied data source. The system displays the pasted data source immediately in the Data Sources folder and automatically generates a new data source alias (for example DS_2).

- **rename**
  Use **Rename**, if you want to rename an existing data source alias.

- **delete**
  Use **Delete** to delete an existing data source.

- **edit initial view**
  Use **Edit Initial View...**, if you want to change the initial view of a data source. The Edit Initial View of *<your selected data source>* dialog box opens.

- **reset initial view**
  Use **Reset Initial View...**, if you want to reset a previously changed view of a data source. Therefore this context menu function is only activated if the initial view of the data source has been changed.

- **smart paste**
  This context menu function is only displayed if you have created a data source in SAP BusinessObjects Analysis, edition for Microsoft Office and you have used the smart copy function to copy it.

- **find references**
  Use **Find References**, if you want to know to which components the selected data source is assigned to or in which scripts of the application the selected data source is used. The results are displayed in the Search Results view.
Context menu functions for components

For single components in the *Outline* view you find the following functions:

- **copy**
  Use *Copy* to copy the selected component.

- **paste**
  Use *Paste* to paste a copied component. The system displays the pasted component immediately in the *Layout* folder and automatically generates a new name (for example CHART_2).

- **rename**
  Use *Rename* if you want to rename an existing component.

- **arrange**
  Use *Arrange Align Left / Align Right / Align Top / Align Bottom* if you want to arrange components in a special way. To do this, select at least two components in a container (for example, in the application container). Use the context menu or the toolbar of the design studio to arrange them as required in the application.

  
  Depending on the docking behaviour of the selected components and the selected alignment, the docking behavior might be different.

  - **arrange**
    Use *Arrange Distribute Horizontally / Distribute Vertically* if you want to distribute at least three components vertically or horizontally in the application. The selected outermost components keep their positions, whereas the selected inner components are arranged evenly and are the same distance to each other. All selected components keep their docking behaviour.

- **delete**
  Use *Delete* to delete an existing component.

- **hide**
  Use *Hide* to hide the selected component in the editor. This function is especially useful, if your application has many container components that contain other components. The hidden component is grayed out in the *Layout* folder and the total number of all hidden components in the application is displayed in parentheses next to the *Layout* folder.

  If you do not want hidden components to be displayed (grayed out) in the structure of the *Outline* view, press the white arrow in the upper right corner of the *Outline* view and deselect *Display Hidden Components*.

  If you want to display all hidden components in the application, choose *Show All Hidden Components* in the context menu of the *Layout* folder.

- **show**
  Use *Show* if you want to display the hidden component in the editor. This function is only displayed for hidden components.

- **find references**
  Use *Find References* if you want to know to which applications the selected component is referenced to. The results are displayed in the *Search Results* view.
Context menu functions for planning objects

The following functions are available for single planning functions and sequences in the Outline view:

- **copy**
  Use Copy to copy the selected planning sequence or planning function.

- **paste**
  Use Paste to paste a copied planning function or sequence. The system displays the pasted planning object or sequence immediately in the Planning Objects folder and automatically generates a new name (for example PF_2).

- **rename**
  Use Rename if you want to rename an existing planning function or sequence.

- **delete**
  Use Delete to delete an existing planning function or sequence.

- **find references**
  Use Find References if you want to know which applications reference the selected planning function or sequence. The results are displayed in the Search Results view.

Context menu functions for backend connection

- **copy**
  Use Copy to copy the selected backend connection.

- **paste**
  Use Paste to paste a copied backend connection. The system displays the pasted backend connection immediately in the Technical Components folder and automatically generates a new name (for example Connection_2).

- **rename**

- **delete**
  Use Rename if you want to rename an existing backend connection.

- **find references**
  Use Find References if you want to know which applications reference the selected backend connection. The results are displayed in the Search Results view.

Context menu functions for context menu

The following functions are available for the technical component Context Menu:

- **delete**
  Use Delete to delete the context menu from the application.

- **find references**
  Use Find References if you want to know which elements of the application reference the context menu. The results are displayed in the Search Results view.
Context menu functions for global scripts objects

The following functions are available for the technical component *Global Scripts Object*:

- **copy**
  
  Use *Copy* to copy the selected global scripts object.

- **paste**
  
  Use *Paste* to paste a copied global scripts object with its global script functions. The system displays the pasted global scripts object immediately in the *Technical Components* folder and automatically generates a new name (for example, *GLOBAL_SCRIPTS_2*).

- **rename**
  
  Use *Rename* if you want to rename an existing global scripts object.

- **delete**
  
  Use *Delete* to delete an existing global scripts object.

- **create script function**
  
  Use *Create Script Function* to create a new script function.

- **find references**
  
  Use *Find References* if you want to know which applications reference the selected global scripts object. The results are displayed in the *Search Results* view.

**Note**

If you want to change an existing global script function, right-click this function and choose *Edit*. The *Edit Script Function* dialog box opens where you can make your changes.

Context menu functions for pdf

- **delete**
  
  Use *Delete* to delete the functionality to export to PDF from the application.

- **find references**
  
  Use *Find References* if you want to know which elements of the application reference the PDF technical component. The results are displayed in the *Search Results* view.

Context menu functions for text pool

The following functions are available for the technical component *Text Pool*:

- **delete**
  
  Use *Delete* to delete the text pool from the application.

- **find references**
  
  Use *Find References* if you want to know which applications reference the selected textpool. The results are displayed in the *Search Results* view.
6.2 Using the Context Menu (Technical Component)

The technical component **Context Menu** enables the application user to navigate and analyse data at runtime. At design time, the context menu is automatically added to every new application, but can be removed if not required. Each application can have only one context menu. If you want to remove the context menu from the application, right-click **CONTEXT_MENU** in the **Technical Components** folder of the **Outline** view and choose **Delete**.

Depending on the context, the following menu options are available:

- for dimensions or dimension members
  - sort the dimension
    - sort by key or text ascending or descending (**Text/Key Ascending**, **Text/Key Descending**)
    - sort by hierarchy if a hierarchy is assigned to a dimension (**Sort by Hierarchy**)
  - change the display of the dimension members
    - switch between different key and text combinations (**Key**, **Text**, **Key and Text**, **Text and Key**)
    - choice of text presentation type (**Short Text**, **Medium Text**, **Long Text**, **Text**)
  - choose which display attributes are shown in the result set
  - change hierarchy of the dimension
    - select hierarchy (and activate it) (**Select Hierarchy**)
    - expand and collapse the assigned hierarchy (**Expand All**, **Collapse All**)
    - expand to a specific hierarchy level (**Expand to Level**)
  - switch the totals display of the dimension (**show totals, hide totals, Hide Totals if Only One Member**)
  - filter dimension members
    - open the filter to filter the dimension (**Filter Members**)
    - clear the current filter (**Select All Members**)
    - filter the member and move the dimension to the background filter (**Keep Member**) (only available if user has clicked on a dimension member)
    - filter the member and leave the dimension in the drilldown (**Keep Member on Axis**) (only available if user has clicked on a dimension member)
    - filter the member and swap the dimension with another dimension from the free axis (only available if user has clicked on a dimension member) (**Filter Member and Swap**)
  - change drilldown

Application Designer Guide: Designing Analysis Applications
Working with the Outline View
○ add another dimension from the free axis to the drilldown (Drilldown By)
○ swap the current dimension with another dimension (Swap With)
○ remove the current dimension from the drilldown (Remove Drilldown)
○ swap the rows axis and the columns axis with each other (Swap Axes)

● for attributes
○ sort the dimension according to this attribute
○ change the member display of the attribute

● for result set data cells
○ If the query is input-enabled and the data cell is input-enabled, the cell can be locked (Lock Value)

● for structure members
○ sort the dimension members according to the measure values (Measure Value Ascending or Measure Value Descending)
○ change the drilldown (see the context menu entries for dimensions above)
○ filter the structure (see the context menu entries for dimensions above)
○ change the number format of the data cells belonging to this structure member
  ○ change the scaling factor (Scaling Factor)
  ○ change the decimal places (Decimal Places)
○ specify whether units and currencies are displayed in the crosstab and where they are displayed
  ♦ Units and Scaling Factors ♦ Display Both in Header/Display Units in Data Cells/Do Not Display

● navigation to jump targets (Jump to)
  If Report-Report Interface targets (RRI targets) are specified for the query, the Jump to menu entry is displayed. If elements of the query have specific targets, the context menu for the jump targets contains an additional entry: More..., which retrieves these specific targets and displays them to the user, thus enabling the user to navigate to these targets.

Related Information

Creating a Generic Analysis Template for RRI Jump Targets [page 235]
7 Adding a Data Source

Prerequisites

Before you can add data sources to the applications you have to create connections to BI backend, SAP HANA systems or universes containing the business data. SAP BusinessObjects Design Studio can access SAP HANA systems or SAP NetWeaver BW systems as BI backend systems.

Context

You add a data source to be able to connect the various components with data. A data source can either be a query or query view of a BW system, an analytic or calculation view of an SAP HANA system (with the Multidimensional Reporting property set to true. For more information, see “Creating Analytic Views” in the SAP HANA Developer Guide on SAP Help Portal at http://help.sap.com/hana_appliance), or a query based on a universe. The appearance of the dialog boxes can vary, depending on the chosen data source.

Procedure

1. In the design tool, there are several methods and contexts for adding a data source to the application. Choose the method that suits the way you like to work in the design tool:
   - Click Application ➔ Add Data Source... in the menu of the design tool.
   - Right-click Data Sources in the Outline tab of the design tool and click New....
   - If you have already added a component for displaying data (such as crosstab or chart) to your application, you can simply add a data source in the context of the chosen component. You do not need to assign the data source to the component in a further step:
     1. Click on the chart or crosstab component in the design area.
     2. Click on the Data Source property in the Properties view of the design tool and choose Add... in the menu.

   The Add Data Source dialog box opens.

2. In the Add Data Source dialog box, perform the following steps:
   a. Select a connection. For more information, see “Selecting a connection” in the Related Topics section.
   b. Select a data source. For more information, see “Selecting a data source” in the Related Topics section.

   ➤ Tip

To view and select an entry from the history of your last ten choices, press [CTRL] + [Space] on your keyboard. You can filter the entries in the history view by typing the first letters of the required data source.
In the Data Source Alias field, the system generates an alias for the data source by default. However, you can change the data source alias as required.

As you can use the same data source several times within one application, you work in the design tool with data source aliases as reference names.

3. Click OK.

Results

You have added a data source with a data source alias as a reference name. You can now assign this data source to one or more components in your applications.

Related Information

Selecting a Connection [page 559]
Defining Connections to BI Backend Systems [page 560]
Selecting a Data Source [page 53]
Assigning a Data Source to a Component [page 79]

7.1 Selecting a Connection

Prerequisites

Before you can choose a connection, your administrator has to create OLAP data source connections to SAP HANA systems, BW systems or universes containing business data.

If you want to use universe queries as data sources for analysis applications, your administrator needs to create the universes and the corresponding relational connections using the Information Design Tool (IDT) and publish them to the BI platform.

Note

The BI platform server needs to be SAP BusinessObjects Business Intelligence 4.1. If you want to use universe queries as data sources in your analysis applications, you cannot use SAP BusinessObjects Business Intelligence 4.0.

In Design Studio you can use universes that meet the following prerequisites:

- relational universes
- single-source universes

For general information about universe access in SAP BusinessObjects Design Studio, see Universe Access in Design Studio [page 70]
If you want to create a planning application, you have to select a planning connection. For more information see Selecting a Planning Connection [page 121]

Context

Connections represent BW or SAP HANA systems or universes. They are defined and configured by your administrator. Connections have to be active if you want to select a data source and use it immediately. However, you can also work with an inactive connection when designing and then validate the data sources for this connection later when the connection is active. The connection is automatically active when the backend system (SAP HANA, SAP NetWeaver BW, universe) is up and running.

Procedure

1. In the Connection box, click Browse.... The Select Connection dialog box opens where you can choose one of the created connections. The Log on to <BW system> or the Log on to <SAP HANA system> dialog box is displayed.
   If you or your system administrator has created new data source connections on the platform (BI platform or SAP NetWeaver) or locally, and you want these connections to be displayed in the Select Connection dialog box, press Reload. The system adds your new connections to the existing connections list in the Select Connection dialog box.
2. This step depends on your choice of connection (SAP BW or SAP HANA system or universe). For a BW system, enter the client, your user credentials and the language. For an SAP HANA system, enter your user credentials and the language. If you chose a universe as your connection, you can skip step 2.
   If the administrator has created a single sign-on connection for a BW system, you can connect to the BW systems without having to enter the BW username and password in the Logon to <BW system> dialog box (when inserting a BW data source in the design tool). You can change the client and the language of the BW system in this dialog box.

   Note
   ○ Currently single sign-on for SAP HANA system connections is supported for the SAP BusinessObjects platform 4.1 but not for the 4.0 platform.
   ○ Note that for OLAP connections of type SAP BW or SAP HANA the authentication type Prompt is not supported by Design Studio. So either choose Pre-defined or Single-Sign-On when creating an OLAP connection on the BI Platform.

Results

You have selected a connection and can now select a data source based on this connection.
7.2 Selecting a Data Source

Context

You select a data source so that you can assign data to your component. Data sources can be SAP BW queries or query views, SAP HANA analytic or calculation views or universe queries. Depending on your connection choice, you have to perform different steps for BW/ and SAP HANA data sources on the one hand, and universe data sources on the other hand.

Procedure

1. For SAP BW and SAP HANA data sources: In the Data Source box, use one of the following methods:
   - Type the name of a data source that you know.
   - Type the name of a data source that you want to create later but which does not exist at the moment and click Enter. The system asks you to confirm that the data source does not show any data, as long as it is not active for the chosen connection, and that you can use the data source with the corresponding data source alias for further design steps. Choose the type for the data source (query, query view, InfoProvider) and click Add.
   - Click Browse. The Select Data Source dialog box opens. Depending on the chosen connection (BW system or SAP HANA system), this dialog box will have a folder tab and/or search tab.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection to BW system</td>
<td>Folder tab: You can specify if you want to look for queries or query views in the InfoAreas or Roles view. The InfoAreas view displays all InfoObjects (InfoAreas, InfoCubes, queries, and query views) in a tree structure. The Roles view displays your role-based objects in a tree structure. Select the view that suits your needs. Search tab: You can search for the description or technical name of a data source. To retrieve data sources that begin with a specific string, you can type * after a partial string. You can also type a partial string without using *. The system will display any result that includes the partial string.</td>
</tr>
<tr>
<td>connection to SAP</td>
<td>Folder tab: All available data sources are displayed in a hierarchical structure. Select the one that suits your needs.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HANA system</td>
<td><strong>Search</strong> tab: You can search for the description or technical name of a data source. To retrieve data sources that begin with a specific string, you can type * after a partial string. You can also type a partial string without using *. The system will display any result that includes the partial string.</td>
</tr>
</tbody>
</table>

**Remember**

The option of typing a data source name or browsing for a data source depends on how a connection has been configured. If the administrator has set one single data source for a connection, the system automatically displays this data source in the **Data Source** box. In this case, you cannot overwrite the entry or browse for another data source.

2. For universe data sources: Click **Edit Query Specification** to open the Query Panel. You can use the Query Panel to build, test, and preview the results of queries. You edit a query specification by adding dimensions and measures, setting filters, defining prompts, and so on. For further information about using the Query Panel, see **About the Member Selector [page 56]**.

3. Click **OK**.

4. For universe data sources: this step is optional but usually recommended.
   a. Select and right-click the new universe data source in the **Outline** view.
   b. Choose **Edit Initial View...** in the context menu of the new data source and define the initial geometry of your query, set additional filters, and so on.
   a. Click **OK + Create Crosstab** or **OK** to leave the **Edit Initial View** dialog box. For further information about the Initial View dialog box, see **Working with the Initial View Dialog for Data Source Aliases [page 76]**.

**Results**

You have selected a data source that you can now assign to your component. The properties of the chosen data source (name, description and type) are displayed in the **Add Data Source** dialog and the properties view of the data source.

**Tip**

You can easily exchange the data source: Click the symbol right to the data source name in the data source properties view. The **Exchange Data Source** dialog box opens where you can choose another data source.

**Related Information**

Assigning a Data Source to a Component [page 79]
7.2.1 Creating a Query

Prerequisites

This procedure assumes that you have opened the Query Panel (also referred to in the design studio as Edit Query Specification dialog box) on a published universe.

Context

You can use this procedure to create queries and preview the results of universe queries.

For links to more detailed information about each step, see the Related Topics.

Procedure

1. To select the objects you want to include in the query, drag objects from the universe on the left into the Result Objects pane.
2. For hierarchy result objects, select members to include or exclude in the results. To open the Member Selector, click the arrow to the right of the hierarchy object name.
3. To filter the results of the query, drag objects from the universe into the Filter Objects pane.
   - If a mandatory filter is defined on an object, the filter is triggered when you add the object to the Result Objects pane. The mandatory filter is visible in the query script, but not in the Filter Objects pane.
   - Non-mandatory, pre-defined filters are listed in the universe. You can drag these pre-defined filters into the Filter Objects pane to limit the results. The filter is visible in the query script.
   - You can also build business filters, including filters that use prompts. For detailed information, see the related topics.
4. For relational universes, you can build combined queries. To open the Combined Queries pane, click the (Show/Hide) Combined Query Panel icon in the upper left corner of the dialog box.
5. To preview the query results, click the refresh button in the Data Preview pane.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat layout</td>
<td>Displays repeated values for a level in every row.</td>
</tr>
<tr>
<td>Hierarchical layout</td>
<td>Displays repeated values once for a level.</td>
</tr>
</tbody>
</table>

6. Click OK. Now you can use the query as a data source for your application.
   
   To define the initial geometry of your query and set additional filters, we recommend that you open and refine the query in the Edit Initial View... dialog box.
7.2.2 About the Member Selector

The Member Selector lets you visualize and select members in a hierarchy. Use the Member Selector to:

- Select the members you want to appear in the query result set.
- Define members that will be excluded from queries.
- Define prompts to allow the selection of members to appear in the query each time you run the query.
- Select the members for a named set.
- Select the members when defining a Business Security Profile filter.

You open the Member Selector from hierarchy objects that you include in queries in the Query Panel. The Member Selector opens automatically when you edit named sets or filters for a Business Security Profile on a hierarchical business layer.

Related Information

Selecting hierarchy members [page 58]
About selecting hierarchy members [page 56]

7.2.2.1 About selecting hierarchy members

In the Member Selector, you can select members in several ways:

- Select members explicitly in the hierarchy. For example, explicitly select the [California] and [Los Angeles] members of the [Geography] hierarchy.
- Select members implicitly using hierarchy relationships. For example, to select US states, you can select the child members of the [US] member.
- Select members included in a named set, for example Top Cities by Revenue, to include the cities that generate the most revenue.
- Select all members in a hierarchy level.
- Select all members up to a certain level in the hierarchy.
- Select calculated members.

The Member Selector contains three tabs:
<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Members</td>
<td>Displays the members arranged hierarchically. Use this tab to select members explicitly, by hierarchical relationships, and by specifying all members up to a given level.</td>
</tr>
<tr>
<td>Metadata</td>
<td>Shows the hierarchy levels (if the hierarchy supports named levels), named sets, and calculated members.</td>
</tr>
<tr>
<td>Prompts</td>
<td>Lets you define and modify prompts.</td>
</tr>
</tbody>
</table>

For information on how to select, display, search for, and sort hierarchy members, see the related topics.

### Related Information

- Selecting hierarchy members [page 58]
- Selecting members by hierarchy relationship [page 58]
- Selecting hierarchy members by level [page 59]
- Selecting named sets [page 60]
- Selecting calculated members [page 61]
- Searching for hierarchy members [page 61]
- Excluding hierarchy members [page 62]
- Defining a prompt to select members [page 63]
- Showing selected members in the Member Selector [page 63]
- Sorting hierarchy members [page 64]
- Setting display options [page 64]
- Showing estimated child count [page 65]

### 7.2.2.2 Opening the Member Selector in the Query Panel

#### Procedure

1. In the Query Panel, add the hierarchy object to the Result Objects pane.
2. To open the Member Selector, click the arrow to the right of the hierarchy object name.
3. You can now select members in the hierarchy for inclusion or exclusion in a query. For descriptions of different ways to select members, see the related topic.

#### Related Information

- About selecting hierarchy members [page 56]
7.2.2.3 Selecting hierarchy members

Procedure

1. In the Member Selector, click the Members tab to display the hierarchy members.
2. Select members in the hierarchy display.
3. To select all members in the hierarchy, click the Select icon, and select Select All.
4. To select all members up to a specified level in the hierarchy, click the Select icon. You can identify the level in two ways:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a named level</td>
<td>This option is only available if the hierarchy has named levels. Select Select All Members until Named Level and select the level from the submenu.</td>
</tr>
<tr>
<td>Select a number of levels below the root</td>
<td>Select Select All Members until and select the number of levels from the submenu.</td>
</tr>
</tbody>
</table>

5. When you complete your selection, click OK.

Results

The selected members appear below the hierarchy object in the Result Objects pane of the Query Panel. When you run the query, only those members are included in the query result.

Related Information

Opening the Member Selector in the Query Panel [page 57]

7.2.2.4 Selecting members by hierarchy relationship

Procedure

1. In the Member Selector, click the Members tab to display the hierarchy members.
2. In the hierarchy, right-click the member for which you want to define the hierarchy relationship.
3. Select the relationship function from the menu:
### Note

*Children/Descendants* and *Parents/Ancestors* are mutually exclusive pairs. You cannot select both the children and the descendants of a member, and you cannot select both the parents and the ascendants of a member.

<table>
<thead>
<tr>
<th>Relationship Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Self</strong></td>
<td>Includes only the selected member. This is the default setting.</td>
</tr>
<tr>
<td><strong>Children</strong></td>
<td>Includes members one level below the selected member that have the selected member as their parent. The selected member is not included.</td>
</tr>
<tr>
<td><strong>Descendants</strong></td>
<td>Includes all members at all levels below the selected member. The selected member is not included.</td>
</tr>
<tr>
<td><strong>Descendants until Named Level...</strong></td>
<td>Includes the members at levels below the selected member until the named level you select. This option is only available if the hierarchy has named levels.</td>
</tr>
<tr>
<td><strong>Descendants until...</strong></td>
<td>Includes the members at levels below the selected member until the number of levels you select.</td>
</tr>
<tr>
<td><strong>Parent</strong></td>
<td>Includes the member that is one level above the selected member. The selected member is not included.</td>
</tr>
<tr>
<td><strong>Ancestors</strong></td>
<td>Includes all members at all levels above the selected member. The selected member is not included.</td>
</tr>
<tr>
<td><strong>Siblings</strong></td>
<td>Includes members at the same level that have the same parent as the selected member. The selected member is not included.</td>
</tr>
<tr>
<td><strong>Exclude</strong></td>
<td>Excludes members according to the relationship function (Self/Children/Descendants/Parent/Ancestors/Siblings).</td>
</tr>
</tbody>
</table>

### Related Information

**Opening the Member Selector in the Query Panel** [page 57]

### 7.2.2.5 Selecting hierarchy members by level

#### Prerequisites

To select members by level, the hierarchy must have named levels.
Procedure

1. In the Member Selector, click the Metadata tab to display the hierarchy levels.
   
   **Note**
   
   If the Levels folder does not display in the Metadata tab, the hierarchy is not level-based and you cannot select members by level.

2. Select levels in the Levels folder.
3. Click OK.

Related Information

Opening the Member Selector in the Query Panel [page 57]

7.2.2.6 Selecting named sets

Prerequisites

To select members by named set, the hierarchy must have at least one named set defined. Named sets are defined in the business layer of the universe.

Procedure

1. In the Member Selector, click the Metadata tab to display the named sets.
   
   **Note**
   
   If the Named Sets folder does not display in the Metadata tab, the hierarchy has no named sets defined.

2. Select named sets in the Named Sets folder.
3. Click OK.

Related Information

Opening the Member Selector in the Query Panel [page 57]
7.2.2.7 Selecting calculated members

Prerequisites

To select calculated members, the hierarchy must have at least one calculated member defined. Calculated members are defined in the business layer of the universe.

Procedure

1. In the Member Selector, click the Metadata tab to display the calculated members.

   Note
   
   If the Calculated Members folder does not display in the Metadata tab, the hierarchy has no calculated members defined.

2. Select calculated members in the Calculated Members folder.
3. Click OK.

Related Information

Opening the Member Selector in the Query Panel [page 57]

7.2.2.8 Searching for hierarchy members

Context

Use the Search function in the Member Selector to select hierarchy members from a list of search results.

Procedure

1. To open the Member Search dialog box, in the Member Selector Members tab, click the Search icon.
2. Enter text to search for in the Search pattern box.

   You can use wildcards in the search:
### Wildcard

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Matches any string of characters</td>
</tr>
<tr>
<td>?</td>
<td>Matches any one character</td>
</tr>
</tbody>
</table>

3. To search for text in the keys, select the **Search Keys** radio button.
4. Click **Search**.
5. To select members from the search results, select the members in the **Search results** table.
6. Click **OK**.

### 7.2.2.9 Excluding hierarchy members

**Procedure**

1. In the Member Selector, select the members that you want to exclude.
   - You can select members explicitly, by hierarchy relationship, by level, by named set, and calculated members.
   - The selected members are listed in the **Summary** pane of the Member Selector.
2. In the **Summary** pane, select the **Exclude** option next to the members or member sets you want to exclude.
3. Click **OK**.

**Results**

Below the hierarchy object in the **Result Objects** pane of the Query Panel, the excluded members appear with a line drawn through the names to indicate that they are excluded from the query.

**Related Information**

- Opening the Member Selector in the Query Panel [page 57]
- Selecting hierarchy members [page 58]
- Selecting members by hierarchy relationship [page 58]
- Selecting hierarchy members by level [page 59]
- Selecting named sets [page 60]
- Selecting calculated members [page 61]
- Searching for hierarchy members [page 61]
7.2.2.10 Defining a prompt to select members

Context

You can define a prompt to defer member selection to the time the query is run.

i Note

When selecting members in response to a prompt, you can only select members explicitly. You cannot select members by hierarchy relationship.

Procedure

1. In the Member Selector, click the Prompt tab.
2. Select Enable Parameter to defer member selection to when the query is run.
   You cannot access the other tabs in the Member Selector when the Enable Parameter option is selected.
3. Enter text for the prompt in the Prompt Text box.
4. If you want the prompt to select the previously-selected values by default when it displays, select Keep last values selected.
5. To define default values for the prompt, select Set default values and click Edit. In the Select Parameter Values dialog box, select default values for the prompt and click OK.
6. Click OK.

Related Information

Opening the Member Selector in the Query Panel [page 57]

7.2.2.11 Showing selected members in the Member Selector

Context

In the Member Selector Members tab, you can click the Expand tree to show selections icon to show the selected members in the hierarchy display.

The display automatically expands to show the following members:

- Explicitly selected members.
- Members used to select related members. The related members implicitly selected are not necessarily shown. For example, if the member called France was used to select its children, the tree view expands to
show France. If the node France contains no explicitly selected members, the node is not expanded the show the implicitly selected children.

Tip

The *Expand tree to show selections* command does not collapse nodes that are already expanded. To reduce the length of the display, close all open nodes in the hierarchy display before clicking the icon.

Related Information

Opening the Member Selector in the Query Panel [page 57]

7.2.2.12  Sorting hierarchy members

Context

By default in the Member Selector, hierarchy members are displayed sorted in the order they are stored in the database. To help find members in the hierarchy, you can sort the display in ascending or descending alphabetical order.

In the Member Selector *Members* tab, click the *Sort order* icon \[\text{Z} \to A\] and select the desired sort order.

The members are sorted locally in the Member Selector. Display of members in the query is not affected.

Related Information

Opening the Member Selector in the Query Panel [page 57]

7.2.2.13  Setting display options

Context

By default, the Member Selector displays hierarchy member captions. You can set the display options to display unique names, or both captions and unique names.

In the Member Selector *Members* tab, click the *Member display options* icon \[\text{S} \to \text{B}\] and select the desired display option.
7.2.2.14 Showing estimated child count

Context

The Member Selector estimates the number of children for each member. By default, the estimates are hidden. You can show the estimated child count in the hierarchy display.

In the Member Selector Members tab, click the Show/Hide estimated child count icon to toggle the display of child counts.

7.2.3 Filtering Data in the Query Panel

7.2.3.1 Creating a Business Filter

Context

Business filters are filters based on objects in the business layer. They limit the data returned in the query.

Query filters have the following structure: filtered object, operator, operand. For example, in the following filter:

```
[Country] InList (US; France)
```

The [Country] dimension is the filtered object, InList is the operator, and the list of values (US; France) is the operand. The filter removes all values of [Country] other than US and France from the query result.

The following table describes the components of a filter:

<table>
<thead>
<tr>
<th>Filter Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filtered object</td>
<td>The filtered object is the business layer object whose values are filtered. Dimensions, attributes, measures, hierarchies, and hierarchy levels can be used as filtered objects.</td>
</tr>
<tr>
<td>Operator</td>
<td>The operator is used to compare the filtered object with the operand. For example, the Equal To operator retains only those values of the filtered object that correspond exactly to the value of the operand.</td>
</tr>
<tr>
<td>Operand</td>
<td>The operand supplies the value or values used to filter the filtered object.</td>
</tr>
</tbody>
</table>

Procedure

1. In the Edit Query Specification dialog box, drag an object from the business layer to the Filter Objects pane. This is the filtered object.
2. In the Filter Objects pane, select an operator from the list.

3. In the Filter Objects pane, select an operand from the list.

Depending on the type of business layer and purpose of the filter, the following types of operands are available:

Table 2:

<table>
<thead>
<tr>
<th>Operand type</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Constant** | Use the Constant operand to enter values directly into the filter. For example, you can use a constant to enter France into the filter:  

```[Country] Equal To France```

You can also enter the @Variable function to retrieve the value of a system variable or User Attribute. For example, to filter on the current user login name, enter the constant operand as @Variable('BOUSER'). For more information about @Variable, see the related topic. |
| **List of Values** | Use the List of Values operand to select values from the list associated with the filtered object. For example, if the filtered object is [City], you can use the list of values to select one or more of the cities associated with the object. |
| **Object** | Use the Object operand to specify an object in the business layer. Drag and drop a business layer object to the operand position when defining the filter. |

**Note**  
You cannot select an object as an operand on some OLAP data sources or if the filtered object is a hierarchy. |
| **Prompt** | Use the Prompt operand when you want to be prompted for a value when the query is refreshed. See the related topic about filtering using prompts. |

**Note**  
Prompt operands are not available if the business filter is defined for a Business Security Profile. |

4. If you want to filter data on more than one criteria, add an additional filter by dragging another object to the Filter Objects pane.

By default, the filters are combined with the AND operator. To use the OR operator, double-click the box with the And operator.

**Note**  
The OR operator is not supported for OLAP data sources.

5. If you want to nest query filters, drag another business object and drop it onto an existing query filter in the Filter Objects pane.

Nesting query filters allows you to create more complex filter conditions than is possible when you combine filters at the same level. When you nest filters, you set the order in which they are evaluated. Nesting filters only makes sense once you have defined two filters at the same level.
7.2.3.2 Filtering Data Using Prompts

A prompt is a special type of query filter. It is a dynamic filter that displays a question every time you refresh the data in a query. You answer prompts by either typing or selecting the values you want to view before you refresh the data. The query then returns only the values you specified.

Prompts allow multiple users viewing a single document to specify a different sub-set of the database information and display it in the same report tables and charts. Prompts also reduce the time it takes for the data to be retrieved from the database.

When you define a prompt query filter, you can either build a new prompt, or use an existing prompt defined as a parameter in the business layer.

If you define more than one prompt in a query, you can change the order in which prompts are presented. Change prompt order in the query properties.

Merged Prompts

When querying a business layer or universe, similar prompts are merged. For prompts to be merged, the following rules must be true:

- The prompts have the same prompt text.
- The prompts expect answers having the same data type.
- The prompts expect the same number of answers. (The number of answers to be given depends on the operator used to reference the prompt. For example, Equal To expects one answer. Between expects multiple answers.)

A single prompt message appears for merged prompts. The list of values displayed by the merged prompt is the list associated with the prompt that has the most display property constraints.

Note

All prompts in the query are candidates for merging: parameters defined in the business layer or data foundation, prompts defined as query filters, and prompts defined in the query expression of a business layer object with the @Prompt function.

Related Information

Creating a New Prompt to Filter Data [page 68]
Using an Existing Prompt to Filter Data [page 69]
7.2.3.2.1 Creating a New Prompt to Filter Data

**Prerequisites**

This procedure assumes you are creating a business filter in the Edit Query Specification... dialog box (also called Query Panel).

**Procedure**

1. Drag the object you want to filter with a prompt and drop it onto the Query Filters pane. The query filter appears in outline in the Query Filters pane. The outline shows the filtered object, the operator and the type of filter applied to the object. (By default the filter is a constant.)
2. Select the filter operator from the list.

   **Note**

   The list of available operators depends on the type of filtered object.

3. Click the arrow at the right of the outline query filter and select Prompt from the menu to filter the object using a prompt. The Edit Prompt dialog box appears and the New Parameter option is selected by default.
4. Edit the prompt question in the Prompt Text box.
5. Select Prompt with List of Values to allow the user to select from a list of values when answering the prompt. The option is only available if the filtered object has an associated list of values in the universe.
6. Select Select only from list to restrict the user choice to values from the list of values.

   You can select this option only if the Prompt with List of Values option is selected.

7. Select Keep last values if you want the prompt to propose the last value that the user selected on the previous refresh. The first time the query is run, the default value (if set) is proposed.
8. Select Optional prompt to make the prompt optional. If the user does not supply a value for an optional prompt, the prompt is ignored.
9. Select Set default values if you want the prompt to propose values by default when it displays.

   a. To enter or select the default values, click Edit.
   b. If the filter object has an associated list of values, select the default values from the list.
   c. If the filter object has no associated list of values, enter default values.
   d. Click OK to save the default values.
10. Click OK to save the new prompt definition.
7.2.3.2.2 Using an Existing Prompt to Filter Data

Prerequisites

This procedure assumes you are creating a business filter in the Edit Query Specification... dialog box (Query Panel).

Procedure

1. Drag the object on which you want to apply a prompt and drop it onto the Query Filters pane. The query filter appears in outline in the Query Filters pane.
2. Select the filter operator from the list.
3. Click the arrow at the right of the Query Filter and select Prompt from the menu.
4. In the Edit Prompt dialog box, select the Use Universe Parameter option.
5. Select an existing parameter.
   The list displays only those universe prompts that are compatible with the object you are filtering. For example, the filtered object and the universe prompt must have the same data type.
6. Click OK to save the prompt definition.
7.3 Universe Access in Design Studio

Supported Platform Release

SAP BusinessObjects Design Studio only supports universe access for BI Platform 4.1.

Supported Universe Types

With SAP BusinessObjects Design Studio, you can use universes that meet the following prerequisites:

- relational universes
- single-source universes

General Workflow

Select a universe as a connection in the Add Data Source dialog box, define the query using the query panel in the Edit Query Specification dialog box and edit the initial layout of the query by using the Edit Initial View... dialog box.

The query definition in the query panel determines the data to be fetched using the universe. After the initial fetch, the user works with a local copy of the data held in memory for this session and the data is mapped to an OLAP-like result set. The initial fetch also determines the basis for needed projection functions (see the measures and the maximum result set sections below). The initial layout displayed is defined so that a measure dimension with all measures is created and moved to the column axis. All dimensions and attributes are moved to the free axis. You can change the initial layout with the initial view editor, but the standard restrictions of Design Studio with regards to layouts apply.

Business Layer

- General business objects settings
  You can define the access level for every object in the universe. The access level specifies which objects the user can work with. Every object can have an associated list of values. These values are used for prompts (see the prompts section below). Restrictions on whether the object can be used in results/conditions/sort are only relevant for the query definition itself. They have no effect on how the objects can be used in the initial view editor or when executing an analysis application.
- Attributes
  Attributes are interpreted in the same way as dimension objects. This allows you to use attributes regardless of their dimensions in the resulting application.
- Measures
The user can define projection functions for measures. These projection functions are also used by default when navigating/aggregating in the OLAP-like result set after the initial fetch. The following projection functions are supported:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>Data is aggregated with average</td>
</tr>
<tr>
<td>Count</td>
<td>Number of occurrences is calculated</td>
</tr>
<tr>
<td>Max</td>
<td>Data is aggregated with maximum</td>
</tr>
<tr>
<td>Min</td>
<td>Data is aggregated with minimum</td>
</tr>
<tr>
<td>None</td>
<td>No aggregation available; if data needs to be aggregated, no value is displayed</td>
</tr>
<tr>
<td>Sum</td>
<td>Data is aggregated with summation</td>
</tr>
</tbody>
</table>

- **Value help (list of values)**
  Value helps defined in the universe are only used in the prompt context in Design Studio. They have no effect on other areas where member listing is performed. Only keys are supported in the value help. For example, if you have defined an LOV based on a query or static list with multiple columns in the business layer, only the key column is used for displaying the list of values.

- **Supported data types**
  Fields with the following data types are supported for usage in Design Studio:
  - Date
  - DateTime
  - LongText
  - Numeric
  - String

- **Format handling**
  Formats defined in the business layer for a business object are not supported. The formatting of objects depends on the viewing locale and the data type exposed by the universe for example, date formats, numeric formats. Timestamp data is displayed in the context of the user’s timezone. For example, if you have defined a year using numeric as the data type, the year might be displayed as 2.013 ("." is the thousands separator). To avoid this, you can change the data type to String in your business layer.

- **Navigation path**
  Navigation path definitions are not interpreted.

### Query specification

You cannot change query properties. This includes settings like Retrieve duplicate rows, predefining contexts and the order of prompts.

The system retains the sorting definitions from the query when possible. However, in some cases, the definitions can be edited using the initial view editor, in order to guarantee the required sorting.
Execution

- Prompts
  Prompts in Design Studio can appear in different situations:
  - A new parameter is created in the query specification (query filter)
  - A universe parameter is used in the query specification (query filter)
  - In the query specification, you use a filter object, which contains a prompt
  - A business layer object contains a prompt
  - The data foundation contains a prompt
  In addition, prompts can occur in the case of contexts and a value help needs a prompt (see value help chapter above)
  A prompt can have an associated list of values. You select a value from a value help. If you set the property \textit{Select only from list}, values that do not exist in the list of values are not allowed. The property \textit{Keep last value} has no effect; the last value is always kept - as in the Design Studio session. You can set a prompt to optional. If the functionality of the @prompt business layer is used, you can define more settings in the prompt. Here only the selection modes Mono, Multi, and Leaf are allowed.

- Maximum Result Set
  The result set has a fixed limit of 5000 rows or 50000 data cells. If either limit is exceeded, no data is displayed.

- Filter display
  In Design Studio you can display the static filter values with script functions. However, only filter selections that produce a Cartesian product are displayed.

7.4 Selecting Data from a Data Source for Charts and SDK Extensions

Prerequisites

You have added chart or SDK extensions to your application and a data source and assign the data source to the charts or SDK extensions.

\begin{itemize}
  \item \textbf{Note}
    The data selection for one data source is only available for charts and SDK extensions.
\end{itemize}

Context

You can assign multiple charts or SDK extensions to a single data source. Each chart (or SDK extension) can display a different subset of the data contained in a data source, which holds the superset of the displayed data.
For example, a data source could contain data for multiple regions (Europe, America, etc). You can then use different charts; each chart displays the data from a different region. To do this, you use the Data Selection property of the chart component. This also allows you to reduce the number of data sources in an application and thereby improve the application performance.

**Procedure**

1. Choose **Data Selection** in the component's properties sheet. The Select Data from ... dialog box appears where the result set of the data source is displayed.

2. Specify your selection(s).
   - for charts: Select the rows and columns from the result set you want to be displayed in your component and choose **Add Selection**. Your selection is now active.
   - for SDK extensions: The options of data selections offered in the dialog box depend on the used type of data-bound properties. Make your selection(s) to activate it(them).

   **Note**

   Several types of data-bound properties allow you to restrict the selection of data values from a result set. Data-bound property types also help the SDK framework to check the feasibility of your selection and restrict the available selections in the Select Data from... dialog box. The following data-bound property types are available:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Data Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultCell</td>
<td>A single data value</td>
</tr>
<tr>
<td>ResultCellList</td>
<td>A single row or column of data values</td>
</tr>
<tr>
<td>ResultCellSet</td>
<td>A complex selection of data values from rows and columns (a Cartesian selection)</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>For the ResultCellSet property type, there are limitations of selection options in the Select Data from... dialog box.</td>
</tr>
<tr>
<td>ResultSet</td>
<td>All data values of the result set</td>
</tr>
</tbody>
</table>


3. optional: If you want to delete your selection(s), choose the selection in the Active Selections area and choose **Delete**.

4. Click **OK**, to close the Select Data from... dialog box. The selected data is now displayed in the component.
7.5 Deleting a Data Source

Procedure

1. Choose the data source you want to delete in the Data Sources folder of the Outline view.
2. Right-click the chosen data source, and click Delete.
   The system removes the corresponding data source in the Data Source folder.

Results

You have deleted a data source. The system automatically removes this data source in all components that use the data source for data binding in the corresponding application. Other applications that use the same data source are not affected. If you want to use this data source again, click the Undo symbol in the toolbar.

7.6 Inserting a Copied Data source from SAP BusinessObjects Analysis, Edition for Microsoft Office

Prerequisites

To use a data source created in SAP Business Objects Analysis, edition for Microsoft Office, the following prerequisites apply:

- Make sure that you have installed SAP BusinessObjects Analysis, edition for Microsoft Office and SAP BusinessObjects Design Studio on your machine.
- Make sure that you use both tools in either the local mode or with a BI platform. Otherwise you cannot insert an exported data source.

Context

The edition for Microsoft Office enables you to create data sources using methods that are not available in the design studio. You can use calculations, conditional formatting and exceptions to change a data source for example. You may then want to use this changed data source in the design studio.
Procedure

1. In the edition for Microsoft Office: Once you have changed and saved the data source, right-click Smart Copy in the table. The data source is copied to the clipboard.

2. In the design studio, there are two ways to insert this data source:
   - Right-click the Data Sources folder (or any data source listed in this folder) in the Outline view and click Smart Paste. The data source in your clipboard is inserted into the Data Source folder. You can now assign this data source to components in the editor.
   - Right-click the Layout folder (or any element listed in this folder) in the Outline view and click Smart Paste. The data source in your clipboard is automatically inserted into a crosstab component and displayed in the editor.
The Initial View dialog box allows you to modify the initial state of any data source (BW, SAP HANA,...). If you have added a data source to your application, choose Edit Initial View or Reset Initial View (if you have already changed the initial view of the data source) in the context menu of the data source alias in the Outline view. In this dialog box, you can change the following data in the initial state of a data source:

- dimensions in the rows/columns and the order of dimensions
- measures displayed in the result set
- dimensions in the background filter
- display of total rows (above the members) in the rows axis and display of total columns (left of the members) in the column axis

For each measure you can change:

- the number of decimal places displayed
- the scaling factor used
- how totals are calculated
- the sorting order

**Note**

This note is relevant for charts only. The number of decimal places displayed is generated based on the value in the first cell of each measure. If the first cell of a measure is empty, no decimal format can be generated for that measure in the Edit Initial View dialog box. To apply formatting to your measures in this case, you should use the format options available in the chart area Additional Properties tab. You can select format options from the Format dropdown list within the Data Label, Tooltip, X-Axis and Y-Axis additional properties.

For each dimension you can change:

- the additional result set attributes displayed
- the active hierarchy (if available) and its initial expansion level
- the member to be filtered (filter members)
- the filter by input string
- the member presentation (text, key, text/key, key/text)
- the totals display mode (show, hide, hide if only one member)
- the sorting type (by member, by attribute, by hierarchy)

**Note**

The options only appear if you have added the attributes or activated existing hierarchies for the dimension, for example.

For each data source you can change the following global settings by expanding the Global Data Source Settings tile:
• display of negative values
  ○ -X
  ○ X-
  ○ (X)
• display of zero values
  ○ default (with currency and unit)
  ○ without currency/unit
  ○ custom
  If you choose this option, you can change the display of zero values according to your needs.

⚠️ Caution
After you have modified the initial state of a data source using the Initial View dialog box, changes on the data source in the backend system will either result in runtime crashes, or the changes to the query might not be visible in your analysis application. It is also generally not possible to detect that changes in the backend have been made. The only solution to this is to reset the initial view (in the context menu of the data source in the Outline view) and recreate it from scratch.

Live preview

You can see the effects of your changes in a live preview in a crosstab. The view is refreshed after each change.

Pause refresh

If several small changes need to be made at the same time, you can pause all browser refreshes using the checkbox Pause Refresh in the upper right corner of the dialog box. Reactivate rendering once all changes have been made.

Interaction within the dialog box

You can drag and drop the elements onto the trees on the left (structure, columns, rows, background filter), but not onto the live preview. You can also use the context menu for each element.

Leaving the dialog box

If you want to leave the dialog box, you can:
• Click Cancel to discard your changes.
• Click OK to apply all modifications to the data source in your application.
Click **OK + Create Crosstab** to create a new crosstab component with the modified data source assigned.

### Reset the initial view

If you have already modified the initial state of a data source using the *Edit Initial View*...dialog box, the context menu command *Reset Initial View* is enabled. When you select this command, all modifications made to the original state of the data source are reset. This means BW queries are displayed as specified in the BW system/Query Designer. With an SAP HANA data source, the initial view will contain highly aggregated measures, with no dimensions in the rows or columns.

### Related Information

[Chart Area Additional Properties](page 273)
Assigning a Data Source to a Component

Prerequisites

Before assigning a data source to a component, make sure you have completed the following steps:

- You have added at least one data source to your application. The data source is displayed in the Data Sources folder of the Outline view.
- You have added at least one component for displaying data (such as crosstab or chart) to your application.

Context

By assigning a data source to a component you supply the component with data.

Procedure

There are several ways to assign a data source to a component:

- Work in the Outline view: Drag the data source from the Data Sources folder and drop it onto the corresponding component in the Layout folder.
- Drag the data source from the Data Sources folder and drop it onto the corresponding component in the design area of the tool.

Note

If you have added a data source by clicking the Data Source property of the component, you do not need to assign the data source to the component in a further step. In this case the system assigns the data source automatically to the component.

Results

You have assigned a data source to a component. The component displays the data of the data source in the layout editor.

Note

Switch to the Data Binding view in the Outline view to see which components have been assigned to a specific data source.
10  Binding the Properties of Standard Components to Data Sources

Data-bound components (also referred to as analytic components) - like crosstabs, charts or filter components and many SDK components - have a Data Source property that points to a data source. Some of these components also have one or more Data Selection properties that describe the subset of data to be used.

Unlike analytic components, basic components do not have a data source property. If you want basic components to display data from a data source, this must be set by scripting. While data-bound components are updated automatically when data is changed, it might be significantly more difficult to synchronize basic components in the same way. In this case, you have to know all the relevant events where a script has to update the basic component’s properties.

An easier way to achieve this is to bind a property of a basic component in your application to a data source, by using the property-binding function at the top of the Properties view of this component. In this way, you can create analysis applications without using scripts.

In the following chapters, you will find workflows for using the property-binding function:

- Displaying Cell Values
- Configuring a List Box for Filtering Data
- Configuring an SDK Component to Use Multiple Data Sources

Related Information

Displaying Cell Values [page 80]
Configuring a List Box for Filtering Data [page 82]
Configuring an SDK Component to Use Multiple Data Sources [page 82]

10.1  Displaying Cell Values

Context

This workflow describes how to bind a data cell value to a property, in order to display a result set cell value in a text box (component Text), for example. The effect is similar to calling datasource.getData(); in a script.
Procedure

1. Add a basic component, for example the Text component, to your application.
2. Choose a simple property, for example Text in the Property view, for this component.
3. Click on the Edit Property Binding... button at the top of the property view. A dropdown menu is displayed with the selected entry No input binding.
4. Choose Data Cell Binding. The Select Data dialog box is displayed.
5. (optional) If there is more than one data source in your application, choose the data source that you want to bind from in the Data Source dropdown box.
6. In the displayed table view of your data source, click the data cell you want to bind to the property and click OK.
7. The binding property (for example the Text property of the Text component) of the selected component changes to <bound>. If the value does not fit your needs, proceed using a formatter function (step 8 to 12).
8. (optional) Ensure that you have at least one Global Scripts Object in your application. If this is not the case, create one of these objects in the Technical Components folder in the Outline view (right-click Technical Components and choose Create Child Global Scripts Object).
9. (optional) The property you have bound to a data cell can be expanded in the Property view. Expand the property until you see the data source and data selection binding. This way you can see the bound data source and data cell value (data selection) and the Formatter Function.
10. Click on the dropdown box of the Formatter Function and choose Add. The Create Script Function dialog box is displayed.
11. Choose the Global Scripts Object (if you have more than one in the application) and type in a name for the script function. Choose OK.
12. A dialog box is displayed. Write a function body using the given arguments value and formattedValue, for example, return value / 1000 + " \\

The effects of the formatter function are not displayed in the editor of the design tool and are only displayed at runtime.

Note

You can only use those global script functions as formatter functions which were created in the property view as described in step 10. Functions created in this way are marked as having a read-only signature. This means the user cannot add, remove or change the input parameters, which the property-binding function relies on. Global script functions created via the Outline view cannot be used as formatter functions.
10.2 Configuring a List Box for Filtering Data

Context

This workflow describes how to bind a List Box, Dropdown Box, Checkbox Group or Radio Button Group as a filter for a dimension. This is similar to calling `dataSource.getMembers();/getMemberList();` and `dataSource.setFilter();` in some scripts.

Procedure

1. Add a list component (for example, a List Box component) to your application.
2. In the Property view of this component, choose the Items property.
3. Click on the Edit Property Binding... button at the top of the property view. A dropdown menu is displayed that contains two entries: No input binding (selected) and No Output Binding (not selected).
5. Expand the Item property until you see the members binding property.
6. Set the Data Source and Dimension properties as required, as well as all other relevant properties. The chosen dimension members are displayed in the list box component in the editor. To configure filtering, proceed with the following steps.
7. Click again on the Edit Property Binding... button at the top of the property view and choose Filter Binding.
8. Expand the Item property until you see filter binding properties under Output Binding.
9. Choose at least one target data source. It can be the same as the data source you used to fill the list box.
10. Choose a dimension to filter. Usually this will be the same dimension as you used to fill the list box.
11. Execute the application. Choosing one or more entries in the list box will automatically filter the visualization component (for example, the crosstab).

10.3 Configuring an SDK Component to Use Multiple Data Sources

Context

This workflow describes how an SDK component receives data from multiple data sources. Note that this needs to be explicitly supported by an SDK component.
Procedure

1. Add an SDK component or SDK data source that has multiple data-bound properties to your application.
2. If the component has a Data Source property, it can be set as default, but it can also remain set to <None>.
3. Choose a data-bound property and click on the on the Edit Property Binding... button at the top of the property view. The data selection dialog box is displayed.
4. Choose the data source and the data range.
5. Repeat these steps for other data-bound properties.
11 Saving an Application

Prerequisites
You have created a new application or changed an existing one.

Context
You want to permanently save the changes you have made in a new/existing application.

Procedure
Click Application > Save. The system saves your application using the name you specified when you created it.
If you are working on several applications in different editors and you want to save all the changes you have made in one go, click Application > Save all.

Next Steps
If you want to save the application using a different name, use the save as function. For more information, see “Saving an application using a different name” in the Related Topics section.

Tip
While you are working on your application, you can always execute it without needing to save it first. If you want to close your new/changed application, you have to save it first.

Related Information
Executing an Application [page 562]
Saving an Application Using a Different Name [page 561]
12 Working with Templates

SAP BusinessObjects Design Studio offers you a set of predefined templates that serve various design and business needs. When you choose a predefined template, the system automatically creates a copy of this template. You can change the copy according to your needs.

Besides, you can also create user-defined templates and make them available as templates for other application designers. For more information, see Exporting Applications As Templates [page 240].

Depending on your choice of target device (desktop browser, iPhone or iPad) in the New Application dialog box, you can choose from the following templates:

- for desktop browsers
  - Ad-hoc Analysis Template
    This template is used for ad-hoc slicing & dicing of data sources.
  - Basic Analysis Template
    This template is a desktop template with a crosstab, chart, filter area and tabstrip.
  - Data Discovery and Visualization Template
    This template is used for online self-service data discovery and visualization.
  - Generic Analysis Template
    This template is used for generic slicing and dicing of data sources.
  - KPI Dashboard Template
    This is a template for a dashboard with an overview page containing KPI tiles and a KPI details page.
  - KPI Details Template
    This template is for a KPI details page.
  - Online Composition Template
    This template is used to create online custom dashboards using portable fragment bookmarks.

- for iPhones
  - iPhone Template 1
    This is an iPhone template with tabstrip navigation.
  - iPhone Template 2
    This is an iPhone template with swipe navigation.

- for iPads
  - iPad Template 1
    This is an iPad template with swipe and tap navigation
  - iPad Template 2
    This is an iPad template based on iPad Template 1 plus a filter area.
  - iPad Template 3
    This is an iPad template based on iPad Template 1 plus a settings popup.
  - Ad-hoc Analysis Template
    This template is used for ad-hoc slicing & dicing of data sources.
  - KPI Dashboard Template
    This is a template for a dashboard with an overview page containing KPI tiles and a KPI details page.
  - KPI Details Template
    This template is for a KPI details page.
Prerequisites

For the templates Ad-Hoc Analysis, KPI Dashboard and KPI Details specific prerequisites must be met for the CSS files and images used in the templates:

Deployment on SAP NetWeaver

Before application designers can use the predefined templates in the design tool and make their generated analysis applications available on SAP NetWeaver, specific template files need to be uploaded to the MIME repository of SAP NetWeaver BW. For more information, see the chapter “Uploading Template Files to the MIME Repository” in the Administrator Guide: SAP BusinessObjects Design Studio on SAP Help Portal at http://help.sap.com/boad.

Deployment on the BI platform

No tasks need to be performed to use templates on the BI platform.

Deployment on SAP HANA

No tasks need to be performed to use templates on SAP HANA. However, the Ad-hoc Analysis Template is not available for deployment on SAP HANA.

Initialization button

To simplify the enhancement and the design of the predefined templates Ad-Hoc Analysis, KPI Dashboard and KPI Details, there is a hidden Initialization Button (BUTTON_INIT_BUTTONS) for each of the templates in the Outline view of the design tool.

Many scripts of the template components refer to the Initialization Button and therefore need much shorter scripts. Each time a Click Event uses the Initialization Button method, the full script of the Initialization Button is run. This script contains CSS class information, settings and ifelse statements for the whole template (header, footer, table and buttons).
12.1 Using the Ad-Hoc Analysis Template

The Ad-hoc Analysis Template is a predefined template used for ad-hoc slicing and dicing of data sources. It contains one crosstab and one chart by default. The Ad-hoc Analysis Template has the following characteristics and advantages:

- It can be used as an application by using URL string and URL parameters:
  
  [...]&XQUERY=<NAMEOFQUERY>&XSYSTEM=<NAMEOFSYSTEM&XTYPE>=BW

- You can easily adapt and extend the template by using a data source of your choice (simply replace the dummy data source with your data source) and by copying components and elements.

- It has predefined buttons that allow you to change the view on your data and to configure technical settings.

- The template is based on the SAP Fiori style theme SAP Blue Crystal.

Using the header

On the left side of the Ad-Hoc Analysis application header, you will find the button for the Navigation panel. Click on this button to change the data view. You can then select different dimensions or key figures. Click on the OK button to close the navigation panel.

If you click the first button on the right side of the header, the Filter dialog box is displayed, where you can allow or restrict different values.

By clicking the Bookmark button, you can save your favorite views. This button works in local mode and if the Design Studio is deployed on the BI Platform.

If you click on the Refresh button, the system refreshes the data from the back end.

Using the crosstab view

The crosstab view contains predefined buttons like Prompts, Hierarchies, Totals, Decimals and Scaling Factors. The Prompts and Hierarchies buttons are automatically activated if your data source contains variables and/or hierarchies. This means you can specify for example, whether you want to show an existing hierarchy or hide it. As the Decimals and Scaling Factors buttons are not displayed by default, you have to set the property Visible to true, in order to display them. There is also a predefined script for the On Click event of the Decimals and Scaling Factors buttons, in order to connect these buttons to the data source.

Using the personalize button

This button offers you additional functions like switching on/off the Analysis Mode for hierarchies, showing/hiding Conditional Formatting and display options for totals.
Using the footer

The *Settings* button on the left side of the footer contains additional information about the data source (for example, the technical query name and the date of the last update). You can also set your current view as the initial view for the whole application or you can reset the initial view.

By clicking on the *Action* button on the right side of the footer, you can export the data to a Microsoft Excel or CSV file. You can also print the application and share a bookmark for the application using an Internet link.

Tips and tricks for working with the Ad-Hoc Analysis Template

- Bookmarking works for local mode and deployment on the BI Platform.
- If the data source contains hierarchies, be careful when making modifications in the initial view.
- Apart from the restrictions for certain functions described above, this template works in general when deployed on the BI Platform and in SAP NetWeaver.

12.2 Using the Data Discovery and Visualization Template

The *Data Discovery and Visualization Template* is used for online self-service data discovery and visualization based on SAP BW and SAP HANA data sources.

Data enters the application as a dataset selected at runtime and different chart types are selected to create a collection of visualizations based on the chosen dataset. You can create a dashboard that consists of visualizations associated with different datasets, offering a very rich graphical representation of your data. A WYSIWYG (What You See Is What You Get) version of the currently displayed page of the dashboard can be exported to PDF and the entire application can be shared via email within an organization. The template consists of two pages called *Visualize* and *Compose*.

**i Note**

Before working with the *Data Discovery and Visualization Template*, you should refer to the following SAP Note:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2169104</td>
<td>Data Discovery and Visualization Template platform support in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
</tbody>
</table>

**Visualize Page**

The *Visualize* page contains the following components:

- **Chart Type Picker**
  The *Chart Type Picker* is a palette of chart types, from which you can select the visualizations you want to use in your dashboards. You can change your chart type at runtime and immediately view your data in a different
way. For more information on using the Chart Type Picker, you can refer to the chapter in this guide called Chart Type Picker within the Basic Components section.

- **Navigation Panel**
  Using the navigation panel, you can easily change how you drilldown into the data and see at a glance the navigational state of the data source at runtime. You can drag and drop dimensions from the Navigation Panel to the columns or rows area. By default all the measures are displayed in the visualization. To display only some of the measures, you must use the context menu to filter out the measures you do not need. For more information on using the Navigation Panel, you can refer to the chapter in this guide called Navigation Panel within the Analytic Components section.

- **Context Menu**
  The context menu offers you a variety of functions that help you create your visualizations and work with them efficiently at runtime, for example, enabling and expanding hierarchies. For more information on using the Context Menu, you can refer to the chapter in this guide called Using the Context Menu (Technical Component) within the Working with the Outline View section.

- **Filter Line**
  The Filter Line offers you a way to add, remove, view and edit filters and measures defined for your assigned data source. For more information on using the Filter Line, you can refer to the chapter in this guide called Filter Line within the Analytic Components section.

- **Layout Editor**
  The layout editor provides a What-You-See-Is-What-You-Get display, which enables you to view the working application during the creation process. As you change chart types or apply filters, your changes are immediately reflected in the Layout Editor.

- **Fragment Gallery**
  In the Visualize page, the Fragment Gallery is configured horizontally. It stores and displays portable fragment bookmarks or visualizations created by the user based on their selected dataset.

**Compose Page**

The Compose page contains the following components:

- **Fragment Gallery**
  In the Compose page, the Fragment Gallery is configured vertically. It stores the collection of portable fragment bookmarks or visualizations, created in the Visualize page. You can drag and drop these visualizations from the Fragment Gallery into the Split Cell to create your own customized dashboards.

- **Split Cell**
  The Split Cell enables you to place and arrange portable fragment bookmarks, dragged from the Fragment Gallery, in a tabular format.

**Selecting an Initial Data Source**

The Data Discovery and Visualization Template has no data source assigned by default. You can keep the template as it is, execute it and then assign a data source at runtime. To assign a data source, run the template application and select Recently Used Queries or Select a System. Recently Used Queries will display a list of all the queries you have already used to date. If there are no recently used queries listed, you can select the data source you want to assign to your template by using the Select a System panel. Once you have selected your data source from the
list, select Next. You can then select a query by searching or by selecting from a folder structure. Select OK. The Visualize page opens.

Changing a Data Source within an Application

It is possible to create dashboards online using data from multiple data sources. This offers you greater flexibility and insight into your data. The Fragment Gallery displays a collection of visualizations associated with the selected dataset. However, you can display a different collection of visualizations in the Fragment Gallery in either the Visualize or Compose pages.

If you select a different dataset from the dropdown list at the top of the Visualize or Compose pages, the collection of visualizations in the Fragment Gallery changes to display the visualizations associated with the newly selected dataset. You can then manipulate those visualizations as you choose, for example, changing chart types or adding filters. Every time you change your selection in the dropdown list you are then working on the visualizations based on a different dataset.

Creating Visualizations

When you drag and drop dimensions to the rows or columns area, the default visualization in the Layout Editor changes automatically to reflect your selection. By default all the measures are displayed in the visualization. To display only some of the measures, you must use the context menu to filter out the measures you do not need.

You can then select a different chart type from the Chart Type Picker and the visualization changes accordingly. To create another visualization, select the + icon in the Fragment Gallery, manipulate the dimensions and measures, and select a chart type. Your visualizations are saved automatically as portable fragment bookmarks and are stored in the Fragment Gallery. Images for the portable fragment bookmarks are automatically generated depending on the chart family type selected. Continue to add to your Fragment Gallery until you have created the number of visualizations required.

Types of Visualizations

<table>
<thead>
<tr>
<th>Type of Analysis</th>
<th>Description</th>
<th>Charts available</th>
</tr>
</thead>
<tbody>
<tr>
<td>comparison</td>
<td>Compares differences between values or shows a simple comparison of categorical divisions of measures. For example, use a bar chart to compare the differences in sales revenue between countries.</td>
<td>• bar chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• bar chart with 2 x-axes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• column chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• column chart with 2 y-axes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• stacked bar chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 100% stacked bar chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• stacked column chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 100% stacked column chart</td>
</tr>
</tbody>
</table>
### Type of Analysis

<table>
<thead>
<tr>
<th>Description</th>
<th>Charts available</th>
</tr>
</thead>
</table>
|             | ● combined bar line chart  
|             | ● combined bar line chart 2 x-axes |
| **trend**   | Shows a trend in the data values (especially for dimensions that are time-based, such as Year) or the progression of your data and possible patterns. For example, you can use a line chart to view sales revenue trends of a product throughout a range of years. | ● line chart  
|             | ● horizontal line chart  
|             | ● area chart  
|             | ● horizontal area chart  
|             | ● combined column line chart  
|             | ● line chart with 2 y-axes  
|             | ● combined column line chart with 2 y-axes |
| **percentage** | Shows the percentage of parts in a whole or values as ratios to a whole. The legend shows the percentage and the total values. For example, use a pie chart to see who had the highest sales as part of a total sales value directly: Total sales = $200, Paul had 10% ($20), David had 65% ($130), and Susan had 25% ($50) | ● pie chart  
|             | ● multiple pie chart |
| **correlation** | Shows the relationship between values or compares multiple measure values. For example, you can view the correlation of two measures and understand the impact of the first measure on the second measure. | ● scatter plot  
|             | ● bubble chart |
| **other charts** | Shows a selection of other chart types available | ● radar chart  
|             | ● multiple radar chart  
|             | ● waterfall chart  
|             | ● horizontal waterfall chart  
|             | ● stacked waterfall chart |
| **additional types** | Other types of visualizations apart from charts. | ● crosstab |

### Renaming a Visualization

Visualization titles are based on the query used to create them. To change the title of your visualization, select the cogwheel to the top right of the Layout Editor and select Rename. Select Restore Default Title to change the title back to the generated title. The generated title is based on the data source selection.
Show or Hide Title

To hide or show the title of your visualization, you must select the menu option Show/Hide Title from the cogwheel to the upper right of the Layout Editor.

Adding and Removing Filters

To add, remove, view and edit filters and measures defined for your assigned data source, select the + icon to the left of Add Filters. Select a dimension from the list in the dialog box. If a dimension contains a dimension hierarchy, select to expand the list to show all dimensions included in the hierarchy. The filters you select to apply to your data source will appear across the top of the Layout Editor. You can return to the dialog box to further modify your dimension filter, by selecting the applied filter from the Add Filters panel. You can remove a filter with the “X” icon beside the applied filter.

Open Prompt Dialog

You can select Open Prompt Dialog from the cogwheel in the top right-hand corner of the Layout Editor. It displays a list of required and optional variables. The values you select for the variables set filters that determine which data is included in a visualization. Multiple values can be selected when a variable allows for multiple values. The prompt values are saved automatically with the visualization.

Creating Custom Applications

When you have created the collection of portable fragment bookmarks or visualizations you require, you can select the Compose page at the top of the template. The Compose page allows you to create custom dashboards using portable fragment bookmarks that are stored in the Fragment Gallery. Select a title for your application and select Create. You can then drag and drop portable fragment bookmarks from your vertical Fragment Gallery onto the Split Cell to create your own custom dashboard. The application is saved automatically. To create a new dashboard, simply select New Story button and repeat the process to create as many applications as you wish.

Changing a Data Source within the Compose Page

If you select a different dataset from the dropdown list at the top of the Compose page, the collection of visualizations in the Fragment Gallery also changes to display the visualizations associated with the newly selected dataset. This allows you to then drag and drop into the Split Cell a combination of visualizations from different datasets.
Switching Between Dashboards

To switch between applications, you can select from the dropdown list that appears at the top of the Compose page.

Sharing a Dashboard

You can share your dashboard as a bookmark with other users by selecting the Share button. This allows you to do one of the following:

- Open the bookmark link directly in a separate browser by selecting the link icon.
- Select and copy the bookmark URL to their preferred browser. They can then use native browser functionality to add the URL to their favorites or share the bookmark by copying and pasting the URL into an email.
- Select the Send by email button to open an email directly from their current application browser. The subject field of the generated email is pre-populated with the title of both the application and the bookmark the user wants to share. The body of the email contains a copy of the actual bookmark link. The user can then edit the email as required and share a bookmark of their application by sending the email to the appropriate recipients.

Export to PDF

Select the Export to PDF button to export to PDF, a WYSIWYG version of the currently displayed page of your dashboard. You can customize the following details before exporting:

- Page size
- Orientation
- Header Title
- Footer Text

12.3 Using the Generic Analysis Template

The Generic Analysis Template is a predefined template used for slicing and dicing data sources. It contains one crosstab and one chart by default and is only available for desktop applications. A context menu is also available on the crosstab, which allows you to easily navigate and analyze your data.

The Generic Analysis Template has no data source assigned by default. You can keep the template as it is, execute it and then assign a data source at runtime by enhancing the application URL in the Web browser. This means the template remains generic and independent of any data sources. Enhance the application URL by entering the following data source URL parameter in the relevant form for the platform deployment of your design studio:

- for SAP NetWeaver: &XQUERY=<TechnicalNameOfYourQuery>
Example: \( \text{http://<applicationURL>&XQUERY=<queryID>} \)

For SAP NetWeaver deployment, we recommend that you use the Generic Analysis Template as template for RRI jump targets. For more information, see the chapter *Creating a Generic Analysis Template for RRI Jump Targets* under the section *Advanced Design Tasks* in this guide.

- for SAP BusinessObjects Platform:
  \( \text{http://<applicationURL>&XSYSTEM=cuid:<sysID>&XQUERY=<queryID>} \)

  For further information on the CUID, see the *Administrator Guide: SAP BusinessObjects Design Studio based on SAP BusinessObjects BI Platform* on the SAP Help Portal at \( \text{http://help.sap.com/boad} \).

You can also assign a data source to the template at design time when you open the template in your design studio. In this case, the assigned data source is automatically displayed when the template is executed.

### Switching the data visualization

To switch the visualization of your data, choose *Crosstab, Chart or Chart/Crosstab* in the right corner of the application.

### Changing the settings for the chart and/or the crosstab

You can display the settings for each visualization type by choosing the arrow icon (*Show Settings*). This enables you to swap the axes in your chart or crosstab, for example, or show any exceptions that may occur on your crosstab.

### Using the Navigation Display

By choosing the *Show Navigation* arrow icon, you can change the display of your data. This means you can change the display of the measures and/or dimension in rows or columns or see dimensions that are not in the drilldown (*FREE*).

**Note**

If you want to reverse a navigation step you have made in the application, you have the following options:

- Choose *Undo* (in the upper left corner of the application) to go back one navigation step; to the navigation state before you changed the application.
- Choose *Back to Start* to go back to the very initial state of the application.

### Using the filter bar

If you want to set a filter, choose *Filter (Q)* above the data visualization icons and choose *Add Filter*. You can set up to 12 dimension members as filter values. The number of dimension members you have included in the filter is...
displayed in brackets for each dimension. The number of dimension filters is displayed in brackets after Filter, for example Filter (2).

To revert to the navigation state and drilldown state before the filter was applied, choose Reset in the filter bar.

**Using prompts and displaying information**

If your data source is designed for setting prompts, click on Prompts in the left corner of the application to set the prompts.

Choose Info in the left corner of the application to display information about your application. Here you will see for example, the technical name of your query, the prompts and filters you have set.

**Using the Actions button**

In the bottom right corner of the application, you will find the Actions button, where you can use the Export to Microsoft Excel, the Export to CSV or the Print function.

**Related Information**

Creating a Generic Analysis Template for RRI Jump Targets [page 235]

### 12.4 Using the KPI Dashboard and KPI Details Templates

The KPI Dashboard Template is a template for a dashboard with an overview page containing KPI tiles and a KPI details page. It contains six tiles, each showing a chart and one KPI value by default. In the header and footer of this template you will find the same functions as in the Ad-hoc Analysis Template. There is also a back button in the header that can be activated. As the KPI Dashboard template is built on the pagebook concept, it is highly recommended for use on mobile devices.

The headings and charts are ready to be filled with your text and your selected data source. Change or exchange the given components (for example, exchange a chart for a crosstab) in the tiles and use/fill the predefined scripts for user interaction. You can adjust the template as required.

If the user clicks on one of the charts at runtime, the KPI Details page (template) is displayed. This template is a predefined template showing details of the KPI Dashboard, but it can also be selected and executed as a standalone template in the New Application dialog box. It contains a crosstab and buttons for choosing different chart types.
12.5 Using the Online Composition Template

The Online Composition Template is a predefined template used to allow end users to create dashboards using portable fragment bookmarks.

The template comprises two pages. The first page contains a grid component that represents the saved state of the second page in the form of tiles and a menu. The second page contains the following:

- split cell
- fragment gallery
- drop down list which is connected to the bookmark folders
- editable text box
- menu

The Online Composition Template has no data source assigned by default. It uses the data sources stored with the bookmarks and can read them dynamically when the user runs the application.

Creating a Composition

To create your virtual dashboard select the Create menu item at the bottom of the first page. This brings you to the second page, where you can drag and drop the required fragments from the Fragment Gallery onto the Split Cell. When you select Save, you will see your application represented by a tile on the first page.

Sharing your Dashboard

The template includes a way for users to share their newly created dashboards with colleagues. To share a dashboard simply select a tile and select the Share menu option on the first page. This opens a new window that includes a generated URL for you to share and a menu item allowing you to generate an email containing the URL.

Opening a Dashboard

To open a dashboard in read mode, select a dashboard tile that you have already created and select Open. This will open the dashboard in read mode.

Deleting a Dashboard

To delete a dashboard select one from your available tiles on the first page and then select the menu option Delete.
Menu Items

- **Save As**
  Allows you to save changes to an existing dashboard.

- **Save**
  Allows you to save changes to an existing dashboard.

- **Refresh Gallery**
  Refreshes the Fragment Gallery to make sure that you are looking at the most up to date list of fragment bookmarks.

- **Cancel**
  Reverts you back to read mode.

- **Edit**
  Enables you to edit your online composition.

- **Back**
  Reverts you back to the first page.

Filtering the Fragment Gallery

You can populate the Fragment Gallery with different bookmarks by using the dropdown list of bookmark folders above the gallery. Selecting a folder from this list updates the Fragment Gallery with the bookmarks available to you from that particular folder. This gives you a different selection of bookmarks from which to choose, to create your online composition.

Exporting to PDF

Select the Export to PDF button to export to PDF, a WYSIWYG version of the currently displayed page of your dashboard. You can customize the following details before exporting:

- **Page size**
- **Orientation**
- **Header Title**
- **Footer Text**

Related Information

Working with Online Composition [page 145]
Working with Fragment Gallery Components [page 148]
Working with Split Cell Containers [page 150]
12.6 Using the Planning Template

Context

The Planning Template is a predefined template used for planning business data. Besides the functions of the Generic Analysis Template, the Planning template also has the following functions designed for planning applications:

- Display
- Back to Saved State
- Planning Functions
- Refresh
- Save

The Planning Template cannot be used as a standard analysis template but has to be modified. As an application designer you need to perform the following steps after you have chosen the Planning Template in the New Application dialog box.

Procedure

1. Assign a data source to the application using a planning query.
2. Define a planning connection in the Properties view under Planning. The planning connection specifies the backend system, which the planning application refers to.
3. Depending to your planning scenario, add one or more planning functions to the application. For a Sales Revenue Planning application, you can add functions such as Copy Actuals to Plan, Delete Plan Data and Reevaluate Plan Data by XY Percent. To do this, choose Planning Objects Add Planning Function... in the Outline view of the design studio and type in or search for the planning function you need. For each planning function, add a Script Alias in the Add Planning Function dialog box. For example, PF_COPY.
4. Now you can add the planning functions to the corresponding UI elements (Planning Functions). In the planning template, the planning functions are bound to buttons on the UI. This means that the planning functions you add are executed when the application user clicks on the buttons. In the design tool, you will find the corresponding buttons under the PANEL_PLANNING_FUNCTION section in the Outline view:
   a. Change the names of the planning function buttons by editing the Text property of each button, for example Copy Initial Data for the Copy planning function you have chosen in step 3.
   b. Change the On Click event of each button as required by using the relevant method for the planning function. For example, for the Copy Initial Data button and the Delete button, you can add scripts like these:

```javascript
PF_COPY.copyFilters(DS_1)
PF_COPY.execute();
```

or
PF_DELETE.execute();

If you do not know the planning function script alias, press **CTRL + Space** to see the list of all components and elements that can be used for the script.

5. If you do not use all planning function buttons, delete the functions that you do not use in the **Outline** view.

You have made the necessary changes in the template. You can now run the application.

6. Save your changes and execute the planning template. The planning application is displayed in the planning mode by default. Click on **Display** if you want to view the data in read mode. If you want to plan data, switch back to the **Plan** mode.

7. Optional: In the planning mode, you can perform manual planning actions such as bottom up and top down planning by entering the relevant data in the input-ready cells. To see the changes, click on **Refresh** (recalculate). Click on **Save** if you want to save the changed data in the backend.

8. To see the planning functions, click on **Planning Functions**. Use the functions as required.

### Results

You have changed the planning template according to your business needs and executed it in the Web browser.

**Note**

Keep the following points in mind:

- In the display mode, the planning template behaves like the generic analysis template. This enables the application user to use the filter panel and analyze data and perform drilldowns.
- Planning functions (or manual planning) can only be executed with a defined drilldown of dimensions.
- **Export to Microsoft Excel** and **Export to CSV** in the **Actions** menu in the bottom right corner of the application is only enabled in the display mode and is only available for saved data.
13 Executing an Application

Context

In the design tool, there are several ways to execute an application:

- Choose **Application ➤ Execute Locally**. The application is displayed in a separate Web browser window, using a local Web server embedded within the design tool. This allows you to check the changes you made to the application before saving.

- Choose **Application ➤ Execute on BI Platform**. The application is displayed in a separate Web browser window, using the document link of the application and the current session of the BI platform. You do not have to log the session to the BI platform when executing the application on the BI platform. The analysis application is executed with the credentials you provided when logging on to the design tool.

  **Note**
  
  If you have made unsaved changes to your application and you execute the application on the BI platform, the system informs you that the currently persisted (and probably outdated) version of the application will be displayed. You can still save your changes before executing the application.

- Choose **Application ➤ Execute on SAP NetWeaver**. The application is displayed in a separate Web browser window. You are prompted to log on to SAP NetWeaver Portal.

- Choose **Application ➤ Execute on SAP HANA**. The application is displayed in a separate Web Browser window. You are prompted to log on to the SAP HANA system again.
14 Executing an Application on a Mobile Device

Procedure

1. Click *Send to Mobile Device (using QR code)*. The dialog box *QR Code* is displayed with the URL of the current application encoded.

2. If there are several network adapters active for your computer, the dropdown box *IP address to use* is displayed. Select the IP address you need.

3. Scan the QR code with a mobile device, for example an iPad (iPad 2 or higher), using one of the various QR code scanner apps.
   - When you click on the QR code, a new Web browser window with the document link opens, but without the current session of the platform. You have to log on to the platform. This is useful when you want to execute the application with different credentials, for example, to check access rights or personalized result sets.
   - To copy the document link to the clipboard, click the *Copy URL to clipboard* button. You can paste the URL into an e-mail, send the e-mail and the URL can be opened on a mobile device. If you want to add this URL to your bookmarks, you need to do this manually (and not by choosing the corresponding button on the mobile device).
15  Recording Applications for Offline Use

Application users can record any application workflow for offline use later on by pressing a key combination in the executed application. In general, the displayed state after each interaction step is saved during recording. After recording the application user can download the recorded steps as a .ZIP file and replay the interaction steps. During playback, it is possible to move forward or backward through the set of saved steps. Apart from moving through the steps, no other type of interaction is possible during playback.

Prerequisites

Recording applications is possible with a design studio deployment on SAP NetWeaver, the BI platform or in the local mode. If you want to record applications, note the following points:

- The maximum number of recorded steps per application is 100. Once this limit is reached, the application user will be warned, and no further steps will be recorded. It will still be possible to download the currently recorded steps in this case.
- The Prompt dialog box as well as the context menu itself will not be shown during playback; any interaction steps containing the prompts dialog box are skipped. This means the playback skips from the step before the variable dialog box is shown to the step immediately after the dialog box is submitted or canceled.

Enabling recording

The application user can press the key combination $\text{Ctrl} + \text{Alt} + R$ to enable recording. Once the recording mode has been started, a control bar is available at the bottom of the application window. It opens when the user hovers with the mouse at the bottom of the window.

The control bar contains the following buttons (from left to right shown as in the picture above):

- Reset: Deletes all steps captured so far, but does not switch recording on or off.
- Toggle Button Recording Active: switches recording on and off. This icon changes appearance depending on whether recording is active. If recording is currently active, the icon is displayed as a pause button (like in a media player). If recording is currently inactive, the icon is displayed as a record button. This toggle button allows the user to skip steps, which are not required in the recording. When the user presses the pause button, the step immediately before the pause button was pressed is included in the recording, but the pausing step itself is not. When the user presses the resume button, the resuming step is included in the recording.
- Stop: Turns off the recording feature altogether. This also removes the recording control bar and clears all recorded steps.
• Download: Downloads the recorded steps as a .ZIP file. This opens a browser download window. After the file has been downloaded, the set of captured steps is reset (the result is the same as clicking the Reset button).

Opening the downloaded .ZIP playback file

The downloaded .ZIP file has the same name as the application. In order to start playback, the application user has to unzip the .ZIP file and open the file `index.html` contained in the ZIP file. All other files should not be directly accessed by the application user.

Controlling playback

The playback control bar contains three buttons and a page/pages indicator:

From left to right as shown in the picture above:
• Go to first step: Displays the first recorded step.
• Previous step: Displays the step before the currently displayed step.
• Next Step: Displays the step after the currently displayed step.
• Current / Total page indicator
# Printing an Analysis Application to a Browser

## Prerequisites

You have created an application and you have configured the settings for printing a copy of your application from your preferred browser. Calling `APPLICATION.print();` and printing produces a WYSIWYG (What You See Is What You Get) version of the analysis application. This may cause truncation depending on the dimensions of the application. The following table describes the print settings (for different browsers) that need to be applied, in order to ensure that the whole application is printed.

<table>
<thead>
<tr>
<th>Mode</th>
<th>IE(9)</th>
<th>Chrome</th>
<th>Firefox</th>
<th>Safari</th>
</tr>
</thead>
<tbody>
<tr>
<td>local</td>
<td>1. Choose Settings Print Print preview</td>
<td>Open the <code>Print</code> dialog box by calling <code>APPLICATION.print()</code>:</td>
<td>1. Choose Menu Print Print preview</td>
<td>Open the <code>Print</code> dialog box by calling <code>APPLICATION.print()</code></td>
</tr>
<tr>
<td></td>
<td>2. See row of icons below <code>Print preview</code> dialog box title.</td>
<td>1. In the Layout Section, select portrait or landscape as required.</td>
<td>2. Choose Page setup Format &amp; Options tab.</td>
<td>1. At the bottom of the dialog box, choose Show Details.</td>
</tr>
<tr>
<td></td>
<td>3. Choose Page setup. You may need to adjust some or all of the</td>
<td>2. In the Margins section, choose Custom.</td>
<td>3. Select the required page orientation and adjust the scale.</td>
<td>2. Select the required page orientation and adjust the scale.</td>
</tr>
<tr>
<td></td>
<td>following settings to ensure that the Design Studio application is</td>
<td>3. Hover over the Page preview and enter the required values for the</td>
<td>4. Ensure that Print Background Colors and Images are selected.</td>
<td>3. Ensure that Print Backgrounds is selected.</td>
</tr>
<tr>
<td></td>
<td>printed correctly:</td>
<td>margins in the four black boxes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ Select portrait or landscape as required.</td>
<td>4. In Options, ensure that Background Colors and Images are selected.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ Ensure that Print Background Colors and Images are selected.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ Set all margins to 0 mm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BI platform</td>
<td>Same as above.</td>
<td></td>
<td>Same as above.</td>
<td>Same as above.</td>
</tr>
</tbody>
</table>

*Note: Printing in Chrome has been disabled.*
Note

Printing is not supported when using the SAP BusinessObjects Mobile solution. You must read the following SAP Note before printing:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2037087</td>
<td>Considerations when printing a SAP BusinessObjects Design Studio Analysis Application.</td>
</tr>
</tbody>
</table>

Context

To call the browser print dialog box from an application, follow the steps below:

Procedure

1. Create a component with an `onClick` property.
2. Include `APPLICATION.print();` call in the `onClick` control of the component.
3. Execute the application.
4. In the application, select the component to call `print()`.
17 Exporting to PDF

You can export some or all elements of your analysis application to PDF.

There are three different techniques you can apply when exporting to PDF. Each method provides different results, depending on your specific requirements.

Exporting to PDF is supported on the following platforms - BI platform, SAP NetWeaver and local mode.

You can use scripting methods, to allow the end user to export to PDF from their analysis applications to produce one of the following types of output:

- A WYSIWYG version of a selected panel of the application.
- A report style export of all crosstabs and charts native to Design Studio, and custom SDK components contained in the application.

Exporting to PDF is activated through the optional Technical Component called PDF. It must be added in the Outline view of the designer application before this export to PDF functionality is available.

For all three types of export to PDF, page size, orientation, header title and footer text can be specified in the dialog box that appears when the export to PDF functionality is called by the user. For the report style export of all components in an application, users can also select to display optional metadata and a page number per PDF page. An optional header image can be specified in the PDF technical component designer properties. This header image then automatically appears at the top of each PDF page when the application is exported to PDF.

i Note

- Legacy applications can also apply this export to PDF functionality.

### Note

Before working with the export to PDF feature, you should refer to the following SAP Notes:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2150460</td>
<td>Using CSS with Export to PDF</td>
</tr>
<tr>
<td>2169049</td>
<td>Considerations when exporting to PDF in Design Studio 1.5</td>
</tr>
</tbody>
</table>

Related Information

- [Enabling Export to PDF in Analysis Applications](#)
- [WYSIWYG Export to PDF](#)
- [Scripting WYSIWYG Export to PDF](#)
- [Application Export to PDF](#)
17.1 Enabling Export to PDF in Analysis Applications

Context

Exporting to PDF is activated through the optional Technical Component called PDF. It must be added in the Outline view of the designer application before this export to PDF functionality is available.

To activate the PDF technical component, follow these steps:

Procedure

1. In the design tool, go to the Outline view.
2. Right-click Technical Components and choose Create Child.
3. Choose PDF.

Results

The PDF technical component will now appear in the script editor as a global object, allowing you to script the export to PDF functionality into your application.

Related Information

Exporting to PDF [page 106]

17.2 WYSIWYG Export to PDF

WYSIWYG Application Export to PDF

You can export to PDF a WYSIWYG (What You See Is What You Get) version of your application. This will export to PDF a WYSIWYG version of the entire application screen, exactly as it appears on your screen. To call this export to PDF functionality, you can script a basic component using the scripting method

PDF.exportApplicationScreen();

You can then print your exported PDF using the print functionality of your PDF viewer of choice.
WYSIWYG Panel Export to PDF

Rather than export all the elements of your application, you can also select to export to PDF only the contents of a specific panel in your application. To call this export to PDF functionality, you can script a component using the scripting method `PDF.exportPanelScreen(panel);`.

You can then print your exported PDF using the print functionality of your PDF viewer of choice.

Note
- Hidden or non-active pages and tabs of Pagebook and Tabstrip components will not be exported to PDF.
- If a component to be exported to PDF shows an error, it will still be exported.

Related Information
- Exporting to PDF [page 106]
- Scripting WYSIWYG Export to PDF [page 108]

17.2.1 Scripting WYSIWYG Export to PDF

Context
You have created an analysis application and wish to export to PDF a screenshot of your application or a specific panel within your application, exactly as it appears on the screen. You have added the PDF technical component in the Outline view of your application.

Procedure
In the scripting editor of the On Select property of a button component, add one of the following scripting methods. Your choice will depend on whether you wish to enable a WYSIWYG export to PDF of your application or a specific panel only.

- `PDF.exportApplicationScreen();`
- `PDF.exportPanelScreen(panel);`

When the user clicks the button in their application, the Export Application Screen to PDF dialog box is called. The user can then specify the following, before selecting the Export to PDF button in the dialog box:

- **Page Size**:
  - A2
  - A3
  - A4 (default)
Results

You have scripted a basic component to enable the user to call the WYSIWYG export to PDF functionality.

Related Information

WYSIWYG Export to PDF [page 107]
Enabling Export to PDF in Analysis Applications [page 107]

17.3 Application Export to PDF

You can allow end users to export to PDF a report style version of their application. Native charts and crosstabs, and custom SDK components are exported to PDF with an optional header image, metadata, page number, header title and footer text. The header image is specified in the PDF technical component designer properties. The metadata, page number, header title and footer text are specified by the user in the Export to PDF dialog box that appears in their application when the export to PDF functionality is called. To call this export to PDF functionality, you can script a basic component using the scripting method `PDF.exportApplication();`

Native charts are exported as SVGs. Custom SDK components are exported as images. Native charts and custom SDK components are exported as one application component per page. Native crosstabs are exported with all of the data (no scrollbars), which will be represented as a formatted table in the PDF. Conditional formatting will be included in the exported PDF. For example, if a cell’s background color is highlighted in red, then this will also be exported. The crosstab will be split by columns and rows where it does not fully fit onto the page. Row and column headers will be repeated for ease of use.

Note

Before working with the export to PDF feature, you should refer to the following SAP Notes:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2150460</td>
<td>Using CSS with Export to PDF</td>
</tr>
<tr>
<td>2169049</td>
<td>Considerations when exporting to PDF in Design Studio 1.5</td>
</tr>
</tbody>
</table>
17.3.1 Scripting Application Export to PDF

Context

You have created an analysis application and wish to export to PDF your entire application, resulting in the export of one component per page. You have added the PDF technical component in the Outline view of your application.

Procedure

1. Add `PDF.exportApplication();` in the scripting editor. For example, on the On Click property of a button component.

2. In the PDF technical component designer property called Header Image, specify the location of the header image that will appear at the top of each page when the application is exported to PDF. Header Images are optional.

When the user calls the export functionality in their application, the Export Application Screen to PDF dialog box is called. The user can then specify the following, before selecting the Export to PDF button in the dialog box:

- **Page Size:**
  - A2
  - A3
  - A4 (default)
  - A5
- **Orientation:**
  - portrait
  - landscape
- **Metadata**
  - Optional information displayed on the PDF for charts and crosstabs. The length of metadata text for variables, static and dynamic filters, is limited to 250 characters, or a maximum of two lines. Types of optional metadata include:
    - date
    - query name
    - variables
    - static filters
    - dynamic filters
- **Header Title:**
- **Page Number:**
- **Footer Text:**

Results

You have scripted a basic component to enable the user to call the report style export to PDF functionality.
Related Information

Application Export to PDF [page 109]
Exporting to PDF [page 106]
You can allow the user to export their data source to a .lums Lumira file. This file can then be opened in SAP Lumira Desktop and used in the same way as if it had been imported from a CSV file.

The exported file is named `EXPORT_LUMSddmmyyhhmmssss`, which refers to the day, month, year, hour, minute, second and millisecond when the file was created. This level of detail is applied in order to guarantee a unique file name.

To implement this feature you can add a data source and from another component, for example, a button, call the `DS_1.export(exportType)` scripting method. When you execute in the browser, you can select the `Save As` option and this will save the generated .lums file to a local directory. In Lumira Desktop, you must select `Import to Folder`. Select the .lums file and select `Yes` to open the dataset in Lumira Desktop.

### Note
You must read the following SAP Note before using the exporting to SAP Lumira Desktop:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2169080</td>
<td>Considerations when exporting to SAP Lumira Desktop.</td>
</tr>
</tbody>
</table>

### Note
The functionality is supported on the BI platform, SAP NetWeaver and local mode.

### How the data is exported

The data is exported with the following changes:

- The data is flattened. All measures are moved to the columns and all dimensions are moved to the rows.
- All total values are not included.
- Values are exported without scaling factors and with up to 9 decimal places.
- Hierarchies will be output in the current selected state, but flattened.
- If the data source has been modified by filtering or removing measures via scripting, the export will reflect this.
19 Changing an Analysis Application

Procedure

1. To open an existing application, click Application ➔ Open, select the relevant application and click Open.
2. Change the application according to your needs, for example, by adding, changing or removing components and data sources.
3. Save the changed application or click Application ➔ Save As to save the changed application using a different name.
   You can undo/redo any change you have made to an application by clicking the Edit menu actions, even after saving the application. The system updates the menu entries in the Edit menu dynamically according to the type of modification.

Related Information

Saving an Application Using a Different Name [page 561]

19.1 Saving an Application Using a Different Name

Prerequisites

You have made changes to an existing application.

Context

You want to save the changed application using a different name. Perform one of the following steps:

Procedure

Click Application ➔ Save as... The Save as dialog box opens. The system suggests the current name for the application.
- For the BI platform mode, proceed as follows:
○ Select the folder on the BI platform where you want to save your application to. Make sure that you have the relevant access rights for the chosen folder. If you encounter problems, contact your system administrator.

○ In the Name box, type in a new name for the application and click Save. Or select the application you want to rename in the list of existing applications and click the symbol (Rename the application). Now you can edit the relevant name directly in the list. Click Save. The new application name is displayed in the outline view of the editor.

○ In the list of existing applications, select the one that you want to use for your application and click Save. The system asks you to confirm that you want to overwrite an existing application. Click Overwrite. The chosen application name is displayed in the relevant tab in the editor.

○ For the SAP NetWeaver or the local mode, proceed as follows:
  ○ In the Name box, type in a new name for the application and click Save. The new application name is displayed in the outline view of the editor.
  ○ In the list of existing applications, select the one that you want to use for your application and click Save. The system asks you to confirm that you want to overwrite an existing application. Click Overwrite. The chosen application name is displayed in the relevant tab in the editor.

⚠️ Caution

Once you have confirmed that you want to overwrite an existing application, you cannot undo the changes.

You cannot select the name of an application that is open in another editor. Select another application name.

Results

You have now saved the application using a new or different name.

19.2 Maximum Number of Steps Back or Resetting within an Application

Prerequisites

You wish to allow users undo some steps or to revert all changes they made within their analysis application.

Context

The scripting API can be used to allow application users to undo a configurable number of steps or revert all changes in their current analysis application. For example, if a user makes any changes to the data source, such as filtering on KeyFigure or Type, these changes can be undone or reset within the application.
Procedure

1. Set Application Properties Behavior Maximum Number of Steps Back to an integer value more than zero. This property is set to zero or disabled by default. So, if you set the integer value to 10, the user will be able to undo 10 steps within their current application. The recommended maximum value is 20. Once the property is set to a value more than zero, the reset feature is then also enabled.

2. Create an analysis application.

3. Script a basic component, such as a Button using the State.backOneStep(); method. This allows the user to revert back one step at a time with each selection of the basic component. The number of steps allowed, depends on the configuration of the Maximum Number of Steps Back application property.

4. Script another basic component, such as a Button using the State.backToStart(); method. This allows the user to revert all changes in the current application, back to the original state of the application.

5. To disable the basic component for reverting back one step at a time, apply the method State.isBackOneStepAvailable();

6. To disable the basic component for resetting the application apply the method State.isBackToStartAvailable().

Once the user has undone the number of steps configured by the Maximum Number of Steps Back application property, or already reset the application to the original state, the basic component will appear greyed out.

Results

You have used API methods to allow users to revert changes made to their analysis application.

- **Note**
  BW Integrated Planning write-back is not compatible with the maximum number of steps back or reset functionality described here.

- **Note**
  Only changes made to the application in memory, such as changing a Chart type, can be undone or reset. Anything that persists to disk or platform, such as saving a bookmark, will not be undone or reset with this feature.

- **Note**
  If the Maximum Number of Steps Back application property is set to a value greater than 20, an information message occurs and the property value is set back to the last acceptable number.

- **Note**
  The maximum number of steps back and reset features are supported on the following platforms - BI platform, SAP NetWeaver and local. They are not supported on SAP HANA.
Related Information

- Back One Step (backOneStep) [page 540]
- Back To Start (backToStart) [page 540]
- Is Back To Start Available (isBackToStartAvailable) [page 541]
- Is Back One Step Available (isBackOneStepAvailable) [page 541]
20 Deleting an Analysis Application

Prerequisites

You have opened the application you want to delete.

Procedure

1. Click Application > Delete. The system asks you to confirm that you want to delete the application permanently.
2. Click Delete.
21 Searching in Applications

Context

You can search within an analysis application and all its content (components, data sources, properties values and scripts) for a certain string or number.

Procedure

1. Open the application that you want to search.
2. Select in the menu of the design studio Search > Search Application... The Search Application dialog box is displayed.
3. Enter your search string. The system displays the search results while you are typing the search string and highlights them.
4. If you want the system to perform a case-sensitive search and/or search any hidden components, select Options and select the respective checkbox.
5. If you want to keep the search result, press Keep Results in the dialog box. The results are displayed in the Search Results view.
Assigning Analysis Applications to the Mobile Category

Context

Before users can access analysis applications on a mobile device using the SAP BusinessObjects Mobile solution, you need to assign the analysis applications to the mobile category on the BI platform.

Procedure

1. In the BI launch pad, go to the folder that contains the analysis application that you want to assign to the mobile category.
2. Select the analysis application and click More Actions Categories. The Categories dialog box appears.
3. Choose the mobile category.
4. Click OK.
23 Creating Planning Applications

With SAP BusinessObjects Design Studio you can create desktop browser applications for planning business data. These planning applications support both manual and automated data entry and changes to data. The application user can enter the planning data manually in the crosstab (in cells or rows) or use planning functions and planning sequences (also known as planning objects) in SAP NetWeaver BW Integrated Planning to enter data automatically. For planning data, you have to use a BW backend system as the planning system.

After entering planning data, the application user can recalculate, reset or save the data. As an application designer, you integrate these functions into the planning application by using the following script methods for planning:

- clientReset
- hasClientChanges
- hasUnsavedChanges
- recalculate
- reset
- save

Prerequisites

For creating planning applications, the following conditions must be met:

- You are using the design studio with BI Platform or SAP NetWeaver as the platform
- You are using a BW back end system as the planning system
- You are using a BW query or query view as a data source, which is defined as a planning query in the BEx Query Designer
- You are using a desktop browser application as planning application


Related Information

- Entering Data in the Crosstab (Manual Planning) [page 122]
- Using Planning Functions and Sequences (Automated Planning) [page 123]
- Selecting a Planning Connection [page 121]
23.1 Selecting a Planning Connection

Context

A planning connection (planning system) is required for all planning actions. If you have not selected a planning connection, you cannot change data, edit data or execute planning functions.

Procedure

1. In the Outline view of your planning application, select the application in the structure.
2. Select Planning Connection in the Properties view of the application and choose a BW system from the dropdown box.
3. Select a suitable data source for the planning data by selecting Add Data Source in the Outline view or in the menu.
   If you have not yet integrated data sources into your application, you can choose from all BW systems that provide data sources. If you have already added data sources, you can only choose between the systems already used in a data source. Although you can use data sources from several different backend systems in a planning application, there can only be ONE planning system. This means you can only plan for one system per application.
4. (optional) If you want to use a planning model that is available for the chosen planning connection, select the required model from the list of environments under the property Planning Model. Under this property, you can select a planning model (Business Planning and Consolidation (BPC) environment and model) of SAP Business Planning and Consolidation, version for SAP NetWeaver, Unified. This makes it possible to create integrated planning solutions in a BW system, thereby providing high flexibility and usability for specialist users. Using this property causes the design studio’s planning functionality on the server to behave differently. For more information, see SAP Business Planning and Consolidation, version for SAP NetWeaver on the SAP Help Portal.

Related Information

Selecting a Data Source [page 53]
23.2 Entering Data in the Crosstab (Manual Planning)

In order to have input ready cells or rows in a crosstab, you have to bind the crosstab component to an input-enabled data source. Whether input ready cells are present also depends on the model in the BW back end and the initial view of the data source.

To enable the input readiness of the data source, you also have to use the data source alias method `configureInputReadiness`.

Input ready cells display an edit field which, when clicked on, allows the application user to enter text. When the user presses Enter or leaves the cell by navigating away from it, the system validates the input. If the input is invalid, the relevant input area of the cell is highlighted. If the input is valid, the entered value is formatted for the relevant data type (for example, a unit is added, the decimal display format is changed, ...). Besides selecting a cell by clicking it, the user can also use the tab key to move forward from cell to cell in a given row. If any value in an input ready cell has been modified, the first press of Tab validates the input, and the second press of Tab moves the focus to the next cell.

**Note**

It is not possible to delete data in input-ready cells. When you enter a blank (empty) input, the cell is reset to its initial value. If you enter a 0 (zero) value (if permitted by the data type), the 0 is set as the new cell value.

In order to get new rows and configure their position in the crosstab, the following prerequisites must be met:

- Use a BW query or query view as a data source, which has a suitable initial view for getting new rows. For more information see “Planning Business Data Manually” on SAP Help Portal at [http://help.sap.com](http://help.sap.com)
- Use the `Number of New Rows` and `Position of New Rows` properties of the crosstab. New rows can either be displayed at the Bottom or the Top of the crosstab.

In input-ready rows, it may also be possible to enter values in row header cells. These have an input help, which when clicked on, opens a value help for dimension members. When a user enters a value without using the value help, the external key of the dimension member must be entered.

**Related Information**

Configure Input Readiness (configureInputReadiness) [page 431]

23.3 Editing Short Texts in Queries

With SAP BW 7.40 SPS 08, you can edit short texts in input-ready queries in the crosstab. This enables you to use characteristics (dimensions) of a suitable InfoProvider as key figures (measures) in the query definition, in order to change characteristic values (for example, classifications) in queries or to write comments (short texts) on key figure values in queries.
In the following example, free text entry is possible for cells in the Comment (Amount) column. Columns, in which you can edit short texts, do not support any settings. This means these columns cannot be sorted and they do not display a sorting icon. No formatting or sorting functions are available in the context menu.

![Table Example](image)

**Figure 1: Example**

---

### Note

When editing short texts in queries, note the following points:

- Editing short texts in queries is not possible in a design studio deployed on SAP HANA. This is only possible for SAP BW-based data sources.
- There is no value help available for attribute fields that only allow you to enter specific values.
- You can only enter single-line texts. The application user cannot enter text with line breaks.
- Automatic text line wrapping in a cell is not supported.

---

For more information about editing short texts in queries and the definition in the query designer, see “Editing Short Texts in Queries” on SAP Help Portal at [http://help.sap.com](http://help.sap.com)

### 23.4 Using Planning Functions and Sequences (Automated Planning)

Planning functions and sequences are planning objects defined in the Planning Modeler of SAP NetWeaver BW. These objects enable system-based processing and data generation. Functions can be executed immediately or in the background as a planning sequence. Various standard planning function types are delivered with the BW system. You can also define your own function types. A planning sequence is a sequence of planning functions. For more information about planning functions and sequences, see the following chapters on the SAP Help Portal:

- Planning Functions
- Implementing a Planning Function Type
- Planning Sequence
In the outline view, you can add an existing planning function or sequence for the selected planning system by choosing Planning Models Add Planning Function... or Add Planning Sequence... In the Add Planning Function or Add Planning Sequence dialog box, enter the name of an existing planning function or sequence, or alternatively use the Browse... button. Now you can use this planning object in the events of other components in your application, for example in the OnClick event of a button component. The scripting methods relating to the planning function or planning object are listed under PF_* or PS_* in the content assistance of the script editor.

The following methods are available for planning functions:

- clearAllFilters
- clearFilter
- copyFilter
- execute
- getDimensionText
- getDimensions
- getFilterExt
- getFilterText
- getMemberList
- getMembers
- getVariableValueExt
- getVariableValueText
- getVariables
- setFilter
- setFilterExt
- setVariableValue
- setVariableValueExt

The following methods are available for planning sequences:

- execute
- getVariableValueExt
- getVariableValueText
- getVariables
- setVariableValue
- setVariableValueExt

23.5 Cell Locking

In analysis applications, you can lock input-ready cells in the crosstab to protect these cells from being changed. A locked cell cannot be changed manually when the crosstab is recalculated. Cell locking is a time-limited setting that only applies to the current user session. Locked cells are displayed as not editable with a lock icon.

You can also undo cell locks.

There are two underlying implementations of locking cells in planning applications.
front-end cell locks (local cell locks in a query)
The user sets the cell lock at runtime for the current result set. The cell locks remain in effect as long as no significant changes are made to the result set.

back-end cell locks (global cell lock in a planning application)
You or your administrator manage the cell locks only in the back-end system for one single query or all input-ready queries in the planning application. Locking cells in all queries of the planning application means a cell that is locked in one query can be displayed as locked in other queries and can be handled as locked.

Caution
Once you have activated the global cell lock on the back end, the local cell lock will not work any more on the front end (design studio).

Remember that application users use the context menu entry Lock Value if they want to protect cells from being changed. They do not necessarily know which kind of cell lock implementation is used for the query.

For more information about cell locking in SAP BW, see “Cell Locks” in the SAP BW documentation on SAP Help Portal at http://help.sap.com/.

23.5.1 Front-End Cell Locks

Front-end cell locks in a query remain in effect until the application user makes significant changes to the crosstab. The cell locks remain in effect if the user performs any of the following activities:

- Sorting the crosstab
- Expanding or collapsing nodes in BW hierarchies
- Changing display settings for dimensions

If the user performs one of the following activities, the cells locks are removed:

- Adding/removing dimensions or measures to the crosstab
- Calling the prompting dialog to change variable values
- Swapping row and column axes
- Hierarchical axis display
- Zero suppression
- Setting a filter
- Executing a planning function or sequence

For more information about front-end cell locking in SAP BW, see “Local Cell Locks in a Query (Front End)” in the SAP BW documentation on SAP Help Portal at http://help.sap.com/.

23.5.2 Back-End Cell Locks

Back-end cell locks in a planning application allow you to use more navigation steps in a query while retaining the cells locks as front-end cell locks.
To activate the global cell lock on the back end, you or your administrator must set a parameter in the table `RSADMIN`. You can use the program `SAP_RSADMIN_MAINTAIN` to do this. Set the following parameter:

```
OBJECT = RSPLS_PQ_BACKEND_CELL_LOCKING
VALUE = X
```

⚠️ Caution

Once you have activated the global cell lock on the back end, the local cell lock will not work any more on the front end (design studio).

Cross query cell locks can be particularly suitable if a planning application contains multiple tabs or pages, and its queries are closely linked but are used for different aspects of the planning application.

The back-end cell locks remain in effect if the user performs any of the following activities:

- Displaying the properties of the dimensions (like Key or Text)
- Displaying attributes of the dimensions
- Sorting the result set by dimension members, texts or measures (key figures)
- Swapping row and column axes
- Expanding or collapsing nodes in hierarchies
- Hiding/showing structure elements (filter structure elements, for example)
- Adding a dimension to the crosstab: At the lowest point on the right in the rows and the furthest point inside in the columns
- Removing a dimension from the crosstab: At the lowest point on the right in the rows and the furthest point inside in the columns

ℹ️ Note

Some of the operations listed above hide locked cells. This does not undo the cell locks however. These “hidden” cell locks also remain active and are taken into account by the system when calculating inverse formulas and during disaggregation.

If the user performs any of the following activities, all cell locks are removed:

- Changing the order of the dimensions included in a cell lock
- Changing the axis of the dimension included in a cell lock
- Changing the settings for BW hierarchies (on/off) for dimensions included in a cell lock
- Changing the query’s dynamic filter (by restricting the variable values without restarting the query for example)
- Changing the filter using the Prompts dialog box
- Performing a planning function or planning sequence

For more information about back-end cell locking in SAP BW, see “Global Cell Locks in a Planning Application (Back End)” in the SAP BW documentation on SAP Help Portal at [http://help.sap.com/](http://help.sap.com/).
23.5.3 Locking Cells

Context

You can lock input-ready crosstab and total cells in planning applications. A locked cell cannot be changed manually when the crosstab is recalculated. The cells are locked for the current user session only. You can also unlock the cells during the session.

Procedure

1. Select the crosstab cell you want to lock.
   The selected cell must be input-enabled and must not have been changed since the last recalculation.
2. Open the context menu of the selected cell and choose Lock cell.
   If you want to unlock the cell, select it and open the context menu. You can now deselect the menu entry Lock cell.

Results

The selected cell is locked for changes and is displayed with a lock icon.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.
24 Working with Bookmarks

There are three types of bookmarks that can be applied when working in Design Studio analysis applications – standard bookmarks, fragment bookmarks and portable fragment bookmarks.

If an application user wishes to serialize the state of their entire application and persist that state in their BI platform or SAP NetWeaver backend, you can apply a standard bookmark to their analysis application. If they wish to serialize only a selected part of their analysis application, you can apply a fragment bookmark or portable fragment bookmark to their application. A user can also decide to create or edit one of their own applications online by working with the online composition feature. The online composition feature combines the functionality of the portable fragment bookmark, the Fragment Gallery and the Split Cell to empower the user to autonomously create their own online dashboards on demand.

All bookmark types created by users online can be assigned to a bookmark folder structure. This folder structure is created by the administrator on the BI platform in the Central Management Console and has user security applied on a folder level. Access is restricted according to the rights of individual users. Using the online composition feature, users can select a folder from a list component representing this folder structure to filter the portable fragment bookmarks they wish to use in their online application. If a user does not have access rights to a particular folder, they will not see it appear as an option for them to select. If a user has view rights only, they will be able to see the folder, but not overwrite the folder content in any way.

For more information on working with the online composition feature, you can refer to the chapter in this guide called Using the Online Composition Feature.

All bookmark types can be saved in local mode for testing and development purposes. You can use scripting methods to allow application users perform various functions online with their own standard and fragment bookmarks or with portable fragment bookmarks created by them or by other users for the same Group Identifier. All bookmark types have to be specifically called by the scripting API, unless they are being loaded from a Url.

For more information about the individual types of bookmarks, how to script for them and how to apply them, you can refer to the following chapters - Standard Bookmarks, Fragment Bookmarks, Portable Fragment Bookmarks, Scripting for All Bookmark Types.

**Note**

Before working with bookmarks, you should refer to the following SAP Notes:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2168984</td>
<td>Considerations when working with bookmarks in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
<tr>
<td>2157239</td>
<td>Performing more than one free text search from your bookmark filter list will return zero items.</td>
</tr>
</tbody>
</table>

**Note**

1. Bookmark Urls can only be loaded in local mode within a browser launched from the users own analysis application in the following circumstances:
   - While the designer is running
Location of bookmarks persisted in the BW system

Bookmarks of analysis applications are persisted in the BW system in the following tables:

- RSWR_DATA (Type = Z)
- RSWR_DATA_XREF
- RSAO_T_BOOKM_EXT

Note

The first two tables are also used by BEx bookmarks. For more information about the required SAP NetWeaver Support Packages, see the following SAP Note:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2069853</td>
<td>Design Studio Netweaver: Bookmarking</td>
</tr>
</tbody>
</table>

Related Information

Standard Bookmarks [page 129]
Portable Fragment Bookmarks [page 131]
Fragment Bookmarks [page 130]
Scripting for All Bookmark Types [page 133]
Obsolete Bookmarks [page 142]
Working with Online Composition [page 145]

24.1 Standard Bookmarks

Standard bookmarks are used to bookmark an entire analysis application. They are defined by the unique combination of analysis application, analysis application version and application user. You can apply standard bookmark scripting methods that allow the application user to serialize the state of their entire application and then persist that state in the BI platform and SAP NetWeaver backend. A standard bookmark has to be specifically called by the scripting API. As the entire application is bookmarked, there is no requirement to select individual
elements within the application. Standard bookmarks can also be saved in local mode (stored on disk) for testing and development purposes.

Standard bookmarks are always private. This means that they can only be shared by explicitly generating a bookmark URL and sending it to other users via email. Standard bookmarks are only visible via scripting to the user that created them.

They can be assigned to the bookmark folder structure created by the administrator.

Standard Bookmarks created and shared by other application users can be loaded via a URL in the browser bar in BI platform and SAP NetWeaver mode.

**Note**

Before working with bookmarks, you should refer to the following SAP Note:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2168984</td>
<td>Considerations when working with bookmarks in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
</tbody>
</table>

**Related Information**

Fragment Bookmarks [page 130]  
Portable Fragment Bookmarks [page 131]  
Scripting for All Bookmark Types [page 133]  
Deleting Bookmarks [page 134]  
Listing Bookmarks [page 135]  
Saving a Bookmark [page 138]  
Sharing a Bookmark [page 140]  
Obsolete Bookmarks [page 142]  
Assigning Bookmarks to Folders [page 141]

### 24.2 Fragment Bookmarks

Users may want to identify parts of their application that are to remain unchanged from a design perspective, while other parts of their application may change from time to time, such as *Text* fields. In order to achieve this, fragment bookmarks are used to serialize a fragment or part of an analysis application. If you wish to serialize only a selected part of an analysis application, you can choose any container component which contains the element(s) of the analysis application that are to be persisted. Any component or components inside the specified container, including the container itself, will be serialized.

Data sources connected to the data bound component in the selection will be included in the fragment bookmark. Any other data sources outside the serialized component will be excluded.

Fragment bookmarks can be assigned to a bookmark folder structure, which is created by the administrator.
It is important that the bookmarked container name remain unchanged and that the container persists as the host application versions change. For example, if you have saved your fragment bookmark within a Panel container, this Panel container must not be removed from the application into which you want to reload your fragment bookmark. Fragment bookmarks are application and user specific. They are loaded only within the context of an application and so cannot be loaded standalone. Fragment bookmarks are available in local mode (stored on disk), BI platform mode (stored in CMS) and SAP NetWeaver mode (stored in BW tables).

The container components that can be specified to accommodate a fragment selection include one of the following:

- grid layout
- pagebook
- panel
- tabstrip
- split cell

**Note**

Fragment bookmarks created and shared by other application users can be loaded within the context of an application via a Url in the browser bar in BI platform and SAP NetWeaver mode.

**Note**

Before working with bookmarks, you should refer to the following SAP Note:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2168984</td>
<td>Considerations when working with bookmarks in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
</tbody>
</table>

**Related Information**

- Standard Bookmarks [page 129]
- Portable Fragment Bookmarks [page 131]
- Scripting for All Bookmark Types [page 133]
- Assigning Bookmarks to Folders [page 141]

## 24.3 Portable Fragment Bookmarks

A portable fragment bookmark is a type of bookmark that can be shared and reused between application users using a common reference called a Group Identifier.

You can decide which part of an analysis application can be saved as a portable fragment bookmark. This allows users to capture the state of a selected area of their application and make that available for use by other users in their applications. To allow other users to consume these portable fragment bookmarks in their own applications online, they must use the same Group Identifier.
Portable fragment bookmarks can be shared with other users like any other bookmark type, by using the share bookmark URL. For more information on sharing a bookmark, you can refer to the chapter in this guide called *Sharing a Bookmark*. Portable fragment bookmarks are available in local mode (stored on disk), BI platform mode (stored in CMS) and SAP NetWeaver mode (stored in BW tables).

### Note

Before working with bookmarks, you should refer to the following SAP Notes:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2168984</td>
<td>Considerations when working with bookmarks in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
<tr>
<td>2157239</td>
<td>Performing more than one free text search from your bookmark filter list will return zero items.</td>
</tr>
</tbody>
</table>

### Note

- If there are multiple parts of the application to be saved, multiple portable fragment bookmarks must be created.
- Portable fragment bookmarks can only be loaded in the application where they were created. To load a portable fragment bookmark from a different application, refer to the chapter called *Using the Online Composition Feature*.

For further information on the online composition feature, you can also refer to the following chapters within this guide - *Working with Fragment Gallery Components* and *Working with Split Cell Containers*.

### Group Identifier

The `groupId` allows you to group or to apply a group to a set of portable fragment bookmarks. It is a common reference used to allow application users to make their portable fragment bookmarks available for consumption online by other application user groups. The Group Identifier is a way for an administrator to add another layer of authorization by ensuring that users from a particular business unit can only see portable fragment bookmarks relative to their business unit. As an application designer, you must associate all portable fragment bookmarks with a Group Identifier. User groups have access to the bookmark folders that store portable fragment bookmarks associated with the Group Identifier.

When scripting using the Group Identifier, a number of rules apply that must be followed. Whenever you enter the Group Identifier into the appropriate scripting method, a check will occur. Once the check passes, the script is executed. If the check fails, an information message appears.

The following are the rules that apply when scripting with the Group Identifier:

- a minimum of 8 and a maximum of 64 characters must be used
- a minimum of 4 alphabetical characters must be used (upper or lower case acceptable)
- numbers are allowed
- underscores are allowed
- special characters are not allowed
- spaces are not allowed
24.4 Scripting for All Bookmark Types

The scripting API can be used to allow application users to delete, list, load, save, share their bookmarks, and assign them to folders. For more detail, you can refer to the following chapters in this guide:

- Deleting Bookmarks
- Listing Bookmarks
- Loading Bookmarks via Scripting
- Loading Bookmarks via Url
- Saving a Bookmark
- Sharing a Bookmark
- Assigning Bookmarks to Folders

**Note**

Before working with bookmarks, you should refer to the following SAP Notes:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2168984</td>
<td>Considerations when working with bookmarks in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
<tr>
<td>2157239</td>
<td>Performing more than one free text search from your bookmark filter list will return zero items.</td>
</tr>
</tbody>
</table>

Related Information

- Deleting Bookmarks [page 134]
- Listing Bookmarks [page 135]
- Loading Bookmarks via Scripting [page 137]
- Loading Shared Bookmarks via Url [page 138]
- Saving a Bookmark [page 138]
- Sharing a Bookmark [page 140]
24.4.1 Deleting Bookmarks

All bookmark types have a parent application. If the parent application is deleted, child bookmarks are deleted. You can use the scripting API to allow application users to delete their own bookmarks. The following scripting is used, depending on the type of bookmark involved:

Deleting Standard Bookmarks

The following scripting is used to allow the application user to delete their own standard bookmarks:

- `Bookmark.deleteBookmark(id);`
- `Bookmark.deleteAllBookmarks();`

Deleting Fragment Bookmarks

The following scripting is used to allow the application user to delete their own fragment bookmarks:

- `Bookmark.FragmentBookmark.deleteBookmark(id);`
- `Bookmark.FragmentBookmark.deleteAllBookmarks();`

Deleting Portable Fragment Bookmarks

The following scripting is used to allow the application user to delete their own portable fragment bookmarks:

- `Bookmark.PortableFragmentBookmark.deleteBookmark(id)`
- `Bookmark.PortableFragmentBookmark.deleteAllBookmarks(groupIdentifier)`

Note

1. Application users cannot delete bookmarks created by other application users.
2. Deleting a bookmark does not automatically remove it from a selection component. The selection component needs to be scripted separately to allow for the list of bookmarks to be refreshed after a bookmark is deleted.

Related Information

Delete All Bookmarks (deleteAllBookmarks) [page 347]
24.4.2 Listing Bookmarks

You can use the scripting API to allow application users to retrieve a list of their own bookmarks at runtime. The following script methods are used, depending on the type of bookmark involved:

**Listing Standard Bookmarks**

The returned array contains a list of `BookmarkInfo` object types. A `BookmarkInfo` object contains `BookmarkId id`, `String name` and `String text`. Both `String name` and `BookmarkId id` refer to the bookmark id. `String text` refers to the bookmark title. Any selection component like a `Dropdown box` or a `List box` can be populated with the array returned from the `Bookmark.getAllBookmarks()` method.

```javascript
var array = Bookmark.getAllBookmarks();
array.forEach(function(element, index) {
  DROPDOWN_1.addItem(element.name, element.text);
});
```

**Listing Fragment Bookmarks**

Listing fragment bookmarks returns a list of all fragment bookmarks for the analysis application.

```javascript
var array = Bookmark.FragmentBookmark.getAllBookmarkInfos();
array.forEach(function(element, index) {
  DROPDOWN_1.addItem(element.id, element.title);
});
```
Listing Portable Fragment Bookmarks for all Applications

Applying the scripting method `Bookmark.PortableFragmentBookmark.getAllBookmarkInfos();` returns a list of all portable fragment bookmarks specified by the Group Identifier parameter. You can populate the Fragment Gallery with all portable fragment bookmarks created by the user using the `FragmentGallery_1.addItems();` scripting method.

```javascript
var array = Bookmark.PortableFragmentBookmark.getAllBookmarkInfos("groupIdentifier");
FRAGMENTGALLERY_1.addItems(array);
```

Listing Portable Fragment Bookmarks for Current Application

Applying the scripting method `Bookmark.PortableFragmentBookmark.getAllBookmarkInfosForApplication();` returns a list of all portable fragment bookmarks for the current application only.

```javascript
var array = Bookmark.PortableFragmentBookmark.getAllBookmarkInfosForApplication("groupIdentifier");
FRAGMENTGALLERY_1.addItems(array);
```

For more information on the rules applied when using the Group Identifier, you can refer to the chapter in this guide called Portable Fragment Bookmarks.

For more information on working with the Fragment Gallery and using the online composition feature, you can refer to the chapter in this guide called Using the Online Composition Feature.

**Note**

Deleting a bookmark does not automatically remove it from a selection component, so a deleted bookmark may still appear in a list. The selection component needs to be scripted separately to allow for the list of bookmarks to be refreshed after a bookmark is deleted.

Related Information

- Portable Fragment Bookmarks [page 131]
- Working with Online Composition [page 145]
- Working with Fragment Gallery Components [page 148]
- Get All Bookmarks (getAllBookmarks) [page 348]
- Get All Bookmark Infos (getAllBookmarkInfos) [page 354]
- Get All Bookmark Infos For Application (getAllBookmarkInfosForApplication) [page 361]
- Get All Bookmark Infos (getAllBookmarkInfos) [page 360]
24.4.3 Loading Bookmarks via Scripting

You can use a scripting method to allow application users to load their own bookmarks within a running analysis application. The following script methods are used, depending on the type of bookmark involved:

### Loading Standard Bookmarks

To load a standard bookmark based on id, you can use the following method:

```javascript
Bookmark.loadBookmark(id);
```

### Selecting a Standard Bookmark id from Dropdown List and Loading

```javascript
var id = DROPDOWN_1.getSelectedValue();

Bookmark.loadBookmark(id);
```

### Loading Fragment Bookmarks

Applying the scripting method `Bookmark.FragmentBookmark.loadBookmark(id)` loads the state of an analysis application from a fragment bookmark.

### Loading Portable Fragment Bookmarks

Applying the scripting method `Bookmark.PortableFragmentBookmark.loadBookmark(id)` loads the state of an analysis application from a portable fragment bookmark.

**Note**

It is important that the bookmarked container name remain unchanged and that the container persists. For example, if you have saved your bookmark within a `Panel` container, this container must not be removed from the application into which you want to load your bookmark.

**Note**

Portable fragment bookmarks can only be loaded in the application where they were created. To load a portable fragment bookmark from a different application, refer to the chapter called *Using the Online Composition Feature*. 
24.4.4 Loading Shared Bookmarks via Url

Application users can load bookmarks shared by other application users via Url in the browser bar. When they open the Url from their preferred browser, they will see the shared analysis application. All bookmark types shared via Url cannot be added directly to the users list of bookmarks. However, they can bookmark the shared analysis application and that bookmark is then listed when calling the script method `Bookmark.getAllBookmarks()` or `Bookmark.FragmentBookmark.getAllBookmarkInfos()` or `Bookmark.PortableFragmentBookmark.getAllBookmarkInfos(groupIdentifier)`; Alternatively, the Url can be saved to their browser favorites.

### Note
- It is important that the bookmarked container name remain unchanged and that the container persists. For example, if you have saved your bookmark within a `Panel` container, this container must not be removed from the application into which you want to load your bookmark.
- Obsolete bookmarks, can still be loaded if they have been saved as a Url. A message will appear when you run the application telling you that the bookmark is obsolete.

## Related Information

Loading Bookmarks via Scripting [page 137]
Sharing a Bookmark [page 140]
Obsolete Bookmarks [page 142]

24.4.5 Saving a Bookmark

### Saving a Bookmark

You can use the scripting API to allow application users to save their own bookmarks. A new standard bookmark is created with an optional title for the current state and version of the analysis application. A new fragment or portable fragment will be created with an optional title for a part or parts of the analysis application. A default title will be automatically generated if not supplied. Via scripting, you can provide a way to allow users to create their own title.
Saving Standard Bookmarks

The following scripting is used to allow the application user to save their own standard bookmarks with a unique, automatically generated title or with a title specified by them at runtime:

```javascript
var id = Bookmark.saveBookmark();
var id = Bookmark.saveBookmark("title")
```

**Note**
- When the application user saves their bookmark with a title that already exists, the corresponding bookmark with that title is overwritten.
- If a title is specified it must not contain an empty string.

Saving Fragment Bookmarks

Scripting is used to allow the application user to save their own fragment bookmarks. Title is optional. If the title is not specified then a unique default title will be generated automatically. Optional `BookmarkInfo toOverWrite` is used if you want to overwrite an existing fragment bookmark.

```javascript
var fragInfo = Bookmark.FragmentBookmark.saveBookmark(ContainerComponent);
```

**Signature:** `FragmentBookmarkInfo saveBookmark(ContainerComponent component, optional String title ,optional String description, optional BookmarkId toOverWrite);`

**Note**
- Title and description are not unique identifiers, so the user can apply any title or description.

Saving Portable Fragment Bookmarks

Portable fragment bookmarks behave technically the same way as a fragment bookmark. The difference between fragment and portable fragment bookmarks, is that they are the only bookmark types that can be used within the Fragment Gallery as part of the online composition feature. For more information on online composition, see the following chapters - Online Composition, Working with Fragment Gallery Components and Working with Split Cell Containers.

Title is optional. If the title is not specified then a unique default title will be generated automatically. The `groupId` allows you to group or to apply a group to a set of portable fragment bookmarks. This allows users to share portable fragment bookmarks across applications.

```javascript
Bookmark.PortableFragmentBookmark.saveBookmark(groupId, ContainerComponent);
```
When you save a bookmark it is saved to disk. Applying the scripting methods for the undo or reset features will not remove the bookmark. To remove the bookmark the appropriate method to delete bookmarks needs to be called.

When scripting using the Group Identifier, a number of rules apply that must be followed. For more details on these rules, you can refer to the chapter called Portable Fragment Bookmarks.

If a bookmark is saved while a popup is open, the popup will be visible when reloading the analysis application from that bookmark. Call the method POPUP.hide() to ensure the popup is closed on loading the bookmark.

**Related Information**

- Fragment Bookmarks [page 130]
- Working with Online Composition [page 145]
- Working with Fragment Gallery Components [page 148]
- Working with Split Cell Containers [page 150]
- Maximum Number of Steps Back or Resetting within an Application [page 114]
- Back One Step (backOneStep) [page 540]
- Back To Start (backToStart) [page 540]
- Fragment Gallery [page 308]

### 24.4.6 Sharing a Bookmark

To enable application users to share any bookmark type Url, you can implement one of the following scripting methods, depending on the type of bookmark being shared:

- To share a standard bookmark, you can applying the `Bookmark.shareBookmark(String URL)` scripting method.
- To share a fragment bookmark, you can apply the `Bookmark.FragmentBookmark.shareBookmark(String URL)` scripting method.
- To share a portable fragment bookmark, you can apply the `Bookmark.PortableFragmentBookmark.shareBookmark(String URL)` scripting method.

The Url parameter is the text to be displayed. At runtime when this script is called, application users will be presented with a *Share Bookmark* dialog box, containing the generated bookmark Url. They have three options when using the *Share Bookmark* dialog box:

- Open the bookmark link directly in a separate browser by selecting the link icon. This will bring the user to the log in screen of the bookmarked application.
• Select and copy the bookmark URL to their preferred browser. They can then use native browser functionality to add the URL to their favorites or share the bookmark by copying and pasting the URL into an email.

• Select the **Send by email** button to open an email directly from their current application browser. The subject field of the generated email is pre-populated with the title of both the application and the bookmark the user wants to share. The body of the email contains a copy of the actual bookmark link. The user can then edit the email as required and share a bookmark of their application by sending the email to the appropriate recipients.

To exit the *Share Bookmark* dialog box, application users should select the **Close** button.

**Note**

• When sharing a fragment bookmark or portable fragment bookmark, it is important to note that users can only share one fragment bookmark ID. If users intend to share their fragment bookmark, they must ensure that everything they want to share is contained within one container component. Only one bookmark URL can be loaded at a time.

• Sharing a bookmark is most applicable in BI platform and SAP NetWeaver mode. However, it is technically possible to share a bookmark in local mode with other users on a different computer for testing and development purposes. Within the design tool under **Tools > Preferences > Application Design**, users should select the *Allow external access to embedded Web server* checkbox. To allow other users to view this bookmark URL, the designer session ID should also be manually added to the URL.

**Related Information**

Share Bookmark (shareBookmark) [page 352]
Share Bookmark (shareBookmark) [page 358]
Share Bookmark (shareBookmark) [page 365]

### 24.4.7 Assigning Bookmarks to Folders

You can script a list component to display to the application user, a list of all bookmark folders available to them, based on their Group Identifier. You can also script a basic component to allow the application user to save one of their own bookmarks to a selected folder. If a user does not have access rights to a particular folder, they will not see it appear as an option for them to select. If a user has view rights only, they will be able to see the folder, but not overwrite the folder content in any way.

**Assigning to a Folder**

This sample script could be used to assign bookmarks to a folder:

```javascript
var folderId = DROPDOWN_1.getSelectedValue();

var bookmarkId = DROPDOWN_2.getSelectedValue();
```
Get Bookmark Folders and Assign to a Dropdown Box

This sample script could be used to assign bookmarks to a folder:

```javascript
var folders = Bookmark.getBookmarkFolders();
DROPDOWN_1.removeAllItems();
folders.forEach(function(element, index) {
  DROPDOWN_1.addItem(element.id, element.title);
});
```

### Note

Before working with bookmarks, you should refer to the following SAP Notes:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2168984</td>
<td>Considerations when working with bookmarks in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
<tr>
<td>2157239</td>
<td>Performing more than one free text search from your bookmark filter list will return zero items.</td>
</tr>
</tbody>
</table>

### 24.5 Obsolete Bookmarks

#### Obsolete Standard Bookmarks

Standard bookmarks can become obsolete when design time changes have been made to the analysis application and the application is then re-published. As a result of these changes, obsolete standard bookmarks are not returned to the user via the `Bookmark.getAllBookmarks();` method. However, the bookmark may still be loaded if the bookmark id is known. If an application user makes a change to their analysis application and tries to load one of their own standard bookmarks saved before this change is made, an information message appears. The information message informs the application user that the bookmark is obsolete and that a new bookmark should be created.

#### Obsolete Personalization

If a user personalizes an application and the application subsequently changes, the personalization will be obsolete.
Obsolete Fragment Bookmarks

Fragment bookmarks are never obsolete. This gives maximum flexibility.

**Note**

It is highly recommended that you only create a fragment bookmark using areas of an application that are unlikely to change in the future. If a change is made in the area that has been bookmarked, new user interface changes will not be reflected in the older version of this fragment bookmark.

Obsolete Portable Fragment Bookmarks

In terms of becoming obsolete, portable fragment bookmarks behave in the same way as fragment bookmarks.

**Note**

- Obsolete bookmarks, can still be loaded if they have been saved as a Url. A message will appear when you run the application telling you that the bookmark is obsolete.

Related Information

Personalization [page 143]
Get All Bookmarks (getAllBookmarks) [page 348]

24.6 Personalization

The personalization feature is available at runtime in local, BI platform and SAP NetWeaver mode. It allows application users to save the state of their analysis application and later restore that default state when reloading the analysis application. Personalization is defined by the unique combination of analysis application, analysis application version and application user. Everytime personalization is set, the current analysis application state is overwritten by the last personalization saved. There is no scripting method to load personalization. The following script methods apply to personalization:

- `State.setPersonalization();`. This method allows the application user to personalize their current analysis application.
- `State.deletePersonalization();`. This method allows the application user to delete the personalization of their analysis application.

**Note**

1. When scripting the personalization feature, you must provide application users with a way to call the `State.deletePersonalization();` method, to allow them to remove personalization when required.
You must script the `State.setPersonalization();` and `State.deletePersonalization();` methods at the same time.

2. If a user personalizes an application and the application subsequently changes, the personalized bookmark will be obsolete. The application is then no longer personalized.

3. On Startup scripts and On Variable initialization do not run when personalizing an analysis application. This prevents startup scripts and variable initialization settings from overriding personalization settings.

Related Information

Set Personalization (setPersonalization) [page 542]
Delete Personalization (deletePersonalization) [page 542]
Obsolete Bookmarks [page 142]
25 Working with Online Composition

The online composition feature allows application users to create and edit their own applications online, based on a selection of portable fragment bookmarks created by any user creating portable fragment bookmarks for the same group identifier.

To create their own online applications, the user applies the drag and drop paradigm to drag portable fragment bookmarks from a Fragment Gallery component and drop them into a Split Cell.

Selecting Portable Fragment Bookmark Folders

The content of the Fragment Gallery can be filtered using a list component populated with bookmark folders. When the user selects one of these folders, the content of the Fragment Gallery changes to reflect the list of portable fragment bookmarks in the folder selected. If a user does not have access rights to a particular folder, they will not see it appear as an option for them to select. If a user has view rights only, they will be able to see the folder, but not overwrite the folder content in any way.

Drag and Drop

Online composition relies on the portable fragment bookmark API methods. Portable fragment bookmarks only store the state of the container which has been expressly chosen by the application designer to be saved. As part of online composition, the Split Cell area adjusts itself to accommodate the portable fragment bookmarks as they are dropped in.

It is important to note that there are prescribed user interactions with the Split Cell. An sample of those include the following:

- You can create content by dragging a portable fragment bookmark from the Fragment Gallery into a cell in the Split Cell.
- You can replace the content in an existing cell by dropping a portable fragment bookmark from the Fragment Gallery into the same cell.
- When adding content, you should drop the portable fragment bookmark to the right or below an existing cell.
- If you add content to the right, your content is added as a new column in the Split Cell.
- If you add content below, your content is added as a new row in the Split Cell.

Note

For more detailed information on populating the Split Cell, you can refer to the chapter in this guide called Working with Split Cell Containers.
Online composition use case:

The online composition feature hinges on the roles played by different personas within your organization. The scenario below describes a possible online composition workflow using three different personas - designer, power user and end user or consumer.

**Designer**

As application designer, you can create a producer application and a consumer application and assign these applications to the same Group Identifier. You can also upload these applications to folders, which can be accessed by users with this same Group Identifier. Within the producer application, you can define the container that will be saved as a portable fragment bookmark. You can script a basic component to save portable fragment bookmarks with the `Bookmark.PortableFragmentBookmark.saveBookmark(groupIdentifier, ContainerComponent)`, scripting method. The portable fragment bookmarks can be added to the Fragment Gallery in the consumer application by using `getAllFragmentBookmarks(GroupIdentifier);` and then `FRAGMENTGALLERY_1.addItems(array);`.

**Power User**

The power user creates a producer application to establish the content to be saved as portable fragments by defining the data source, filters and variable to be applied to the data. The power user can also define how this data will be visualized. These fragments created in the production application are then assigned to the appropriate bookmark folder and thereby available to any end user or consumer that shares the same Group Identifier.

**End User or Consumer**

The end user opens the consumer application online and will select from the Fragment Gallery which portable fragment bookmarks they will drop into their own custom-made online application. If they wish to change the portable fragments available to them, they can select a different folder from within a list component. This will then populate the Fragment Gallery with a new list of portable fragment bookmarks from which to choose. The end user drags an item from the Fragment Gallery into the Split Cell and rearranges accordingly.

The end user can then save this new composition as a bookmark and call it up again by loading the bookmark.

**Using Templates**

Examples of a consumer and producer applications can be found in the designer under the Generic Analysis Template and the Online Composition Template.
Note

For more information on using these templates, you can refer to the chapters called *Using the Generic Analysis Template* and *Using the Online Composition Template*.

Online Composition Notes

Note

Before working with bookmarks, you should refer to the following SAP Notes:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2168984</td>
<td>Considerations when working with bookmarks in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
<tr>
<td>2157239</td>
<td>Performing more than one free text search from your bookmark filter list will return zero items.</td>
</tr>
</tbody>
</table>

You must also be aware of the following when using the online composition feature:

Note

If you save a chart component and another component in a container as a portable fragment bookmark, a multiple component icon will be generated to represent your portable fragment bookmark in the *Fragment Gallery*. If your additional component is simply a *Textbox* and you expect your icon to reflect a chart, it is recommended that you do not include the *Textbox* within the same panel as your chart. You should put the *Textbox* component outside the panel. Then a chart component icon will be generated to represent your portable fragment bookmark in the *Fragment Gallery*.

Related Information

Working with Fragment Gallery Components [page 148]
Working with Split Cell Containers [page 150]
Portable Fragment Bookmarks [page 131]
Assigning Bookmarks to Folders [page 141]

25.1 Creating a Bookmark Based on a Query

You can use the script editor to assign a query and a system to a bookmarked application. This allows users running the application to distinguish between bookmarks that are associated with an application and bookmarks that are associated with the current query in the application.
When creating a bookmark, you can add an optional application identifier parameter called `appIdentifier`. This acts as a filter when calling for a list of bookmarks. Instead of returning a list of all bookmarks for an application, this added parameter allows you to filter on bookmarks that refer to a certain query only.

When you save a bookmark, you can set a string as an attribute of the bookmark. When you then use the `getAllBookmarks` or `deleteAllBookmarks` scripting methods, you can supply that string and it is used to decide which bookmarks are listed or deleted. For example:

```javascript
saveBookmark("Week5_net_sales, XSYSTEM + "_" + XQUERY);
```

### Note
Before working with bookmarks, you should refer to the following SAP Notes:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2168984</td>
<td>Considerations when working with bookmarks in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
<tr>
<td>2157239</td>
<td>Performing more than one free text search from your bookmark filter list will return zero items.</td>
</tr>
</tbody>
</table>

### Note
- When adding an `appIdentifier` parameter, you must apply certain criteria, such as the following:
  - alpha-numeric characters only
  - must be between 8 and 64 characters in length
- This parameter can be applied to any application, including the Generic Analysis Template.

## 25.2 Working with Fragment Gallery Components

The **Fragment Gallery** is a basic component that is used to store portable fragment bookmarks for the user while they are using the online composition feature. The user can drag portable fragment bookmarks from the **Fragment Gallery** into the **Split Cell** to create their own virtual dashboard. If they wish to change the portable fragments available to them, they can select a different folder from within a list component in their online application. This will then populate the **Fragment Gallery** with a new list of portable fragment bookmarks from which to choose. The end user can again drag an item from the **Fragment Gallery** into the **Split Cell** and rearrange accordingly.

### Configuration

The **Fragment Gallery** can be configured to display in the following ways:

- Single or Dual columns
- Vertical or Horizontal
- Display icons as:
Adding Portable Fragment Bookmarks to the Fragment Gallery

**Example**

```javascript
var bookmarks = Bookmark.PortableFragmentBookmark.getAllBookmarkInfosForApplication("group Identifier");

bookmarks.forEach(function(element, index) {
  FRAGMENTGALLERY_1.addItem(element);
});
```

**Example**

```javascript
var bookmarks = Bookmark.PortableFragmentBookmark.getAllBookmarkInfosForApplication("group Identifier");

FRAGMENTGALLERY_1.addItems(bookmarks);
```

**Note**

1. If components are displaying off-centre after being dropped into the *Split Cell*, you must make sure that the component positioning within the bookmarked container is absolute.
2. Images for the portable fragment bookmarks are automatically generated depending on the component types in the bookmarked container. You can also apply scripting to allow a user specify an image from their own repository.

**Related Information**

- Working with Split Cell Containers [page 150]
- Working with Online Composition [page 145]
25.3 Working with Split Cell Containers

The Split Cell container component enables users to place and arrange portable fragment bookmarks, dragged from the fragment gallery, in a tabular format. Therefore the Split Cell container can only be used with the Fragment Gallery component and the portable fragment bookmarks, which can be created using the corresponding API methods.

Note

The split cell container is not available for the SAP HANA mode.

Note

When portable fragment bookmarks are inserted into a splitcell container, the components contained in them may be renamed. In general, script references will be adapted to the new names. However, in case the name is given as a string, this renaming will not take place and could result in unexpected behaviour.

- working example: PAGEBOOK_1.setSelectedPageIndex(0)
- non-working example: PAGEBOOK_1.setSelectedPageByName("PAGE_1")

Adding content to a split cell container

The split cell container component is initially always empty. If you drag an item from the fragment gallery onto the split cell container, the content of the portable fragment bookmark that this item represents is placed in a cell. This cell is added to the container. To fill the split cell container with an initial cell, you can drop an item anywhere. Depending on where you drop any other item, the layout of the container will change. If you drop an item onto the top border of an existing cell, a new cell will be created above it. The same is true for left, right and bottom drops. You can also drop items onto the resize bar (see below) to create a cell between two cells. If you drop an item in the center of a filled cell, the content of this cell will be replaced.

Tip

Components that are stored in a fragment bookmark should be docked at all layout properties; set a numeric value for the left, top, right and bottom margin, and the value <auto> for the width and height. Otherwise the content of a split cell container cell can overlap with another cell.
Resizing cells

At runtime the user can freely resize the cells by moving the mouse pointer between two cells. A resize bar is displayed. When the user drags the resize bar, the cells adjacent to the bar are changed in size accordingly.

Deleting cells

If the user hovers with the mouse pointer over the top of a cell, a menu bar is displayed. To delete the cell, the user has to click the delete symbol in this bar.

Moving cells

The user can move cells by hovering with the mouse pointer over the top of a cell until the menu bar is displayed. By dragging the bar, the user can move the cells and drop them within the container as described above.
26 Working with Charts

You can add a chart component to an analysis application to display the data in a data source.

Context

Procedure

1. From the Components view, drag and drop the Chart component into the layout editor.
2. From the Outline view, drag and drop a data source onto the chart component. The chart appears according to the properties of the default chart.
3. Select the chart. In the Properties and Additional Properties views, you can modify and configure the chart properties.

26.1 Chart Settings

Several factors influence the structure and appearance of data in a chart.

To design a chart well, you need to take into account both the measures and dimensions of a data source. Assigning these as columns or rows has a great impact on the appearance of your chart.

Other important factors that influence a chart's appearance are the chart type and its properties in the Properties and Additional Properties views.

Related Information

Chart [page 268]
26.2 Configuring Initial Chart Settings

Use the initial chart settings tables to control the display of data in a chart.

Context

To work with columns and rows, use the Edit Initial View... dialog box to drag measures and dimensions to the Columns or Rows areas.

i Note

1. When measures and dimensions are added together in either the Columns area or the Rows area, we assume in the following tables that measures are always the last item.
2. In the Edit Initial View... dialog box, in the pane on the left, open the Measures node to see the full list of measures in your selected data source.
3. In Edit Initial View..., if you have an empty measure or measures only in the Columns area, the crosstab will display metadata in the form of headers, but no rows or cells. When you run the analysis application, the chart will display dimensions in the axes, but no data.
4. If you have an empty dimension or dimensions only in the Rows area, the crosstab will display metadata in the form of one column header for each dimension and rows with metadata. When you run the analysis application, the chart will display dimensions in the legend, but no data.
5. If there are empty dimensions in the Rows and Columns areas, the crosstab will display a column header for each of the dimensions in the rows and columns. The crosstab will have no data. When you run the analysis application, the chart will display the legend and axes, but no data.
6. Dual axis, scatter and bubble charts always require measures.

Procedure

1. In the Outline view, right-click a data source, and in the context menu, select Edit Initial View.
2. In the Edit Initial View... dialog box, from the pane on the left side of the screen, drag measures and dimensions to the Columns or Rows areas.
3. Do one of the following:
   ○ Choose the OK button to return to the layout editor.
   ○ Choose the OK + Create Crosstab button to create a crosstab and return to the layout editor.
4. Use the information in the chart tables below to configure your chart settings.
26.2.1 100% Stacked Bar

Use 100% stacked bar charts to display the percentage that each component contributes to a total across categories.

The table below contains the configuration chart display details for creating a 100% stacked bar chart. You can enter the details in the Edit Initial View... dialog box.

Note

Each stacked bar displays the stack segments as percentages of an entire bar, which always represents 100 in length.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures are displayed as stack segments in one bar, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One bar appears. Measures and dimensions appear as stack segments, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One bar appears for each measure. No stack segments appear.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Each measure appears as a bar. The dimensions appear as stack segments of each bar, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Each dimension or combination of dimensions appears as a bar. The measures appear as stack segments of each bar, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Each dimension or combination of dimensions in the Columns area appears as a bar. The measures and dimensions in the Rows area appear as stack segments of each bar, and in the legend.</td>
</tr>
</tbody>
</table>
26.2.2 100% Stacked Column

Use 100% stacked column charts to display the percentage that each component contributes to a total across categories.

The table below contains the configuration chart display details for creating a 100% stacked column chart. You can enter the details in the Edit Initial View... dialog box.

Note
Each stacked column displays the stack segments as percentages of an entire column, which always represents 100 in length.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Each measure appears as a bar. The bars are clustered for each dimension. No stack segments appear.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Each measure in the Columns area appears as a bar. Bars are clustered by dimensions in the Columns area. Each measure colors a group of bar clusters. Measures appear in the legend.</td>
</tr>
</tbody>
</table>

Table 6: 100% Stacked Column Chart Initial Settings
### 26.2.3 Area

Use area charts to emphasize an amount of change over a certain period or to show total values for a trend. When an area chart includes sums of values, you can see the relationship of the individual components to a whole.

The table below contains the configuration chart display details for creating an area chart. You can enter the details in the *Edit Initial View...* dialog box.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Each measure appears as a column. The dimensions appear as stack segments of each column, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Each dimension or combination of dimensions appears as a column. The measures appear as stack segments of each column, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Each dimension or combination of dimensions in the <em>Columns</em> area appears as a column. The measures and dimensions in the <em>Rows</em> area appear as stack segments of each column, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Each measure appears as a column. The columns are clustered for each dimension. No stack segments appear.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Each measure in the <em>Columns</em> area appears as a column. Columns are clustered by dimensions in the <em>Columns</em> area. Each measure colors a group of column clusters. Measures appear in the legend.</td>
</tr>
</tbody>
</table>
Table 7: Area Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures appear as a vertical line, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Measures and dimensions appear as a vertical line, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Measures appear on the label axis. The area from 0 on the value axis to the plotted measure amounts is filled with color.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Measures appear on the label axis. Dimensions are plotted from 0 on the value axis and each area is filled with a different color. Dimensions appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Dimensions appear on the label axis. Measures are plotted from 0 on the value axis and each is filled with a different color. Measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Dimensions in the <strong>Columns</strong> area appear on the label axis. Measures are clustered by dimensions and plotted from 0 on the value axis. Each measure area is filled with a different color. Measures and dimensions in the <strong>Rows</strong> area appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Measures are clustered by dimensions on the label axis. Measures are plotted from 0 on the value axis and the area is filled with color.</td>
</tr>
</tbody>
</table>
### 26.2.4 Bar

Use bar charts to display the differences between items. Bar charts are useful for displaying durations.

The table below contains the configuration chart display details for creating a bar chart. You can enter the details in the *Edit Initial View...* dialog box.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Measures are clustered on the label axis by the dimensions in the <em>Columns</em> area. Measures are plotted according to the dimensions in the <em>Rows</em> area. The measure areas are filled with color according to these row dimensions, whose colors appear in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Dimensions appear as bars for each measure. Each dimension has a color that is repeated for each measure. Dimensions and measures appear in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Each measure appears as a bar.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Bars are displayed in clusters. Each measure represents a cluster. Dimensions in the <em>Rows</em> area appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Bars are displayed in clusters. Each dimension represents a cluster. Measures appear in the legend.</td>
</tr>
</tbody>
</table>
26.2.5 Bar Combination

A bar combination chart combines the features of a bar chart and a line chart. Use bar combination charts to display the values in different categories.

The table below contains the configuration chart display details for creating a bar combination chart. You can enter the details in the Edit Initial View... dialog box.

Note
In the table below, in the Chart Display column, certain settings cause either dots or lines to be displayed. In these cases, you can display lines by adding more data in the Edit Initial View... dialog box, in the Rows or Columns area.

Table 9: Bar Combination Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Bars appear in clusters. Each dimension represents a cluster. The measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Bars appear as clusters of dimensions for each measure.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Bars appear as clusters of dimensions for each dimension in the Columns area. Each measure represents a cluster. Dimensions in the Rows area appear in the legend.</td>
</tr>
</tbody>
</table>

Table 9: Bar Combination Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>The first measure appears as a bar, and the subsequent measures as dots or lines.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>The first dimension appears as bars. Subsequent dimensions and measures appear as horizontally placed dots or lines that are colored according to the dimensions. Measures and dimensions appear in the legend.</td>
</tr>
</tbody>
</table>
26.2.6 Bubble

Bubble charts can display three or four measures of data. Each bubble is plotted using the first two values as its label-axis / data-axis location and the third and fourth as bubble size.

The tables below contain the configuration chart display details for creating a bubble chart. You can enter the details in the Edit Initial View... dialog box.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>A bar appears for each measure. No line appears.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Measures appear on the value axis. Dimensions appear as bars that are clustered by dimensions and measures. No lines appear. Dimensions appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>The first measure appears as bars. Subsequent measures appear as horizontal lines that are plotted by dimensions. Measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>The first measure appears as bars that are clustered according to dimensions. Subsequent measures appear as dots or lines.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>A bar appears for each measure. Measures are clustered by dimensions. No line appears.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Clusters of bars appear according to measures and dimensions. No line appears.</td>
</tr>
</tbody>
</table>

Note

When you are working with the bubble chart initial settings, bear the following in mind:

- You must use a minimum of three measures. The first measure becomes the label-axis setting, the second measure becomes the value-axis setting, and the third and fourth measures become the bubble width and height.
Location of the measures does not have any impact on the chart. However, you usually add measures in the *Rows* area. If you add measures in the *Columns* area, you must set the *Swap Axes* property to *true*.

You can add dimensions to the same area to which you have added measures.

The first dimension in the *Columns* area displays as different colors. If you add more dimensions, more bubbles appear in the chart.

The first dimension in the *Rows* area displays as different shapes. If you add more dimensions, these appear as extra bubbles or other shapes.

If you have less than three measures, a scatter chart is recommended.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Three measures selected.</td>
<td>The chart displays one bubble.</td>
</tr>
<tr>
<td>None selected.</td>
<td>Three measures selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>The chart displays one bubble.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Three measures selected.</td>
<td>The chart displays more than one bubble.</td>
</tr>
<tr>
<td>None selected.</td>
<td>Three measures selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>The chart displays more than one bubble.</td>
</tr>
</tbody>
</table>

**Note**

For this setting you must set the *Swap Axes* property to *true*.

### 26.2.7 Column

Use column charts to display the differences between items or to display durations.

The table below contains the configuration chart display details for creating a column chart. You can enter the details in the *Edit Initial View...* dialog box.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures are displayed as columns, and in the legend.</td>
</tr>
<tr>
<td>Column Dimensions</td>
<td>Column Measures</td>
<td>Row Dimensions</td>
<td>Row Measures</td>
<td>Chart Display</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Dimensions appear as columns for each measure. Each dimension has a color that is repeated for each measure. Dimensions and measures appear in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Each measure appears as a column.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Columns are displayed in clusters. Each measure represents a cluster. Measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Columns are displayed in clusters. Each dimension represents a cluster. Measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Columns appear in clusters. Each dimension represents a cluster. The measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Columns appear as clusters of dimensions for each measure.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Columns appear as clusters of dimensions for each dimension in the Columns area. Each measure represents a cluster. Dimensions in the Rows area appear in the legend.</td>
</tr>
</tbody>
</table>

### 26.2.8 Column Combination

A column combination chart combines the features of a column chart and a line chart. Use column combination charts to display the values in different categories.

The table below contains the configuration chart display details for creating a column combination chart. You can enter the details in the *Edit Initial View...* dialog box.
Note

In the table below, in the Chart Display column, certain settings cause either dots or lines to be displayed. In these cases, you can display lines by adding more data in the Edit Initial View... dialog box, in the Rows or Columns area.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>The first measure appears as a column, and the subsequent measures as dots or lines.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>The first dimension appears as columns. Subsequent dimensions and measures appear as vertically placed dots or lines that are colored according to the dimensions. Measures and dimensions appear in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>A column appears for each measure. No line appears.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Measures appear on the label axis. Dimensions appear as columns that are clustered by dimensions and measures. No lines appear. Dimensions appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>The first measure appears as columns. Subsequent measures appear as horizontal lines that are plotted by dimensions. Measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>The first measure appears as columns that are clustered according to dimensions. Subsequent measures appear as dots or lines.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>A column appears for each measure. Measures are clustered by dimensions. No line appears.</td>
</tr>
</tbody>
</table>
26.2.9 Dual Axis

Use dual axis charts to assign a different measure to each of the two vertical axes. This provides a broader view of the data. The following charts are available in dual axis mode:

- **Bar**
- **Column**
- **Combination**
- **Horizontal Combination**
- **Line**
- **Horizontal Line**

The **Dual Axis** chart data mapping works in the same way as for the single axis equivalent of the chart type with one exception; in order for the axis colors to match the displayed measures colors, measures cannot be in the **Columns** (x-axis). Put the measures in the **Legend**.

- You can set the color for each axis and you can change which measure shows on each axis.
- In combination dual axis charts, you can change the axis and the shape of a given measure.
- To create a dual axis chart, you enter the details in the **Edit Initial View**... dialog box.
- For more information, see the relevant chart type documentation for initial settings for each chart type.

**Note**

There need to be at least 2 measures in the initial view to display a dual axis chart. When there is 1 or no measure in the **Columns** or **Rows** area, no chart appears in the layout editor.

26.2.10 Horizontal Area

Horizontal area charts are similar to area charts. The difference is that they display the data area horizontally across the chart’s label axis. Use horizontal area charts to emphasize an amount of change over a certain period or to show total values for a trend. When a horizontal area chart includes sums of values, you can see the relationship of the individual components to a whole.

The table below contains the configuration chart display details for creating a horizontal area chart. You can enter the details in the **Edit Initial View**... dialog box.
<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures appear as a horizontal line, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Measures and dimensions appear as a horizontal line, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Measures appear on the label axis. The area from 0 on the value axis to the plotted measure amounts is filled with color.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Measures appear on the label axis. Dimensions are plotted from 0 on the value axis and each area is filled with a different color. Dimensions appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Dimensions appear on the label axis. Measures are plotted from 0 on the value axis and each is filled with a different color. Measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Dimensions in the Columns area appear on the label axis. Measures are clustered by dimensions and plotted from 0 on the value axis. Each measure area is filled with a different color. Measures and dimensions in the Rows area appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Measures are clustered by dimensions on the label axis. Measures are plotted from 0 on the value axis and the area is filled with color.</td>
</tr>
</tbody>
</table>
26.2.11 Horizontal Line

Horizontal line charts are similar to line charts. The difference is that they display the data area horizontally across the chart’s label axis. Use horizontal line charts to display trends in data at equal intervals. Also, horizontal line charts can be useful for displaying durations. In a horizontal line chart, the data label axis is vertical and the data value axis is horizontal.

The table below contains the configuration chart display details for creating a horizontal line chart. You can enter the details in the *Edit Initial View*... dialog box.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures appear as dots and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Measures appear as horizontally placed dots that are colored according to the dimensions. Measures and dimensions appear in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Measures appear on the label axis. Line is plotted according to values on the value axis.</td>
</tr>
</tbody>
</table>

**Note**

If a measure or dimension contains no value (null), the line will end at the previous value and begin again with the following value. The line will appear to be broken at the null point.
### 26.2.12 Horizontal Waterfall

Horizontal waterfall charts are similar to waterfall charts. The difference is that they display the data area horizontally across the chart’s label axis. Use horizontal waterfall charts to display how an initial value is affected by a series of intermediate positive or negative values.

The tables below contain the configuration chart display details for creating a horizontal waterfall chart. You can enter the details in the *Edit Initial View...* dialog box.

#### Table 15: Horizontal Waterfall Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures appear as vertical lines that are plotted by dimensions. Measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Dimensions in the <em>Columns</em> area appear on the label axis. Measures and dimensions in the <em>Rows</em> area appear as dots, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>A line is plotted according to measures, which are clustered by dimensions.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Dimensions are clustered by measures and appear as dots or lines. Measures appear in the legend.</td>
</tr>
<tr>
<td>Column Dimensions</td>
<td>Column Measures</td>
<td>Row Dimensions</td>
<td>Row Measures</td>
<td>Chart Display</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td>---------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>Measures in Columns area are displayed as bars according to the dimension in the Rows area.</td>
</tr>
<tr>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures in the Rows area are displayed as bars according to the dimension in the Columns area.</td>
</tr>
<tr>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Only the first measure in the Rows area is displayed as one bar.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>From the Columns area, a combination of the first measure and the first value of each dimension is displayed as a bar.</td>
</tr>
</tbody>
</table>

**Caution**

When the following settings are applied, no chart appears in the layout editor.

Table 16: Horizontal Waterfall Chart Initial Settings: Not Supported

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
</tbody>
</table>
Use line charts to display trends in data at equal intervals, or to display durations.

The table below contains the configuration chart display details for creating a line chart. You can enter the details in the Edit Initial View... dialog box.

**Note**

In the table below, in the Chart Display column, certain settings cause either dots or lines to be displayed. In these cases, you can display lines by adding more data in the Edit Initial View... dialog box, in the Rows or Columns area.

**Note**

If a measure or dimension contains no value (null), the line will end at the previous value and begin again with the following value. The line will appear to be broken at the null point.

### Table 17: Line Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>Two or more are selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>Two or more are selected.</td>
<td>None selected.</td>
<td>One is selected.</td>
<td>One or more selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Two or more are selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
</tbody>
</table>

Measures appear as dots or lines and in the legend.

Measures appear as vertically placed dots or lines that are colored according to the dimensions. Measures and dimensions appear in the legend.

Measures appear on the label axis. Line is plotted according to values on the value axis.
26.2.14 Multiple Pie

Use multiple pie charts to display several sets of data at the same time.

Note the following regarding multiple pie charts:

- Dimensions in the *Rows* area of the *Edit Initial View*... dialog box appear as pie slices in the chart.
- Dimensions in the *Columns* area of the *Edit Initial View*... dialog box appear as a vertical column of pie charts in the chart.
- Each measure in the *Columns* area in the *Edit Initial View*... dialog box appears as a horizontal row of pie charts in the chart.

The table below contains the configuration chart display details for creating a multiple pie chart. You can enter the details in the *Edit Initial View*... dialog box.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Measures appear on the label axis. Lines representing dimensions appear in different colors. Dimensions appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures appear as horizontal lines that are plotted by dimensions. Measures appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Dimensions in the <em>Columns</em> area appear on the label axis and are colored according to the dimensions in the <em>Rows</em> area. Measures and dimensions appear as dots or lines, and in the legend, according to the dimensions in the <em>Rows</em> area.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>A line is plotted according to measures, which are clustered by dimensions.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Dimensions are clustered by measures and appear as dots or lines. Measures appear in the legend.</td>
</tr>
</tbody>
</table>
### Table 18: Multiple Pie Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One column of pies without pie slices is displayed.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>A column of pie charts appears, one for each measure. Dimensions are displayed in the legend and as pie slices.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>A column of pie charts appears, one for each measure. No pie slices appear.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>A column of pie charts appears, with one pie chart for each measure. Dimensions appear as pie slices, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Rows and columns of pie charts appear according to the number of measures and dimensions. No pie slices are displayed.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Rows and columns of pie charts appear. The number of rows is determined by the number of measures, and the columns are determined by the dimensions in the Rows area. The dimensions in the Rows area appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Rows and columns of pies without pie slices appear according to the number of measures and dimensions.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Rows and columns of pie charts appear according to the measures and dimensions in the Columns area. These dimensions appear in the legend, and the pie charts are colored accordingly with no slices if there is only one dimension in the Rows area. If there is more than one dimension in the Rows area, they appear as pie slices.</td>
</tr>
</tbody>
</table>

### 26.2.15 Multiple Radar

Multiple radar charts display more than one radar chart, where each individual chart represents one measure.

The table below contains the configuration chart display details for creating a multiple radar chart. You can enter the details in the *Edit Initial View...* dialog box.

> **Note**

When you are working with the multiple radar chart initial settings, note the following:

- You must use at least two measures. The measures are plotted on the dimension radii, creating polygons.
- The more dimensions you add, the more radii appear in the chart.
- You cannot add dimensions to the same area that you have added measures.
- Usually, you add measures to the *Rows* area. If you add measures in the *Columns* area, you must set the *Swap Axes* property to `true`. 
Table 19: Multiple Radar Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
</table>
| One or more selected. | None selected. | None selected. | Two or more measures selected. | The chart displays multiple radar charts:  
  ● Each measure represents one individual radar chart.  
  ● The data for each measure is plotted around the radii of the chart.  
  ● Each dimension represents radii in the individual charts. |

26.2.16 Pie

Use pie charts to display the size of items in a data series relative to the sum of the items. Each item is displayed as a slice of the pie.

The table below contains the configuration chart display details for creating a pie chart. You can enter the details in the Edit Initial View... dialog box.

**Note**

When there are no measures in the Columns nor the Rows area, only a dot appears in the layout editor.

Table 20: Pie Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures are displayed as pie slices. Measures appear in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Measures and dimensions are displayed as pie slices.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Measures are displayed as pie slices.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Dimensions are displayed as pie slices.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Dimensions are displayed as pie slices, and in the legend.</td>
</tr>
</tbody>
</table>
### 26.2.17 Radar

Radar charts compare the cumulative values of more than one data series. A radar chart displays the data of measures on axes starting from the same point. The radar chart contains a sequence of spoke-like elements, called radii, with each radius representing one of the measures. A line connects the data values for each radius.

The table below contains the configuration chart display details for creating a radar chart. You can enter the details in the `Edit Initial View...` dialog box.

#### i Note

When you are working with the radar chart initial settings, note the following:

- You must use at least two measures. The measures are plotted on the dimension radii, creating polygons.
- You cannot add dimensions to the same area that you have added measures.
- Usually, you add measures in the `Rows` area. If you add measures in the `Columns` area, you must set the `Swap Axes` property to `true`.
- The more dimensions you add, the more radii appear in the chart.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Dimensions in the <code>Columns</code> area are displayed as slices, and appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Dimensions and measures appear as pie slices. Measures appear in the legend according to dimensions.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Measures appear as pie slices only if there are at least two dimensions in the <code>Rows</code> area. Otherwise, no pie slices appear.</td>
</tr>
</tbody>
</table>
26.2.18 Scatter

Scatter charts display a group of dots. Scatter charts can display two measures of data. Each dot is plotted using measure values as its label axis / value axis location.

The table below contains the configuration chart display details for creating a scatter chart. You can enter the details in the *Edit Initial View...* dialog box.

### Note

When you are working with the scatter chart initial settings, note the following:

- You must use two measures. The first measure becomes the label axis setting; the second measure becomes the value axis setting.
- Usually, you add measures in the *Rows* area. If you add measures in the *Columns* area, you must set the *Swap Axes* property to **true**.
- You cannot add dimensions to the same area that you have added measures.
- In rows, the first dimension sets the legend color.
- In columns, the first dimension sets the legend shape.
- Extra dimensions are used in the tooltip.
- The more dimensions you add, the more dots appear in the chart.
- For more complex data sources with measures and dimensions in columns and dimensions in rows, the result is that the dimensions map to the legend but are rendered as shapes. This chart is similar to the reverse scenario with measures and dimensions in rows and measures in columns, which are then rendered as shapes.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Three measures selected.</td>
<td>The chart displays one dot.</td>
</tr>
<tr>
<td>None selected.</td>
<td>Three measures selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>The chart displays one dot.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Three measures selected.</td>
<td>The chart displays more than one dot.</td>
</tr>
<tr>
<td>None selected.</td>
<td>Three measures selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>The chart displays more than one dot.</td>
</tr>
</tbody>
</table>

### Note

For this setting you must set the *Swap Axes* property to **true**.
26.2.19 Stacked Bar

Use stacked bar charts to display the relationship of each item to the whole, comparing the amount of each component to a total across categories.

The table below contains the configuration chart display details for creating a stacked bar chart. You can enter the details in the Edit Initial View... dialog box.

Table 23: Stacked Bar Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures appear as stack segments in one bar, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One bar appears. Measures and dimensions appear as stack segments, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One bar appears for each measure. No stack segments appear.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Each measure appears as a bar. The dimensions appear as stack segments of each bar, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Each dimension or combination of dimensions appears as a bar. The measures appear as stack segments of each bar, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Each dimension or combination of dimensions in the Columns area appears as a bar. The measures and dimensions in the Rows area appear as stack segments of each bar, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Each measure appears as a bar. The bars are clustered for each dimension. No stack segments appear.</td>
</tr>
</tbody>
</table>
26.2.20 Stacked Column

Use stacked column charts to display the relationship of each item to the whole, comparing the amount of each component to a total across categories.

The table below contains the configuration chart display details for creating a stacked column chart. You can enter the details in the Edit Initial View... dialog box.

Table 24: Stacked Column Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Measures appear as stack segments in one column, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One column appears. Measures and dimensions appear as stack segments, and in the legend.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One column appears for each measure. No stack segments appear.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Each measure appears as a column. The dimensions appear as stack segments of each column, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Each dimension or combination of dimensions appears as a column. The measures appear as stack segments of each column, and in the legend.</td>
</tr>
</tbody>
</table>
26.2.21 Stacked Waterfall

Use stacked waterfall charts to display how an initial value is affected by a series of intermediate positive or negative values. Chart elements for each data series are displayed as stack components. Each column represents the cumulative value of the components it contains.

The tables below contain the configuration chart display details for creating a stacked waterfall chart. You can enter the details in the Edit Initial View... dialog box.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Each dimension or combination of dimensions in the Columns area appears as a column. The measures and dimensions in the Rows area appear as stack segments of each column, and in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Each measure appears as a column. The columns are clustered for each dimension. No stack segments appear.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Each measure in the Columns area appears as a column. Columns are clustered by dimensions in the Columns area. Each measure colors a group of column clusters. Measures appear in the legend.</td>
</tr>
<tr>
<td>Column Dimensions</td>
<td>Column Measures</td>
<td>Row Dimensions</td>
<td>Row Measures</td>
<td>Chart Display</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>Measures in Columns area are displayed as columns according to the dimension in the Rows area. No stack segments appear.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>When you add measures to the Columns area, you must add either no dimensions or one dimension in the Rows area; otherwise the chart cannot be displayed.</td>
</tr>
<tr>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One is selected.</td>
<td>Measures in the Rows area are displayed as columns according to the dimension in the Columns area. No stack segments appear.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>When you add measures to the Rows area, you must add either no dimensions or one dimension in the Columns area; otherwise the chart cannot be displayed.</td>
</tr>
<tr>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Dimensions in the Columns area are displayed as columns. Measures in the Rows area appear as stack segments according to the dimensions in the Rows area, which also appear in the legend.</td>
</tr>
<tr>
<td>Column Dimensions</td>
<td>Column Measures</td>
<td>Row Dimensions</td>
<td>Row Measures</td>
<td>Chart Display</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>Dimensions in the Rows area appear as columns according to the values of the first measure. The columns are colored according to the dimensions in the Columns area, which also appear in the legend.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Two or more selected.</td>
<td>One is selected.</td>
<td>Dimensions in the Columns area are displayed as columns. In the Rows area, both dimensions of the measure are displayed as stack segments and appear in the legend.</td>
</tr>
<tr>
<td>One is selected.</td>
<td>One is selected.</td>
<td>One is selected.</td>
<td>None selected.</td>
<td>Dimensions in the Columns area appear as columns according to the values of the measure. The columns are colored according to the dimensions in the Rows area, which also appear in the legend.</td>
</tr>
<tr>
<td>Two or more are selected.</td>
<td>One is selected.</td>
<td>One is selected.</td>
<td>None selected.</td>
<td>Dimensions in the Rows area are displayed as columns. In the Columns area, the dimensions of the measure are displayed as stack segments and in the legend.</td>
</tr>
</tbody>
</table>

⚠️ Caution

When the following settings are applied, no chart appears in the layout editor.

Table 26: Stacked Waterfall Chart Initial Settings: Not Supported

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One is selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One is selected.</td>
<td>One is selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One is selected.</td>
<td>Two or more selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>One is selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>Two or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One is selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>Two or more selected.</td>
<td>None selected.</td>
<td>One is selected.</td>
<td>One is selected.</td>
<td>Chart is not displayed.</td>
</tr>
</tbody>
</table>
26.2.22 Waterfall

Use waterfall charts to display how an initial value is affected by a series of intermediate positive or negative values. Generally, waterfall charts are used for time or duration-related data, and horizontal waterfall charts are used for other types of data.

The tables below contain the configuration chart display details for creating a waterfall chart. You can enter the details in the Edit Initial View... dialog box.

Table 27: Waterfall Chart Initial Settings

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Measures in the Rows area are displayed as columns.</td>
</tr>
<tr>
<td>None selected.</td>
<td>One or more selected.</td>
<td>No more than one is selected.</td>
<td>None selected.</td>
<td>Measures in the Columns area are displayed as columns according to the dimension in the Rows area.</td>
</tr>
</tbody>
</table>

**Note**

When you add measures to the Columns area, you must add either no dimensions or one dimension in the Rows area; otherwise the chart cannot be displayed.
<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>No more than one is</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more</td>
<td>Measures in the <em>Rows</em> area are displayed as columns according to the dimension in the <em>Columns</em> area.</td>
</tr>
<tr>
<td>selected.</td>
<td></td>
<td></td>
<td>selected.</td>
<td></td>
</tr>
<tr>
<td>No more than one is</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more</td>
<td>Only the first measure in the <em>Rows</em> area is displayed as one column.</td>
</tr>
<tr>
<td>selected.</td>
<td></td>
<td></td>
<td>selected.</td>
<td></td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>No more than one is</td>
<td>None selected.</td>
<td>From the <em>Columns</em> area, a combination of the first measure and the first value of each dimension is displayed as a column.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>selected.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Caution**

When the following settings are applied, no chart appears in the layout editor.

<table>
<thead>
<tr>
<th>Column Dimensions</th>
<th>Column Measures</th>
<th>Row Dimensions</th>
<th>Row Measures</th>
<th>Chart Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>Two or more are selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>Two or more are selected.</td>
<td>None selected.</td>
<td>One is selected.</td>
<td>One or more selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>None selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
<tr>
<td>One or more selected.</td>
<td>One or more selected.</td>
<td>Two or more are selected.</td>
<td>None selected.</td>
<td>Chart is not displayed.</td>
</tr>
</tbody>
</table>
26.3 Configuring Additional Chart Types

A selection of Additional Chart Types can be selected from the Chart Type property dropdown list. Use the Additional Charts dialog box to populate these additional chart types.

Context

You have created an application and wish to select and populate an additional chart type.

Note

- If you select and populate a chart type from the Additional Chart Types dropdown list, it is recommended that you reconfigure your axes using the Additional Charts dialog box. You should not reconfigure your axes by changing the value of the Swap Axes property in the Properties tab. Changing the Swap Axis property here may result in your chart being rendered incorrectly.
- If you have created and installed new chart extensions using the SAP Lumira SDK, these extensions will appear in the Additional Chart Types dropdown list. For more information about creating SAP Lumira SDK extensions, see the SAP Lumira Visualization Extensions Developer Guide on the SAP Help Portal at http://help.sap.com/lumira. Extensions developed with the SAP Lumira SDK can only be deployed to the BI platform and SAP NetWeaver. They cannot be deployed to SAP HANA.
- The Time Based Line chart is only available after you install the Real-Time package. For more information, you can refer to the chapters called “Working with Real-Time Dashboards” and “Time Based Line Chart” in this guide.
- If you have dimensions in rows and/or columns in the Initial View Definition pane, the Chart Population pane must reflect the identical assignment of dimensions. For example, if Column Dimensions contains the dimension COUNTRY and Row Dimensions contains the dimension REGION, the dimensions COUNTRY and REGION must be assigned in the Chart Population pane. If, for example, the dimension COUNTRY is not assigned in the Chart Population pane, only the data from the first member of the dimension COUNTRY will display in your chart. To ensure that your Initial View Definition pane only displays the dimensions you need for your chart, you must ensure that the dimensions in your Edit Initial View are assigned appropriately.

To configure Additional Chart Types, you must follow the steps below.

Procedure

1. In the Properties view, under Chart Type, select the last element in the dropdown list called Additional Chart Types.
2. In the Additional Charts dialog box, select the required Chart Type from the dropdown list.
3. From the Initial View Definition pane on the left side, move measures, columns and rows to the Chart Population pane on the right side of the dialog box. To move measures, columns and rows into the correct area within the Chart Population pane, use the Assign Measure here or Assign Column or Row Axis here prompts.
4. If the *Initial View Definition* pane has dimensions in rows and columns, the *Chart Population* pane should also contain the same rows or columns in an axis.

5. The assignment of measures, rows and columns is based on the additional chart type you have selected.

6. Choose |Apply| and |OK|.

7. The additional chart types currently available by default include:
   - **trellis**: A trellis chart is a series or grid of small similar graphics or charts, allowing them to be easily compared. The following are trellis chart types:
     - bar
     - column
     - line
     - area
     - horizontal line
     - horizontal area
   - **heat map**: A heat map is a chart that represents by colors, the individual values contained in a matrix.
   - **treemap**: Treemaps display hierarchical (tree-structured) data as a set of nested rectangles. Each branch of the tree is given a rectangle, which is then tiled with smaller rectangles representing sub-branches.
   - **time based line**: Time based line charts decouple the plot area from the X-Axis, allowing the spacings in the plot area to be independent from the spacings on the axis. The data shown in the plot area is placed according to its time dimension rather than the index in the data. For more information, you can refer to the chapters called “Working with Real-Time Dashboards” and “Time Based Line Chart” in this guide.

**Results**

You have configured an additional chart type.

**Related Information**

*Installing CVOM Chart Extensions for SAP BusinessObjects Design Studio* [page 233]
*Deploying SDK Extensions* [page 232]
*Working with Real-Time Dashboards* [page 194]
26.4 Conditional Formatting

You can apply conditional formatting rules to measures or dimension members within a selected chart. These rules change the appearance of the chart when specific conditions are met. You can set conditional formatting rules within the Chart Properties > Conditional Formatting.

Context

Your analysis application contains a chart to which you wish to apply conditional formatting rules.

To apply conditional formatting to your chart, follow the steps below:

Procedure

1. Within the Layout Editor, select the chart to which you wish to apply conditional formatting.
2. Within the Properties panel, choose the Conditional Formatting ellipsis button. The Conditional Formatting dialog box opens.
4. In the Name textbox, enter a title to identify your new conditional formatting rule.
5. Within the Rule Description area of the Conditional Formatting dialog box, select either Measure or Dimension Member from the dropdown list. The dropdown list is populated from the data set you have applied to your chart.
6. There are three types of conditional formatting rules that you can apply to your chart. Select one of the following:
   - is equal to
   - is greater than
   - is less than
7. Input the appropriate value to be assigned to the new rule.
8. Within the Preview area of the dialog box, choose the Format button to assign a color to your rule. Choose a color from the pre-defined color picker or define a custom color by choosing the Define Custom Colors button. You can then save your custom color to the color picker Choose OK. The color you selected is previewed in the Preview area.
9. Choose OK. Your new rule is listed in the Conditional Formatting dialog box. The Conditional Formatting property value field is populated with the JSON text.
10. To edit existing conditional formatting rules, highlight the rule to be edited and choose the Edit Rule button. You can also activate the edit function by doubleclicking the rule within the Conditional Formatting dialog box.
11. To delete existing conditional formatting rules, highlight the rule to be deleted and choose the Delete Rule button.
12. Use the up-down arrows to change the order in which the rules appear in the list. The rules are applied to the chart in the order in which they appear in the Conditional Formatting dialog box list.

Results

You have applied conditional formatting to your chart.

Related Information

Chart [page 268]
27 Working with Geo Maps

You can add a Geo Map analytic component to an analysis application to allow users to display different layers of geographical information on a map.

Context

Depending on how you configure the Geo Map properties and apply scripting, each layer displays different information when the user interacts with it. Each layer is connected to a different data source.

You must connect to data sources with geo-specific dimensions or measures, for example, countries, longitude and latitude.

Note

You must read the following SAP Note before using the Geo Map component:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2166923</td>
<td>Working with SAP BusinessObjects Design Studio 1.5 Geo Maps.</td>
</tr>
</tbody>
</table>

Procedure

1. From the Components view, drag and drop the Geo Map component into the layout editor.
2. From the Outline view, add the data sources you wish to work with in your Geo Map into the Data Sources folder.
   The added data sources are listed in the Data Source dropdown list in the Additional Properties tab.
3. Before the user can interact with the layers of Geo Map, you must configure the Edit Initial View... for each data source to ensure that only one geo dimension appears in the Rows. The measures should appear in the Columns area.
4. Select the Geo Map component in the layout panel. In the Properties and Additional Properties views, you can configure the Geo Map properties. For more information on each of these properties, you can refer to the chapter in this guide called Geo Map.
5. You can allow users to interact in multiple ways with the different Geo Map layers by using the onSelect(); scripting method. For more information on interacting with the Geo Map, you can refer to the chapter in this guide called Interacting with Geo Maps.
27.1 Geo Map Settings

To create a well-designed Geo Map, you must use data sources containing geo-specific dimensions or measures and you need to have a very good idea of how you would like to display your data. You can choose to display data as shapes (polygons and multi-polygons), points or bubble charts.

Other important factors that influence a Geo Map’s appearance are the selected properties in the Properties and Additional Properties views.

27.2 Configuring Initial Geo Map Settings

Context

You must use the Edit Initial View.. to configure the initial settings for the Geo Map.

Note

- There must be only one geo-specific dimension located in the Rows area of the Edit Initial View... of your SAP BW or SAP HANA query. The measures should appear in the Columns area.
- If your geo dimension SAP BW attributes contain longitude and latitude, longitude and latitude are assigned to your map layer based on these BW attributes. Your SAP BW attributes, should then only be used to get longitude and latitude values and should not be used elsewhere. If your geo dimension does not contain latitude and longitude BW attributes, then longitude and latitude values for your Geo Map should be contained in a measure. It is not possible to display shapes based on SAP BW attributes, as there can be many duplicates. Shapes can only be based on dimension members, not SAP BW attributes.

Procedure

1. In the Outline view, right-click a data source, and in the context menu, select Edit Initial View...
2. In the **Edit Initial View...** dialog box, from the pane on the left side of the screen, drag one geo dimension to the **Rows** area.

3. If your SAP BW attributes contain longitude and latitude, expand the geo dimension and its attributes. Right click on the underlying longitude and latitude and select **Add**.

4. Drag the geo dimension to the **Rows** pane.

5. You should hide totals. To do this, right click on the dimension, select **Totals Display > Hide Totals**.

6. Select the **Geo Map** in the layout pane. In the **Properties** tab, enter the URL for your selected basemap in the **Basemap URL** property. It is also an option to leave this blank, if no basemap is required.

7. If required by the basemap provider, you should enter a reference to the relevant basemap copyright information.

8. The **Map Legend Visible** property is set to true by default. If you wish to hide the legend on all your **Geo Map** layers, you can set this property value to false.

### 27.3 Configuring Geo Map Layers

Once you have selected your data sources, configured your initial view and set the basemap url, basemap copyright (when required) and set your map legend visibility in the **Properties** tab, you can now begin to add layers to your **Geo Map**. Adding or removing layers can be done by using the relevant buttons in the **Additional Properties** tab. This layered approach to adding data to your **Geo Map** allows the application user to drill down into and interact with the data visualization. Each layer has a unique ID. You can show, hide, rename, and remove the different layers in your **Geo Map**. You can also change the order in which the layers are applied to the **Geo Map** by using drag and drop in the **Additional Properties** tab. When you then run the application, the **Geo Map** legend reflects the new order of the layers. For each layer, you can select, whether you want to display data in the form of shapes, points or charts. Data sources are assigned to individual layers of your **Geo Map**.

#### Note

- When configuring each layer, you must ensure that the **ID** property contains valid css selectors. A name must begin with an underscore, a hyphen, or a letter, followed by any number of hyphens, underscores, letters, or numbers. If the first character is a hyphen, the second character must be a letter or underscore. The name must be at least two characters long.

- When scripting the **Geo Map**, the scripting methods `getSelectedLayer();` and `getSelectedMember();` should be used together for optimal results.

### Binding Data Sources

You can bind different data sources to different layers of your **Geo Map**. One data source is assigned per layer. However, the same data source can also be used in different layers. Data source selection can be accessed from the **Geo Map Additional Properties**. You can also use the **Data Selection** ellipsis button if you want to have a more fine-grain control over the selection of data in the data source. For example, if you are looking at a layer in the **Geo Map** that displays capital cities in the world, you might want to restrict the data selection. When you select your data source, you need to ensure that in the **Edit Initial View...** you have a geographic dimension in the **Rows** area.
The measures should appear in the **Columns** area. For more information on using the **Data Selection** property, you can refer to the chapter called **Chart** within the **User Interface Reference** section.

**GeoJSON**

GeoJSON is a standard format for encoding collections of simple geographical features. A GeoJSON file contains a list of polygons, each of which contains a list of properties or metadata that define the polygons and a list of the coordinates, that will be drawn on the map. You need to define how to match the dimension member to the polygon within the GeoJSON file. You have the option of using a custom GeoJSON file, which can be used in local mode, or uploaded onto your BI platform or SAP NetWeaver platform. This file is then accessed from the **Custom GeoJSON File** property within the **Additional Properties** tab. If you do not specify a custom GeoJSON file, a default GeoJSON file containing countries is used. It contains a standard set of properties such as ISO names and full names. For example, ‘DE’ is the ISO name for ‘Germany’. In the **GeoJSON Mapping Property** within the **Additional Properties** tab, you will see by default, sample measure names aligned with the corresponding GeoJSON name. You can use this dropdown list to select the appropriate GeoJSON property name for use in your **Geo Map**. You can use the **GeoJSON Mapping Type** to determine how the dimension members will be mapped by selecting either key or text.

**Related Information**

Chart [page 268]

**27.3.1 Configuring a Shapes Layer**

**Context**

A shape layer or choropleth layer is a collection of polygons or multi-polygons on a map, for example a country, region, sales region etc. In the **Geo Map**, the shape layer is the default layer type. To display shapes in a **Geo Map**, a shapes layer needs to reference a GeoJSON file containing the required shapes. You can use a custom GeoJSON file or you can use the default shapes provided with Design Studio. Shapes are based on dimension members in the **Edit Initial View**... For example, if you want to display the states of a country, the dimension 'State' should be in the **Rows** area.

**Procedure**

1. Select the **Geo Map** component in the **Layout** pane and select the **Additional Properties** tab.
   A shapes layer is open by default.
2. If you have already added a layer, and wish to add another one, select the **Add Layer** button.
   A new layer is opened above the previous layer.
3. Provide a unique ID for your shapes layer in order to make it easy to identify when all layers are collapsed.

4. The *Show Layer* checkbox is selected by default. If you want this layer to be invisible to the user online, you must uncheck this checkbox.

5. Select the data source you wish to assign to this shapes layer. Once you have selected the data source, you can also use the *Data Selection* ellipsis button to further restrict or modify the number of measures that appear in the dropdown list for selection for the *Geo Map*.

6. Select from the *Measure* dropdown list, the measure you wish to assign to the shape layer. Measures determine the color of the shape to be represented on the *Geo Map*.

7. Select the shape color gradient using the *Start Color* and *End Color* color pickers or by entering HEX values. The color gradient is displayed on the map based on the assigned measure value from the data source specified. The color properties are black and white by default. The highest value will use the *End Color*.

   The shapes in this geo map layer will display on the *Geo Map* in color selected.

8. If you are using a custom GeoJSON file in local mode, the file must be copied to the following application folder: *C:\Users\<username>\Analysis-workspace\com.sap.ip.bi.zen\repository\<my_app>\shapes.geojson*. This filename must then be referenced in the *Custom GeoJSON File* property.

9. If you are on the BI platform or the *SAP NetWeaver* platform, you can select the ellipsis button within the *Custom GeoJSON File* property to select the location of your custom GeoJSON file.

10. You can use the *GeoJSON Mapping Property* dropdown list to select the property in the GeoJSON file that will be used to display the shape.

11. You can specify how the dimension members will be mapped by selecting either key or text from the *GeoJSON Mapping Type* property.

**Related Information**

Configuring Geo Map Layers [page 188]

### 27.3.2 Configuring a Points Layer

**Context**

You can display data as precise locations on a *Geo Map* points layer using values for longitude and latitude. It is necessary that the data source assigned to the layer contains latitude and longitude. To ensure that latitude and longitude coordinates are included in your *Geo Map*, the SAP BW attributes must be displayed in the *Edit Initial View...* If the values for longitude and latitude are in the measures, the geographic dimension should be in the *Rows*.

There are two options available to you when assigning data to a points layer in your *Geo Map*. You can assign them from the *Latitude* and *Longitude* measure list in the *Geo Map Additional Properties*. This displays dimension members as measures representing longitude and latitude as custom or default points on a map. Or, they can also be assigned from the SAP BW attributes in the *Edit Initial View...*
Procedure

1. Select the Geo Map component in the **Layout** pane and select the **Additional Properties** tab. A shapes layer is open by default.

2. If you have already added a layer, and wish to add another one, select the **Add Layer** button. A new layer is opened above the previous layer.

3. Change the layer type to a points layer type by selecting the center icon. The **Additional Properties** are changed to refer only to the points layer type.

4. Provide a unique ID for your points layer in order to make it easy to identify when all layers are collapsed.

5. The **Show Layer** checkbox is selected by default. If you want this layer to be invisible to the user online, you must uncheck this checkbox.

6. **Note**

   The longitude and latitude will not appear in the **Data Selection** dialog box, if you select the ellipsis button beside the **Data Source** property.

   Select the data source you wish to assign to this points layer. Once you have selected the data source, you can also use the **Data Selection** ellipsis button to further restrict or modify the number of measures that appear in the dropdown list for selection for the Geo Map.

7. Select from the dropdown lists, the longitude value for the **Longitude** property and the latitude value for the **Latitude** property. The longitude and latitude both appear for the two dropdown lists for the longitude and latitude above of the data selection made.

8. Select a color to customize the color of the points or markers on the Geo Map.

Related Information

Configuring Geo Map Layers [page 188]

27.3.3 Configuring a Charts Layer

Context

In the context of the Geo Map component, charts layers are similar to points layers, but with a measure assigned to them. The data source assigned to a charts layer must contain longitude and latitude like points or reference a GeoJSON file. If longitude and latitude are present, the bubble charts will be displayed at these points or coordinates. If a GeoJSON file is referenced, the bubble charts will be positioned in the middle of the shape. The bubble size changes, depending on the measure. Bubble charts are the only chart type available for the Geo Map.
Procedure

1. Select the Geo Map component in the Layout pane and select the Additional Properties tab.
   A shapes layer is open by default.
2. If you have already added a layer, and wish to add another one, select the Add Layer button.
   A new layer is opened above the previous layer.
3. Change the layer type to a charts layer type by selecting the last icon.
   The Additional Properties are changed to refer only to the charts layer type.
4. Provide a unique ID for your charts layer in order to make it easy to identify when all layers are collapsed.
5. The Show Layer checkbox is selected by default. If you want this layer to be invisible to the user online, you must uncheck this checkbox.
6. Select the data source you wish to assign to this charts layer. Once you have selected the data source, you can also use the Data Selection ellipsis button to further restrict or modify the number of measures that appear in the dropdown list for selection for the Geo Map.
7. If you want to display the chart as a location, in the Longitude and Latitude dropdown menus, select the relevant measure, containing latitude and longitude values.
8. Select a color to customize the color of the bubbles on the Geo Map.
9. In the Measure dropdown menu, select a measure.
10. If you want to display the chart at the center of a shape or polygon, and are using a custom GeoJSON file in local mode, enter the filename in the Custom GeoJSON property. For BI platform and SAP NetWeaver, you can select the location of your custom GeoJSON file within the Additional Properties tab. You can use the GeoJSON Mapping Property dropdown list to select one of the available properties to be mapped to the geospecific dimension available in the rows of the data source. For example, a GeoJSON property could be "NAME".
11. You can specify whether key or text of the geo dimension member is used when mapping to the GeoJSON file by selecting from the GeoJSON Mapping Type property.

Related Information

Configuring Geo Map Layers [page 188]

27.4 Interacting with Geo Maps

Through scripting and working with the Geo Map properties, you can allow the application user to interact with the map in a number of ways.
Drilldown into different layers

You can script the Geo Map so that when the user clicks on the map, the map returns a different layer. This hides or shows a layer based on an OnSelect event and allows the user to drilldown into the map layers to reveal different data.

Change the basemap

You can script a component in the Geo Map to allow the user to select a different basemap while running the application.

Center the map

You can script the Geo Map to change the center the map around the data contained in a layer, based on where the user clicks on the map. This repositions the map, so that the area that contains the data becomes the center of the map.

Pan and zoom

The user can use the mouse and/or the default zoom buttons to pan across the entire map and to zoom in and out on the selected area on the map.

Tooltips and legends

As the user hovers over different areas on the map, the tooltip changes to display the geo-specific dimension and its corresponding measure value. If you have scripted the layers to show the legend, this information is also displayed in the legend.
28 Working with Real-Time Dashboards

The Real-Time extension package consists of components that support the ability to create visualizations with streaming data (push based), but also allow users to create visualizations, which have a near Real-Time connection to SAP HANA or SAP BW (pull based).

There are many applications for the Real-Time package including in the area of operational systems, where data may only be relevant as it occurs and where it may not be necessary to persist the data, for example utility companies monitoring smart buildings and the financial capital markets. The real-time package contains the following:

- Design Studio extension for an SAP ESP (Event Stream Processor) data source, which allows streaming data from SAP ESP to be connected to standard charts.
- Time Based Line chart as a chart extension which supports time axis scaling as well as a number of other formatting properties.
- A Timer custom component. The Timer custom component allows users to create near Real-Time visualizations using the OOTB Chart component while connected to SAP HANA or SAP BW.

Note

Before working with Real-Time feature, you should refer to the following SAP Note:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2169103</td>
<td>Working with the Design Studio Real-Time Add-On.</td>
</tr>
</tbody>
</table>

Related Information

- How to Create Real-Time Dashboards with Streaming Data Sources [page 194]
- Streaming Data Source Additional Properties [page 198]
- Time Based Line Chart [page 200]
- Timer [page 321]

28.1 How to Create Real-Time Dashboards with Streaming Data Sources

A streaming data source component allows you to connect to the streaming data source SAP ESP and then connect to Design Studio charts to create visualizations of these streams of Real-Time data. SAP ESP enables you to create and run your own complex event processing (CEP) applications to derive continuous intelligence from streaming event data in real time.
Context

You have an instance within SAP ESP 5.1 (SP09) and a project is running. You have the Real-Time package installed in Design Studio.

Note

For more details on the Additional Properties of the Streaming Data Source, you can refer to the chapter in this guide called “Streaming Data Source Additional Properties”.

Procedure

1. Within Design Studio go to Outline View Data Sources.
2. Right-click to view the option to select Add Custom Data Source.
3. Select Streaming Data Source.
4. In the Properties view, select Additional Properties for the Streaming Data Source component. Three Additional Properties tabs appear. The first tab is called ESP Configuration. It enables you to configure your Streaming Data Source.
5. Configure your Streaming Data Source by entering the Host and REST port on which your SAP ESP project is running.
6. Enter your user SAP ESP password and username credentials. Use your ESP Studio username and password. The Connect button will be enabled once you have entered values for all the configuration and credential fields.
7. Select Connect. The Credentials icon will turn green indicating that a connection to the Streaming Data Source has been made.

Note

If credential values are incorrect, the Credentials icon will turn red. The tooltip over this icon will provide you with details of the status of the connection.

8. Workspace, Project and Topic are automatically defined based on the projects running in SAP ESP. You can modify these values if required. Your Workspace refers to the location of all your stored projects. The projects stored in that workspace will be loaded into the Projects drop down list. A project can contain multiple topics.
9. If the Topic you selected is a stream, you must set the Retention Policy. This option will allow you to specify the maximum number of rows that the Chart will display.
10. Configure the Update Interval (ms) property to set the frequency with which new data updates are displayed. The default setting is 1,000 milliseconds.
11. Select the second Additional Properties tab, which is called Measure/Dimension Configuration. This tab displays the measures and dimensions from your SAP ESP project.
12. Review the data types and update measure or dimension configuration if required.
13. Select the third Additional Properties tab, which is called Data Selection for Rows and Columns. This tab displays the data selection for your rows and columns.
14. Select your measures, rows and columns to configure your query for the Streaming Data Source initial view.
15. Select a Chart component and add to your canvas.
16. Assign the Streaming Data Source to your selected Chart component.
17. If you wish to view the data at design time, select the Live Preview checkbox in the Data Selection for Rows and Columns tab for the Streaming Data Source.
18. You can change the Chart type at anytime to get a modified view of your data.

Related Information

Streaming Data Source Additional Properties [page 198]
Configuring Additional Chart Types [page 182]
Working with Real-Time Dashboards [page 194]

28.2 How to Create Real-Time Dashboards with a Single Pull-Based Data Source

Context

You wish to have your dashboards update at a set interval rather than every time there is a new event. You can create near Real-Time visualizations when connected to SAP HANA or SAP BW (pull based) data sources by using the Timer custom component in conjunction with the standard out of the box charts within Design Studio. The Timer custom component allows you to set an interval in milliseconds, at which point it runs a user-defined script. This script can be used to reload a single data source.

To reload a single data source, follow the steps below:

Procedure

1. Add a Chart component onto your canvas and assign a data source. The data source can be either SAP BW or SAP HANA.
2. Add the Timer, which is located under Customer Components.
   You will see a Timer icon on your canvas area. This icon can be hidden through the properties of the Timer component. To hide the Timer icon, change the value of the Show Icon in Application property to false.
3. Within the Timer properties, change the Interval in Milliseconds property to your chosen interval value.
4. To make the Timer reload the data source, begin by opening the Script Editor for the Timer components event On Timer.
5. To reload the data source use the reloadData(); scripting method in the Script Editor. For example, DS_1.reloadData();
6. To start the Timer, configure the On Startup property of the application. Add to the Script Editor for the On Startup event to start the timer component. For example, TIMER_1.start();
Once executed, this dashboard will now update the dashboard by the defined number of seconds you set for the **Timer** component property **Interval in Milliseconds**.

**Related Information**

Assigning a Data Source to a Component [page 79]
Timer [page 321]

### 28.3 How to Create Real-Time Dashboards with Multiple Pull-Based Data Sources

**Context**

You have multiple data sources with multiple chart types in a dashboard that needs to update at regular intervals. It may not be necessary to load the data at regular intervals as it might not contain new data. To increase performance you can implement a separate data source that executes quickly and determines if there is new data. For example, this “trigger” data source can return a single row, containing the Id of the last entry inserted into the database. This Id can be compared to a previous value. If they are different, the other data sources should be reloaded.

To reload two data sources, follow the steps below:

**Procedure**

1. Add two **Chart** components on to your canvas and assign a data source to each. The data source can be either SAP BW or SAP HANA.
2. Set up and add a trigger data source, which has only a single measure containing the Id of the last inserted event.
3. To keep track of the last ID of the trigger data source, add a **Text** component to the outline. This component can have the **Visible** property set to **false**.
4. Add the **Timer**, which is located under **Custom Components**. You will see a **Timer** icon on your canvas area. This icon can be hidden through the properties of the **Timer** component by setting the value of the **Show Icon in Application** property to **false**.
5. Within the **Timer** properties, change the **Interval in Milliseconds** property to your chosen interval value.
6. Since reloading a slower performing data source may impede user interaction, move the reloading to the background processing. To move the reloading to the background processing, open the **Script Editor** for the **Timer** components event **On Timer** and use the `doBackgroundProcessing();` scripting method. For example: `TIMER_1.stop(); APPLICATION.doBackgroundProcessing();`
7. To start the **Timer** component, configure the **On Startup** property of the application. Add to the **Script Editor** for the **On Startup**. For example: `TIMER_1.start();`
8. Configure the *On Background Processing* event for the application within the *Script Editor*.

Once executed, this dashboard will now check if there are updates for the dashboard by the defined number of seconds you set for the *Timer* component property *Interval in Milliseconds* and make updates if changes have been found.

### Example

```javascript
DS_TRIGGER.reloadData();
var oldID = TEXT_1.getText();
var newID = DS_TRIGGER.getDataAsString("MAX_EVENT_ID", {});
if(oldID != newID)
{
    DS_1.reloadData();
    DS_2.reloadData();
    TEXT_1.setText(newID);
}
TIMER_1.start();
```

### Related Information

- How to Create Real-Time Dashboards with a Single Pull-Based Data Source [page 196]
- Working with Real-Time Dashboards [page 194]
- Timer [page 321]

### 28.4 Streaming Data Source Additional Properties

The *Streaming Data Source* additional properties are described in the table below.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuration</td>
<td>host, port</td>
<td>Host and REST port on which your project is running.</td>
</tr>
<tr>
<td>Property Name</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>security</td>
<td>● HTTPS</td>
<td>The default value is True</td>
</tr>
<tr>
<td>credentials</td>
<td>● username</td>
<td>ESP Studio username and password.</td>
</tr>
<tr>
<td></td>
<td>● password</td>
<td>Connects to SAP ESP instance.</td>
</tr>
<tr>
<td></td>
<td>● Connect button</td>
<td></td>
</tr>
<tr>
<td>datasource</td>
<td>● workspace</td>
<td>ESP projects are stored within the workspace.</td>
</tr>
<tr>
<td></td>
<td>● project</td>
<td>Projects within a workspace.</td>
</tr>
<tr>
<td></td>
<td>● topic</td>
<td>Retention Policy: In SAP ESP a topic is either a stream or a window. If the topic you select is a stream, a Text field will appear. You will be required to specify the Retention Policy for a stream. This option will allow the you to specify the maximum number of rows that the Chart will display. If a new event occurs when the chart is already displaying its maximum number of rows, the oldest row will be deleted in order to make room. If the topic you select is a window the Retention Policy text field will not appear as SAP ESP handles the retention policy of windows.</td>
</tr>
<tr>
<td></td>
<td>● retention policy</td>
<td>Update Interval (ms): The interval for displaying new events in milliseconds. By default the display updates with new data every second.</td>
</tr>
<tr>
<td></td>
<td>● update interval (ms)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A list of Data Types</td>
<td>Currently SAP ESP has no concept of a measure or a dimension. As a results of this, some default rules have been applied to decide if a column is a measure or dimension. These rules are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● If a Column is Primary Key in SAP ESP it is a dimension.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● If a column is a number in SAP ESP it is a measure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Any remaining columns are dimensions.</td>
</tr>
</tbody>
</table>

Note
You will also be able to configure some columns.
### Property Name | Property Value | Property Description
--- | --- | ---
measures | Primary Values | A number or quantity that records a directly observable value or performance. They typically represent some time dependent measure of the stream being visualized, for example, stock price, temperature, pressure.
dimensions | Rows, Columns | A collection of related data members, which represents one aspect of a business. In relation to streaming data sources, time would be a typical dimension.
live preview | Checkbox | To preview the live stream you must select the Live Preview checkbox. Until you select Live Preview, any charts connected to the data source will show no data available. Live Preview will not persist and if connection is lost or the chart is closed and reopened it must be selected again.

**Note**
Live Preview is unchecked by default.

---

**Related Information**

[Working with Real-Time Dashboards](#) [page 194]
[How to Create Real-Time Dashboards with Streaming Data Sources](#) [page 194]

### 28.5 Time Based Line Chart

The *Time Based Line* chart is a chart extension to which the streaming data source can be connected. Effectively the chart decouples the plot area from the X-Axis, allowing the spacing in the plot area be independent from the spacing on the axis. The data shown in the plot area is placed according to its time dimension rather than index in the data.

*Time Based Line* charts can handle null values. If there is a break in the stream of data for a period of time, the *Time Based Line* chart will represent this on the plot area. It joins the points between the null, but it also shows clearly to the user that a period of time has passed. Standard *Line* charts cannot handle null values. Standard *Line* charts will always evenly space the points on the axis, not based on the time.

The *Time Based Line* chart is one of the main features of the Real-Time Package that can be installed in Design Studio. For more information, you can refer to the chapter in this guide called *Working with Real-Time Dashboards*. 
The *Time Based Line* chart properties are described in the table below.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Name</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot Area</td>
<td>• Scrolling</td>
<td>• Scrolling: Enables the scrolling of the axis in relation to time.</td>
</tr>
<tr>
<td></td>
<td>• Scrolling Time Scale</td>
<td>• Scrolling Time Scale: If the previous scrolling property is enabled, sets the time scale that the chart shows. For example, if set to &quot;1 Minute&quot;, the scrolling chart shows the previous 1 minute of data.</td>
</tr>
<tr>
<td></td>
<td>• Scrolling Time Zone Offset</td>
<td>• Scrolling Time Zone Offset: If the scrolling property is enabled, this sets the time-zone offset of the incoming data. By default the chart tries to figure this out for itself. For example, if the incoming data is GMT+5 and its time code does not contain the time zone, the time-zone offset should be -5 Hours.</td>
</tr>
<tr>
<td>Time Axis</td>
<td>• Major Ticks</td>
<td>• Major Ticks: Sets the time scale of the x-axis major ticks. There are also options for &quot;None&quot;, where no major ticks are shown, and &quot;Auto&quot; where, depending on the time scale of the chart, determines the best scale for the major ticks.</td>
</tr>
<tr>
<td></td>
<td>• Major Gridline Visible</td>
<td>• Major Gridline Visible: Sets the visibility of vertical gridlines where there are major ticks.</td>
</tr>
<tr>
<td></td>
<td>• Major Gridline Type</td>
<td>• Major Gridline Type: Sets the type of gridlines shown on the major ticks.</td>
</tr>
<tr>
<td></td>
<td>• Major Gridline Size</td>
<td>• Major Gridline Size: Sets the width of the gridlines shown on the major ticks.</td>
</tr>
<tr>
<td></td>
<td>• Minor Ticks</td>
<td>• Minor Ticks: Sets the time scale of the time-axis minor ticks. The same options that are available for the major ticks are available for the minor ticks.</td>
</tr>
<tr>
<td></td>
<td>• Minor Gridline Visible</td>
<td>• Minor Gridline Visible: Sets the visibility of vertical gridlines where there are minor ticks.</td>
</tr>
<tr>
<td></td>
<td>• Minor Gridline Type</td>
<td>• Minor Gridline Type: Sets the type of gridlines shown on the minor ticks.</td>
</tr>
<tr>
<td></td>
<td>• Minor Gridline Size</td>
<td>• Minor Gridline Size: Sets the width of the gridlines shown on the minor ticks.</td>
</tr>
<tr>
<td></td>
<td>• Label -&gt; Time Format</td>
<td>• Label -&gt; Time Format: Sets the format of the time value labels on the major ticks.</td>
</tr>
</tbody>
</table>
Related Information

Working with Real-Time Dashboards [page 194]
How to Create Real-Time Dashboards with Streaming Data Sources [page 194]
How to Create Real-Time Dashboards with a Single Pull-Based Data Source [page 196]
Streaming Data Source Additional Properties [page 198]
29 Advanced Design Tasks

29.1 Scripting for User Interaction Enablement

SAP BusinessObjects Design Studio is a design tool for creating interactive analysis applications. To enable interactivity, you write scripts that are executed when the user performs an action in the executed application. For example, you can place the button component in the application and assign a script to the button’s On Click event.

Scripts consist of one or more statements written in a JavaScript-based language that follow a specific syntax. For writing scripts the design tool offers two approaches: the statement wizard and the script editor. All objects, methods and expression types are listed in the API reference.

Script language

You write scripts in the BI Action Language (BIAL). BIAL is a true subset of the JavaScript/ECMAScript standard described here: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf. The scripts are executed on the Analysis Application Design Service (unlike JavaScript that is executed in the Web browser). Scripts have a clear definition of the supported language constructs, objects and methods.

Scripts, statements and events

Scripts consist of statements. Each statement is typically written in a single line. The execution of scripts is triggered by user interaction with the component. This interaction is also referred to as an event and the script executed in response to this event is the event handler. Each component has its own set of one or more events that are displayed in the Properties view of the component.

➤ Remember

Events on components are only triggered by user interaction and not implicitly by scripting. For example, when changing the selection in a list box by using the script LB.setSelectedItem(), the On Select event of the list box is not triggered. An exception to this rule is the On Result Set Changed event, which can be triggered implicitly by a script method like setFilter() or reloadData().

SAP BusinessObjects Design Studio currently supports the following statement types:

- call statements
- conditional execution statements
- assignment statements
Syntax of call statements

Call statements execute an API method of an object and have the following format:

```
<Component>.<method>(<arguments>);
```

<Component> is the name of a data source alias or component in your application, for example, DS_1 or Button_1.

<method> (also referred to as method) is an operation that is applied to the object specified on the left of the period. The available functions depend on component type. DataSourceAlias objects, for example, provide functions to filter data, and visual components provide functions to modify visibility, enablement, etc.

<arguments> is a comma-separated list of expressions. The passed expressions must match the requirements of the method.

Each statement ends with a semicolon (";").

You can use functions as arguments for other methods as long as the return type matches the argument type.

Syntax of conditional execution statements

Conditional execution statements have one of the following formats:

- first format

  ```
  if (<condition>) {
  <sequence of statements to execute when condition is met>
  }
  ```

- second format

  ```
  if (<condition>) {
  <sequence of statements to execute when condition is met>
  } else {
  <sequence of statements to execute when condition is NOT met>
  }
  ```

<condition> is a Boolean expression - one of the following:

- true or false as value literals (constants)
- a method call statement returning a Boolean value
- a comparison for equality in the form `a == b` or
- a comparison for inequality in the form `a != b`
- multiple conditions combined with `&&` and `||`, parentheses are optional
Assignment statements

Assignment statements have one of the following formats:

- `var <variable> = <expression>;`
  This format defines a script variable (see Script Variables in this chapter) and assigns the result of an expression to this script variable.

- `<variable> = <expression>;`
  This format assigns the result of an expression to a script variable that has been already defined.

Methods and object types

Methods are operations that are applied to the object specified on the left of the period in the statement. The available methods depend on the object type. For example, DataSourceAlias objects provide methods to filter data. Visual components have methods to modify visibility, position and so on. SAP BusinessObjects Design Studio supports methods that depend on the following object types:

- data source alias (DataSourceAlias)
- application (Application)
- component (Component)
- individual components like button, text, image, etc. (Button, Text, Image, etc.)
- information objects (ApplicationInfo, DataCell, Convert, DataSourceInfo)

Expressions

Expressions compute a result (also referred to as returned value), similar to a formula. Expressions can appear as method arguments and as conditions (if-statements or conditional execution statements). Results of an expression have a type. An expression may consist of literals (String constants, integer numbers, Arrays, JSONs), operators, method calls and parentheses.

Operators supported by BIAL:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Argument Type</th>
<th>Result Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Concatenates strings</td>
<td>String, (Integer, Float, Boolean)</td>
<td>String</td>
<td>&quot;ab&quot;+&quot;cd&quot; (=&quot;abcd&quot;)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Boolean and Integer will be converted to String automatically)</td>
<td></td>
<td>&quot;ab&quot;+1 (=&quot;ab1&quot;)</td>
</tr>
<tr>
<td>+</td>
<td>Adds two integer values or floating point</td>
<td>Integer, Float</td>
<td>Integer</td>
<td>1+2 (=3)</td>
</tr>
<tr>
<td>-</td>
<td>Subtracts two integer values or floating point</td>
<td>Integer, Float</td>
<td>Integer</td>
<td>3-2 (=1)</td>
</tr>
<tr>
<td>*</td>
<td>Multiplies two integer values or floating point</td>
<td>Integer, Float</td>
<td>Integer, Float</td>
<td>3*2 (=6)</td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td>Argument Type</td>
<td>Result Type</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>/</td>
<td>Divides one integer value by the other or one floating point by the other</td>
<td>Integer, Float</td>
<td>Integer, Float</td>
<td>8/2 (=4)</td>
</tr>
</tbody>
</table>
| ==       | Checks if the two operands are equal | Any | Boolean | 1 == 1 (=true)  
"a" == "b" (=false) |
| !=       | Checks if the two operands are not equal | Any | Boolean | 1 != 2 (=true)  
"a" != "a" (=false) |
| &&       | Logical AND | Boolean | Boolean | true && false (=false)  
true && true (=true)  
if (<condition1> && <condition2>)  
{  
<statements>  
}  
Statements will be executed if both conditions are true. |
| ||       | Logical OR | Boolean | Boolean | true || false (=true)  
false || false (=false)  
if (<condition1> || <condition2>)  
{  
<statements>  
}  
Statements will be executed if at least one of the conditions is true. |
| !        | Logical NOT | Boolean | Boolean | !true (=false)  
!false (=true)  
if (! <condition>) {  
<statements>  
} |
Expression type system

Expression results can be used as method arguments. The expression type must match the argument type. In some cases, BIAL will automatically convert between types. This is typically only done for certain strings, where BIAL can check that the string value is valid (see below). The +Operator automatically converts Integer and Boolean arguments to String. In all other cases, the type system is strict and error messages are displayed.

There are four different types:
- primitive types (String, Integer, Boolean, Float)
- BI types (DataSourceAlias, Dimension, Measure, …) enable technically appropriate input help for API method arguments in the script editor. Such special types are sometimes written in Enum, Array, or JSON format (or even a combination of these). To view examples, refer to the API reference.
- component types (Button, Text, Pagebook…)

The visual component types correspond to the list of components in the Components view of the design tool.

Enums

In many cases, the input of a function can be a fixed set of values. This set of values is referred to as an enumeration or simply "Enum". Enums are BI types. An Enum value is written as <EnumType>.<EnumValue>, for example, "ChartType.PIE".

Script Variables

Script variables store expression results. They are useful, for example, to store intermediate results that are used repeatedly later in a script. Script variables have a name composed of the characters "A"-"Z", "a"-"z", "0"-"9", and
The name must not begin with a digit character “0”-“9”. Like expressions, script variables have a type. There are local and global script variables:

- **Local script variables**
  
  Local script variables are script variables that can be used only in the script in which they were defined, but not in other scripts. To define a local script variable, open a script with the Script Editor and add a line of the following format:

  ```
  var <variable> = <expression>;
  ```

  The type of the script variable is automatically determined by the type of the expression.

- **Global script variables**
  
  Global script variables are script variables that can be used in any script of your application. To define a global script variable, first click the application in the Outline view. Next click the item **Scripting > Global Script Variables** in the Property sheet. Click the button .... This opens the **Edit Global Script Variable** dialog box. You can insert a new global script variable and define its name, type, and default value. In addition you can decide if the global script variable is a URL parameter. If you decide to make it a URL parameter, then you can set the value of this global script variable by adding the global script variable and its value as a URL parameter to the applications URL.

  **Note**

  Script variable names used as a URL parameter must start with a capital x (X) and must not end with an underscore and a digit (_<digit>).

### Methods calling the event handler

To avoid event handlers calling each other an infinite number of times (which could result in runtime crashes), events on components are only triggered by user interaction on the component and not by scripting. For example, if a user swipes the pages of a pagebook, the On Select event is raised, and the relevant event script is executed. In contrast, if the selected page index is set by script using the method setSelectedPageIndex of the pagebook, the On Select event is not raised. Another example is if a user selects a value from a dropdown box, the On Select event is raised and the event script is executed. In contrast, if the selected item is set using the setSelectedValue method of the dropdown box, the event is not raised. However there are situations when you want to execute the relevant event script without duplicating script code. In these cases, each component with an event (On Click, On Select) has a corresponding method that allows the event handler to be called from another handler’s code. For example, the statement `BUTTON_1.onClick();` calls the **On Click** event handler of the button component with the name BUTTON_1. Another example is the statement `TABSTRIP_1.onSelect();` that calls the **On Select** event handler of the tabstrip component with the name TABSTRIP_1.

### Example

You have created an application with a dropdown box DROPDOWN_1 and a button BUTTON_1. If the user selects a value in the dropdown box, the selected value filters the dimension MYDIM in the data source aliases DS_1 and DS_2. For this scenario, the **On Select** event script of the dropdown box looks like this:

```javascript
DS_1.setFilter("MYDIM", DROPDOWN_1.getSelectedValue());
DS_2.setFilter("MYDIM", DROPDOWN_1.getSelectedValue());
```
If the user pushes the button, a specific value in the dropdown box should be selected programmatically using the same logic defined for user interaction. In other words, the selected value filters the dimension MYDIM in the data source aliases DS_1 and DS_2. Instead of duplicating the script code of the On Select event of DROPPDOWN_1, you add the following statements in the On Click event handler of the button:

```
DROPDOWN_1.setSelectedValue("MYDIMVALUE");
DROPDOWN_1.onSelect();
```

The benefits of calling the onSelect method increase as you start to work with more use cases in the application (where you want to set the selected value programmatically) and more statements in the On Select event handler of the dropdown box.

**Member key format**

The most frequently used API methods deal with dimension members in the context of setting filters or variable values. Often the filter or variable values that need to be set are returned from the selection of a UI component.

In SAP Business Information Warehouse (BW) there are multiple key types; the most important are the INTERNAL and the EXTERNAL key formats. The INTERNAL key is a unique identifier for all users (user-locale independent), whereas the EXTERNAL key can be user-locale dependent.

By default, all methods that deal with member keys (as returned values or as parameters) work with the INTERNAL key format. This keeps applications language-independent to ensure that they can work for multilingual user groups. However, you might want to choose the EXTERNAL key format due to various situations or reasons:

- The EXTERNAL key format is more compact and either the application is known to be only used by users sharing the same locale, or the EXTERNAL key format is known to be locale-independent.
- The number of complex selections is more than the single values that need to be passed as parameters for the API methods setFilter or setVariableValue. In this case, the SAP Business Explorer selection syntax ("INPUT_STRING") can be used, which references individual members by their EXTERNAL key.

For these special use cases, SAP BusinessObjects Design Studio provides Ext variants of methods, for example setFilterExt, setVariableValueExt.

**Writing script statements**

In the design tool, there are two different approaches for writing script statements:

- You can write scripts in a free form fashion with the script editor and activate the content assistance at any time by clicking `CTRL + Space`.
- If you activate the content assistance right from the beginning, you can also use the statement wizard. This guides you through the necessary steps and then creates a script statement.

The statement wizard is functionally limited compared to free form scripting. However, if you are unfamiliar with scripting, it might be helpful to use the statement wizard and learn from the generated script statement.
API Reference

In the API reference, you will find all the methods and their descriptions.

Related Information

Using the Script Editor [page 210]
Using the Statement Wizard [page 214]

29.1.1 Using the Script Editor

Prerequisites

Before you can use the script editor, you must add the components that enable user interaction (such as button or pagebook) to your application. Also, you might have to add the appropriate data sources and assign them to the components, if this is required by your scenario.

Context

Procedure

1. Click the On Click or On Select property of the corresponding component that enables user interaction.
2. Click the Edit the script button next to the On Click or On select property.
   The script editor opens and displays the name of the property (event) (example: "On Click") and the component to which the script will be assigned (example: "Button_1").
3. Type in one or more statements in this supported format:
   <ComponentVariable>.<function>(<arguments>);
   
   Tip
   You can activate the content assistance at any place in the script by pressing \text{CTRL} + \text{Space} on your keyboard. If you activate the content assistance right from the beginning, you can also switch to the statement wizard that guides you through the statement creation process.
4. Click OK to close the script editor.
Results

You have created a script for a component that enables the user to interact in the application.

Related Information

Content Assistance in the Script Editor [page 211]
Error Analysis in Scripting [page 212]
API Reference [page 323]

29.1.1.1 Content Assistance in the Script Editor

The script editor enables you to easily create scripts and use them for user interaction enablement. It supports you by providing a variety of useful functions, hints and checks:

Opening the script editor

You open the script editor by editing the On Click or On Select property of the component that you have dragged and dropped into the editor.

Automatic syntax check

The system checks the syntax of your script in the background. Whenever there is a mistake, you will see an error marker on the left hand side and the bad code is underlined. You can hover over the error marker or the underlined text to see the error message.

Auto correction

In some cases you will find a “Quick fix” that fixes the error automatically. Just check the correction links in the error message popup.

Display of additional information

When you hover over the script text with the mouse and keep the mouse pointer still, you see additional information about the text below the mouse pointer. For example, if you hover over a data source alias name, you
see the corresponding query or query view data source. Or if you hover over a dimension, you see its display name.

**Intelligent proposal offering by content assistance**

When entering the script text, you can press $CTRL + Space$ to see proposals on how to continue. The editor is an intelligent tool and knows which continuations make sense and which do not. If you press $CTRL + Space$ in an empty script editor, you will see all the components and data source aliases of the application as proposals. You can select one and continue typing the script or press $CTRL + Space$ again to get the next proposal.

In some cases the editor will show you proposals without you pressing $CTRL + Space$. For example, if you type the name of a data source alias and press $\cdot\cdot\cdot$, all available functions for the data source alias will be displayed automatically.

**Relevance-ordered proposals and context-sensitive documentation**

The proposals are ordered by relevance. The most likely proposals are displayed first. Proposals that have the same relevance are sorted alphabetically. If you select a proposal with the mouse or with the up/down keys, you will see some documentation about the proposal. For example, if you show proposals for a data source alias “DS_1.”, you see all available functions. Each function displays documentation.

**Linked mode and value help for member selection**

If you have chosen a proposal for a function, the editor will automatically insert the function name, the opening and closing brackets “(...)” and placeholders for all arguments. The editor is now in the “linked mode” that allows you to enter the parameters. You can use the Tab key to switch to the next parameter and the Shift key to go back to the previous parameter. You will also recognize the little green bar behind the closing bracket. Proceeding to this bar using the tab, arrow keys or the mouse will exit the “linked mode”. To fill the parameters, you press $CTRL + Space$ again to see proposals for parameters.

Proposals for parameters depend on the function. The function “setFilter” will propose you dimensions in place. Members will be shown as proposals if the available number is not too high. Otherwise there will be a proposal that allows you to open the value help for member selection.

**29.1.1.2 Error Analysis in Scripting**

**Script validation**

Each script is validated before it is executed. If an error is found, for example, in the syntax, the script is not executed. Instead a log entry and a message are created and displayed. The log entry in the Error Log view
contains detailed error messages for the script. At runtime, the end user will see a message in the message view stating that there is a problem in the script. The message contains a log ID that allows you - the application designer - to find detailed error information.

For performance reasons, validation is performed only once per script. However, a script with errors is never executed.

As an application designer, you can use the following application methods to analyze the general behavior of your application:

- log
- alert
- createErrorMessage
- createWarningMessage
- createInfoMessage

If you use the methods log and alert, the results are displayed in the design tool. For example, if you want to find out whether or not a certain script has been aborted, you can add the log or alert call statements with a user-defined message as the last line of the script. If the message does not appear in a dialog box (application method alert) or in the Error Log view of the design tool (application method log), this means the script has been aborted. You can add more messages to the script, to find the exact statement that causes the script to be aborted.

If you use the methods createErrorMessage, createWarningMessage or createInfoMessage the results of the analysis are displayed in the executed application and can be viewed on external machines or mobile devices (for example, iPads).

**Note**

The log and alert methods only work if the design tool is running. Messages will only appear on the machine where the design tool is installed and running and where user confirmation is expected. Therefore, any applications executed on an external machine or a mobile device are blocked if they encounter an alert statement in a script (assuming that you have activated external access for these applications on the preferences page).

**Using the Problems view**

The Problems view displays script errors encountered during script validation at design time, after the application was saved. The Problems view shows errors and warning markers for all script-related issues, even if the script editor is currently not open. To display the Problems view, click View Problems in the menu of the design tool. To examine and resolve the errors displayed in the view, simply double-click an error. This automatically selects the component with the bad script and the script editor is opened.

The Problem view is refreshed when you open and save an application. Therefore a marker can remain even if the problem is already fixed. This marker disappears when you save the application the next time.

**Note**

Differences between working in local mode and with the Business Intelligence platform:
If you work locally, the marker remains when you close the application window or the design studio. You can double-click the marker to open the application window and navigate to the error. The Problems view displays all problems for all analysis applications in the local workspace.

If you work with the BI platform, the markers of an application disappear when the application is closed. They are displayed again when the application is opened again. Therefore the Problems view only shows markers of open applications.

Related Information

Log (log) [page 337]
Alert (alert) [page 330]
Create Error Message (createErrorMessage) [page 333]
Create Info Message (createInfoMessage) [page 333]
Create Warning Message (createWarningMessage) [page 334]

29.1.2 Using the Statement Wizard

Prerequisites

Before you can use the script editor or the statement wizard, you must add the components that enable user interaction (such as button or pagebook) to your application. Also, you might have to add the appropriate data sources and assign them to the components, if this is required by your scenario.

Context

The statement wizard guides you through the script statement creation process, step by step.

Procedure

1. Click the On Click or On Select property of the relevant component that enables user interaction.
2. Click the Edit the script button next to the On Click or On select property.
   The script editor opens and displays the name of the property (event) (for example, "On Click") and the component to which the script will be assigned (for example, "Button_1").
3. Press [CTRL] + [Space] to activate the content assistance.
4. Double click New Statement Wizard... in the content assistance window.
   The New Statement dialog box opens.
5. Select a statement from the available statements list and click Next.
Tip

You can also filter for statements by typing in the first characters of the statement name in the Available Statements field. The system automatically filters the statements while you are typing. The matching parts of the statement name are highlighted in the list. Click the eraser icon on the right side of the filter field to clear the filter. The eraser icon is only displayed after you have typed in characters in the filter field.

6. Set the parameters for your chosen statement and click Finish.
   Click Back, to return to the previous step.
   The statement you have created is displayed in the script editor.
7. Click OK to close the script editor.

Results

You have created a script statement using the wizard. Note the following: You can modify the script afterwards in the script editor. However, if you make changes to a wizard-generated script, which are not supported by the wizard, the wizard cannot modify the statement anymore.

29.1.3 Selecting Members of a Dimension

When you use statements like setFilter for a dropdown box, you need to select single members of a dimension. You can select the members in the content assistance of the Script Editor dialog box, or, if you use the statement wizard, in the Select Member dialog box. The capabilities and options available in the Select Member dialog box depend on the type of the member’s dimension. Dimension types currently supported:

- flat dimensions
- dimensions with a related hierarchy

The capabilities and options available in the Select Member dialog box also depend on the number of members that a dimension has. You can display a small or large quantity of members by adjusting the maximum threshold number of members. You can set this (default) threshold in the Preferences dialog box in the design tool.

Restriction

With input help for member selection, you can only select single members. Multiple selections or ranges are not supported at present.

Filtering members in flat dimensions

If the number of members is less than or equal to the maximum threshold number defined in the Preferences dialog box, the Select Member dialog box displays the members list with each member’s text and key. You can now filter the members by typing a part of a member’s text or key into the filter/input field. The list is filtered
while you type. Matching parts of member names or keys are highlighted in the list. The number of matches is displayed below the list.

**Tip**

Filtering is case-insensitive. You can use the following wildcard characters for filtering:

- An asterisk (*) matches any sequence of zero, one or multiple characters.
- A question mark (?) matches a single character.

If your filter entry matches several members, the first member is always selected. You can use the currently selected member by pressing **OK**, double-clicking the selected member, or pressing **Enter** on your keyboard.

You can discard the currently selected member by pressing **Cancel** or by pressing **ESC** on your keyboard.

To clear the filter field, click the eraser symbol on the right side of the filter field. This will display all members again. The eraser symbol only appears if you have entered characters in the filter field.

### Changing the sorting display and the threshold

You can switch the display sequence for the member text and key in the members list by selecting **Sort by key** or **Sort by text** in the popup menu. This menu appears when you click the arrow symbol above the input field. The system sorts the list automatically according to your choice.

You can define and change the maximum number of members displayed in the content assistance for the *Script Editor* dialog box and in the *Select Member* dialog box. Enter the required number in the **Maximum number of members to fetch from backend in content assistance** checkbox and/or in the **Maximum number of members to fetch from backend in dialog**. The default threshold is 20. If the number of available members exceeds this threshold number, the content assistance does not list single members. Instead it offers the **Select Member...** entry, which opens the *Select Member* dialog box.

### Searching for members

If the number of members exceeds the maximum threshold number, the *Member Selection* dialog box displays the members list with each member’s text and key. The number of displayed members is cut off at the maximum threshold number. A message below the list indicates that the maximum threshold number of members has been exceeded and provides a link to the *Preferences* menu in the design tool, where you can configure this threshold number.

To search for members, first enter a part of a member’s text or key into the search field. This enables the **Search** button. Then click the **Search** button or press **Enter** to perform the search.

**Tip**

The case-sensitivity of the search functionality is dependent on the backend system.
Filtering and searching for members in dimensions with related hierarchies

If the number of members and the hierarchy nodes is less than or equal to the maximum threshold number, the Select Member dialog box displays the members in a hierarchical tree. The initial expansion level is determined by the corresponding setting in the BW query. Depending on how you close the dialog (table view or hierarchical view), the dialog opens in the mode again the next time it is opened. The total number of members is displayed below the tree.

If the number of members assigned to the hierarchy is less than or equal to the threshold, but the total number of members exceeds the threshold, then only the assigned nodes are displayed in the tree. The unassigned members are suppressed, and a dummy node is displayed for them instead. You can only filter for assigned members. If you want to filter for unassigned members, you need to switch to the leaves/table view.

If the number of assigned members exceeds the threshold, the Select Member dialog box initially displays the collapsed members tree. A message below the tree indicates that the maximum threshold number has been exceeded. You can now collapse and expand the nodes.

To expand or collapse the nodes and the entire hierarchy, press Expand All (+) or Collapse All (-).

You can now filter the members as described above. You can also change the sorting display of the member keys and texts, adjust the threshold number and display the hierarchy leaves as a table.

29.1.4 Script Processing Flow in Applications

The following graphic describes the processing flow for scripts that you create and insert into your analysis application.

Figure 2: Script Processing Flow in Applications
While executing an event script, the system checks on the server if there are mandatory variables in the application without values, or if a variable is set to a wrong value. If this is true for one of these cases, the prompt dialog box for entering the values appears. After the application user has entered the variable values, or if there are no mandatory variables or if the values are correct, the components are rendered before background processing is triggered. If there is a script in the application that should be processed in the background, the system checks again whether there are mandatory variables in the application without values or whether a variable is set to a wrong value. As a result, all the steps described above are repeated again. This can cause an endless loop. If there is no script to be executed in the background, the event script is finally executed.

### Note

If you use the method `APPLICATION.doBackgroundProcessing()` in the event *On Background Processing*, you create a loop. This can be useful, for example, when loading data from different data sources - one after the
other. However, to avoid an endless loop, you must integrate a condition (in your script) that terminates the loop. For example, you can specify the number of calls by using a script like this: In this case, a global variable counter has been defined and initialized with 0. After the third call of the On Background Processing event, no further calls are executed. For more information, see On Background Processing event in Properties of the Application [page 257]

```java
counter = counter + 1;
if (counter < 3 ) {
  APPLICATION.doBackgroundProcessing();
}
```

### 29.1.5 Process Flow at Application Start

This graphic describes the overall process flow in applications. It contains information about which application elements are initialized at a certain point in time, when scripts and dialog boxes are displayed and under what certain circumstances.

**Figure 3: Process Flow at Application Start**
First, the data sources are initialized on the server. This is followed by the event execution of **On Variable Initialization**. After executing **On Variable Initialization**, the system checks on the server if prompts have to be forced, if there are mandatory variables without values, or if a variable is set to a wrong value. If one of these cases is true, the prompt dialog box for entering values appears. After the application user has entered the variable values, or if there are no prompts to be forced, or if there are no mandatory variables, or if the values are correct, the application is initialized on the server and the **On Startup** event is executed. While executing the **On Startup** event, the system checks on the server if there are mandatory variables in the application without values, or if a variable is set to a wrong value. If one of these cases is true, the prompt dialog box for entering values appears. After the application user has entered the variable values, or if there are no mandatory variables, or if the values are correct, the components are rendered before background processing is triggered. If there is a script in the application that should be processed in the background, the system again checks whether there are mandatory variables in the application without values or whether a variable is set to a wrong value. As a result, all the steps described above are repeated again. This can cause an endless loop. If there is no script to be executed in the background, the event script is finally executed.

### 29.1.6 Business Cases

#### 29.1.6.1 Setting a Crosstab Member as a Filter on Another Component

**Context**

The following business case example describes how to select a member from a crosstab, how to use the member as a filter for another component and how to jump from one tab to another tab in a tabstrip.

In this example, you create an application with two tabs in a tabstrip. In the first tab TAB_1 you embed a crosstab MAIN_CROSSTAB with data source DS_1. In the second tab TAB_2 you embed a chart MAIN_CHART with data source DS_2. The two data sources should have common dimensions that can be used for selection. In this example, both data sources have the dimension "country".

In the MAIN_CROSSTAB, you allow users to select a member and set this member as a filter on the MAIN_CHART. You also allow users to jump from TAB_1 to TAB_2.

For this scenario perform the following steps:

**Procedure**

1. Use a tabstrip component with two tabs (TAB_1 and TAB_2).
2. Embed a crosstab component in TAB_1, name the crosstab MAIN_CROSSTAB and assign the data source DS_1 to the crosstab. Embed a chart component in TAB_2, name the chart MAIN_CHART and assign the data source DS_2 to the chart.
3. To enable user interaction in the crosstab MAIN_CROSSTAB by using events, set the property **Enable Selection to True** and write the following script for the On Select event of the crosstab:

   ```javascript
dS_2.setFilter( "country" , MAIN_CROSSTAB.getSelectedMember( "country" ));
TABSTRIP_1.setSelectedTabIndex(1);
```
29.1.6.2 Using Two Buttons with Toggle Function

Context

In this example use case, you create an application with two different buttons - one below the other. Each button triggers a different function when the user pushes the button. The first button BUTTON_FILTERON allows the user to see the filter setting area, whereas the second BUTTON_FILTEROFF allows the user to hide the filter setting area. The buttons are embedded in the text component TOOLBAR_BACKGROUND_1 that represents the background color of the buttons. A second text component TOOLBAR_BACKGROUND_2 represents the background color for the filter area and is only displayed when the user clicks on BUTTON_FILTERON. The filter setting area itself is embedded in the grid layout component TOOLBAR_LAYOUT.

For this scenario perform the following steps:

Procedure

1. Use a text component as the background color for the buttons (TOOLBAR_BACKGROUND_1).
2. Place two buttons, one above the other, inside the text component.

   ➤ Tip
   To match the exact size and position of the two buttons, you can copy BUTTONFILTER_ON (by using its context menu in the Outline view) and paste it under the Layout folder in the Outline view.

3. Use another text component for the background color of the filter setting area (TOOLBAR_BACKGROUND_2).
4. Create your filter settings area within the grid layout component TOOLBAR_LAYOUT.

   ➤ Note
   For easier layout structuring, embed the two text components in a cell of the grid layout component (TOOLBAR_LAYOUT).

5. To enable user interaction, write a script for the On Click events of the buttons.

   For BUTTONFILTER_ON:
   ```javascript
   TOOLBAR_BACKGROUND_2.setVisible(true);
   TOOLBAR_LAYOUT.setVisible(true);
   BUTTON_FILTEROFF.setVisible(true);
   BUTTON_FILTERON.setVisible(false);
   ```

   For BUTTONFILTER_OFF:
   ```javascript
   TOOLBAR_BACKGROUND_2.setVisible(false);
   TOOLBAR_LAYOUT.setVisible(false);
   BUTTON_FILTERON.setVisible(true);
   BUTTON_FILTEROFF.setVisible(false);
   ```
29.1.6.3 Swiping in Pages

Context

In this example use case, you create an application containing a pagebook (PAGEBOOK_1) with four pages (PAGE_1, PAGE_2, PAGE_3, PAGE_4). The user switches between the pages by swiping the pages horizontally. Set the relevant properties as described below:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected page index</td>
<td>0 (specifies the visible page of the pagebook. In this case the visible page is the first page.)</td>
</tr>
<tr>
<td>transition effect</td>
<td>slide in</td>
</tr>
<tr>
<td>transition direction</td>
<td>horizontal</td>
</tr>
<tr>
<td>page caching</td>
<td>none</td>
</tr>
</tbody>
</table>

Note

The first page of the pagebook has the page index 0, the second page has the index 1, the third page has the index 2, and so on. By setting the selected page index on 0, the pagebook starts with the first page.

29.1.6.4 Switching Pages by Clicking Images

Context

In addition to the scenario described in Swiping in pages, you want to enable the user to switch between pages by clicking images. You also want to display a toolbar in the header of the application that changes the text (for example, from Page 1 to Page 2 or to Page 3 or to Page 4) whenever the user clicks the image of a different page. Each image also has a special text (in this example, it is logical to use the texts Page1... Page4). Therefore you add four image components (PAGEICON_1, PAGEICON_2, PAGEICON_3, PAGEICON_4) to your application. Each component represents the corresponding page of the pagebook and each component has its own text component. You also add a text component to the header of the application. When the user clicks on an image, the relevant page is displayed and at the same time the text in the toolbar changes accordingly.

To configure this layout scenario proceed as follows:

Procedure

1. Prepare two images for each page (one for the selected image and one for the normal image). Save the images in the image folder or subfolder of the application directory.
2. Create a text component for the toolbar title (here TEXT_TOOLBAR_TITLE).
3. Create an image component for each page (here: PAGEICON_n).

4. Create a text component (here: TEXT_n) for each page. The text component and the image describe the page.

5. Write the following script statements for the On Click event of each image:

```
PAGEBOOK_1.setSelectedPageIndex(0);
TEXT_TOOLBAR_TITLE.setText(TEXT_1.getText());
PAGEICON_1.setImage("images/Icon_1_selected.png");
PAGEICON_2.setImage("images/Icon_2.png");
PAGEICON_3.setImage("images/Icon_3.png");
PAGEICON_4.setImage("images/Icon_4.png");
```

Adapt the script for each image to the corresponding page. For example, the script for the second page should look like this:

```
PAGEBOOK_1.setSelectedPageIndex(1);
TEXT_TOOLBAR_TITLE.setText(TEXT_2.getText());
PAGEICON_1.setImage("images/Icon_1.png");
PAGEICON_2.setImage("images/Icon_2_selected.png");
PAGEICON_3.setImage("images/Icon_3.png");
PAGEICON_4.setImage("images/Icon_4.png");
```

6. To enable the user to switch pages by clicking the images, you have to use the callable On Click event of the image component in the On Select event of the pagebook.

For this use case, the script for the On Select event of the pagebook should look like this:

```
if (PAGEBOOK_1.getSelectedPageIndex() == 0) {
    PAGEICON_1.onClick();
}
if (PAGEBOOK_1.getSelectedPageIndex() == 1) {
    PAGEICON_2.onClick();
}
if (PAGEBOOK_1.getSelectedPageIndex() == 2) {
    PAGEICON_3.onClick();
}
if (PAGEBOOK_1.getSelectedPageIndex() == 3) {
    PAGEICON_4.onClick();
}
```

### 29.1.6.5 `getDataAsString` for BW Queries with Structures

If you use BW queries with structures as data sources, you may want to read specific cells using the `getDataAsString` method. In the following business cases, this query is used (simplified presentation):

<table>
<thead>
<tr>
<th>Structure in the Rows</th>
<th>Structure in the Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW Key Figure 1 (Measure 1 = Sales)</td>
<td>Actual</td>
</tr>
<tr>
<td>Hierarchy Node 1 ( = World)</td>
<td>Cell 1</td>
</tr>
<tr>
<td>Hierarchy Node 2 ( = Europe)</td>
<td>Cell 2</td>
</tr>
<tr>
<td>Hierarchy Node 3 ( = DE)</td>
<td>Cell 3</td>
</tr>
</tbody>
</table>

© 2015 SAP SE or an SAP affiliate company. All rights reserved.
As this business cases deal with BW queries, BW terminology is used. In BW, key figures are measures and characteristics are dimensions.

**Business Case 1: Read Cell 1**

In the following business case, the key figure `Sales` is selected for a specific structure element in the columns (for example, Actual):

```java
DS_3.getDataAsString("DCNFKHLVQUD5AIFYGF4HQF4J",
{"DCNFKHLVQG4FC4BHDPM8JRN":"DCNFKHLVQSUORHJ7MGXE26MR"});
```

In this business case a specific structure element is selected within a structure in the columns with "DCNFKHLVQUD5AIFYGF4HQF4J" as the UID (unique ID) of the key figure `Sales` in the BW key figure structure in the rows.

Annotations:
- `DCNFKHLVQUD5AIFYGF4HQF4J` is the UID (unique ID) of the key figure `Sales` in the BW key figure structure in the rows.
- `DCNFKHLVQG4FC4BHDPM8JRN` is the UID (unique ID) of the structure in the columns.
- `DCNFKHLVQSUORHJ7MGXE26MR` is the UID (unique ID) of a structure element in the columns (for example, Actual).

**Business Case 2: Read Cell 2**

In the following business case the key figure `Sales` is selected and a specific structure element in the columns (for example, Actual) is filtered in combination with a hierarchy node (for example, Europe, which represents a text node):

```java
DS_3.getDataAsString("DCNFKHLVQUD5AIFYGF4HQF4J",
{"DCNFKHLVQG4FC4BHDPM8JRN":"DCNFKHLVQSUORHJ7MGXE26MR"});
```

In this business case a specific structure element is selected within a structure in the columns with "DCNFKHLVQUD5AIFYGF4HQF4J", "DCNFKHLVQG4FC4BHDPM8JRN", and "DCNFKHLVQSUORHJ7MGXE26MR" as the UID (unique ID) of the key figure `Sales` in the BW key figure structure in the rows.

Annotations:
- `DCNFKHLVQUD5AIFYGF4HQF4J` is the UID (unique ID) of the key figure `Sales` in the BW key figure structure in the rows.
- `DCNFKHLVQG4FC4BHDPM8JRN` is the UID (unique ID) of the structure in the columns.
- `DCNFKHLVQSUORHJ7MGXE26MR` is the UID (unique ID) of a structure element in the columns (for example, Actual).

In addition, a specific hierarchy node (for example, EUROPE, which represents a text node in the hierarchy) is filtered. The value of Cell 2 is read.

Annotations:
- `DCNFKHLVQUD5AIFYGF4HQF4J` is the UID (unique ID) of the key figure `Sales` in the BW key figure structure in the rows.
• DCNFKHLVQVTUD5AIFYGF4HQF4J is the UID (unique ID) of the key figure Sales in the BW key figure structure in the rows.
• DCNFKHLVQVGT4FC4BHDP8JRN is the UID (unique ID) of the structure in the columns
• DCNFKHLVQSUZORHJ7MGXE26MR is the UID (unique ID) of a structure element in the columns (for example, Actual)
• 0PROFIT_CTR__ZKBUSAREA=LS is the technical name of the characteristic value to be filtered (for example, DE, which is part of the hierarchy)

Business Case 3: Read Cell 3

In the following business case, the key figure Sales is selected and a specific structure element in the columns (for example, Actual) is filtered in combination with a characteristic value (for example DE) which is part of a hierarchy:

```javascript
DS_3.getDataAsString("DCNFKHLVQVTUD5AIFYGF4HQF4J",
{"DCNFKHLVQVGT4FC4BHDP8JRN":"DCNFKHLVQSUZORHJ7MGXE26MR","0PROFIT_CTR":"DE"});
```

In this business case, a specific structure element is selected within a structure in the columns with "DCNFKHLVQVGT4FC4BHDP8JRN" : "DCNFKHLVQSUZORHJ7MGXE26MR" (for example, Actual). In addition, a specific characteristic value (for example, DE, which is part of the hierarchy) is filtered. The value of Cell 3 is read.

Annotations:
• DCNFKHLVQVTUD5AIFYGF4HQF4J is the UID (unique ID) of the key figure Sales in the BW key figure structure in the rows.
• DCNFKHLVQVGT4FC4BHDP8JRN is the UID (unique ID) of the structure in the columns
• DCNFKHLVQSUZORHJ7MGXE26MR is the UID (unique ID) of a structure element in the columns (for example, Actual)
• 0PROFIT_CTR__ZKBUSAREA=LS is the technical name of the characteristic value to be filtered (for example, DE, which is part of the hierarchy)

29.1.6.6 Export to Microsoft Excel

Enables the user to specify and export a crosstab from SAP BusinessObjects Design Studio into Microsoft Excel. You can add scripting to basic components to export data from a crosstab to a Microsoft Excel file.

Example

To export crosstab_1 to Microsoft Excel 2004, add the following script to a button:

```javascript
APPLICATION.export(ExportType.EXCEL_xls,[CROSSTAB_1]).
```

The application user can export data from a crosstab to a Microsoft Excel file.

Note

• In Microsoft Excel 2000 and Microsoft Excel 2007, the metadata is separated from the crosstab data and displayed in a separate tab in the MS Excel spreadsheet. For example, the Crosstab itself is represented in
You must read the following SAP Notes when exporting to Microsoft Excel.

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1917890</td>
<td>Listing the Microsoft Excel export feature as the last statement in the scripting.</td>
</tr>
<tr>
<td>1917891</td>
<td>Increasing jvm memory for large data export.</td>
</tr>
<tr>
<td>1917892</td>
<td>Using Microsoft Excel 2007 if a large column export is required.</td>
</tr>
<tr>
<td>1917943</td>
<td>Limiting the number of formatting elements in each crosstab.</td>
</tr>
<tr>
<td>1917944</td>
<td>The third parameter export setting currently not corresponding with the first parameter export type.</td>
</tr>
<tr>
<td>1917946</td>
<td>Microsoft Excel 2007 export feature not supported on BI platform 4.0.</td>
</tr>
<tr>
<td>2167615</td>
<td>Design Studio 1.5 Microsoft Excel export cell size.</td>
</tr>
</tbody>
</table>

Related Information

Export (export) [page 330]

29.2 Working with Global Scripts Objects and Global Script Functions

You can create any number of Global Scripts Objects, a technical component type, which provide a grouping of global script functions.

On each global scripts object, you can create any number of script functions. Each script function has a configurable return type and any number of typed input parameters. The following types can be used for return value and input parameters:

- primitive types (int, Boolean, String, float)
- `<none>` for return values only
- DataSourceAlias
- all UI component types (Button, Tabstrip, ...) including SDK extension components

i Note
Array types are currently not supported.
Creating new global scripts objects and script functions

To create a new scripts object, right-click the Technical Components type folder in the Outline view and choose Create Child Global Scripts Object. The new object is displayed in the Technical Components folder.

To create a new global script function, right-click a global scripts object in the Outline view and choose Create Global Script Function... Enter the function name in the Create Script Function dialog box and click OK. The Create Script Function dialog box is displayed again. You can now enter a description for the function and enter the code for the global script function. Click OK, when you have finished the global script function. The function is now displayed under the corresponding global scripts object.

Note
Due to security reasons, it is not possible to use `onClick` and other `on<ACTION>` methods in the script.

Editing, renaming, deleting scripts objects and script functions

Right-click the scripts object or script function to:

- copy and paste scripts objects and script functions
- edit existing script functions by reopening the Create Script Function dialog box
- rename existing objects and functions (which automatically refactors each occurrence in scripts)
- delete scripts objects and script functions
- find all references to the scripts objects and script functions in scripts

Example

- Example for return type and input parameters for the script function computeAverage
  Script function: `computeAverage`
  Return Type: `float`
  Input Parameters: `value1`, `value2` and `value3` (all of type `float`)
  Script code:
  ```java
  return (value1 + value2 + value3) / 3.0;
  ```

- Example for restyling a button using CSS classes. The style changes when the button is enabled or disabled
  Script function: `styleButton`
  Return Type: `<none>`
  Input Parameters: `button` of type `Button`, `enabled` of type `boolean`
  Script code:
  ```java
  if (enabled) {
      button.setCSSClass("enabledCSSClass");
  } else {
      button.setCSSClass("disabledCSSClass");
  }
  ```
29.3 Enabling Text Translation in Analysis Applications

Context

In addition to the texts from the data sources, analysis applications can contain translatable texts, like labels on buttons or messages, that are created by you, the application designer. If you want to provide your analysis applications in different languages, you need to configure the analysis applications accordingly.

Note

Texts from the data sources are provided in localized form. You do not need to have these texts translated (for example, master data or metadata of the selected data source).

The translatable texts created by you can be divided into static and dynamic texts:

<table>
<thead>
<tr>
<th>Text Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static text</td>
<td>Button label that stays constant during application lifetime</td>
</tr>
<tr>
<td>Dynamic text</td>
<td>Message text that a script joins together from multiple parts</td>
</tr>
</tbody>
</table>

Table 30:

For both text types, you need to flag your application to be translatable by adding a Text Pool component to your application.

Procedure

1. In the design tool, go to the Outline view.
2. Right-click Technical Components and choose Create Child.

Results

The system collects all translation-relevant texts that you enter as property values in the Properties view for different components (like buttons or tooltips in this Text Pool component), and saves these texts for translation. The procedure for storing and translating the texts depends on the platform you use.
Next Steps

In addition to the static texts you enter as property values in the Properties view for different components like buttons or tooltips, you can create dynamic texts that consist of different translation-relevant text parts. You create these dynamic texts with scripting. The dynamic texts are also collected in the Text Pool component.

Related Information

Using Scripts for Translatable Dynamic Texts [page 230]
Working With Translatable Texts in Analysis Applications [page 231]

29.3.1 Using Scripts for Translatable Dynamic Texts

Context

Application designers can write a script that joins together translatable texts from multiple parts. You can use this function to create message texts, for example.

Procedure

1. After adding a Text Pool component to your application, select your TEXT_POOL in the Outline view.
2. Go to the Properties view and under Texts, click the Edit Texts... button.
3. In the Edit Text Pool Entries dialog box, choose Insert.
4. Enter a key for your text.
   You need this key in your script to retrieve the text.
5. Enter your text in your original language.

   Tip
   You can use placeholders (like {0} or {1}, for example) for dynamic parts of the text.

6. After inserting the required texts to the text pool, choose OK.
7. To retrieve the translated texts, replace the placeholders and display the whole text, create a script as follows:

   ```javascript
   var translated = TEXT_POOL.Key1;
   var placeholdersReplaced = Convert.formatString(translated,
   [ "Replacement1" , "Replacement2" ]);
   TEXT_1.setText(placeholdersReplaced);
   ```
29.3.2 Working With Translatable Texts in Analysis Applications

In analysis applications, texts created by the application designers (for example, button texts) are translatable. When the BI platform is used, these texts can be translated using the Translation Management Tool and stored in the applications InfoObject for each analysis application. When SAP NetWeaver is used as the platform, these texts are stored in the BW system in the TLOGO object table RSAO_T_TEXT for each analysis application. You can translate the texts with standard translation tools, for example, transaction SE63. For more information, see “Transaction SE63” on SAP Help Portal at http://help.sap.com.

At design time, with SAP NetWeaver, **application designers need to enter the texts in English**. When they save the analysis application, the design tool automatically saves the texts with language key EN in the TLOGO object table RSAO_T_TEXT. The texts are stored separately for each analysis application.

**Note**

If the text key changes, the system deletes all texts with the unused old key in all languages. The text key consists of the name of the component and the property.

Example of an analysis application with a button:

<table>
<thead>
<tr>
<th>Text_Key</th>
<th>Text</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUTTON_1.TEXT</td>
<td>Action</td>
<td>EN</td>
</tr>
<tr>
<td>BUTTON_1.TEXT</td>
<td>Aktion</td>
<td>DE</td>
</tr>
</tbody>
</table>

The application designer changes the text key of the button to BUTTON_2.TEXT. The system deletes all BUTTON_1.TEXT entries in EN and DE and saves the following new entry:

<table>
<thead>
<tr>
<th>Text_Key</th>
<th>Text</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUTTON_2.TEXT</td>
<td>Action</td>
<td>EN</td>
</tr>
</tbody>
</table>

**In the design tool, the original texts entered by the application designer are always displayed.**

At runtime, the texts of the analysis application appear in the BW logon language of the application user. If there are no translated texts available in this language, the texts of the analysis application (.biapp file) are displayed. These are the original texts that the application designer created in English.

With the BI platform, when transporting analysis applications from one system to another, the texts are automatically transported together with the analysis application.

With SAP NetWeaver, when transporting analysis applications from one BW system to another, the texts are automatically transported together with the analysis application (TLOGO object of type AZAP).

For more information about the required SAP NetWeaver Support Packages, see SAP Note 2051284.
29.4 Deploying SDK Extensions

In addition to the standard palette of components in SAP BusinessObjects Design Studio, you can install 3rd party components developed with the Design Studio SDK. Adding these 3rd party components, known as Design Studio SDK extensions, to your SAP BusinessObjects Design Studio installation enables you to create and execute local analysis applications containing these SDK extensions.

Furthermore, you can install new chart types developed with the SAP Lumira SDK. These SDK extensions, also known as CVOM chart extensions, are added to the list of additional chart types for the standard chart component. You can create and execute local analysis applications containing these new chart types.

Note

CVOM chart extensions are not supported in SAP BusinessObjects Design Studio if SAP HANA is used as the platform.

Before application users can launch analysis applications containing SDK extensions from one of the supported platforms (SAP NetWeaver, BI platform or SAP HANA), your administrator needs to deploy the SDK extensions to the required platform. For more information, see “Deploying SDK Extensions” in the corresponding Administrator Guide: SAP BusinessObjects Design Studio on SAP Help Portal at http://help.sap.com/boad.


For more information about creating CVOM chart extensions using the SAP Lumira SDK, see the SAP Lumira Visualization Extensions Developer Guide on SAP Help Portal at http://help.sap.com/lumira.

Related Information

Configuring Additional Chart Types [page 182]

29.4.1 Installing Design Studio SDK Extensions to SAP BusinessObjects Design Studio

Context

You can add extensions developed with the Design Studio SDK to your SAP BusinessObjects Design Studio installation as new components.
Procedure

1. In SAP BusinessObjects Design Studio, choose Tools Install Extension to Design Studio...
2. Depending on where the SDK extension is located, proceed as follows:
   - For locally saved extensions, choose Archive... and select the archive file containing the SDK extension, under C:\SampleExtension.zip, for example.
   - For extensions stored on a Web server, enter the URL of the Web server.
3. Choose OK.
4. Select the required feature, for example, SampleExtensionFeature.
5. Select the Design Studio extensions that you want to install.
6. Choose Finish to proceed with the installation.
7. Choose Next and again Next to confirm the installation.
8. Accept the terms of the license agreement and choose Finish.
9. Choose Yes to allow SAP BusinessObjects Design Studio to restart.

Results

The SDK extension components appear in the Components view of SAP BusinessObjects Design Studio as new components.

The components are stored under <user home directory>\Analysis-config.

29.4.2 Installing CVOM Chart Extensions for SAP BusinessObjects Design Studio

Context

You can add CVOM chart extensions, developed with the SAP Lumira SDK, to your SAP BusinessObjects Design Studio installation as new chart types for the standard chart component.

Procedure

1. In SAP BusinessObjects Design Studio, choose Tools Install CVOM Chart Extension...
2. Depending on where the CVOM chart extension is located, proceed as follows:
   - For locally saved extensions, choose Archive... and select the archive file containing the extension, for example C:\SampleChartExtension.zip.
   - For extensions stored on a Web server, enter the URL of the Web server.
3. Choose OK and then choose OK again to confirm the installation.
4. To restart SAP BusinessObjects Design Studio, choose Yes.

Results

When you create or open an analysis application with a chart component, the CVOM chart extensions you have added are listed as new chart types.

Next Steps

In the Properties view of the chart component, under Chart Type, choose Additional Chart Types.... The CVOM chart extensions are listed as new chart types in the Additional Charts dialog box.

Related Information

Configuring Additional Chart Types [page 182]

29.4.3 Removing Extensions from SAP BusinessObjects Design Studio

Context

You can remove SDK extensions that you have added to your SAP BusinessObjects Design Studio installation as follows:

Procedure

1. In SAP BusinessObjects Design Studio, choose Help > About...
2. Click the Installation Details button.
3. Select the feature containing the SDK extension, for example, SampleExtensionFeature.
4. Choose Uninstall....
5. In the Uninstall wizard, choose Finish.
6. Choose Yes to allow SAP BusinessObjects Design Studio to restart.
Results

The SDK extension components are removed from the Components view of SAP BusinessObjects Design Studio. CVOM chart extensions are removed from the list in the Additional Charts dialog box.

29.5 Creating a Generic Analysis Template for RRI Jump Targets

Context

Before application users can use the report-report interface (RRI) in analysis applications, you or your administrator need to configure the jump targets for the query that is used as data source in the analysis application. For more information, see
- “BEx Query as Recipient” on SAP Help Portal at http://help.sap.com

If the jump target, which is configured for the data source query, is also a query, the target query is launched as a BEx Web application by default. In order to avoid this and ensure that the jump is handled by Design Studio and the target query is displayed as an analysis application, you or your administrator need to specify a generic analysis template on your platform.

Procedure

1. In the design tool, create an analysis application that can be used as the generic analysis template.
   As this analysis application will be used for all query jumps, it should be a very generic application, which can handle basic analysis of an arbitrary query. In general, it must conform to the following constraints:
   ○ Contains only one data source, which is loaded in script.
   ○ Accepts query ID as URL parameter XQUERY (for SAP NetWeaver mode).
   ○ Accepts query ID and system ID as URL parameters XQUERY and XSYSTEM respectively (for the BI platform mode).
   ○ Loads the single data source (usually in the On Startup script) by calling assignDataSource() and passing the XQUERY and XSYSTEM parameters to the corresponding parameters of assignDataSource().

   Tip
   A good starting point, and an example of a valid generic analysis template, would be to select Generic Analysis Template when creating an application in the design tool. Note that you do not need to assign a data source to this Generic Analysis Template. When application users jump to the target query, the
necessary query information is automatically added to the Generic Analysis Template as URL parameters (XQUERY, XSYSTEM).

2. Save your generic analysis template on the platform.

3. To configure the runtime in order to use this analysis application as the generic analysis template for query jumps, you need to set this template accordingly on the platform. For more information, see “Configuring the Report-Report Interface for Analysis Applications” in the Administrator Guide: SAP BusinessObjects Design Studio on SAP NetWeaver or Administrator Guide: SAP BusinessObjects Design Studio on BI Platform.

29.6 Exporting Analysis Applications

Context

Application designers can export analysis applications to their PC or a network share. You can use this function for the following purposes:

- It makes it possible to provide the original source code of analysis applications to SAP and helps us to resolve problems that you report to SAP as customer messages.
- You can export applications, in order to import them to a different platform.

**Note**

For example, you can export an application from one BI platform and import it to a different BI platform. You can also use it for exporting an application from the SAP NetWeaver platform to the BI platform, or even from one local mode platform to another local mode platform.

- You can use the export for regular backups of your complex analysis applications.

**Note**

Exporting analysis applications is available in all modes (BI platform, SAP NetWeaver, SAP HANA and local mode).

Procedure


2. In the Export Application dialog box, choose [Browse...] and select the required analysis application.

   If an application is currently open, this application is automatically preselected, but can be changed using the [Browse...] button.

3. If you want to have the MIME objects (for example, images or CSS files), which are referenced in the application, exported, keep the Export referenced files checkbox activated.

   The system analyzes the selected application and auto-detects all MIME objects referenced by the application (including MIME objects referenced by script coding). If you use string concatenation in a script to construct
an image file name, or if the application uses a CSS file that internally references images, for example, the auto-detection might fail. In such cases, you can add the files manually to the export.

4. (optional) If you have MIME files that the system cannot auto-detect, choose Add File... and select the required files.

   **Note**

   When exporting the same application on a regular basis (for backup purposes), you only need to fill this list once. For subsequent exports, the list is already pre-filled from the export history.

5. Choose Next.

6. To select the target folder for the ZIP file (containing the application and MIME objects) on your local PC or on a network share, choose Browse...

   For subsequent exports, the folder is already pre-filled from the export history.

7. (optional) If you want to change the default name of the ZIP file, enter the required name under ZIP File Name.

   The default name of the ZIP file is the name of the application suffixed with a timestamp.

   **Note**

   The timestamp suffix prevents from overwriting previous ZIP file versions of the same application. For regular backups of the same application, we recommend that you use the default name with the timestamp.

8. (optional) Under Export description, you can enter a change list description.

   For example, First working version of data source filtering.

   This description is displayed in the Import Application dialog box and can help you identify the correct version to be imported.

9. Click Finish.

   The system creates a ZIP file, which at least contains two files on root level: content.biapp and export.properties. The export.properties file contains information about the application (like contained MIME objects, used back end connections and the source platform), which will be used when importing the application.

10. (optional) To quickly access the created ZIP file, click Open Export Folder in the pop-up window on the right bottom corner.

### Results

You can use the ZIP file for backup purposes of your application, or you can attach it to your customer message for troubleshooting purposes. If you exported the application, in order to import it to a different platform, proceed with the import process under [Application ➤ Import...](#).
Related Information

Importing Analysis Applications [page 238]

29.7 Importing Analysis Applications

Prerequisites

You have exported an application, which you have created on a different platform, and you want to import it to another platform or the local mode. Note that for importing (uploading) applications, you should always use the same design studio version that you used to export (downloading) the application. For example, to import a 1.4 biapp file, you should use design studio version 1.4.

Note

Importing analysis applications is possible in all modes (BI platform, SAP NetWeaver, SAP HANA and local mode).

Once you have exported the application as a ZIP file to a folder of your choice, you are now logged on to another platform or the local mode.

Context

You can import applications, which you have previously exported, to a different platform or mode.

Note

For example, you can export an application from one BI platform and import it to a different BI platform. You can also use it for exporting an application from the SAP NetWeaver platform to the BI platform, or even from one local mode platform to another local mode platform.

Procedure

1. Choose Application Import in the design tool.
2. In the Import Application dialog box, choose Browse... to select the ZIP file of the application you have exported. Once you have selected a valid ZIP file (the ZIP file must contain content.biapp and export.properties), information about the application is displayed. This enables you to check if it is the correct application to import. Choose Next.
3. Specify the **Name** and **Description** of the target application.

Choose **Next**.

4. **(optional) The next dialog box page is only displayed if the exported ZIP file contains referenced files (images, CSS). You can choose whether to import these files.**

**Note**

In local mode and HANA mode, the default option is to import the files (otherwise the application will be incomplete after the import), whereas in BI platform mode and the SAP NetWeaver mode, the default option is not to import the files. These modes only support shared global MIME objects. This means that an import without MIME objects can still result in a complete application, and you might not want to overwrite existing MIME objects because they could also be used by others. If you choose to import the files, additional options are displayed that depend on the capabilities of the target platform:

- In local mode, you can either keep the original folder structure of the files, or create a subfolder in the application’s directory, into which all files are imported (in a flat way).
- In SAP HANA mode, the target folder for the files cannot be selected, and an information text is displayed, which indicates that all files are imported into the application’s package.
- In BI platform mode, you are offered three options. The first option, to keep the original folder structure, might be disabled if the folders in the ZIP file are not BI platform compliant (in other words, if the folders do not start with `Root Folder/` or `User Folders/<user>/`). This is always the case if the source platform was not a BI platform. The second option is to import all files into a new or existing folder (in a flat way which means without keeping the folder structure). The third option is a checkbox where you can choose to overwrite existing files.
- In NetWeaver mode, the MIME import wizard page is displayed, but an information text may appear to indicate that the import of MIME objects is not supported, and refer you to SAP Note 2111546. The MIME import is only enabled if the minimum support package is implemented in the SAP BW system. This Note contains a number of ABAP function modules that are required by Design Studio to upload MIME objects, delete MIME objects, create new folders in the MIME repository, and so on. If the MIME object upload is supported, the same options are displayed as in BIP mode. The difference is that the option to import into the original location is always enabled.

Choose **Next**.

5. **(optional) The next dialog box page is only displayed if the exported application contains SAP BW, SAP HANA or Universe data sources (not SDK data sources).**

On the **Map Backend Connections** page, you can map the backend connections used in the exported application to backend connections available in the target platform. This is especially useful if the application contains a lot of data sources from the same system, because on the mapping page only one entry needs to be mapped. There is no need to go through every data source and change the backend system manually using the property sheet. Normally the mapping page suggests an appropriate mapping. For each backend connection, only connections that are of the same type as the exported connection can be selected as mapping targets. This means if the exported connection is, for example, a SAP BW connection, it can only be mapped to a SAP BW connection in the target platform. To choose another connection than the proposed one, click on the proposed connection in the **Should Be Mapped To** field and open the dropdown box. Now you can choose another suitable connection for the platform/mode that you are logged on to. This wizard page is optional; you are not required to map anything.

Choose **Next**.

6. The **Summary** page lists all changes that will be performed once you choose **Finish**. Since the import might overwrite existing files, you should check the list before finishing the wizard.
Choose Finish.

Results

You have imported an application to another platform or local mode.

29.8 Exporting Applications As Templates

Context

You can create analysis applications and provide them as templates for your fellow application designers.

Procedure

1. Choose Application ➔ Export as Template... in the design tool.
2. In the Export Application As Template dialog box, choose Browse... and select the required analysis application.
   If an application is currently open, this application is automatically preselected, but can be changed using the Browse... button.
3. If you want to have the MIME objects (for example, images or CSS files), which are referenced in the application, exported, keep the Export referenced files checkbox activated.
   The system analyzes the selected application and auto-detects all MIME objects referenced by the application (including MIME objects referenced by script coding). If you use string concatenation in a script to construct an image file name, or if the application uses a CSS file that internally references images, for example, the auto-detection might fail. In such cases, you can add the files manually to the export.
4. (optional) If you have MIME files that the system cannot auto-detect, choose Add File... and select the required files.
   
   **Note**
   
   When exporting the same application again, you do not need to fill this file list again. For subsequent exports, the list is already pre-filled from the export history.
5. Choose Next.
6. To select the template folder on your local PC or on a network share, choose Browse....
   For subsequent exports, the folder is already pre-filled from the export history.
7. Enter a template name.
8. Select the required template category: Desktop Browser, iPad, iPhone, or a user-defined category.
If you need a new user-defined template category, you can jump to the Preferences page by clicking Configure Categories.... Here you can add your own template categories that indicate the target device types recommended for a specific template.

9. Under Template Caption, enter the name of the template to be displayed in the New Application dialog box.

Note that this name is case-sensitive.

10. Under Template Description, you can describe the content and purpose of your template.

This description is displayed in the New Application dialog box and can help other application designers identify which template to select.

11. Click Finish.

In the selected template folder, the system creates a new folder with the content.biapp file and an .info file with the template description.

12. (optional) To quickly access the created content.biapp file, click Open Template Folder in the pop-up window on the right bottom corner.

Results

When creating new applications, other application designers can choose from the templates provided by SAP BusinessObjects Design Studio, and can also select the template you have exported.

Related Information

Maintaining Settings in the Design Tool [page 29]

29.9 Using the Backend Connection Component for Data Source Browsing

The technical component Backend Connection enables the application user to select a generic data source at runtime by means of a data source selection dialog box. You can either use the predefined data source selection dialog box for runtime and configure it in the properties of the Backend Connection, or you can use the API to create your own user interface for browsing data sources. Before you can use the Backend Connection, you have to assign a system at design time. This can be done in the Property view of the design tool or using the API.
Adding a Backend Connection component to an application

To add a backend connection component to an application right-click on the Technical Components folder of the Outline view and choose Create Child Backend Connection. The backend connection component is displayed in the Technical Components folder.

Prerequisites: Assigning a backend system

Before you can use the Backend Connection, you have to assign a system at design time. This can be done in the Properties view of the design tool or using the API.

Note

Each backend connection always relates to one backend system. If you need multiple systems in your application, we recommend that you create multiple backend connection components rather than reassigning the system each time at runtime.

As DSL data sources are not supported on the BI platform 4.0, the use of the backend connection component is not supported for DSL data sources.

Configuring the Data Source Browser dialog box

Using the technical component Backend Connection in an application automatically provides you with a predefined data source browser dialog box for the runtime. You can configure the dialog box by setting the properties of the Backend Connection component, or by using the API:

- You can give the dialog box a user-defined name (Titel property).
- You can specify which tab is shown by default when the dialog box is opened (Default Tab property). Decide whether the dialog box is opened with the Search, Roles, Workspaces or Folders/InfoAreas tab.
- You can specify which tabs are visible in the dialog box. By default all tabs (Search, Roles, Workspaces or Folders/InfoAreas tab) are visible, but you can change the visibility of the tabs as required.
- You can change the size of the dialog box (Width and Height properties). By default, the dialog box size is automatically adjusted to suit the screen size.

The UI of the data source browsing dialog box can have different appearances, depending on the chosen system for the Backend Connection component:
Table 33:

<table>
<thead>
<tr>
<th>Data Source Browser dialog box for SAP HANA connections</th>
<th>Data Source Browser dialog box for SAP BW connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>For SAP HANA, the dialog box contains a maximum of two tabs, the Search tab, where you can search for data sources, and the Folders tab, which shows the folder/tree structure of the SAP HANA data repository.</td>
<td>For SAP BW the dialog box contains up to four tabs, the Search tab (just like in HANA), the Roles tab showing SAP BW roles, the Workspaces tab showing SAP BW workspaces in a tree structure, and the InfoArea tab, which shows the whole InfoArea tree of the SAP BW system</td>
</tr>
</tbody>
</table>

Opening the Data Source Browser dialog box

To open the dialog box in the application, you have to use the API method `connection.showDataSourceBrowser();`, for example, in the On Startup event of the application. The user interactions in the dialog box can be evaluated using two events - On Data Source Browser Confirm and On Data Source Browser Cancel.

For further information on all methods relating to the object Backend Component, see the relevant section “Backend Connection” in the API documentation in this guide.

Example

You use the Backend Connection component for an application, where the application user sets the system in a dropdown box and searches for a data source relating to the chosen data source, by opening the data source browsing dialog box with a click of a button.

For this example application, you have to add the following objects to your application:

- a crosstab
- a dropdown box with a tooltip (property Tooltip) Select System
- a button with a title (property Text) Data Sources

Perform the following steps:

1. Write the following script in the On Startup event of the application

   ```javascript
   var conns = CONNECTION_1.getConnections();
   conns.forEach(function(element, index) {
     DROPDOWN_1.addItem(element.name, element.text);
   });
   ```

2. Write the following script in the On Select event of the dropdown box with the tooltip Select System

   ```javascript
   var selected = me.getSelectedValue();
   CONNECTION_1.setSystem(selected);
   ```

3. Write the following script in the On Click event of the button with the title Data Sources

   `CONNECTION_1.showDataSourceBrowser();`

If you want to use the selected query for further purposes, proceed in the following way:
1. Write the following script in the On Data Source Browser Confirmed event of the backend connection component:

```javascript
var ds = CONNECTION.getSelectedDataSource();
DS_2.assignDataSource(ds.connection, ds.type, ds.name, true);
```

**Note**

As the script keeps running after the dialog box for data source browsing opens, you should use the script `CONNECTION.getSelectedDataSource();` in the event of the backend connection component and not in the event of the button component.

### 29.10 Using Processing Groups for Parallel Query Execution

Usually queries are executed sequentially (classic runtime). But even queries with short backend runtimes can add up to long server roundtrips when multiple queries are used in one application, and therefore you may need to execute queries in parallel. For this reason, with SAP BusinessObjects Design Studio 1.5 you can define groups of data sources, by using the data source property `Processing Group`. All of these groups can be executed in parallel.

With this property, you can assign each data source used in your application to one processing group. Each group is associated with a session. This means different groups are executed in parallel and all groups in one application run in parallel, whereas all data sources in one group still run sequentially.

Scripts are also executed sequentially even if parallel query execution is used in the application. To avoid the sequential execution of scripts, you can use the method `APPLICATION.loadDataSources();` or the data binding function. See Load Data Sources (loadDataSources) [page 336] and Binding the Properties of Standard Components to Data Sources [page 80]

### Prerequisites

When using parallel query execution, note the following points:

- Parallel query execution can only be used for a design studio deployed on the BI platform or locally.
- Parallel query execution can be used with all data sources (SAP BW, SAP HANA, DSL).
- Parallel query execution is only relevant for the following runtime phases:
  - during initialization of data sources on start-up
  - when fetching result sets during rendering
  - when submitting variables
  - during data binding
- Parallel query execution cannot be used with the following functions:
  - with planning-enabled queries:
    - As planning-enabled queries need to run in one session and parallel query execution requires multiple sessions, these two functions cannot be used together. Therefore the system ignores planning-enabled queries as data sources if they are contained in any of the non-default processing groups.
General considerations and guidelines for using parallel query execution

As an application designer you need to be aware that each processing group allocates additional resources. Therefore we recommend that you think about how and when to use additional processing groups. Take the following points into account when you want to run queries in parallel in your application:

- In general, you should not execute queries with a very short runtime in parallel as the overhead might be bigger than the performance improvement achieved by parallelization.
- When designing applications with SAP BW data sources, be aware that using additional processing groups causes additional load in the BW system.
- The decision about when processing groups are used and for which data sources should be mainly UI-based. This means that data sources, which should be visible at the same time on the UI, should potentially be categorized for different processing groups.
- When you use merged variables in the application and the variable processing takes a lot of time, it might not always be beneficial to change the application in order to use parallelization. In this case, the variables are not merged anymore and the variable processing needs to run multiple times (for each query) instead of once for the merged container.
- We recommend that you always test different combinations of processing groups and design aspects to validate the best approach for each application.

Example

The following image shows an example of a UI-driven configuration for an application with a tabstrip component and five data sources. The initial screen shows Tab 1 with two data sources (DS_2 and DS_3) plus an additional data source outside of the tabstrip which is always visible (DS_1). For maximum parallelization, all three initially visible data sources were assigned to different groups (DS_1 is contained in the default group, DS_2 in G1 and DS_3 in G2).

Neue Formulierung: The second tab contains two additional data sources. These data sources could reuse the groups from the data sources of the first tab, which means the number of parallel sessions would be kept low. As these additional data sources are not displayed initially, they should be initialized later by using the script method Application.loadDataSources(). Otherwise the initialization of these data sources slows down the start up time of the application.
29.11 Unmerging Prompts (Variables)

Prerequisites

When unmerging prompts, note the following points:

- Unmerging prompts works only for a design studio deployment on SAP NetWeaver or the BI platform or for the local mode.
- Unmerging prompts for SAP HANA data sources only works if all data sources rely on the same analytic or calculation view.

Technical Background

In SAP BW, variables (prompts) can either be merged or unmerged.

If variables are merged, the system creates one merged variable container for all data sources in an analysis application. This means, if the same variable is used in multiple queries, it can be entered only once.
If variables are unmerged, the system creates separate variable processors for each data source. This means, if the same variable is used in multiple queries, it has to be entered for each SAP BW query separately, and different values can be entered for different queries.

For further information on SAP BW variables, see “Variables” on SAP Help Portal at Variables.

When creating analysis applications, you may have two reasons for unmerging variables:

- Business scenario: You want separate variable handling for the same variable (for example, you want to run the same query on different countries).
- Technical scenario: You want to unmerge variables for technical reasons (for example, performance requirements, parallel query execution). In this case, you might want to hide the unmerge option from the application user (for example, you can fill variables automatically in the background).

Properties Setting in the design studio for (un)merging variables

In the design studio, there are two properties that relate to the functionality for unmerging variables:

- application property Merge Prompts with the values true and false; default: true
- data source properties Text. With this property you can add a descriptive text that is displayed in the Prompts dialog box if prompts dialog box is filled (otherwise the data source alias specified in the name property will be used, for example, DS_1).

Prompts dialog box at runtime or design time (for mandatory variables)

The Prompts dialog box shows all variables for all queries in one application. In the Prompts dialog box, a prefix is added to the variable name, to enable you to identify which query the variable belongs to. By default this is the data source alias, for example DS_1, or the text you entered in the Text property of the data source. For example, if you enter myText1 in the Text property of the data source DS_1, every instance of DS_1 in the dialog box is replaced with myText1.

Prompt Settings / Select Prompts dialog box in the application properties

This dialog box shows all variables for all queries in one application. Variables can be removed and filled automatically using the On Before Prompts Submit event of the application. Using this dialog box, you can remove (hide) mandatory variables. However, the mandatory variables will be displayed in the Prompts dialog box at runtime unless you set the Merge Prompts application property to false. This means mandatory variables will NOT be displayed in the prompts dialog box after they have been removed.

If mandatory variables are not filled, this will cause an error that you can avoid by filling the variable automatically using the On Before Prompts Submit event.
Using the On Before Prompts Submit event

This event is called when the OK button in the Prompts dialog box is pressed and BEFORE variables are submitted. This allows users to copy variable values between data sources, which means application users do not necessarily have to enter the same variable for each query manually. For example,

```javascript
DS_2.setVariableValueExt('0BC_HIER_MAND', DS_1.getVariableValueExt('0BC_HIER_MAND'));
```

Using Load in Script property

When you use the Load in Script data source property in combination with the unmerge variables function, the prompts dialog box might be displayed multiple times, as in this case the variables are handled separately for each data source. To avoid this, you can copy variable values from another data source before loading a new data source. For example,

```javascript
DS_2.setVariableValueExt('0BC_HIER_MAND', DS_1.getVariableValueExt('0BC_HIER_MAND'));
DS_2.loadDataSource();
```
30 Troubleshooting

30.1 Auto-Recovering Applications

When you create an application in the design studio, the system automatically saves your application in the background as a snapshot at regular intervals. You can configure this interval in the Preference page under Application Recovery. If the system crashes and you open the application you have worked on again, the dialog box Application Recovery appears. Here you can decide how to proceed with the auto-saved version. You can choose from the following options:

- **Save here:**
  Click this option, if you want to save the auto-saved version using a different name. Use the Browse button to choose or create a different name. This option opens both the last actively saved version and the auto-saved version. If you use an existing name for the auto-saved version only this application will be opened.

- **Overwrite last actively saved version**
  Click this option, if you want to overwrite the last actively saved version with the auto-saved version of the application.

- **Discard auto-saved version**
  Click this option, if you want to use the last actively saved version and discard the auto-saved version.

30.2 Using the Script Problems View

**Context**

To find, analyze and resolve script errors and problems at design time, you can use the Script Problems view in Design Studio. This view displays script errors in a list of lines, each connected to one single error. Each line provides information about the following:

- Error description
- Location of the script error
- Event script the error relates to
- Component the error relates to
- Application the error relates to
- Type of error
Procedure

1. To display the Script Problems view, click View > Script Problems in the View design tool menu. The view is displayed below the layout editor.

2. Create the scripts for your application.
   If the scripts you write contain errors, these are displayed immediately in the Script Problems view.

3. Save the application. Any scripts errors that have occurred are now displayed in the view.

4. In the Script Problems view double-click the script error line you want to resolve. The system displays the Script Editor dialog box containing the script with the error.

5. Correct the script error and save the application. The relevant script error line is no longer displayed in the Script Problems view.

30.3 Working with Large eval() Statements

Sometimes Microsoft Internet Explorer® 9 has trouble with large eval() statements.

Microsoft Internet Explorer® 9 sometimes has trouble with large eval() statements in either of the following two cases:

- When the developer tools are open.
- When the one of the script debugging options is selected in the Microsoft Internet Explorer® 9 browser, on the Advanced tab.

To avoid this issue, we recommend not using the developer tools in the Microsoft Internet Explorer® 9 browser.

30.4 Setting Network Connections If Logon Problems Occur

Context

After installing or deinstalling SDK extensions and restarting the design tool, you might not be able to log on to your BI platform. The Authentication dropdown box in the logon dialog box for the design tool is disabled.

This problem occurs when exceptions are specified with a "*" wildcard character in the proxy settings of Internet Explorer under Internet options > Connections > LAN settings > Advanced > Exceptions.

Example

The exceptions are defined as localhost;127.0.0.1;*.mycompany.com.

To communicate with the BI platform, SAP BusinessObjects Design Studio uses a third-party library, which can experience connectivity problems when "*" wildcard characters are used.
To solve this issue, proceed as follows:

**Procedure**

1. Start the design tool in local mode by pressing *Skip* in the logon dialog box.
3. Choose one of the following options:
   - Set the Active Provider to Direct. This disables proxies completely.
   - Set the Active Provider to Manual and specify the proxy manually, omitting the problematic "*" entries.
4. Restart the design tool.

**Results**

This procedure only needs to be performed once.

For more information, see “Network Connection Preferences” in the online help. You can access this chapter by pressing F1 or the Help button in the Preferences dialog box.

### 30.5 Activating Runtime Traces

**Context**

You can record traces to analyze problems in the design tool.

**Procedure**

1. In the design tool, choose Tools > Preferences > Application Design > Support Settings.
2. Select the Activate runtime trace checkbox.
3. Click OK.

**Results**

The system will create a trace file on your local hard drive. The trace file is a text file that contains a log of the activities performed in the design tool's layout editor and in the executed analysis application. It is stored in the C:\<user>\Analysis-workspace\.metadata\plugins\com.sap.ip.bi.zen\logs\RSTT folder.
The design tool also connects to the SAP NetWeaver BW system. This checks if the user requesting the trace has sufficient authorization to log traces (standard authorization object S_RS_RSTT). If this is the case, the SAP NetWeaver BW trace tool environment (transaction code RSTT in the connected BW system) is activated. With SAP NetWeaver as platform, a new trace containing your navigation steps will be created in the BW system. In BI platform mode or local mode with processing groups for parallel query execution in usage, for each session (processing group), a new trace containing your navigation steps will be created in the BW system. In addition, the main analysis application has its own trace.

**Note**

For example, an analysis application with four processing groups starts five different traces in the BW system.


The recorded traces help us to resolve problems that you report to SAP as customer messages.

### 30.6 Activating SAP JCo Traces

#### Context

You can record SAP JCo traces to analyze problems in the design tool.

#### Procedure

1. In the design tool, choose **Tools > Preferences > Application Design > Support Settings**.
2. Select the **Activate SAP JCo Trace** checkbox.
3. Click **OK**.
4. Restart the design tool to enable the SAP JCo trace.

#### Results

The SAP JCo trace is activated and the trace level is set to 8. You can collect all the SAP JCo trace files using the Collect Support Information function. The recorded traces help us to resolve problems that you report to SAP.

#### Related Information

Collecting Support Information [page 254]
30.7 Viewing And Collecting Statistics Data At Runtime

Context

If you encounter performance issues in your analysis applications, you can activate a runtime profiling function for your applications, which helps you to analyze the performance-critical processing steps.

Procedure

1. In the design tool, go to Tools > Preferences > Application Design > Support Settings and choose Activate Runtime Profiling.

2. Execute the required analysis application either in local mode, or on your platform (SAP NetWeaver or BI platform).

   The system automatically adds the following parameter to the URL: PROFILING=X

   You can also add this parameter to the URL manually without activating the runtime profiling setting in the Preferences dialog box.

3. In the analysis application, a Statistics dialog box appears, where you can view the following statistics data:

   ○ Under Java Statistics, you can view how much time is required to process each navigation step (and initial loading) on the server. This includes the processing time on the backend system (SAP NetWeaver BW or SAP HANA) and on the used platform (BI platform or SAP NetWeaver).

   With applications that use parallel query execution, the system displays the execution steps for each processing group separately. Whenever parallel execution starts, Execute Processing Groups asynchronously is displayed, followed by separate lines showing the execution of each processing group. The separation into processing groups is also reflected in the downloaded content.

   ○ Under Rendering Statistics, you can view how much time is required to render the components of an analysis application in the Web browser for each navigation step (and initial loading).

   ○ Under General Information, you can view the following information:

     ○ Timestamp of the application execution
     ○ Name and description of the application
     ○ Details about the data sources of the application. For each data source, the data source alias, the name of the object (for example, the query name in BW systems or the view name in SAP HANA systems) representing the data source, the processing group (when parallel query execution is used), the connection type, and the initialization state are listed.

4. After reproducing the performance-critical navigation steps in your analysis application, choose Refresh.

   All the relevant statistics data is downloaded and displayed.

5. To download the statistics data, choose Download as Csv or Download as Text.

Results

You can attach the recorded statistics data (*.csv or *.txt file) to customer messages that you send to SAP.
30.8 Collecting Support Information

Context

If you encounter problems in the design tool, you can collect the relevant information to send to SAP in a zip file.

Procedure

1. In the design tool, choose Help > Support > Collect Support Information...
2. Select the target folder for the zip file.
3. Click OK.
   - The support information is saved in the file DS_Support.zip.
4. To view the content of the zip file, click View...
5. Click OK.

Results

You can attach the zip file to a customer message and send it to SAP.

30.9 Managing Logs in the Design Tool

Defining the Log Level

You can specify how much information is stored in the log file. Under Tools > Preferences > Support Settings, the following options are available:

- **Warning**: The system stores exceptions, error messages, and warnings in the log file. This is the default option.
- **Error**: The system stores exceptions and error messages in the log file.
- **Information**: The system stores exceptions, error messages, warnings, and information messages in the log file.
- **Debugging Information**: The system stores exceptions, error messages, warnings, information messages and debugging information in the log file.
Viewing Error Messages

You can view error messages in the design tool’s Error Log view. To open this view, choose View Error Log.

Saving Error Messages

Press the Export Log button (in the upper right corner of the Error Log view) to save the error messages. The exported error log files help us to resolve problems that you report to SAP in customer messages.

30.10 Best Practices

30.10.1 Increasing Performance of Analysis Applications with Multiple Data Sources

Background

To prevent deadlock scenarios while processing multiple data sources, SAP BusinessObjects Design Studio processes all data source calls sequentially. This may reduce the performance of analysis applications that contain multiple data sources.

Solution Possibilities

There are three ways to increase the performance of analysis applications that contain multiple data sources.

i Note

The second and third solutions are only available as of SAP BusinessObjects Design Studio version 1.2.

1. Selection of data sources initialized at startup

   By default, data sources are initialized at application startup. However, in many business scenarios, it is often a requirement that applications do not display all screens (parts of an application) at startup. You can achieve this by using the components Tabstrip (where only one of the tabs is visible) or Pagebook (where only one page is visible) or simply by setting the component’s property Visible to false. Data sources used to display information that is not immediately visible at application startup should be initialized later (not at application startup). For these data sources, you must set the property Load in Script to true. The script method loadDataSource() must be called when the data source is needed.
Example: In this example, the application contains a tabstrip component, where the first tab (tab with index 0) is visible at startup. The second and third tab refer to data contained in the data sources DS_2 and DS_3 respectively. DS_2 and DS_3 should have Load in Script set to true. The On Select event of the Tabstrip component should contain code similar to this:

```java
if (TABSTRIP_1.getSelectedTabIndex() == 0) {
    // DS_1 is already loaded at the beginning. Nothing to do.
} else if (TABSTRIP_1.getSelectedTabIndex() == 1) {
    DS_2.loadDataSource();
} else if (TABSTRIP_1.getSelectedTabIndex() == 2) {
    DS_3.loadDataSource();
}
```

2. Sharing of data sources among multiple charts
With SAP BusinessObjects Design Studio 1.2, multiple charts can be assigned to a single data source. Each chart can display a different subset of the data contained in a data source, which holds the superset of the displayed data. For example, a data source could contain data for multiple regions (Europe, America, etc). You can then use different charts; each chart displays the data from a different region. To do this, use the Data Selection property of the chart component. The Data Selection property allows you to choose a subset (filter or projection) of the overall data contained in the data source. This allows you to reduce the number of data sources in an application and thereby improve the application performance.

3. Immediate component rendering after data source loading
With SAP BusinessObjects Design Studio 1.2, you, the application designer, can control the order in which the data sources are initialized. In addition, screen rendering of loaded data sources can be enforced, even if other data sources are still being loaded. To do this, use the On Background Processing event. This technology does not increase the overall performance, but the perceived performance of the application user will be improved, because the most important screen parts become visible quickly. While the application user is looking at these visible areas, the other parts of the screen/application are loaded without being perceived as waiting time by the application user.
# 31 User Interface Reference

## 31.1 Properties of the Application

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>NameOfYourApplication</td>
<td>Displays the application name that you entered when creating the application.</td>
</tr>
<tr>
<td>type</td>
<td>NameOfObject</td>
<td>Specifies the object type, which the properties belong to. For an application, the type is <em>Analysis Application</em>.</td>
</tr>
<tr>
<td>description</td>
<td>DescriptionOfYourApplication</td>
<td>Displays the application description that you entered when creating the application. If you did not enter a description, you can do so now.</td>
</tr>
<tr>
<td>folder</td>
<td>YourFolderOnTheBIPlatform</td>
<td>Specifies the folder on the platform, where the application is stored (only relevant for deployment on the BI platform)</td>
</tr>
<tr>
<td>created by</td>
<td>UserNameForOperatingSystem</td>
<td>Displays the user name in the operating system.</td>
</tr>
<tr>
<td>creation time</td>
<td>none</td>
<td>Displays the time stamp when the application was created.</td>
</tr>
<tr>
<td>last modified by</td>
<td>none</td>
<td>Displays the user who made the last change.</td>
</tr>
<tr>
<td>last modification time</td>
<td>none</td>
<td>Displays the time stamp of the last change.</td>
</tr>
<tr>
<td>content version</td>
<td>number</td>
<td>Displays the version number of the application.</td>
</tr>
<tr>
<td>maximum number of steps back</td>
<td>integer</td>
<td>Specifies the number of undo steps permitted in an application. When set to a value more than zero, it also determines if the user can reset their application back to the original state. The default value is zero or disabled. The recommended maximum value is 20.</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>theme</td>
<td>SAP High Contrast Black, SAP Platinum, SAP Blue Crystal, SAP Mobile, SAP Gold Reflection (deprecated), SAP Mobile Black (deprecated)</td>
<td>Specifies the theme of the application. The SAP Platinum theme is recommended for desktop applications and the SAP Mobile theme is recommended for iPhone applications. SAP Gold Reflection and SAP Mobile Black are deprecated themes that may have been used for applications created with a design studio prior to 1.5. If applications with these themes are opened in the design tool, only the technical keys (and not the text) are displayed for these themes in the Theme property of the application.</td>
</tr>
<tr>
<td>custom CSS</td>
<td>none</td>
<td>Specifies the CSS style file for the application. You can change the default CSS style by entering the CSS style file of your choice. Using this function requires detailed knowledge of cascading style sheets (CSS) and Web design.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Restriction</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you use this function, you should be aware that SAP does not guarantee that custom CSS stylesheets will work properly. SAP does not guarantee that no UI-related and/or functionality-related problems will occur. SAP also does not guarantee that custom CSS stylesheets will work properly after software upgrades.</td>
</tr>
<tr>
<td>position of message button</td>
<td>bottom right, bottom left, top right, top left</td>
<td>Specifies the position of the message button. The message button displays errors, warnings and information relating to the application at runtime.</td>
</tr>
<tr>
<td>loading indicator delay</td>
<td>default: 1000</td>
<td>Specifies the delay in milliseconds before the loading indicator is displayed.</td>
</tr>
<tr>
<td>position of message window</td>
<td>right, left</td>
<td>Specifies the position of the message window that is displayed. when the user clicks the message button at runtime.</td>
</tr>
<tr>
<td>displayed message types</td>
<td>none, errors, warnings and errors, all</td>
<td>Specifies the content in the message button.</td>
</tr>
</tbody>
</table>
| merge prompts        | true, false; default: true                                                      | Specifies if the same prompts (variables) that are used in different
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>data sources in an analysis application are merged. If this property is set to true, the same variable used in multiple queries can be entered only once. If this property is set to false, the same variable used in multiple queries must be entered for each SAP BW query separately, and different values can be entered for different queries. For further information, see chapter Unmerging Prompts (Variables) [page 246].</td>
<td></td>
</tr>
<tr>
<td>force prompts on start up</td>
<td>true, false</td>
<td>Specifies whether the prompts dialog box is displayed at runtime as soon as the application is displayed.</td>
</tr>
<tr>
<td>prompt settings</td>
<td></td>
<td>Specifies which prompts should be displayed in which order.</td>
</tr>
<tr>
<td>maximum number of members</td>
<td>your number, default: 100</td>
<td>Specifies the maximum number of members displayed in the value help for prompts. If the number of members is greater than the value for this property, no values are displayed. Instead, the application user has to limit the number by searching for the text or key of one or more values.</td>
</tr>
<tr>
<td>planning connection</td>
<td>none or listed planning system</td>
<td>Specifies the back end connection used for planning scenarios.</td>
</tr>
<tr>
<td>planning model</td>
<td>none or listed planning models</td>
<td>Specifies the planning model (Business Planning and Consolidation (BPC) environment and model) of SAP Business Planning and Consolidation, version for SAP NetWeaver, Unified. This makes it possible to create integrated planning solutions in a BW system, thereby providing high flexibility and usability for specialist users. Using this property causes the design studio’s planning functionality on the server to behave differently. For more information, see SAP Business Planning and Consolidation, version for SAP NetWeaver on the SAP Help Portal.</td>
</tr>
<tr>
<td>global script variables</td>
<td>none</td>
<td>Specifies global script variables that you can use for the whole application. To enter global script variables, choose Edit Global Script Variables. A dialog box appears. Choose Insert. You can now enter the name, type and default value of the global script variable and decide whether you want the variable to be used as a URL parameter.</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| on variable initialization| none           | You can use this event to set query variables. If this event is used to set all mandatory variables, the variables will only be submitted once. The event is executed as follows:  
   ● before the event On Startup  
   ● before the first variable submit  
   Opens the script editor. With this property/event, you can enable user interaction with the application by writing scripts. Click `CTRL + Space` to see the list of available methods for the application, data source alias and the components. Choose one of the following methods to use for this event:  
   ● for the object APPLICATION  
     ○ setVariableValue  
     ○ setVariableValueExt  
     ○ alert  
     ○ createErrorMessage  
     ○ createWarningMessage  
   ● global variables  
   ● all methods for the object CONVERT |

**Note**  
- If all mandatory variables are set by using the On Variable Initialization event, no variable screen is displayed.  
- If there are mandatory variables that are not set by using the On Variable Initialization event, the variable screen is displayed.  
- If the property Force Prompts on Start Up is set to true, the variable screen is displayed regardless of whether one or more mandatory variables are set. All variables set using event On Variable Initialization are set
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>on startup</td>
<td>none</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the application by writing scripts. Click <strong>CTRL</strong> + Space to see the list of available methods for the application, data source alias and the components. Choose one of these methods or use the statement wizard to guide you through the statement creation process. For further information, see the method descriptions in the API reference.</td>
</tr>
</tbody>
</table>
| on background processing                           | none           | You can use this event to load data sources in the background. Although you can use any method in the API, it is recommended that you use this event with the methods for the data source alias. \[Note\] To trigger this event, you must use the script method `doBackgroundProcessing`. This method executes the contents of the event. The execution is run after the script (which calls `doBackgroundProcessing`) has finished and the result has been sent to the front end. Click **CTRL** + Space to see the list of available methods for the application, data source alias and the components. Choose one of these methods (the most suitable methods are the methods for the data source alias) or use the statement wizard to guide you through the statement creation process. For further information, see the method descriptions in the API reference. There are two main scenarios in which the On Background Processing might be especially useful: \mai

There are two main scenarios in which the On Background Processing might be especially useful:

- guided navigation

- unguided navigation
A tree navigation refers to an application concept, where a user can only choose from a limited number of paths at a time, depending on what the user has selected previously. Depending on the measure that the user selects on the first page, three different pages can be shown afterwards. In this example, there are four data sources used, one initially selected and three others, depending on the user’s selection. On Background Processing can be used here to load all three possible data sources in the background, while the user is still looking at the first data source to decide which path he/she will take.

- **tile effect**
  To use a tile effect, where data sources are loaded one after the other, the application designer can use the recursion functionality. If four different data sources are used in the application, and you want them to show their data as soon as one is loaded, the following script could be used:

```javascript
if(Variable1 == 0){
  DS_1.loadDataSource();
}
if(Variable1 == 1){
  DS_2.loadDataSource();
}
if(Variable1 == 2){
  DS_3.loadDataSource();
}
if(Variable1 == 3){
  DS_4.loadDataSource();
}  
Variable1 = Variable1 +1;
if(Variable1 < 4){
  APPLICATION.doBackgroundProcessing();
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>on before prompts submit</td>
<td>none</td>
<td>Your can use this event in combination with the Prompts Settings property. For</td>
</tr>
</tbody>
</table>
### 31.2 Properties of the Data Source Alias

**Syntax**

Data source aliases have the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>name of data source alias</td>
<td>Displays the name of the data source alias, for example, DS_1.</td>
</tr>
<tr>
<td>type</td>
<td>type of data source alias</td>
<td>Displays the type of the data source alias.</td>
</tr>
<tr>
<td>load in script</td>
<td>false, true</td>
<td>Specifies whether the data source alias is initialized in the script, instead of automatically loading the data source when the application is started.</td>
</tr>
<tr>
<td>data source: name</td>
<td>name of underlying data source</td>
<td>Displays the name of the underlying data source, for example, the name of query or query view.</td>
</tr>
<tr>
<td>data source: connection</td>
<td>name of connection</td>
<td>Displays the connection of the data source.</td>
</tr>
<tr>
<td>data source: type</td>
<td>type of data source</td>
<td>Displays the type of the data source, for example, query view.</td>
</tr>
<tr>
<td>processing group</td>
<td></td>
<td>This property can only be used if the design studio runs in BI platform mode or local. Also this property cannot be used for planning enabled queries and for merged variables. Specifies the processing group of a data source. Data sources in the same group are processed sequentially. Each group allocates additional resources (for example, session in backend or thread on application server). As an application designer you need to decide which data sources should run in parallel and therefore need to be assigned to different processing groups:</td>
</tr>
</tbody>
</table>

For more information, see chapter [Unmerging Prompts (Variables)][page 246].
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>● For specifying a group at the beginning of design process, type in capital letters a group name, for example GROUP1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● The first typed in group name will be listed in the property if you want to set the property for another data source.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● If you don’t specify a group name, the so called default group is used. This default group does not allocate a new thread since it uses the current thread. Therefore the following configurations are behaving different:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Configuration 1: DS_1 (&lt;no group&gt;), DS_2 (GROUP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Configuration 2: DS_1 (GROUP1), DS_2 (GROUP2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configurations like (2) where all data sources are assigned to a non-default group are not recommended since it unnecessarily allocates an additional new thread.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information on limiting the number of sessions and threads, see “Configuring the Number of Sessions for Parallel Query Execution” in the Administrator Guide: SAP BusinessObjects Design Studio based on BI Platform.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For further background information on working the parallel query execution, see XXXX.</td>
</tr>
<tr>
<td>text</td>
<td>your Text</td>
<td>This property enables you to add a descriptive text that is displayed in the Prompts dialog box. If no descriptive text is entered, the data source alias specified in the name property will be used, for example, DS_1. This property can be useful when unmerging variables. For further information, see Unmerging Prompts (Variables) [page 246].</td>
</tr>
</tbody>
</table>
on result set changed | none | Opens the script editor. With this property/event, you can enable user interaction with the application by writing scripts. Click `CTRL + Space` to see the list of available methods for the application, data source alias and the components. Choose one of these methods or use the statement wizard to guide you through the statement creation process. For further information, see the method descriptions in the API reference.

The script assigned to this event is executed if the result set for the data source is changed during a roundtrip. The script will be called in the following circumstances:

- if a data source is initially assigned.
- if the query model was changed (for example, by using a filter command) during the roundtrip.

The script is called after all other scripts have been executed. That means that multiple actions that modify this data source lead to one execution of the script.

Note
An exception to this rule: if this data source is modified in an On Result Set Changed handler (of this data source or of a different data source), the script is called again.

### 31.3 General Properties for All Components

The following general properties are available for all components:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Defines the unique name of a component. If you do not enter a name, the system takes the default name (for example CHART_1).</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>Displays the type of the component, for example, crosstab, chart, checkbox.</td>
</tr>
<tr>
<td>visible</td>
<td>true / false</td>
<td>Specifies whether a component is visible or not.</td>
</tr>
<tr>
<td>enabled</td>
<td>true / false</td>
<td>Specifies whether a component is enabled. Disabled components do not allow user interaction.</td>
</tr>
</tbody>
</table>

**Note**

This property is not available in all components.

**Example**

**Hidden buttons become visible**

In an application with a dropdown box for filtering calendar years, the hidden buttons showing the quarters of the calendar year become visible. If the user selects 2011 for example, four buttons appear for filtering the data according to the four quarters of 2011.

**Example**

**Disabled buttons**

An application has a dropdown box for filtering calendar years and four buttons for the quarters of the calendar year. The buttons for quarters - for which no business data is available - are disabled. If the user selects 2012 at the beginning of July 2012 for example, only the first two quarters have business data available. Buttons Q1 and Q2 are enabled, and the user can filter the data for the first two quarters of 2012. Buttons Q3 and Q4 are disabled. The user sees that these buttons exist. However, the fact that these buttons are grayed out indicates that filtering for business data in Q3 and Q4 is not possible. Later on, in August or September for example, the Q3 button becomes active, as business data now exists for this period of time.

### 31.4 Display Properties for All Components

You use the display properties to specify the display of the component at runtime.

The following display properties are available for all components:
Table 34:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSS class</td>
<td>Your CSS class</td>
<td>Defines an additional CSS class for custom CSS. The CSS classes must have the format <code>myclass</code> and not <code>.myclass</code>. You do not have to use this property to be able to use Custom CSS.</td>
</tr>
</tbody>
</table>

### 31.5 Layout Properties for All Components

You can define the layout properties by entering the values manually in the Properties view, or by dragging the borders of a component in the layout editor.

The following properties defining the layout of a component are available for all components:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Margin</td>
<td>Numeric value in pixels or auto</td>
<td>Specifies the distance between the top margin of the component and the top margin of the application. Enter the numeric value in pixels, or set the value to auto.</td>
</tr>
<tr>
<td>Left Margin</td>
<td>Numeric value in pixels or auto</td>
<td>Specifies the distance between the left margin of the component and the left margin of the application. Enter the numeric value in pixels, or set the value to auto.</td>
</tr>
<tr>
<td>Bottom Margin</td>
<td>Numeric value in pixels or auto</td>
<td>Specifies the distance between the bottom margin of the component and the bottom margin of the application. Enter the numeric value in pixels, or set the value to auto.</td>
</tr>
<tr>
<td>Right Margin</td>
<td>Numeric value in pixels or auto</td>
<td>Specifies the distance between the right margin of the component and the right margin of the application. Enter the numeric value in pixels, or set the value to auto.</td>
</tr>
<tr>
<td>Height</td>
<td>Numeric value in pixels or auto</td>
<td>Specifies the height of a component. Enter the numeric value in pixels, or set the value to auto.</td>
</tr>
<tr>
<td>Width</td>
<td>Numeric value in pixels or auto</td>
<td>Specifies the width of a component. Enter the numeric value in pixels, or set the value to auto.</td>
</tr>
</tbody>
</table>
### Example

**Crosstab with fixed margins**

- Top margin = 60
- Left margin = 120
- Bottom margin = 116
- Right margin = 455

The values for width and height are set to `auto`. This means that the distances between the margins of the crosstab and the margins of the application are fixed, while the height and width of the crosstab are variable. The height and width of the application vary according to different screen or window sizes. In this case the height and width of the crosstab vary accordingly.

### Example

**Crosstab with fixed width and height**

If you set the height and width of the crosstab to fixed values, one property of each axis is set to `auto`.

- Top margin = 60
- Left margin = 120
- Bottom margin = `auto`
- Right margin = `auto`
- Width = 600
- Height = 400

In this case, the left margin and the width of the crosstab are fixed, while the third property of the horizontal axis, namely the right margin, is variable. The top margin and the height of the crosstab are fixed, while the third property of the vertical axis, namely the bottom margin, is variable. With different screen or window sizes, the height and width of the application vary. In this case, the bottom margin and the right margin vary accordingly.

**Note**

It is not possible to set all three properties of one axis to fixed values. One property of each axis is always set to `auto`.

### 31.6 Analytic Components

#### 31.6.1 Chart

Use the chart properties in the **Properties** and **Additional Properties** views to configure the settings of the **Chart** component.

You can add charts to your application to present your data graphically. Charts can often emphasize irregularities or trends in your data, and help you focus your business analysis on those areas. When you drag and drop the
Chart component into the layout editor, the component displays a graphic image of a generic chart. When you assign a data source to the Chart component, it then displays the data using the Column chart type, unless you have modified the chart type.

The properties of the Chart component include the following views:

- Properties
- Additional Properties:
  - Chart Area
  - Data Series
  - Chart CSS

The chart properties in the Properties view are described in the table below. The chart properties in the Additional Properties view are described in the Chart Area Additional Properties, Data Series Additional Properties and Chart CSS Additional Properties chapters, which are referenced below the Properties table.

### Properties

Table 35:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data source</td>
<td>Name of data source alias, such as DS_1.</td>
<td>Displays all data source aliases. If you have created several data sources for the application, you can change the data source for the chart by choosing the corresponding data source alias.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Because you can use the same data source several times within one application, you work in the layout editor using data source aliases as reference names.</td>
</tr>
<tr>
<td>data selection</td>
<td>Selection string expressed in a JSON notation generated from the data selection.</td>
<td>Dialog box allows you to select multiple rows or columns from the data result set to create a separate chart. Click the Add Selection button to select the columns or rows that you want to appear in a separate chart.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If your first data selection is in a row, your subsequent data selections must only be in rows. Similarly, if your first data selection is in a column, your subsequent data selections must only be in columns.</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| chart type        | The property values as shown in the Property Description column to the right.                                                                      | - Specifies the type of chart in a drop-down list. Many chart types, such as column, bar, area, bubble, and waterfall, are available to help you visualize your data.  
|                   |                | - The last element in the dropdown list is Additional Chart Types... Selecting this option opens up an Additional Charts dialog box from where you can assign measures and dimensions to additional chart types. |
|                   |                | Note: If you have created and installed new chart extensions using the SAP Lumira SDK, these extensions will also appear in the Additional Chart Types... dropdown list. For more information about creating SAP Lumira SDK extensions, see the SAP Lumira Visualization Extensions Developer Guide on the SAP Help Portal at http://help.sap.com/lumira. Extensions developed with the SAP Lumira SDK can only be deployed to the BI platform and to SAP NetWeaver. They cannot be deployed to SAP HANA. |
| conditional formatting | Selection string expressed in a JSON notation generated from the conditional formatting applied to the selected chart. | Dialog box allows you to apply conditional formatting rules to measures or dimension members within a selected chart. These rules change the appearance of the chart when specific conditions are met. |
| swap axes         | • true  
|                   | • false  | Select true to switch the column and row content. Select false to display the columns and rows as defined in the Edit Initial View... dialog box. The default setting is false. |
| show totals       | • true  
<p>|                   | • false  | When you select true, the totals for each dimension that you have added in the Edit Initial View... dialog box are displayed in the chart. Select false to hide these totals. The default setting is false. |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show scaling factors</td>
<td>● true</td>
<td>Select <code>true</code> to show scaling factors in the chart. The scaling factors displayed are the ones defined for the measures in the <em>Edit Initial View...</em> dialog box. Select <code>false</code> to hide scaling factors. The default setting is <code>false</code>.</td>
</tr>
<tr>
<td></td>
<td>● false</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

1. If your measures are in the columns, the scaling factors will display in brackets in the axes, to the right of the measures. If your measures are in the rows, the scaling factors will display in brackets in the legend, to the right of the measures.

**Example**

Measure A (*1000)

2. When you swap axes, the scaling factors will follow the measures.

3. If there are mixed measures or units, they are not displayed on the chart, but are displayed in the crosstab.

**Example**

If a measure shows Net Sales in US$ and Euro in the crosstab and *Show Scaling Factors* is set to `true` in the chart, the scaling factor will appear beside the measures without the currency value.

Defining scaling factors: There are 2 other ways to define scaling factors.

- Right-click on the measure in the *Edit Initial View...* and select the required scale.
- Script a component with an on-click event.
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension label</td>
<td>• initial view definitions</td>
<td>• Select Initial View Definitions to use the definitions that are set in the Edit Initial View... dialog box.</td>
</tr>
<tr>
<td></td>
<td>• text</td>
<td>• Select the Text option to set all the members to be displayed as text.</td>
</tr>
<tr>
<td></td>
<td>• key</td>
<td>Note: This setting overrides the settings in the Edit Initial View... dialog box.</td>
</tr>
<tr>
<td>on select</td>
<td>none</td>
<td>The On Select event of the chart is triggered when a value is selected or deselected. This property enables you to assign a custom handler to the OnSelect event. To enable user interaction with the chart, use this property to assign a custom handler to the OnSelect event.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Choose the Browse button to open the Script Editor dialog box.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Press [CTRL+SPACEBAR] to see the list of available methods for the application, the data source alias, and the chart.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Choose one of them or the New Statement Wizard option. The New Statement wizard guides you through the process of creating a statement.</td>
</tr>
</tbody>
</table>

**Related Information**

- Working with Charts [page 152]
- Adding a Data Source [page 50]
- Chart CSS Additional Properties [page 281]
- Chart Area Additional Properties [page 273]
- Data Series Additional Properties [page 280]
- Configuring Additional Chart Types [page 182]
- Conditional Formatting [page 184]
- Installing CVOM Chart Extensions for SAP BusinessObjects Design Studio [page 233]
### 31.6.1.1 Chart Area Additional Properties

The *Chart Area* additional properties are described in the table below.

#### Table 36: Chart Area Additional Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>show title</td>
<td>Select the check box <em>Show Title</em> to show the chart title in the chart area.</td>
</tr>
<tr>
<td></td>
<td>title</td>
<td>Populate the <em>Title</em> textbox with a customized chart title.</td>
</tr>
<tr>
<td></td>
<td>align</td>
<td>Select from the three options available to align the chart title within the chart area.</td>
</tr>
<tr>
<td></td>
<td>left</td>
<td></td>
</tr>
<tr>
<td></td>
<td>centred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>right</td>
<td></td>
</tr>
<tr>
<td>background</td>
<td>color:</td>
<td>Select <em>Solid Fill</em> from the dropdown to make the chart background a solid color as defined by the color selection in the color picker.</td>
</tr>
<tr>
<td></td>
<td>solid fill</td>
<td>Select <em>No Fill</em> from the dropdown to make the background transparent.</td>
</tr>
<tr>
<td></td>
<td>no fill</td>
<td>Select the <em>Color Picker</em> button to change the chart background color.</td>
</tr>
<tr>
<td></td>
<td>color picker</td>
<td>Choose the <em>Reset Default</em> button to reset the background color to the default theme color.</td>
</tr>
<tr>
<td></td>
<td>reset default</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

If a chart has a custom background color, changing the theme will not update the color.
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>axes labels and values</td>
<td>● show x-axis</td>
<td>Select the Show X-Axis checkbox to display an axis line with labels and title. Unchecking the checkbox removes the axis line, the labels and the title.</td>
</tr>
<tr>
<td></td>
<td>● show title</td>
<td>Select the Show Title checkbox to display the axis title beside an axis line.</td>
</tr>
<tr>
<td></td>
<td>● title</td>
<td>Populate the Title textbox with a customized axis line title.</td>
</tr>
<tr>
<td></td>
<td>● show x-axis line</td>
<td>Select the Show X-Axis Line checkbox to display an axis line in the chart. Unchecking the checkbox removes only the axis line. Axis line values and title remain visible.</td>
</tr>
<tr>
<td></td>
<td>● line size</td>
<td>Populate the Line Size input field to determine the thickness of an axis line in the chart.</td>
</tr>
<tr>
<td></td>
<td>● show x-axis labels</td>
<td>Select the Show X-Axis Labels checkbox to display the measure names on an axis line.</td>
</tr>
<tr>
<td></td>
<td>● format</td>
<td>Select one of the following suggested format options from the Format dropdown list to define how the data in the chart is formatted:</td>
</tr>
<tr>
<td></td>
<td>○ Default</td>
<td>Applies the same formatting in the chart as applied in the crosstab.</td>
</tr>
<tr>
<td></td>
<td>○ None</td>
<td>Applies no formatting</td>
</tr>
<tr>
<td></td>
<td>○ #,##0</td>
<td>Formats the number with a thousands separator</td>
</tr>
<tr>
<td></td>
<td>○ #,##0.00</td>
<td>Formats the number with a thousands separator, period and two decimal places</td>
</tr>
<tr>
<td></td>
<td>○ $#,##0</td>
<td>Formats the number as a dollar amount</td>
</tr>
<tr>
<td></td>
<td>● axis scaling:</td>
<td>You can also define your own formatting style by manually editing the values displayed in the Format dropdown list.</td>
</tr>
<tr>
<td></td>
<td>○ min value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ max value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Extend Label Capacity</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>● Select the <em>Show Gridline</em> checkbox to display the gridlines on the chart area.</td>
<td>□ Choose from the <em>Type</em> drop-down list, the style of gridline required.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Use the <em>Color Picker</em> button to select the required grid line color. Alternatively, the HEX color value can be entered manually in the input field beside the <em>Color Picker</em> button.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ The <em>Size</em> input field determines the thickness of the gridline in the chart.</td>
<td></td>
</tr>
<tr>
<td>● Select the <em>Axis Scaling</em> checkbox to limit the value axis range displayed in your chart to the range defined in the <em>Min Value</em> and the <em>Max Value</em> input fields. The default value of the <em>Min Value</em> and <em>Max Value</em> is &quot;0&quot;. Removing the default value &quot;0&quot; from the <em>Min Value</em> input field and leaving it blank, ensures that your chart will display the minimum value of your data set. If the <em>Axis Scaling</em> checkbox is not checked, the range reverts back to an automatic range for the axis.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ The <em>Extend Label Capacity</em> checkbox to extend the maximum space taken by the axis labels from the default 25% to 75% of the entire chart area.</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

Axis scaling is available for all chart types except the following:

- 100% Stacked Chart
- Pie Chart
- Select the *Extend Label Capacity* checkbox to extend the maximum space taken by the axis labels from the default 25% to 75% of the entire chart area.
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
</table>
| legend     | • show legend  
• show title  
• title  
• hierarchical  
• position:  
  ○ top  
  ○ bottom  
  ○ right  
  ○ left | • Select the **Show Legend** checkbox to specify if a legend should be displayed in the chart area.  
• Select the **Show Title** checkbox to specify that a legend title should be displayed in the chart area.  
• Populate the **Title** textbox with a customized legend title.  
• Select the **Hierarchical** checkbox to display the legend in a hierarchical way. Selecting this checkbox will display the chart dimensions in the legend in a horizontal list, with each item separated by a backslash.  

**Note**  
The legend must be located to the right of the chart area for the hierarchical property to function correctly.  
• Select one of the **Position** buttons to select the default location of the legend in the chart area.  
  ○ Select **Top** to display the legend above the chart.  
  ○ Select **Bottom** to display the legend below the chart.  
  ○ Select **Right** to display the legend to the right of the chart.  
  ○ Select **Left** to display the legend to the left of the chart. |
| data label | • show data labels:  
  ○ position:  
    ○ inside  
    ○ outside  
  ○ format | • Select the **Show Data Labels** checkbox to display the measure names and dimension members in the chart area.  
  ○ Select **Inside** from the dropdown list to set your preference to display the numeric values inside the chart bars or delineated areas.  
  ○ Select **Outside** from the dropdown list to set your preference to display the numeric values outside the chart bars or delineated areas.  
• For a property description of **Format** see the axes labels and values section above. |
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tooltip</td>
<td>● show tooltips&lt;br&gt;● format&lt;br&gt;● always show within the chart area</td>
<td>● Select the <strong>Show Tooltips</strong> checkbox to display tooltips in the analysis application during runtime.&lt;br&gt;● For a property description of <strong>Format</strong> see the axes labels and values section above.&lt;br&gt;● Select the <strong>Always Show within the Chart Area</strong> to set the position of the tooltip within the chart area.</td>
</tr>
</tbody>
</table>
| plot area  | ● animate on data loading<br>● animate on data updating<br>● animate on data resizing | Select **Plot Area** properties to allow application users to interact with the elements in the chart.  

**Note**  
Plot area properties differ according to the chart type selected.
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interaction</td>
<td>● selection mode:</td>
<td>Define the selection mode the application user will apply at runtime to interact with a chart. Depending on the selection mode, the application user can make their member selections using the legend, the axes or the individual chart elements.</td>
</tr>
<tr>
<td></td>
<td>○ inclusive</td>
<td>Note: It is recommended that the scripting method <code>getSelectedMembers()</code> is used for all selection modes except for <code>single</code>.</td>
</tr>
<tr>
<td></td>
<td>○ exclusive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ single</td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ multiple</td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ none</td>
<td></td>
</tr>
</tbody>
</table>

Select one of the following selection modes:

- Select **inclusive** to allow multiple members to be selected at the same time. The application user can set their member selections by using the click or lasso function in the legend, the axes or the individual chart elements. This persists the selection of previously selected members, whether the member is already selected or not. The default selection mode value is **inclusive**.

- Select **exclusive** to limit the selection option to one member only. Using the lasso function it is also possible to select multiple members in the legend, the axes or the individual chart elements.

- Select **single** to allow the user to make one member selection only by selecting the individual chart elements. This selection mode does not allow legend or axis selection.
### Related Information

- Chart [page 268]
- Data Series Additional Properties [page 280]
- Chart CSS Additional Properties [page 281]
- Clear Selection (clearSelection) [page 369]
### 31.6.1.2 Data Series Additional Properties

The Data series additional properties are described in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayed series format</td>
<td>• axis 1</td>
<td>By default all the existing data series that have been added in the Edit Initial View window, are displayed in the Displayed Series Format panel.</td>
</tr>
<tr>
<td></td>
<td>• axis 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• color picker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• restore defaults</td>
<td></td>
</tr>
<tr>
<td>1. Note</td>
<td></td>
<td>The property values Axis 1 and Axis 2 apply to all dual axis charts only. For dual axis charts the color picker applies to the axes only. The measures automatically get assigned a color based on the color range of the axis color</td>
</tr>
<tr>
<td>2. Note</td>
<td></td>
<td>For all other chart types, the color picker applies to the whole data series.</td>
</tr>
<tr>
<td>displayed measures</td>
<td>• bar</td>
<td>Use the drop-down box to select bar or line to display the measure as a bar or line (depending on the chart type selected) in the chart.</td>
</tr>
<tr>
<td></td>
<td>• line</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• axis 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• axis 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• restore defaults</td>
<td></td>
</tr>
<tr>
<td>1. Note</td>
<td></td>
<td>This property and property value options appear only for the combination dual axis charts.</td>
</tr>
<tr>
<td>data series sequence</td>
<td>• cumulative</td>
<td>Choose Cumulative to display a cumulative total for the dimensions in your Chart.</td>
</tr>
<tr>
<td></td>
<td>• total</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• restore default</td>
<td></td>
</tr>
<tr>
<td>1. Note</td>
<td></td>
<td>This property and property value options appear for waterfall charts only.</td>
</tr>
</tbody>
</table>
31.6.1.3 Chart CSS Additional Properties

The Chart CSS tab contains the list of CSS tags that allow you to manipulate specific formats within the chart component. Table 1 gives a list of the CSS property groups available. Table 2 gives an example of a CSS property group, the properties within that group and the associated default values.

<table>
<thead>
<tr>
<th>Property Group</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.v-m-title .v-title</td>
<td>Defines the style for the title label.</td>
</tr>
<tr>
<td>.v-m-xAxis .v-title</td>
<td>Defines the style for the x-axis title.</td>
</tr>
<tr>
<td>.v-m-xAxis .v-label</td>
<td>Define the style of the y-axis label.</td>
</tr>
<tr>
<td>.v-m-yAxis .v-title</td>
<td>Define the style of the x-axis title.</td>
</tr>
<tr>
<td>.v-m-yAxis .v-label</td>
<td>Define the style of the y-axis title.</td>
</tr>
<tr>
<td>.v-m-legend .v-title</td>
<td>Define style for the legend title.</td>
</tr>
<tr>
<td>.v-m-legend .v-label</td>
<td>Define style for the legend label.</td>
</tr>
<tr>
<td>.v-m-legend .v-scrollbarThumb</td>
<td>Define style for legend scrollbar thumb.</td>
</tr>
<tr>
<td>.v-m-datalabel .v-datalabel</td>
<td>Define style for the data label.</td>
</tr>
</tbody>
</table>

Example
### 31.6.2 Crosstab

The crosstab displays multi-dimensional data in a grid with analytic functions. After you have dragged and dropped a crosstab in the layout editor, the crosstab is initially displayed with scrollbars and dummy data. As soon as you assign a data source to the crosstab, the data of the data source is displayed in the crosstab.

- **Note**
  - The scrollbars of the crosstab displayed in the editor are not intended to be used during design time. They are only intended for layout and design purposes.
  - The scrollbars are only displayed if the crosstab has a minimum size (row headers, column headers and a specific number of data cells).

The **Crosstab** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data source</td>
<td>name of data source alias (for example, “DS_1”)</td>
<td>Displays all data source aliases. If you have created multiple data sources for the application, you can change the data source for the crosstab by choosing the relevant data source alias.</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pixel-based scrolling</td>
<td>true, false</td>
<td>Enables smooth scrolling experience. We recommend using this property when creating mobile applications and/or applications with low data volume. If you do not activate pixel-based scrolling, then row-based/column-based scrolling will be applied by default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>As you can use the same data source several times within one application, you work in the design tool with data source aliases as reference names.</td>
</tr>
<tr>
<td>row limit</td>
<td>numeric value for rows; default: 20 (max)</td>
<td>Specifies the maximum number of rows for pixel-based scrolling. With this property, you can set the number of rows to be displayed. For iPad applications, SAP recommends a maximum of 20 rows.</td>
</tr>
<tr>
<td>column limit</td>
<td>numeric value for columns; default: 20 (max)</td>
<td>Specifies the maximum number of columns for pixel-based scrolling. With this property, you can set the number of columns to be displayed. For iPad applications, SAP recommends a maximum of 20 columns.</td>
</tr>
<tr>
<td>navigating hierarchies enabled</td>
<td>true, false</td>
<td>Specifies whether the application user can expand or collapse hierarchy nodes,</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>if the crosstab contains one or more hierarchies or hierarchical structures. If this property is set to false, the hierarchy expand/collapse icons (plus and minus symbols) are not shown in the crosstab. However, the end user can still see that there is a hierarchy because of the indentation of the nodes and leaves.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sorting enabled</td>
<td>true, false</td>
<td>Specifies whether the end user can sort columns. If the property is set to false, the icons for sorting are not shown in the column headers.</td>
</tr>
<tr>
<td>column resizing enabled</td>
<td>true, false</td>
<td>Specifies whether the end user can resize columns by double-clicking on the right border of the column header. If this property is set to false, the hover spot for the double-click resize will not be available.</td>
</tr>
<tr>
<td>horizontal header resizing enabled</td>
<td>true, false</td>
<td>Specifies whether the application user can move the vertical boundary between data area and row header area and/or dimension header area and column header area interactively. The default value is false. When set to true, the user will be able to move the boundary by clicking and dragging the line between the respective boundary (see above).</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td>If you have specified a maximum header width in pixels, the user cannot extend the width of the header area beyond this value.</td>
</tr>
<tr>
<td>enable horizontal scrolling for header</td>
<td>true, false</td>
<td>Specifies whether the header area will have a horizontal scrollbar. When set to false (default), the crosstab will not have a scrollable header area at runtime. If set to true, the header area in the crosstab will have a horizontal scrollbar (if required).</td>
</tr>
<tr>
<td>context menu enabled</td>
<td>true, false</td>
<td>Specifies whether a context menu can be displayed on the crosstab. The context menu can only be displayed on</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>the crosstab, if the property is set to true and the technical component CONTEXT_MENU is available in the application (in the Outline view in the Technical Components folder). If the property is set to false, the context menu cannot be displayed regardless of whether the technical component CONTEXT_MENU is part of the application or not. The entries of the context menu depend on the element of the data source the user clicks on. For more information, see</td>
<td></td>
<td></td>
</tr>
<tr>
<td>units and scaling factors</td>
<td>display both in header, display units in data cells, do not display</td>
<td>Specifies how units and scaling factors are shown in the crosstab. Measures can have scaling factors and units or currencies. If you have chosen Display both In Header, an additional header row is added to the row or column header. This additional header row contains information about the scaling factor and the unit or currency - provided that the information is consistent (for example, EUR 1,000).</td>
</tr>
<tr>
<td>always fill</td>
<td>true, false</td>
<td>Specifies whether the component space should be filled as defined by the layout properties. If you set the property value to false, the crosstab is displayed in its actual size, but within a frame that has the size defined in the layout properties. If you set the property to true, the crosstab is adjusted and displayed in the exact size defined in the layout properties.</td>
</tr>
<tr>
<td>conditional formatting visible</td>
<td>true, false</td>
<td>Specifies whether conditional formats (exceptions) that have been defined in the BEX Query Designer or in SAP BusinessObjects Analysis, edition for Microsoft Office (Analysis), are shown in the crosstab.</td>
</tr>
</tbody>
</table>

**Note**
- To return the names of conditional formats, this property must be set to true
- In the BEx Query Designer you can define threshold values (exceptions) for a query. Data that deviates from these exceptions is marked in different colors. You can use these
exceptions to spot deviations from expected results straight away. The exception visualization is based on nine alert levels. For each alert level, the affected cells are displayed in the corresponding background color. For more information, see the documentation for the BEx Query Designer on SAP Help Portal at http://help.sap.com.

- In Analysis, any results that fall outside a set of predetermined threshold values (rules for conditional formatting) are highlighted in color or designated with symbols. For more information, see the SAP BusinessObjects Analysis, edition for Microsoft Office User’s Guide on SAP Help Portal at http://help.sap.com/boaa.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
</table>
| column widths  | numeric values >= 0 | Sets the column widths of the crosstab. If you select this property, the Edit Columns Widths dialog box is displayed. Choose Insert to set the column widths. A valid column width setting consists of two parts:

- **Column Index**: The index of the addressed column. The index must be a number >= 0. The leftmost column of the crosstab is column 0, the second is column 1 and so on. If the column index addresses a column outside the total number of columns in the crosstab, the setting will be ignored.

- **Column Width**: The column width must be a number >= 0. The width unit is pixels. There is a minimum column width for each column even when a column width is set to 0, depending on the theme and the applied custom CSS styles. The minimum width depends on the
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>settings for borders, padding,</td>
<td></td>
<td>settings for borders, padding, margin and similar style attributes of the crosstab cells that can be changed by the customer. Therefore it is not possible to specify the exact minimum width.</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td>It is not possible to set a column width that is smaller than this minimum width. You cannot use this feature to hide columns by trying to set a width of 0 pixels for a column in the crosstab.</td>
</tr>
<tr>
<td>maximum width of header area</td>
<td>auto, integer values &gt; 0</td>
<td>Specifies the header area width of the crosstab. If the property is set to auto (default setting), the header uses as much horizontal space as it needs. You can limit the crosstab header area by entering an positive integer value &gt; 0, which represents the maximum row and/or dimension header width in pixels.</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td>This property cannot be used to hide the header area.</td>
</tr>
<tr>
<td>show repeated texts</td>
<td>true, false</td>
<td>Specifies if multiple header cells with the same texts are merged into single cells with row spans or column spans</td>
</tr>
<tr>
<td>number of new rows</td>
<td>numeric value</td>
<td>Specifies the number of new rows for manual planning applications.</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>position of new rows</td>
<td>bottom, top</td>
<td>Specifies the position of new rows for planning applications.</td>
</tr>
<tr>
<td>maximum number of value help</td>
<td>numeric value (default: 100)</td>
<td>Specifies the maximum number of members displayed in a value help for non-hierarchical dimensions.</td>
</tr>
<tr>
<td>members</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enable selection</td>
<td>none (default), single, multi, data cell</td>
<td>Specifies if and which kind of data selection is offered for the application user at runtime:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If set to value <strong>None</strong>, selection is completely disabled. The user cannot select anything in the crosstab. There is no hovering effect even if the property <strong>Enable Hover Effect</strong> is set to <strong>true</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If set to value <strong>Single</strong>, the user can select rows and columns in the crosstab by hovering over and clicking the members of the required dimensions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If set to value <strong>Multi</strong>, multiple selection is enabled. This means you can select multiple rows and columns by pressing <strong>Ctrl + Left</strong> mouse button on the required rows and columns. To remove a single selected row or column from multiple selected rows or columns, press <strong>Ctrl + Left</strong> mouse button on the selected row or column again. You</td>
</tr>
</tbody>
</table>

**Note**

If you open an application with the formerly used property **Selection Enabled** set to **true** on the crosstab but the application was created with an older version of the design tool (prior to SAP BusinessObjects Design Studio 1.5), the formerly used **Enable Selection** property automatically transferred to **Enable Selection** with value **Single**.
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>can also achieve multi selection by choosing a member of an outer dimension with children.</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td></td>
</tr>
<tr>
<td>○</td>
<td></td>
<td>Multiple selection is not possible on mobile devices.</td>
</tr>
<tr>
<td>○</td>
<td></td>
<td>It is not possible to specify multiple selection by clicking the left mouse key on a start row/column and then clicking <code>Shift</code> + <code>Left</code> mouse button on the end row/column.</td>
</tr>
<tr>
<td>○</td>
<td></td>
<td>Each time the user performs an individual selection, an On Select event is triggered.</td>
</tr>
<tr>
<td>○</td>
<td></td>
<td>You can only select multiple members that have a parent dimension (for example all regions of Germany). If you change the parent dimension (for example, from Germany to Cuba), the already selected members get deselected as they do not belong to this group.</td>
</tr>
<tr>
<td>●</td>
<td></td>
<td>If set to value Data cell, the selection of a single data cell is enabled. This means all members on the row and column axes that specify this data cell are highlighted and selected as well. The application designer can get the full selection with the API method <code>getSelection</code> or get the relevant members using <code>getSelectedMember</code>. It is not possible to select more than one data cell.</td>
</tr>
<tr>
<td>selection area</td>
<td>all (default), rows, columns</td>
<td>Specifies if the application user can select the rows or columns in the</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>header area. However, a selection cannot exist at the same time for both rows and columns. If rows are selected, all previously made column selections are undone (same applies for the other way round).</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td>This property is only useful, if you have set the property Enable Selection to Multi or Single.</td>
</tr>
<tr>
<td>enable hover effect</td>
<td>true (default), false</td>
<td>This property does not have any effect if selection is disabled in the crosstab. If selection is generally enabled, the property specifies if hover effects should be displayed when the mouse pointer is moved over a selectable cell.</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td>On mobile devices, there are no hover effects as there is no mouse pointer.</td>
</tr>
<tr>
<td>on select</td>
<td>none</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the crosstab by writing scripts. Click CTRL + Space to see the list of available methods for the application, the data source alias and the crosstab. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. The following specific methods are currently available for the crosstab:</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td>For further information, read the relevant method description in the API reference.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The On Select event of the crosstab is triggered when a value is selected and also when a value is deselected.</td>
</tr>
</tbody>
</table>
### 31.6.3 Dimension Filter

#### Syntax

Using the dimension filter, you can add a generic filter for one dimension to an application, without the need to use scripting. The dimension filter provides an easy way for the end user to define a filter for a single dimension of a data source. Click on the dimension filter at runtime to open a popup with the "selection view" of the dimension.

The **Dimension Filter** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data source</td>
<td>name of data source alias (for example &quot;DS_1&quot;)</td>
<td>Displays all data source aliases. If you have created several data sources for the application, you can change the data source for the filter panel by choosing the relevant data source alias.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- As you can use the same data source several times within one application, you need to work in the design tool using data source aliases as reference names.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Keep in mind that the data source is filtered automatically and you do not have to add it to the Target Data Sources property.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you change the data source, the system automatically refreshes the list of the Target Data Sources.</td>
</tr>
<tr>
<td>target data sources (optional)</td>
<td>names of target data source aliases (for example, &quot;DS_1; DS_2&quot;)</td>
<td>If you want the filter to be applied to other, additional data sources, you can also define target data sources. If the target data sources contain the same dimension as the (source) data source, the filter applies the dimension to the target data sources in the same way as in the (source) data source. The target data sources must be existing data source aliases added to the application and separated by a semicolon.</td>
</tr>
<tr>
<td>dimension</td>
<td>dimension</td>
<td>Specifies the dimension to be filtered. You can select a dimension using the Dimension Selection dialog box or search for a dimension.</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dimension name</td>
<td>true, false</td>
<td>Specifies whether the dimension name of the selected dimension is displayed in the dimension filter.</td>
</tr>
<tr>
<td>display mode</td>
<td>filter list, filter count</td>
<td>Specifies the display mode of the dimension members. If you choose Filter List, the applied members (= filter values) are displayed as a comma-separated list (for example, Fax, Internet, Phone). If you choose Filter Count, the number of the applied members (= filter values) is displayed in parentheses (for example, (3)).</td>
</tr>
<tr>
<td>member display</td>
<td>key, text, key and text, text and key</td>
<td>Specifies the dimension member display.</td>
</tr>
<tr>
<td>popup width</td>
<td>numeric value in pixel</td>
<td>Specifies the width of the popup displayed when the user clicks on the dimension filter at runtime.</td>
</tr>
<tr>
<td>popup height</td>
<td>numeric value in pixel</td>
<td>Specifies the height of the popup displayed when the user clicks on the dimension filter at runtime.</td>
</tr>
<tr>
<td>popup position</td>
<td>bottom right, bottom left, center right, center left, top right, top left, bottom center, top center</td>
<td>Specifies the position of the popup displayed when the user clicks on the dimension filter at runtime.</td>
</tr>
<tr>
<td>popup is modal</td>
<td>true, false (default)</td>
<td>Specifies whether the value help popup dialog box of the dimension filter is modal. If the property is set to true, the user cannot interact with other UI elements in the background of the application (for example, click a button or dropdown box). The user can only interact with the elements in the value help popup dialog box. If the property is set to false, users can interact with other UI elements.</td>
</tr>
<tr>
<td>remove redundant selections</td>
<td>true, false</td>
<td>Specifies whether the system removes redundant selections in hierarchical dimensions. If the property is set to true and if the application user selects a node and a subnode, the system removes the subnode because it is automatically selected when the parent node is selected.</td>
</tr>
<tr>
<td>auto apply</td>
<td>true, false (default)</td>
<td>Specifies if changes made by the application user at runtime are applied automatically. If set to false, the application user has to click on Apply after making changes in the filter panel. The filtering changes are also applied if the application user uses the direct</td>
</tr>
</tbody>
</table>
### Filter Line

The **Filter Line** analytic component offers the user a list of filters that are applied to the assigned data source. It behaves in a similar way to the **Filter Panel**. With the **Filter Line**, the user can add, remove, view and edit the dimensions and measures to which they can apply filters. When the user selects to add or edit a filter, the **Dimension Filter** will appear. The filter on the selected dimension and member(s) is added to the row of filters in the **Filter Line** and the filter is applied to your data source.

Filter items can be directly removed from the **Filter Line**, without having to drill down into the **Dimension Filter**.

On SAP HANA the measure dimension will be **Measures**. On SAP BW the measure dimension will be **Key Figures**.

**Note**

Before working with the **Filter Line**, you should refer to the following SAP Note:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>input and then clicks on the Back button or OK button</td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum number of members</td>
<td>your number, default: 100</td>
<td>Specifies the maximum number of members displayed in the value help for non-hierarchical dimensions. If the number of members is greater than the value for this property, no values are displayed. Instead, the application user has to limit the number by searching for the text or key of one or more values.</td>
</tr>
<tr>
<td>desktop style for popup</td>
<td>true, false</td>
<td>Specifies the style of the value help popup and enables you to switch between mobile style and desktop style (changes layout of buttons). If the property is set to true, the value help popup is displayed in desktop style.</td>
</tr>
<tr>
<td>on apply</td>
<td>none</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the dimension filter by writing scripts. The <strong>On Apply</strong> event is triggered when the application user presses the <strong>Apply</strong> button of the dimension filter at runtime. In the script editor, click <strong>CTRL + Space</strong> to see the list of available methods for the application, the data source alias and the dimension filter. Choose one of the methods or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference.</td>
</tr>
</tbody>
</table>
The **Filter Line** component has the following specific property:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure visible</td>
<td>true, false</td>
<td>Select true to include measures in the list of data source filters in the <strong>Filter Line</strong> menu.</td>
</tr>
</tbody>
</table>

**Related Information**

Filter Panel [page 294]
Dimension Filter [page 291]

### 31.6.5 Filter Panel

**Syntax**

With the filter panel, you can add a generic filter to an application without using scripting. The filter panel makes it easy for the end user to define a filter for the data source. To use the filter panel, you have to add at least one data source to the application and assign it to the filter panel.

**Note**

If a dimension has more than 100 members, the members are not listed at runtime. Instead the user is asked to use the search function.

The **Filter Panel** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data source</td>
<td>name of data source alias (for example “DS_1”)</td>
<td>Displays all data source aliases. If you have created several data sources for the application, you can change the data source for the filter panel by choosing the relevant data source alias.</td>
</tr>
</tbody>
</table>

**Note**

- As you can use the same data source several times within one
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application, you need to work in the design tool using data source aliases as reference names. Keep in mind that the data source is filtered automatically and you do not have to add it to the Target Data Sources property. If you change the data source, the system automatically refreshes the list of the Target Data Sources.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>target data sources (optional)</td>
<td>names of target data source aliases (for example &quot;DS_1; DS_2&quot;)</td>
<td></td>
</tr>
<tr>
<td>mode</td>
<td>filtering, navigation, filtering and navigation</td>
<td></td>
</tr>
<tr>
<td>dimensions</td>
<td>dimensions</td>
<td></td>
</tr>
<tr>
<td>display mode</td>
<td>filter list, filter count</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Count</td>
<td></td>
<td><em>Count</em>, the number of the applied members (=filter values) is displayed in parentheses (for example, <em>(3)</em>).</td>
</tr>
<tr>
<td>member display</td>
<td>key, text, key and text, text and key</td>
<td>Specifies the display mode of the dimension members.</td>
</tr>
<tr>
<td>title</td>
<td>yourTitel</td>
<td>You can set a title for the filter panel. The title is displayed in the middle of the filter panel header.</td>
</tr>
<tr>
<td>remove redundant selections</td>
<td>true, false</td>
<td>Specifies whether the system removes redundant selections in hierarchical dimensions. If the property is set to <em>true</em> and if the application user selects a node and a subnode, the system removes the subnode as this one is automatically selected if the the parent node is selected.</td>
</tr>
<tr>
<td>drag &amp; drop</td>
<td>true, false</td>
<td>Specifies whether the application user can change the axis of the dimension in navigation mode by using drag &amp; drop.</td>
</tr>
<tr>
<td>direct input for filter</td>
<td>enabled, disabled for measures, disabled</td>
<td>Specifies whether the application user can use direct input for dimension or measure keys. If set to <em>Disabled for Measures</em>, the application user cannot use direct input for measures but only for dimensions.</td>
</tr>
<tr>
<td>auto apply</td>
<td>true, false</td>
<td>Specifies if all changes made by the application user at runtime are applied automatically. If set to <em>false</em>, the application user has to click on <em>Apply</em> after making changes in the filter panel. All changes in navigation will be applied immediately. The changes in the filter panel are also applied if the application user uses the direct input and then clicks on the <em>Back</em> button or <em>OK</em> button.</td>
</tr>
<tr>
<td>maximum number of members</td>
<td>your number. default: 100</td>
<td>Specifies the maximum number of members displayed in the value help for non-hierarchical dimensions. If the number of members is greater than the value for this property, no values are displayed. Instead, the application user has to limit the number by searching for the text or key of one or more values.</td>
</tr>
<tr>
<td>on apply</td>
<td>none</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the filter panel by writing scripts. Click <em>CTRL</em> + <em>Space</em> to see the list of available methods for the application, the data source alias and...</td>
</tr>
</tbody>
</table>
31.6.6 Geo Map

The Geo Map analytic component allows you to display different layers of geographical information on a map and allow users to drill down through the different layers to reveal data in a variety of ways. You can apply three different types of layers to your Geo Map - shapes (polygons and multi-polygons), points and bubble charts.

You can assign a different data source to each layer.

**Note**

You must read the following SAP Note before using the Geo Map component:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2166923</td>
<td>Working with SAP BusinessObjects Design Studio 1.5 Geo Maps.</td>
</tr>
</tbody>
</table>

The properties of the Geo Map component include the following views:

- Properties
- Additional Properties
The Geo Map Properties and the Additional Properties views are described in the tables below.

Properties

Table 40:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>basemap url</td>
<td>url</td>
<td>The URL template which is used to provide the basemap. The parameters LOD, X and Y are expanded to provide a URL for each individual basemap. For example, <code>my.map/{LOD}/{X}/{Y}.png</code>. Selecting to have a basemap is optional.</td>
</tr>
<tr>
<td>basemap copyright</td>
<td>text</td>
<td>Text which is displayed at the bottom right of the map. The basemap provider may require you to specify a reference to the copyright text for the basemap url you have selected to use.</td>
</tr>
<tr>
<td>map legend visible</td>
<td>true, false</td>
<td>Specifies whether the application user can set the map legend to visible.</td>
</tr>
</tbody>
</table>

Additional Properties

Table 41:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>unique layer id</td>
<td>Enables you to assign a unique identifier to each layer of your Geo Map. This ID is then used by the Geo Map scripting methods.</td>
</tr>
</tbody>
</table>

i Note

IDs must be valid css selectors. A name must begin with an underscore, a hyphen, or a letter, followed by any number of hyphens, underscores, letters, or numbers. If the first character is a hyphen, the second character must be a letter or underscore. The name must be at least two characters long.
<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show layer</td>
<td>checkbox</td>
<td>Sets the visibility of the layer. When this checkbox is selected, the layer is visible on the <strong>Geo Map</strong>. This checkbox is selected by default.</td>
</tr>
</tbody>
</table>
| type | ● shape  
● point  
● chart | Specifies the type of layer to display data on the **Geo Map**. The type selected determines which other properties become available. The **Shapes** layer is the default layer type. |
| data source | drop down list of available data sources | Enables you to select a data source to assign to the current layer in the **Geo Map**. |
| measure | drop down list of the available measures | Enables you to assign a measure to a shape or chart layer. |
| start color | hex, color picker | You can use a color picker to specify the start color of a shape layer color gradient. The default start color is white. |
| end color | hex, color picker | You can use a color picker to specify the end color of a shape layer color gradient. The default end color is black. |
| custom geoJSON file | text | Specifies the file path to the GeoJSON file. |
| geoJSON mapping property | drop down list | Provides a list of all available properties in the GeoJSON file. |
| geoJSON mapping type | key, text | Specifies whether key or text of the geo dimension member is used when mapping to GeoJSON. |
| latitude | drop down | Enables you to assign a latitude measure name to points and charts layers. |
| longitude | drop down | Enables you to assign a longitude measure name to points and charts layers. |
| color | hex, color picker | Assigns a color to points and charts layers. |

### 31.6.7 Navigation Panel

Using the navigation panel, you can easily change the drilldown of the data and see at a glance the navigation state of the data source at runtime. To change the drilldown, the application user can drag and drop the dimensions from the list into the rows or columns area. The application user can also rearrange the drilldown by dragging and moving dimensions from the rows into the columns area and/or the other way round. To remove dimensions from the drilldown, the application user can drag a dimension and drop it anywhere in the application.

The **Navigation Panel** component has the following specific properties:
### 31.7 Basic Components

#### 31.7.1 Button

Buttons enable the user to interact within the application. To configure this interaction, you add a script to the button’s On Click property. The script is triggered when the user clicks the button in the application. You can insert any text for the button label and add an icon.

The **Button** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>default text: button</td>
<td>Specifies the text to be displayed on the button. You can overwrite the default text and enter your own text. The text is centered on the button.</td>
</tr>
</tbody>
</table>
| icon               | name of icon; save location              | Specifies the icon to be displayed on the button. Insert the name of the image you want to add. Use one of the following methods:  
  - If the icon is stored in the application directory  
    `<userhome>\Analysis_workspace\com.sap.ip.bi.zen>` |
The icon of a button can be specified in the following ways:

- If the icon is stored in a subfolder of the application directory, type in the relative path, for example, `MySubFolder\MyImage.jpg`.
- If the icon is stored on the Internet/intranet, type in the URL, for example, `http://www.myWebServer.com/myImage.jpg`.
- If your Design Studio is connected to a SAP NetWeaver platform, click on the `Browse` button. The `Open Image` dialog box opens, where you can select the image or icon from the MIME repository.

> **Note**
> In the MIME repository, you can select images and icons uploaded by the system administrator. For more information, see the `Administrator Guide: SAP BusinessObjects Design Studio` at [http://help.sap.com/boad](http://help.sap.com/boad).

If you select a text and an icon for the button, the icon is aligned left of the text. If you only select an icon, it is centered on the button.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repository</td>
<td>\MyApplication</td>
<td>type in the name of the icon (MyIcon.jpg).</td>
</tr>
<tr>
<td></td>
<td>\MyApplication</td>
<td>If the icon is stored in a subfolder of the application directory, type in the relative path, for example, MySubFolder\MyImage.jpg.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the icon is stored on the Internet/intranet, type in the URL, for example, <a href="http://www.myWebServer.com/myImage.jpg">http://www.myWebServer.com/myImage.jpg</a>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If your Design Studio is connected to a SAP NetWeaver platform, click on the Browse button. The Open Image dialog box opens, where you can select the image or icon from the MIME repository.</td>
</tr>
<tr>
<td>on click</td>
<td>script</td>
<td>Opens the script editor. Using this property, you can enable user interaction with the button by writing scripts. The On Click event is triggered when the application user clicks on the button. In the script editor, click <code>CTRL + Space</code> to see the list of available methods for the application, the data source alias and the button. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference.</td>
</tr>
</tbody>
</table>
31.7.2 Chart Type Picker

The Chart Type Picker basic component allows the application user to select an alternative visualization type to display their data, while actually running their application. You can manipulate the Chart Type Picker properties to add additional visualization types to the existing palette of types. You can decide, through scripting, how the visualizations behave when the user runs the application. A dropdown list beside each group of visualizations, displays a list of each visualization type within each group. If you add an additional type, for example, a crosstab or an image component, to the Chart Type Picker, they will be grouped together in a new group.

### Note

Before working with the Chart Type Picker, you should refer to the following SAP Note:

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2169108</td>
<td>Supported Languages for the Filter Line and Chart Type Picker components in SAP BusinessObjects Design Studio 1.5.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chart reference</td>
<td>dropdown list</td>
<td>Specifies which chart component in the application is changed when a different chart type is selected in the Chart Type Picker. The chart component type will change to reflect this chart type selection once the selection made is different to the chart component’s current type.</td>
</tr>
</tbody>
</table>
| additional types | ellipsis button that pops up a dialog box | Dialog is displayed to allow you to add or remove additional visualization types. Each additional visualization type requires the following:  
- name  
- type  
- icon |

### Note

- The additional visualization type entered must be unique. It must not be one of the standard chart types already available
- All additional visualization types added, will be grouped together in a new group within the Chart Type Picker component.
31.7.3 Checkbox

Checkboxes enable the user to interact within the application. To do this, add a script to the On Click property of the checkbox. The script is triggered when the user activates the checkbox in the application. You can use any text for the checkbox.

The Checkbox component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>default text: checkbox</td>
<td>Specifies the text to be displayed on the right of the checkbox. You can overwrite the default text and enter your own text.</td>
</tr>
<tr>
<td>checked</td>
<td>false/true</td>
<td>Specifies whether the checkbox is initially selected.</td>
</tr>
<tr>
<td>on click</td>
<td>script</td>
<td>Opens the script editor. With this property, you can enable user interaction with the checkbox by writing scripts. The On Click event is triggered when the application user activates the checkbox. In the script editor, press <code>CTRL</code>+<code>Space</code> to see the list of available methods for the application, the data source alias and the checkbox. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the method description in the API reference.</td>
</tr>
</tbody>
</table>

31.7.4 Checkbox Group

The checkbox group displays several checkboxes, each for one item. All checkboxes are aligned in a vertical fashion. Checkbox groups enable the user to interact within the application. To do this, add a script to the On Select property of the checkbox group. The script is triggered when the user activates or deactivates one checkbox in the application.

The Checkbox Group component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>items</td>
<td>list of items</td>
<td>With the items property, you can open the Edit items dialog. Here, you can insert items, change the order of items or remove them. Each item has the following properties:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value</td>
</tr>
</tbody>
</table>
### 31.7.5 Date Field

The date field component enables the user to select a date. To activate user interaction with the date field, you need to add a script to the On Select property. The script is triggered when the user picks a date in the date field or when the user enters a date manually at runtime.

The **Date Field** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>date</td>
<td>Specifies the initial date in the format yyyymmdd.</td>
</tr>
<tr>
<td>on select</td>
<td>script</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the date field by writing scripts. Click <code>CTRL + Space</code> to see the list of available methods for the application, the data source alias and the checkbox group. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the method description in the API reference.</td>
</tr>
</tbody>
</table>
31.7.6 Dropdown Box

Dropdown boxes enable the user to select items, for example, to set a filter. To enable user interaction, you need to add a script to the On Select property of the dropdown box. The script is triggered when the user selects an item in the dropdown box.

The **Dropdown Box** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>items</td>
<td>list of items</td>
<td>With the items property, you can open the Edit items dialog box. Here, you can insert items, change the order of items or remove them. Each item has the following properties:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• text (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The text property is optional. If you enter a text for an item, the text is displayed. If no text is displayed, the value for the item is displayed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you want a specific item to be displayed by default, select the relevant item and click <strong>Set as Default</strong>. This is shown as the default entry in the dropdown box.</td>
</tr>
<tr>
<td>on select</td>
<td>script</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the dropdown box by writing scripts. The On Select event is triggered when the application user selects an item from the dropdown box. In the script editor, click <strong>CTRL + Space</strong> to see the list of available methods for</td>
</tr>
</tbody>
</table>
Example

You have created an application with a crosstab. You have assigned the data source alias DS_1 to the crosstab. In addition, you have added a dropdown box to the application. This dropdown box contains items that represent the values of the dimension DEMO_CHA2 of DS_1. Using the on select event of the dropdown box, enter the following script:

```java
DS_1.setFilter("DEMO_CHA2", DROPDOWN_1.getSelectedValue());
```

This event triggers the following action: the crosstab is filtered by the value selected in the dropdown box.

31.7.7 Formatted Text View

The Formatted Text View component enables you to apply text formatting within one control. The set of tags and attributes that can be used within this control are listed in the table below. To use the Formatted Text View component, you must drag the component onto the Layout Panel. The Formatted Text View editor appears in the Additional Properties panel.

The Formatted Text View editor supports the following input:

<table>
<thead>
<tr>
<th>Input Type</th>
<th>Input Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>functions</td>
<td>• cut</td>
</tr>
<tr>
<td></td>
<td>• copy</td>
</tr>
<tr>
<td></td>
<td>• paste</td>
</tr>
<tr>
<td></td>
<td>• paste as plain text</td>
</tr>
<tr>
<td></td>
<td>• undo</td>
</tr>
<tr>
<td></td>
<td>• redo</td>
</tr>
<tr>
<td>font</td>
<td>• bold</td>
</tr>
<tr>
<td></td>
<td>• italic</td>
</tr>
<tr>
<td></td>
<td>• remove format</td>
</tr>
</tbody>
</table>
**Input Type** | **Input Options**
--- | ---
**styles** | • block styles  
  ○ address  
  • inline styles  
  ○ cited work  
  ○ inline quotation
**paragraph format** | • normal  
  • h1  
  • h2  
  • h3  
  • h4
**list** | • insert/remove numbered list  
  • insert/remove bulleted list
**layout** | • decrease indent  
  • increase indent

### Supported HTML Tags

The following table lists all the supported HTML Tags for this component.

<table>
<thead>
<tr>
<th>HTML Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abbr</td>
<td>abbreviation</td>
</tr>
<tr>
<td>acronym</td>
<td>acronym</td>
</tr>
<tr>
<td>address</td>
<td>format as address</td>
</tr>
<tr>
<td>blockquote</td>
<td>blockquote</td>
</tr>
<tr>
<td>br</td>
<td>line break</td>
</tr>
<tr>
<td>cite</td>
<td>cite</td>
</tr>
<tr>
<td>dd</td>
<td>definition item</td>
</tr>
<tr>
<td>dfn</td>
<td>definition</td>
</tr>
<tr>
<td>div</td>
<td>div</td>
</tr>
<tr>
<td>dl</td>
<td>definition list</td>
</tr>
<tr>
<td>dt</td>
<td>definition topic</td>
</tr>
<tr>
<td>em</td>
<td>emphasis</td>
</tr>
<tr>
<td>h1</td>
<td>header level 1</td>
</tr>
<tr>
<td>h2</td>
<td>header level 2</td>
</tr>
<tr>
<td>h3</td>
<td>header level 3</td>
</tr>
<tr>
<td>h4</td>
<td>header level 4</td>
</tr>
<tr>
<td>HTML Tag</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>kbd</td>
<td>text to be entered via keyboard</td>
</tr>
<tr>
<td>li</td>
<td>list item for “ol” and “ul”</td>
</tr>
<tr>
<td>ode</td>
<td>inline code</td>
</tr>
<tr>
<td>ol</td>
<td>ordered list (numbered list)</td>
</tr>
<tr>
<td>p</td>
<td>section symbol</td>
</tr>
<tr>
<td>pre</td>
<td>pre-formatted text</td>
</tr>
<tr>
<td>q</td>
<td>quotation</td>
</tr>
<tr>
<td>samp</td>
<td>sample(block)</td>
</tr>
<tr>
<td>span</td>
<td>span</td>
</tr>
<tr>
<td>strong</td>
<td>strong</td>
</tr>
<tr>
<td>ul</td>
<td>unordered list</td>
</tr>
<tr>
<td>var</td>
<td>text that is variable</td>
</tr>
</tbody>
</table>

**i Note**
- Header tags must be on the top level.
- If styled accordingly in your chosen theme, you can either use `<em>` for italic font and `<strong>` for bold font. Alternatively, use a span tag with a class carrying the desired styling.
- The attributes `class` and `id` can be used with the following tags:
  - `div`
  - `span`

For more information on *Formatted Text View*, see the following SAPUI5 documentation. [SAPUI5 documentation](#)

**Related Information**

Get HTML Text (`getHTMLText`) [page 491]
Set HTML Text (`setHTMLText`) [page 492]

## 31.7.8 Fragment Gallery

The *Fragment Gallery* is a basic component that is used to store portable fragment bookmarks for the user while they are working with them online to create customized dashboards. The *Fragment Gallery* along with the *Split Cell*, plays an integral role in the online composition feature. For more information on using the *Fragment Gallery*, you can refer to the chapter called *Using the Online Composition Feature*. 
### 31.7.9 Image

Using the image component, you can enhance applications by adding images.

The **Image** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
</table>
| image         | image location/name | Specifies the image that is displayed. There are several possible locations where you can store the images and icons that you want to use in your application:  
- in the application directory  
  `<userhome>\Analysis-workspace  
  \com.sap.fb.bi.zen  
  \repository  
  \MyApplication  
  \MyImage.jpg` |

---

**Related Information**

- Working with Fragment Gallery Components [page 148]
- Working with Online Composition [page 145]
- Working with Fragment Gallery Components [page 148]
- Working with Split Cell Containers [page 150]
<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hover image</td>
<td>image location/name</td>
<td>Specifies the image that is displayed when the user hovers with the mouse over the component. Type in the image name, for example, <em>MyImage.jpg</em> or use the Open Image dialog box to select the image or icon from the MIME repository if your Design Studio is connected to SAP NetWeaver.</td>
</tr>
<tr>
<td>click image</td>
<td>image location/name</td>
<td>Specifies the image that is displayed when the user clicks the image. Type in the image name, for example, <em>MyImage.jpg</em> or use the Open Image dialog box to select the image or icon from the MIME repository if your Design Studio is connected to SAP NetWeaver.</td>
</tr>
</tbody>
</table>

Note

In the MIME repository, you can select images and icons uploaded by the system administrator. For more information, see the Administrator Guide: SAP BusinessObjects Design Studio at [http://help.sap.com/boad](http://help.sap.com/boad).
### 31.7.10 Input Field

The input field enables the user to write user-specific content in the input field at runtime.

The **Input Field** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>string</td>
<td>Specifies the initial value that the application user can overwrite during runtime.</td>
</tr>
<tr>
<td>on change</td>
<td>none</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the input field by writing scripts. Click <code>CTRL + Space</code> to see the list of available methods for the application, the data source alias and the input field. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference.</td>
</tr>
</tbody>
</table>

### 31.7.11 List Box

List boxes enable the user to select items, for example, to set a filter. To do this, you need to add a script to the On Select property of the list box. The script is triggered when the user selects an item from the list box.
The **List Box** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
</table>
| items         | list of items  | Opens the *Edit items* dialog box. Here, you can insert items, change the order of items or remove them. Each item has the following properties:
|               |                | - value
|               |                |   The value property is mandatory for all items.
|               |                | - text (optional)
|               |                |   The text property is optional. If you enter a text for an item, this text is displayed. If no text is entered, the value of the item is displayed.
|               |                | If you want a specific item to be displayed by default, select the relevant item and click *Set as Default*. This item is shown as the default entry in the list box. |
| multiple selection | false, true | Allows multiple values to be selected. This property can be used to filter data sources by multiple members. |
| on select     | script         | Opens the script editor. With this property/event, you can enable user interaction with the listbox by writing scripts. The *On Select* event is triggered when the application user selects an item from the list box. In the script editor, click `CTRL` + `Space` to see the list of available methods for the application, the data source alias and the list box. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference. |

### 31.7.12 Radio Button Group

Radio button groups enable the user to select items, to set a filter for example. To do this, you need to add a script to the radio button group’s On Select property. The script is triggered when the user selects an item from the radio button group.

The **Radio Button Group** component has the following specific properties:
<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
</table>
| items         | list of items  | With the items property, you can open the Edit items dialog. Here, you can insert items, change the order of items or remove them. Each item has the following properties:  
  ● value  
    The value property is mandatory for all items.  
  ● text (optional)  
    The text property is optional. If you enter a text for an item, this text is displayed. If no text is entered, the value for the item is displayed.  
  If you want a specific item to be displayed as default, select the relevant item and click Set as Default. This item is then shown as the default entry in the radio button group. |
| columns       | numeric value for columns, default: 1 | Specifies the number of columns for the radio button group. The specified columns are filled by the system with the specified items.  
  Example: You have specified five items and two columns. The items are displayed as follows:  
  ● line 1: item 1 and item 2  
  ● line 2: item 3 and item 4  
  ● line 3: item 5  
  ➤ Remember  
  If you specify several columns, you need to enlarge the width of the component. Otherwise the columns will not be fully or correctly displayed. |
| on select     | script         | Opens the script editor. With this property/event, you can enable user interaction with the radio button group by writing scripts. The On Select event is triggered when the application user selects an item in the radio button group. In the script editor, click \( \text{CTRL} + \text{Space} \) to see the list of available methods for the application, the data source alias and the radio button group. Choose one of them or use the |
You use the **Text** component to add user-defined text to your application. By setting the relevant properties, you can change the style and size of the text.

The **Text** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Some Text</td>
<td>Here you can enter the text that you want to add to your application.</td>
</tr>
<tr>
<td>Style</td>
<td>Standard, Heading1, Heading 2, Heading 3, Heading 4</td>
<td>Specifies a predefined style.</td>
</tr>
</tbody>
</table>
| CSS Style| YourCssStyle   | Defines the CSS style that is applied for this component. Click the browse button to open the CSS Style Editor dialog box and write your CSS style into the white area, for example `color: pink;`. You can specify one of the following CSS attributes:  
- font size  
- font color  
- weight (bold)  
- style (italic)  
- text decoration (underline)  
This property provides advanced options for text formatting. You can type in the CSS style code directly. Alternatively, you can use any external CSS style editor (such as cssmate or any other) to generate code for the CSS attributes you want to use. Copy this code into the CSS style editor of the design tool. |
31.8 Container Components

31.8.1 Grid Layout

You use the grid layout component to group and order the content of your application in a grid that is not displayed at runtime.

The Grid Layout component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of rows</td>
<td>numeric value, default: 1</td>
<td>Specifies the number of rows in the grid.</td>
</tr>
<tr>
<td>number of columns</td>
<td>numeric value, default: 1</td>
<td>Specifies the number of columns in the grid.</td>
</tr>
</tbody>
</table>

31.8.2 Pagebook

The pagebook component enables the user to switch between the different views of an application. You group the components or contents of a view on a separate page. For example, you want to show different data on different pages or you want to display the data in different ways (in a chart on one page and in a crosstab on another page). The pagebook only shows one page at time.
Note

- The pages are shown in the *Outline* view.
- If you drag and drop a pagebook component in the layout editor, the system automatically creates two pages for the pagebook. If you want to add further pages to the pagebook, use the context menu of the pagebook in the *Outline* view and click > Create Child > Page.

The **Pagebook** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected page index</td>
<td>index</td>
<td>Specifies the visible page of the pagebook. The index 0 represents the first page of the pagebook, 1 represents the second page, 2 represents the third page, etc.</td>
</tr>
<tr>
<td>transition effect</td>
<td>none, slide in, fade, flip, cube</td>
<td>Specifies the transition effect when the user switches between the pages of the pagebook.</td>
</tr>
<tr>
<td>transition direction</td>
<td>horizontal, vertical</td>
<td>Specifies the transition direction when the user switches between the pages of the pagebook.</td>
</tr>
<tr>
<td>page caching</td>
<td>none, all, adjacent</td>
<td>Specifies the caching behavior for the pagebook component.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you set the property to <em>none</em>, no pages are cached. This property setting is recommended for applications with high data volume.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you set the property to <em>all</em>, every page of the pagebook is cached. This property setting is useful when the data volume is low.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you set the property to <em>adjacent</em>, the previous page and next page are cached.</td>
</tr>
<tr>
<td>show page indicator</td>
<td>true, false</td>
<td>Specifies whether a page indicator is displayed in the pagebook component. The page indicator displays how many pages the pagebook consists of and indicates which page is active. With the page indicator, you can also change between pages by clicking the indicators.</td>
</tr>
</tbody>
</table>

Note

On the iPad, you can only change to the page right or left of the selected page. It is not possible, for example, to change from page two to page.  

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Application Designer Guide: Designing Analysis Applications

User Interface Reference
The panel component has the following specific properties:

### 31.8.3 Panel

You use the panel component to group other components in your application. The panel component is a very simple container component.

The `Panel` component has the following specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable swiping</td>
<td>true, false</td>
<td>Specifies whether the application user can swipe the pages of the pagebook at runtime. If <code>Enable Swiping</code> is set to <code>true</code> and <code>Transition Effect</code> is set to <code>None</code>, the application user can swipe the pages but without any transition effect.</td>
</tr>
<tr>
<td>on select</td>
<td>none</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the pagebook by writing scripts. Click <code>CTRL + Space</code> to see the list of available methods for the application, the data source alias and the pagebook. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference.</td>
</tr>
<tr>
<td>Property</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
</tbody>
</table>
| CSS style         | CSS style      | Defines the CSS style that is applied to the component. Click the browse button to open the CSS Style Editor dialog box and write your CSS style into the white area, for example, `color: pink;`. You can specify one of the following CSS attributes:  
- font size  
- font color  
- weight (bold)  
- style (italic)  
- text decoration (underline)  
This property provides advanced options for text formatting. You can type in the CSS style code directly. Alternatively, you can use any external CSS style editor (such as cssmate) to generate code for the CSS attributes you want to use. Copy this code into the CSS style editor of the design tool. |
|                  |                | **Note**  
Restriction: Only the CSS attributes listed above are supported for both desktop browser and iPad/iPhone applications. If you use any other CSS attributes in the external CSS style editor, run the application on your target device, to check whether the text is displayed as required and as defined by the attributes. |
| on click          | none           | Opens the script editor. With this property/event, you can enable user interaction with the panel by writing scripts. Click [CTRL + Space] to see the list of available methods for the application, the data source alias and the panel. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference. |
31.8.4 Pop-up

Syntax

The popup component helps users to quickly enter information, perform configurations or make selections. In addition, popups can also be useful for displaying more specific data for a selected item displayed on the main page of the application. As the popup is a container component, you can put any other component into the popup (for example, crosstab, button or checkbox).

Restriction

- The popup can only be nested in the root layout and not within another container component.
- The popup can only be positioned relative to the root layout.

The popup component is initially invisible. It is displayed (or closed) when the user interacts in the application either with another visible component (for example, a button) or a script (for example, at startup).

The **Popup** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>modal</td>
<td>true, false</td>
<td>Specifies whether the popup is modal. If the popup is set to modal, the user can only navigate and perform interactions within the popup. Otherwise the user can also interact within the whole application.</td>
</tr>
<tr>
<td>animation</td>
<td>no animation, flip animation, pop animation, horizontal slide animation, vertical slide animation</td>
<td>Specifies the animation effect when the popup is displayed or closed.</td>
</tr>
<tr>
<td>autoclose</td>
<td>true, false</td>
<td>Specifies whether the component is automatically closed, when the user interacts outside the popup but inside the application.</td>
</tr>
</tbody>
</table>

31.8.5 Tabstrip

You use the tabstrip component to group and order the content of your application in tabs. You can also use it to enable user interaction within the application. To do this, you add a script to the tabstrip’s On select property. The script is triggered when the user selects one of the tabs on the tabstrip in the application.

The **Tabstrip** component has the following specific properties:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected tab index</td>
<td>index</td>
<td>Specifies the initially displayed tab of the tabstrip. The index 0 represents the first tab of the tabstrip, 1 represents the</td>
</tr>
<tr>
<td>Property Type</td>
<td>Property Value</td>
<td>Property Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>second tab, 2 represents the third tab etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>on select</td>
<td>script</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the tabstrip by writing scripts. The On Select event is triggered when the application user selects one tab. In the script editor, click CTRL+Space to see the list of available methods for the application, the data source alias and the tabstrip. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference.</td>
</tr>
</tbody>
</table>

31.8.6 Split Cell

You use the Split Cell container component to enable the user to place and arrange portable fragment bookmarks, dragged from the fragment gallery, in a tabular format. Thus the splitcell container can only be used with the Fragment Gallery component and the portable fragment bookmarks which can be created by using the corresponding API methods.

The Split Cell container component has the following specific properties:

i Note

When portable fragment bookmarks are inserted into a splitcell container, the components contained in them may be renamed. In general, script references will be adapted to the new names. However, in case the name is given as a string, this renaming will not take place and could result in unexpected behaviour.

- working example: PAGEBOOK_1.setSelectedPageIndex(0)
- non-working example: PAGEBOOK_1.setSelectedPageByName("PAGE_1")

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>on drop</td>
<td>script</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the splitcell by writing scripts. The On Drop event is triggered after a drop from the Fragment Gallery has been made. It is not triggered after moving or resizing cells. In the script editor, click CTRL+Space to see the</td>
</tr>
</tbody>
</table>
### Related Information

**Working with Split Cell Containers** [page 150]

### 31.9 Custom Components

#### 31.9.1 Timer

The Timer component is a custom component that can be used with the OOTB Chart component to create near Real-Time dashboards with single or multiple SAP HANA or SAP BW data sources.

**Note**

To avail of this custom component, you must install the Real-Time package. For more information on the Real-Time package, you can refer to the chapter in this guide called “Working with Real-Time Dashboards”.

---

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>list of available methods for the application, the data source alias and the splitcell. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference.</td>
</tr>
<tr>
<td>on delete</td>
<td>script</td>
<td>Opens the script editor. With this property/event, you can enable user interaction with the splitcell by writing scripts. The On Delete event is triggered when the application user deletes a cell. It is not triggered after moving or resizing cells. In the script editor, click <code>CTRL + Space</code> to see the list of available methods for the application, the data source alias and the splitcell. Choose one of them or use the statement wizard. The wizard guides you through the statement creation process. For further information, read the relevant method description in the API reference.</td>
</tr>
</tbody>
</table>
How To Configure the Timer Custom Component

1. Drag and drop a `Timer` into the editor area.
2. Assign a Design Studio script to the `On Timer` property.
3. Assign a time interval in milliseconds to the `Interval in Milliseconds` property.
4. Start and stop the `Timer` using the Timer’s `start()` and `stop()` Design Studio script commands.

Properties

Table 44:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval in Milliseconds</td>
<td>int</td>
<td>The time interval of the <code>Timer</code> (default setting: 1000)</td>
</tr>
<tr>
<td>On Timer</td>
<td>ScriptText</td>
<td>The Design Studio script that is executed periodically, each time the time interval elapses.</td>
</tr>
<tr>
<td>Show Icon in Application</td>
<td>boolean</td>
<td>If set to <code>true</code>, the <code>Timer</code> icon is visible in the analysis application. If set to <code>false</code>, the icon is hidden (default setting: <code>true</code>).</td>
</tr>
</tbody>
</table>

Design Studio Script API

- `void start()`
  Starts the `Timer`. This executes the Design Studio script of the `On Timer` property periodically, each time the time interval elapses.
- `void stop()`
  Stops the `Timer`. This stops the Design Studio script of the `On Timer` property.
- `boolean isRunning()`
  Returns `true` if the `Timer` has been started or `false` if the `Timer` has been stopped.

Related Information

Working with Real-Time Dashboards [page 194]
How to Create Real-Time Dashboards with a Single Pull-Based Data Source [page 196]
How to Create Real-Time Dashboards with Multiple Pull-Based Data Sources [page 197]
SAP BusinessObjects Design Studio is a design tool to create interactive analysis applications. To enable interactivity, you write scripts that are executed when the user performs an action in the executed application. For example, you can place the button component in the application and assign a script to the button's On Click event.

Scripts consist of one or more statements written in a JavaScript-based language that follow a specific syntax. For writing scripts the design tool offers two approaches: the statement wizard and the script editor.

All objects, methods and expression types are listed in this API reference. The methods are grouped by the object types they refer to. These object types are:

- data source alias (DataSourceAlias)
- application (Application)
- component (Component)
- single components like button, text, image (for example, Button, Text, Image)
- information objects (ApplicationInfo, DataCell, Convert)

Script language

You write scripts in the BI Action Language (BIAL). BIAL is a true subset of the JavaScript/ECMAScript standard described here: [http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf](http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf). The scripts are executed on the Analysis Application Design Service (unlike JavaScript that is executed in the Web browser). Scripts have a clear definition of the supported language constructs, objects and methods.

Scripts, statements and events

Scripts consist of statements. Each statement is typically written in a single line. The execution of scripts is triggered by user interaction with the component. This interaction is also referred to as an event and the script executed in response to this event is the event handler. Each component has its own set of one or more events that are displayed in the Properties view of the component.

→ Remember

Events on components are only triggered by user interaction and not implicitly by scripting. For example, when changing the selection in a list box by using the script `LB.setSelectedItem()`, the On Select event of the list box is not triggered. An exception to this rule is the On Result Set Changed event, which can be triggered implicitly by a script method like `setFilter()` or `reloadData()`.

SAP BusinessObjects Design Studio currently supports the following statement types:

- call statements
- conditional execution statements
• assignment statements

Syntax of call statements

Call statements execute an API method of an object and have the following format:

```plaintext
<Component>.<method>({arguments});
```

<Component> is the name of a data source alias or a component in your application, for example, DS_1 or Button_1.

<method> is an operation that is applied to the object specified on the left of the period. The available methods depend on component type. DataSourceAlias objects, for example, provide methods to filter data, and visual components provide methods to modify visibility, enablement and so on.

<arguments> is a comma-separated list of expressions. The passed expressions must match the requirements of the method.

Each statement ends with a semicolon (";").

You can use methods as arguments for other methods as long as the return type matches the argument type.

Syntax of conditional execution statements

Conditional execution statements have one of the following formats:

- first format

```plaintext
if (<condition>) {
<sequence of statements to execute when condition is met>
}
```

- second format

```plaintext
if (<condition>) {
<sequence of statements to execute when condition is met>
} else {
<sequence of statements to execute when condition is NOT met>
}
```

<condition> is a Boolean expression - one of the following:

- true or false as value literals (constants)
- a method call statement returning a Boolean value
- a comparison for equality in the form a == b or
- a comparison for inequality in the form a != b
- multiple conditions joined using && and ||, parentheses are optional
Assignment statements

Assignment statements have one of the following formats:

- `var <variable> = <expression>;`
  This format defines a script variable (see Script Variables in this chapter) and assigns the result of an expression to this script variable.
- `<variable> = <expression>;`
  This format assigns the result of an expression to a script variable, which has been already defined.

Methods and object types

Methods are operations that are applied to the object specified on the left of the period in the statement. The available methods depend on the object type. For example, DataSourceAlias objects provide methods to filter data. Visual components have methods to modify visibility, enablement and so on. SAP BusinessObjects Design Studio supports methods that depend on the following object types:

- data source alias (DataSourceAlias)
- application (Application)
- component (Component)
- individual components like button, text, image (for example, Button, Text, Image)
- information objects (ApplicationInfo, DataCell, DataSourceInfo)

Expressions

Expressions compute a result (also referred to as returned value) in a similar way to a formula. Expressions can appear as method arguments and as conditions ("if" statements or conditional execution statements). Results of an expression have a type. An expression may consist of literals (String constants, integer numbers, Arrays, JSONs), operators, method calls and parentheses.

Operators supported by BIAL:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Argument Type</th>
<th>Result Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Concatenates strings</td>
<td>String, (Integer, Float, Boolean) (Boolean and Integer will be converted to String automatically)</td>
<td>String</td>
<td>&quot;ab&quot;+&quot;cd&quot; (=&quot;abcd&quot;) &quot;ab&quot;+1 (=&quot;ab1&quot;)</td>
</tr>
<tr>
<td>+</td>
<td>Adds two integer values or floating point</td>
<td>Integer, Float</td>
<td>Integer</td>
<td>1+2 (=3)</td>
</tr>
<tr>
<td>-</td>
<td>Subtracts two integer values or floating point</td>
<td>Integer, Float</td>
<td>Integer</td>
<td>3-2 (=1)</td>
</tr>
<tr>
<td>*</td>
<td>Multiplies two integer values or floating point</td>
<td>Integer, Float</td>
<td>Integer, Float</td>
<td>3*2 (=6)</td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td>Argument Type</td>
<td>Result Type</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>/</td>
<td>Divides one integer value by the other or one floating point by the other</td>
<td>Integer, Float</td>
<td>Integer, Float</td>
<td>8/2 (=4)</td>
</tr>
</tbody>
</table>
| ==       | Checks if the two operands are equal | Any | Boolean | 1 == 1 (= true)  
"a" == "b" (=false) |
| !=       | Checks if the two operands are not equal | Any | Boolean | 1 != 2 (= true)  
"a" != "a" (=false) |
| &&       | Logical AND | Boolean | Boolean | true && false (=false)  
true && true (=true)  
if (<condition1> && <condition2>)  
{  
<statements>  
}  
Statements will be executed if both conditions are true. |
| ||       | Logical OR | Boolean | Boolean | true || false (=true)  
false || false (=false)  
if (<condition1> || <condition2>)  
{  
<statements>  
}  
Statements will be executed if at least one of the conditions is true. |
| !        | Logical NOT | Boolean | Boolean | !true (=false)  
!false (=true)  
if (! <condition>) {  
<statements>  
} |
Expression type system

Expression results can be used as method arguments. The expression type must match the argument type. In some cases, BIAL will automatically convert between types. This typically only happens for certain strings, where BIAL can check that the string value is valid (see below). The +Operator automatically converts Integer and Boolean arguments to String. In all other cases, the type system is strict and error messages are displayed if there is a mismatch.

There are four different types:

- **primitive types (String, Integer, Boolean, Float)**

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>&quot;Hello&quot;</td>
</tr>
<tr>
<td>Integer</td>
<td>123</td>
</tr>
<tr>
<td>Float</td>
<td>123</td>
</tr>
<tr>
<td>Boolean</td>
<td>true, false</td>
</tr>
<tr>
<td>String Array</td>
<td>['A', 'B']</td>
</tr>
<tr>
<td>Integer Array</td>
<td>[1, 2]</td>
</tr>
<tr>
<td>JSON</td>
<td>{&quot;key&quot;: &quot;value&quot;}</td>
</tr>
</tbody>
</table>

- **BI types (DataSourceAlias, Dimension, Measure, ...)** enable context-relevant input help for API method arguments in the script editor. These special types are sometimes written in Enum, Array, or JSON format (or even a combination of those). To view examples, refer to the API reference.

- **component types (Button, Text, Pagebook...)**

The visual component types correspond to the list of components in the **Components** view of the design tool.

**Enums**

In many cases, the input for a function can be one of a fixed set of values. This set of values is referred to as an enumeration or simply "Enum". Enums are BI types. An Enum value is written as <EnumType>.<EnumValue>, for example, "ChartType.PIE".
Script Variables

Script variables store expression results. They are useful, for example, for storing intermediate results that are used repeatedly in a script. Script variables have a name composed of the characters "A"-"Z", "a"-"z", "0"-"9", and "_". The name cannot begin with any of the digits "0"-"9". Like expressions, script variables have a type. There are local and global script variables:

- Local script variables
  Local script variables are script variables that can only be used in the script in which they were defined, and not in any other scripts. To define a local script variable, open a script with the Script Editor and add a line with the following format:
  ```javascript
  var <variable> = <expression>;
  ```
  The type of the script variable is automatically determined by the type of the expression.

- Global script variables
  Global script variables are script variables that can be used in any script of your application. To define a global script variable click the Application in the Outline view, then click in the Property sheet the item Scripting Global Script Variables. Click the button ... This opens the Edit Global Script Variable dialog box. You can insert a new global script variable and define its name, type, and default value. In addition you can specify whether the global script variable is a URL parameter. If you make it a URL parameter, then you can set the value of this global script variable by adding the global script variable and its URL parameter value to the applications URL.

  **Note**
  Script variable names used as a URL parameter must start with a capital X (X) and must not end with an underscore and a digit (_<digit>).

Methods calling the event handler

To avoid event handlers calling each other infinitely which could result in runtime crashes, events on components are only triggered by user interaction on the component and not by scripting. For example, if a user swipes the pages of a pagebook, the On Select event is raised, and the relevant event script is executed. In contrast, if the selected page index is set by script using the method setSelectedPageIndex of the pagebook, the On Select event is not raised. Another example is, if a user selects a value from a dropdown box, the On Select event is raised and the event script is executed. In contrast, if the selected item is set using the setSelectedValue method of the dropdown box, the event is not raised. However there are situations when you want to execute the relevant event script without duplicating script code. In these cases, each component with an event (On Click, On Select) has a corresponding method that allows the event handler to be called from another handler’s code. For example, the statement `BUTTON_1.onClick();` calls the On Click event handler of the button component with the name BUTTON_1. Another example is the statement `TABSTRIP_1.onSelect();` that calls the On Select event handler of the tabstrip component with the name TABSTRIP_1.
**Example**

You have created an application with a dropdown box **DROPDOWN_1** and a button **BUTTON_1**. If the user selects a value in the dropdown box, the selected value filters the dimension **MYDIM** in the data source aliases **DS_1** and **DS_2**. For this scenario, the *On Select* event script of the dropdown box looks like this:

```javascript
DS_1.setFilter("MYDIM", DROPDOWN_1.getSelectedValue());

DS_2.setFilter("MYDIM", DROPDOWN_1.getSelectedValue());
```

If the user pushes the button, a specific value in the dropdown box should be selected programmatically using the same logic defined for user interaction. In other words, the selected value filters the dimension **MYDIM** in the data source aliases **DS_1** and **DS_2**. Instead of duplicating the script code of the *On Select* event of **DROPDOWN_1**, you add the following statements in the *On Click* event handler of the button:

```javascript
DROPDOWN_1.setSelectedValue("MYDIMVALUE");

DROPDOWN_1.onSelect();
```

The benefits of calling the `onSelect` method increase as you start to work with more use cases in the application (where you want to set the selected value programmatically) and more statements in the *On Select* event handler of the dropdown box.

**Member key format**

The most frequently used API methods deal with dimension members in the context of setting filters or variable values. Often the filter or variable values to set are returned from the selection of a UI component.

In SAP Business Information Warehouse (BW) there are multiple key types; the most important are the INTERNAL and the EXTERNAL key formats. Whereas the INTERNAL key is a unique identifier for all users (user-locale independent), the EXTERNAL key can be user-locale dependent.

By default, all methods dealing with member keys as returned values or as parameters, work with the INTERNAL key format. This keeps applications language-independent to ensure, that they can work for multilingual user groups. However, you might want to choose the EXTERNAL key format due to various situations or reasons:

- The EXTERNAL key format is more compact and either the application is known to be used only by users sharing the same locale, or the EXTERNAL key format is known to be locale-independent.
- The number of complex selections is more than the single values that need to be passed as parameters for the API methods `setFilter` and/or `setVariableValue`. In this case, the SAP Business Explorer selection syntax ("INPUT_STRING") can be used, which references individual members by their EXTERNAL key.

For these special use cases, SAP BusinessObjects Design Studio provides Ext variants of methods, for example `setFilterExt`, `setVariableValueExt`.
32.1 Application (Application)

32.1.1 Alert (alert)

Opens a message dialog on the computer where the design tool is installed.

Note

This method is intended for debugging a locally executed application. It works if the application is executed locally, but will be ignored if the application is executed directly on the supported platforms.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>String</td>
<td>Message text</td>
</tr>
</tbody>
</table>

Returned value

None

32.1.2 Export (export)

Enables the user to specify and export a crosstab from the SAP BusinessObjects Design Studio into Microsoft Excel.

The following file formats are supported:

- CSV
- Microsoft Excel 2004 (.xls)
- Microsoft Excel 2007 (.xlsx)

The following describes the method of naming used for the exported data and its metadata.

Table 45: Exported Data

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excel file name</td>
<td>Composed of the application name and creation time.</td>
</tr>
<tr>
<td>Excel sheet name</td>
<td>Named according to the component name.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Excel sheet metadata</td>
<td>Export time</td>
</tr>
<tr>
<td>Data Source</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Variables pre-set on the data</td>
<td></td>
</tr>
<tr>
<td>source</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Static Filters pre-set to the</td>
<td></td>
</tr>
<tr>
<td>data source</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Filters set on the data source at</td>
<td></td>
</tr>
<tr>
<td>runtime by the user</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Parameters

Table 46:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export Type</td>
<td>String</td>
<td>• To view all available export types, use <code>Ctrl+Space</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The supported export types are CSV, Microsoft Excel 2004 (.xls) and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Microsoft Excel 2007 (.xlsx)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➤ Tip</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Always select the export types automatically displayed</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>List of Crosstabs</td>
<td>Array</td>
<td>The list and sequence of crosstabs that can be exported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Optional parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● The sequence of elements in the array determines the sequence of sheets in the Excel file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong> [cross_tab1, cross_tab1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong> [cross_tab7, cross_tab3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong> All crosstabs will be exported if this parameter is not entered.</td>
</tr>
<tr>
<td>Export Settings</td>
<td>String</td>
<td>● Optional parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Use Ctrl+space to view all the available options.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● CSV format is currently supported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong> CSV_encoding_UTF8</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong> CSV_SEPARATOR_COMMA</td>
</tr>
</tbody>
</table>

You must read the following SAP Notes when exporting to Microsoft Excel.

<table>
<thead>
<tr>
<th>SAP Note Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1917890</td>
<td>Listing the Microsoft Excel export feature as the last statement in the scripting.</td>
</tr>
<tr>
<td>1917891</td>
<td>Increasing jvm memory for large data export.</td>
</tr>
<tr>
<td>1917892</td>
<td>Using Microsoft Excel 2007 if a large column export is required.</td>
</tr>
<tr>
<td>1917943</td>
<td>Limiting the number of formatting elements in each crosstab.</td>
</tr>
<tr>
<td>1917944</td>
<td>Third parameter export setting currently not corresponding with the first parameter export type.</td>
</tr>
<tr>
<td>1917946</td>
<td>Microsoft Excel 2007 export feature not supported on BI platform 4.0.</td>
</tr>
</tbody>
</table>
Related Information

Export to Microsoft Excel [page 226]

32.1.3 Create Error Message (createErrorMessage)

Creates an error message that is displayed in the message view.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>String</td>
<td>Message text</td>
</tr>
</tbody>
</table>

Returned value

None

32.1.4 Create Info Message (createInfoMessage)

Creates an info message that is displayed in the message view.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>String</td>
<td>Message text</td>
</tr>
</tbody>
</table>

Returned value

None
32.1.5  Create Warning Message (createWarningMessage)

Creates a warning message that is displayed in the message view.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>String</td>
<td>Message text</td>
</tr>
</tbody>
</table>

Returned value

None

32.1.6  Do Background Processing (doBackgroundProcessing)

Triggers the script written in the On Background Processing property of the application. This script is executed in the background when the UI is rendered.

Parameters

None

Returned value

None

32.1.7  Get Info (getInfo)

Returns analysis application information, for example, the current date or the name of the analysis application.
Parameters

None

Returned value

Returns object providing information about the application.

32.1.8 Get Resource String (getResourceString)

Returns a resource string using its id.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>ID of the resource string</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the resource string.

32.1.9 Get Tick Count (getTickCount)

Returns a point in time in milliseconds. This method is useful for performance measurements.

Parameters

None
Returned value

Integer

Example

In the following example, the time for executing method setFilter is displayed:

```javascript
var startTime = APPLICATION.getTickCount();
DS_MYDATA.setFilter(...);
var endTime = APPLICATION.getTickCount();
APPLICATION.alert("Elapsed time: " + (endTime - startTime) + " ms");
```

32.1.10 Get User Agent (getUserAgent)

Returns the user agent of the browser executing the scripting function.

Parameters

None

Returned value

String. It contains the user agent.

32.1.11 Load Data Sources (loadDataSources)

Loads the specified data sources.

This method is equivalent to loadDataSource on a data source but allows loading several data sources at once.

This method respects the Processing Group property of data sources: Data sources can be loaded in parallel if contained in such a group. Therefore it might be faster to call this method instead of calling loadDataSource on each data source.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataSourceAliases</td>
<td>Array of DataSourceAlias</td>
<td>Data sources to load</td>
</tr>
</tbody>
</table>

Returned value

None

32.1.12 Log (log)

Creates a message in the design tool's Error Log view for script error analysis.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>String</td>
<td>Message text</td>
</tr>
</tbody>
</table>

Returned value

None

32.1.13 Open New Window (openNewWindow)

Opens a new browser window with the specified URL.

If you want to open an external domain, you have to specify the protocol explicitly, for example http://www.sap.com. If you do not specify the protocol, the window will open the URL on the same domain as the invoking Analysis Application.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newUrl</td>
<td>String</td>
<td>URL</td>
</tr>
</tbody>
</table>

⚠️ Restriction
This method does not open in a new window with SAP BusinessObjects Mobile 4.1 (navigation continues in the same window).

Returned value

None

Example

**Using an external domain**

In this example, the Web page sap.com is opened in a new browser window:

```javascript
APPLICATION.openNewWindow("http://www.sap.com");
```

32.1.14 Open Prompt Dialog Box (openPromptDialog)

Opens prompt dialog box.

ℹ️ Note

Be aware that the `openPromptDialog` method must be the last method in the script that uses this method.

Parameters

Table 47:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td>Integer</td>
<td>Width of the prompt dialog box in pixels</td>
</tr>
<tr>
<td>height</td>
<td>Integer</td>
<td>Height of the prompt dialog box in pixels</td>
</tr>
</tbody>
</table>
Returned value

None

Example

In the following example, the prompt dialog box is opened:

```java
APPLICATION.openPromptDialog(400, 500);
```

32.1.15 Set Variable Value (setVariableValue)

Sets data source variable values in the internal key format and executes the data source query again. Variable values are set in the application for all data source aliases containing the variable.

Variable values are set in the application for all data source aliases containing the variable. With this method you can set only single members or hierarchy nodes in the internal key format as variable values.

With this method you can set variable values of input-enabled characteristic value variables (only single values) or hierarchy node variables. If you want to set variable values for the following cases, use `setVariableValueExt` instead:

- formula variables
- text variables
- hierarchy variables
- variables representing precalculated values sets (buckets)
- multiple single values
- intervals
- variables representing a selection option
- using external format

For hierarchy nodes, prefix the internal non-compounded presentation of the hierarchy node with `HIERARCHY_NODE/NodeType/`, for example `HIERARCHY_NODE/0HIER_NODE/ROOT`.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable to set</td>
</tr>
<tr>
<td>value</td>
<td>VariableValue</td>
<td>Variable value to set in internal key format</td>
</tr>
</tbody>
</table>

Restriction

It is not possible to use the crosstab and chart method `getSelectedMember` for setting variables for compounded characteristics.
Returned value

None

Restrictions

It is not possible to use the Crosstab component and the Chart component method getSelectedMember for setting variables for compounded dimensions.

Example

In the following example a fixed string value is set to a variable:

```
APPLICATION.setVariableValue("0VAR", "4711");
```

In the following example the value is set to a variable returned from a component selection:

```
APPLICATION.setVariableValue("0VAR", DROPDOWN_1.getSelectedValue());
```

32.1.16 Print (print)

Prints an application.

Parameters

None

Returned value

None

32.1.17 Search Data Sources (searchDataSources)

Searches for all data sources matching a given pattern.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataSourceConnection</td>
<td>DataSourceConnection</td>
<td>Name of data source connection</td>
</tr>
<tr>
<td>searchFor</td>
<td>SearchExpression</td>
<td>Pattern to search for</td>
</tr>
</tbody>
</table>

Returned value

Array of DataSourceDescriptor. It contains information about the matching data sources. None

32.1.18 Set Variable Value Ext (setVariableValueExt)

Sets data source variable values in external key format, then executes the data source query again.

Variable values are set in the application for all data source aliases containing the variable. Use this method if other selections than single members or hierarchy nodes need to be set as variable value.

Note

External key values might be dependent on the user’s regional settings. If the application is intended to be used by a multi-lingual group of users, it is therefore recommended to use the `setVariableValue` method when possible.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable to set</td>
</tr>
<tr>
<td>value</td>
<td>InputString</td>
<td>Variable value to set in external key format</td>
</tr>
</tbody>
</table>

Syntax for entering values

You can specify the input string as variable value. This syntax allows you to specify more complex selections (such as intervals and excluding selections). In contrast to the method `setVariableValue`, the members are specified in the external format (for example, 001.2012).
<table>
<thead>
<tr>
<th><strong>Description</strong></th>
<th><strong>Example</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>equal to</td>
<td>15</td>
</tr>
<tr>
<td>multiple</td>
<td>15:18:20</td>
</tr>
<tr>
<td>exclude value</td>
<td>!22</td>
</tr>
<tr>
<td>value range</td>
<td>1-5</td>
</tr>
<tr>
<td>exclude value range</td>
<td>!6-9</td>
</tr>
<tr>
<td>greater than</td>
<td>&gt;8</td>
</tr>
<tr>
<td>exclude values greater than &lt;value limit&gt;</td>
<td>!&gt;8</td>
</tr>
<tr>
<td>greater than or equal to</td>
<td>&gt;=8</td>
</tr>
<tr>
<td>less than</td>
<td>&lt;12</td>
</tr>
<tr>
<td>exclude values less than &lt;value limit&gt;</td>
<td>!&lt;12</td>
</tr>
<tr>
<td>less than or equal to</td>
<td>&lt;=12</td>
</tr>
<tr>
<td>exclude values less than or equal to &lt;value limit&gt;</td>
<td>!&lt;=12</td>
</tr>
<tr>
<td>complex combination</td>
<td>15:10-15:20-25:1-5;&gt;8; etc.</td>
</tr>
<tr>
<td>dimension hierarchy node</td>
<td>+&lt;Dimension Attribute&gt;(&lt;Technical Name of Dimension&gt;), for example, +ELEMENT1(WBS_ELEMENT)</td>
</tr>
<tr>
<td>text hierarchy node</td>
<td>++Technical Name of Hierarchy Node&gt;(0HIER_ NODE), for example, +EUROPE(0HIER_NODE)</td>
</tr>
</tbody>
</table>

### Returned value

None

#### Example

In the following example a fixed string value is set to a variable:

```java
APPLICATION.setVariableValue("0VAR", "4711");
```

In the following example the value returned from a component selection is set to a variable:

```java
APPLICATION.setVariableValue("0VAR", DROPDOWN_1.getSelectedValue());
```

### 32.2 ApplicationInfo

An object providing information about the application.
**dateNow**

Current date in the user's locale.

**dateNowInternalFormat**

Current date in the format "YYYYMMDD".

**name**

Name of the application.

### 32.3 Array

The object Array has the following sub types:

- DataSourceConnectionArray
- DataSourceDescriptorArray
- DimensionArray
- HierarchyArray
- MemberArray
- MeasureFilterIdArray
- StringArray
- VariableArray
- ComponentArray
- CrosstabArray
- DataSourceAliasArray
- DataSourceDescriptorArray
- DimensionArray
- FilterArray
- FragmentBookmarkArray
- HierarchyArray
- KeyValueArray
- ListValueArray
- MemberArray
- SingleMemberFilter
- StringArray
- BooleanArray
32.3.1 Accessing Array Elements

You can access elements of an array in the `forEach` method of the array. There is no simple way to retrieve an individual array element from outside a `forEach` loop using an array index. However, you can use this workaround: in a `forEach` loop, store the array element you want to index in a local variable.

**Example**

In the following example, the third array element of an array of members is retrieved and its external key value is displayed:

```javascript
var i = 2;
var e = ";
var array = DS_1.getMembers("OD_CUSTOMER", 100);
array.forEach(function(element, index) {
    if (i == index) {
        e = element.externalKey;
    }
});
APPLICATION.alert("Element with index " + i + " is \"" + e + \"\".");
```

Related Information

For Each (forEach) [page 344]

32.3.2 For Each (forEach)

Iterates through the elements of an array.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callback</td>
<td>Function</td>
<td>The function that is executed with each iteration.</td>
</tr>
</tbody>
</table>
Returned value

None

Example

In the following example, the final value of sum is 6:

```javascript
var sum = 0; var array = [1, 2, 3]; array.forEach(function(element, index) {
  sum = sum + element;
});
```

32.3.3 Length (length)

The array's number of elements.

32.4 AttributeMember

Provides access to the representations of an attribute member.

externalKey

The member's representation as external key.

externalNoncompoundedKey

The member's representation as external non-compounded key.

internalKey

The member's representation as internal key.
**internalNoncompoundedKey**

The member’s representation as internal non-compounded key.

**text**

The member’s representation as text.

### 32.5 Bookmark

#### 32.5.1 Assign To Folder (assignToFolder)

Assigns a standard bookmark to a folder. Both BookmarkFolderId and BookmarkId must be provided to identify the target folder directory and target bookmark.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folderId</td>
<td>BookmarkFolderId</td>
<td>Bookmark folder id</td>
</tr>
<tr>
<td>bookmarkId</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

**Returned Value**

None

ℹ️ Note

Supported platforms - BI platform and local.

#### 32.5.2 Bookmark With Title Exists (bookmarkWithTitleExists)

Returns true if a bookmark with specified title exists and false if not.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>String</td>
<td>Bookmark title</td>
</tr>
</tbody>
</table>

Returned value

Boolean. True if a bookmark with the specified title exists and false if not.

Note

Supported platforms - BI platform, SAP NetWeaver, Local

32.5.3 Delete All Bookmarks (deleteAllBookmarks)

Deletes all standard bookmarks in the analysis application.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
</tbody>
</table>

Note

This parameter is not supported on SAP NetWeaver.

Returned value

None

Note

Supported platforms - BI platform, SAP NetWeaver, Local
32.5.4 Delete Bookmark (deleteBookmark)

Deletes a standard bookmark with specified Id.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>BookmarkId id</td>
</tr>
</tbody>
</table>

Returned Value

None

Note

Supported platforms - BI platform, SAP NetWeaver, Local

32.5.5 Get All Bookmarks (getAllBookmarks)

Gets all standard bookmarks for the current version of an analysis application.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
</tbody>
</table>

Note

This parameter is not supported on SAP NetWeaver.

Returned value

Array of BookmarkInfo. It contains a list of standard bookmarks for the current version of an analysis application.
32.5.6  Get All Bookmarks By Folder (getAllBookmarksByFolder)

Gets all standard bookmarks for the current version of an analysis application by folder.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkFolderId</td>
<td>Bookmark folder id</td>
</tr>
</tbody>
</table>

Returned Value

Array of BookmarkInfo. It contains standard bookmarks located within a specified folder for the current version of an analysis application. If the folder identifier is invalid or there are no bookmarks found, an empty array is returned.

Note

Supported platforms - BI platform and local.

32.5.7  Get Bookmark Folders (getBookmarkFolders)

Gets all bookmark folders which have been configured on the platform.

Parameters

None
Returned Value

Array of BookmarkFolderInfo. It contains a list of folders which have been configured on the platform.

Note

Supported platforms - BI platform and local.

32.5.8 Get Bookmark Info (getBookmarkInfo)

Returns a standard bookmark specified by Bookmark Id.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

BookmarkInfo. BookmarkInfo

Note

Supported platforms - BI platform, SAP NetWeaver, Local

32.5.9 Get Bookmark Url (getBookmarkUrl)

Gets the bookmark Url for a given standard bookmark.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>
## Returned value

String. Bookmark Url.

### Note

Supported platforms - BI platform, SAP NetWeaver, Local

## 32.5.10 Load Bookmark (loadBookmark)

Loads the state of an analysis application from a standard bookmark.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

### Returned value

None

### Note

Supported platforms - BI platform, SAP NetWeaver, Local

## 32.5.11 Save Bookmark (saveBookmark)

Creates a new bookmark with an optional title for the current state of an analysis application. If the title is not specified, then a default title will be generated automatically.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) title</td>
<td>String</td>
<td>Bookmark title</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
</tbody>
</table>

**Note**

This parameter is not supported on SAP NetWeaver.

---

**Returned value**

BookmarkId. It contains the saved bookmark.

**Note**

Supported platforms - BI platform, SAP NetWeaver and local mode.

---

### 32.5.12 Share Bookmark (shareBookmark)

Shares an analysis application URL from a standard bookmark.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>String</td>
<td>Bookmark URL</td>
</tr>
</tbody>
</table>

**Returned value**

None

**Note**

Supported platforms - BI platform, SAP NetWeaver, Local
32.6 Fragment Bookmark

32.6.1 Assign To Folder (assignToFolder)

Assigns a fragment bookmark to a folder. Both BookmarkFolderId and BookmarkId must be provided to identify the target folder directory and target bookmark.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folderId</td>
<td>BookmarkFolderId</td>
<td>Bookmark folder id</td>
</tr>
<tr>
<td>bookmarkId</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

None

Note

Supported platforms - BI platform and local.

32.6.2 Delete All Bookmarks (deleteAllBookmarks)

Deletes all fragment bookmarks of the analysis application.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
</tbody>
</table>

Note

This parameter is not supported on SAP NetWeaver.
Returned Value

None

Note
Supported platforms - BI platform, SAP NetWeaver, Local

32.6.3 Delete Bookmark (deleteBookmark)

Deletes a fragment bookmark with specified Id.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>BookmarkId id</td>
</tr>
</tbody>
</table>

Returned Value

None

Note
Supported platforms - BI platform, SAP NetWeaver, Local

32.6.4 Get All Bookmark Infos (getAllBookmarkInfos)

Gets a list of all fragment bookmarks for an analysis application.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td>This parameter is not supported on SAP NetWeaver.</td>
</tr>
</tbody>
</table>

Returned Value

Array of FragmentBookmarkInfo. FragmentBookmarkArray

- **Note**
  
  Supported platforms - BI platform, SAP NetWeaver, Local

### 32.6.5 Get All Bookmarks By Folder (getAllBookmarksByFolder)

 Gets all fragment bookmarks for the current version of an analysis application by folder.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkFolderId</td>
<td>Bookmark folder id</td>
</tr>
</tbody>
</table>

Returned Value

Array of FragmentBookmarkInfo. It contains fragment bookmarks located within a specified folder for the current version of an analysis application. If the folder identifier is invalid or there are no bookmarks found, an empty array is returned.

- **Note**
  
  Supported platforms - BI platform and local.
32.6.6 Get Bookmark Info (getBookmarkInfo)

Returns a fragment bookmark specified by Bookmark Id.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

FragmentBookmarkInfo. FragmentBookmarkInfo

Note

Supported platforms - BI platform, SAP NetWeaver, Local

32.6.7 Get Bookmark Url (getBookmarkUrl)

Gets the bookmark Url for a given fragment bookmark.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

String. Bookmark Url

Note

Supported platforms - BI platform, SAP NetWeaver, Local
32.6.8  Load Bookmark (loadBookmark)

Loads the state of an analysis application from a fragment bookmark.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

None

Note

Supported platforms - BI platform, SAP NetWeaver, Local

32.6.9  Save Bookmark (saveBookmark)

Creates a new fragment bookmark using a container component as a selector. Selection is recursive to any depth in the specified container. If the title is not specified then a unique default title will be generated for the current application.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>selection</td>
<td>Container</td>
<td>Container selection</td>
</tr>
<tr>
<td>(optional) title</td>
<td>String</td>
<td>Bookmark title</td>
</tr>
<tr>
<td>(optional) description</td>
<td>String</td>
<td>Bookmark description</td>
</tr>
<tr>
<td>(optional) toOverwrite</td>
<td>BookmarkId</td>
<td>Bookmark to overwrite</td>
</tr>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note</td>
</tr>
</tbody>
</table>

This parameter is not supported on SAP NetWeaver.

**Returned Value**

FragmentBookmarkInfo.FragmentBookmarkInfo

**Note**

Supported platforms - BI platform, SAP NetWeaver, Local

**32.6.10 Share Bookmark (shareBookmark)**

Shares an analysis application URL from a fragment bookmark.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>String</td>
<td>Bookmark URL</td>
</tr>
</tbody>
</table>

**Returned Value**

None

**Note**

Supported platforms - BI platform, SAP NetWeaver, Local
32.7 Portable Fragment Bookmark

32.7.1 Assign To Folder (assignToFolder)

Assigns a portable fragment bookmark to a folder. Both BookmarkFolderId and BookmarkId must be provided to identify the target folder directory and target bookmark.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folderId</td>
<td>BookmarkFolderId</td>
<td>Bookmark folder id</td>
</tr>
<tr>
<td>bookmarkId</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

None

Note

- Supported platforms - BI platform and local.

32.7.2 Delete All Bookmarks (deleteAllBookmarks)

Deletes all portable fragment bookmarks of all analysis applications specified by group identifier.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupIdentifier</td>
<td>String</td>
<td>Group Identifier</td>
</tr>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
</tbody>
</table>
### Delete Bookmark (deleteBookmark)

Deletes a portable fragment bookmark with specified Id.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>BookmarkId id</td>
</tr>
</tbody>
</table>

#### Returned value

None

**Note**

Supported platforms - BI platform, SAP NetWeaver, Local

### Get All Bookmark Infos (getAllBookmarkInfos)

Gets a list of all portable fragment bookmarks for a group identifier.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupIdentifier</td>
<td>String</td>
<td>Group Identifier</td>
</tr>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
</tbody>
</table>

Note
This parameter is not supported on SAP NetWeaver.

Returned Value
Array of PortableFragmentBookmarkInfo. PortableFragmentBookmarkInfoArray.

Note
Supported platforms - BI platform, SAP NetWeaver, Local

32.7.5 Get All Bookmark Infos For Application (getAllBookmarkInfosForApplication)

Gets a list of all portable fragment bookmarks for the current analysis application.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupIdentifier</td>
<td>String</td>
<td>Group Identifier</td>
</tr>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
</tbody>
</table>

Note
This parameter is not supported on SAP NetWeaver.
Returned Value

Array of PortableFragmentBookmarkInfo. PortableFragmentBookmarkInfoArray.

**Note**

Supported platforms - BI platform, SAP NetWeaver, Local

### 32.7.6 Get All Bookmarks By Folder (getAllBookmarksByFolder)

Gets all portable fragment bookmarks for the current version of an analysis application by folder.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folderId</td>
<td>BookmarkFolderId</td>
<td>Bookmark folder id</td>
</tr>
<tr>
<td>groupIdentifier</td>
<td>String</td>
<td>Group Identifier</td>
</tr>
</tbody>
</table>

**Returned Value**

Array of PortableFragmentBookmarkInfo. It contains portable fragment bookmarks located within a specified folder for the current version of an analysis application. If the folder identifier is invalid or there are no bookmarks found, an empty array is returned.

**Note**

Supported platforms - BI platform and local.

### 32.7.7 Get BookmarkInfo (getBookmarkInfo)

Returns a portable fragment bookmark specified by Bookmark Id.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

PortableFragmentBookmarkInfo. PortableFragmentBookmarkInfo

**Note**

Supported platforms - BI platform, SAP NetWeaver, Local

32.7.8  Get Bookmark Url (getBookmarkUrl)

Gets the bookmark Url for a given portable fragment bookmark.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

String. Bookmark Url

**Note**

Supported platforms - BI platform, SAP NetWeaver, Local

32.7.9  Load Bookmark (loadBookmark)

Loads the state of an analysis application from a portable fragment bookmark.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>BookmarkId</td>
<td>Bookmark id</td>
</tr>
</tbody>
</table>

Returned Value

None

**Note**

Supported platforms - BI platform, SAP NetWeaver, Local

### 32.7.10 Save Bookmark (saveBookmark)

Creates a new portable fragment bookmark using the specified groupIdentifier. The generation of the bookmark uses a container component as a selector. Selection is recursive to any depth in the specified container. Title is optional, and if not present, will be generated automatically. Portable fragment bookmarks are group id and user specific and can be accessed across applications.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupIdentifier</td>
<td>String</td>
<td>Group Identifier</td>
</tr>
<tr>
<td>selection</td>
<td>Container</td>
<td>Container selection</td>
</tr>
<tr>
<td>(optional) title</td>
<td>String</td>
<td>Bookmark title</td>
</tr>
<tr>
<td>(optional) description</td>
<td>String</td>
<td>Bookmark description</td>
</tr>
<tr>
<td>(optional) image</td>
<td>String</td>
<td>Bookmark image</td>
</tr>
<tr>
<td>(optional) toOverwrite</td>
<td>BookmarkId</td>
<td>Bookmark to overwrite</td>
</tr>
<tr>
<td>(optional) appIdentifier</td>
<td>String</td>
<td>Application identifier</td>
</tr>
</tbody>
</table>

**Note**

This parameter is not supported on SAP NetWeaver.
Returned Value

PortableFragmentBookmarkInfo.

Note

Supported platforms - BI platform, SAP NetWeaver, Local

32.7.11 Share Bookmark (shareBookmark)

Shares an analysis application Url from a portable fragment bookmark.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>String</td>
<td>Bookmark Url</td>
</tr>
</tbody>
</table>

Returned Value

None

Note

Supported platforms - BI platform, SAP NetWeaver, Local

32.8 Backend Connection

32.9 Button

32.9.1 Get Text (getText)

Returns the text displayed on the button.
Parameters

None.

Returned value

String. It contains the text displayed on the button.

32.9.2 Get Tooltip (getTooltip)

Returns the tooltip of the component.

Parameters

None.

Returned value

String. It contains the tooltip text.

32.9.3 Is Enabled (isEnabled)

Returns whether component is enabled.

Parameters

None

Returned value type

Boolean. True if component is enabled or false if component is disabled.
32.9.4 On Click (onClick)

Calls the script that is executed when the user clicks the button.

Parameters

None

Returned value

None

32.9.5 Set Enabled (setEnabled)

Enables or disables component.

Disabled components do not allow user interaction.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies whether to enable or disable the component</td>
</tr>
</tbody>
</table>

Returned value

None

32.9.6 Set Text (setText)

Sets the text displayed on the button.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>String</td>
<td>Button text</td>
</tr>
</tbody>
</table>

Returned value

None

32.9.7 Set Tooltip (setTooltip)

Sets the tooltip of the component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tooltip</td>
<td>String</td>
<td>Tooltip text</td>
</tr>
</tbody>
</table>

Returned value

None.

32.10 Chart

32.10.1 Get Chart Type (getChartType)

Returns name of chart type.

Parameters

None
Returned value

String. It contains the name of chart type.

32.10.2 Clear Selection (clearSelection)

Clears the selection from the chart.

Parameters

None

Returned Value

None

32.10.3 Set Axis Scaling (setAxisScaling)

Sets a minimum and maximum value for the desired axis number.

Parameter

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chartAxisScaling</td>
<td>ChartAxisScaling</td>
<td>Property axis number</td>
</tr>
<tr>
<td>minValue</td>
<td>Float</td>
<td>Property min value</td>
</tr>
<tr>
<td>maxValue</td>
<td>Float</td>
<td>Property max value</td>
</tr>
</tbody>
</table>

Returned Value

None
32.10.4 Remove Axis Scaling (removeAxisScaling)

Disables axis scaling for the desired axis number.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chartAxisScaling</td>
<td>ChartAxisScaling</td>
<td>Property axis number</td>
</tr>
</tbody>
</table>

Returned Value

None

32.10.5 Get Axis Scaling Max (getAxisScalingMax)

Gets the maximum value for an axis.

Parameter

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chartAxisScaling</td>
<td>ChartAxisScaling</td>
<td>Property axis number</td>
</tr>
</tbody>
</table>

Returned Value

Float

32.10.6 Get Axis Scaling Min (getAxisScalingMin)

Gets the minimum value for an axis.
### Parameter

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chartAxisScaling</td>
<td>ChartAxisScaling</td>
<td>Property axis number</td>
</tr>
</tbody>
</table>

### Returned Value

Float

### 32.10.7 Get Legend Position (getLegendPosition)

Returns the position of the chart legend.

#### Parameters

None

#### Returned Value

LegendPosition. It contains the position of the chart legend.

### 32.10.8 Get Selected Member (getSelectedMember)

Returns the member that is contained in the current selection of the chart.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the selected members</td>
</tr>
</tbody>
</table>
Returned value

Member. It is the member that has been selected in the chart.

32.10.9 Get Selected Members (getSelectedMembers)

Returns the members that are contained in the current selections of the chart.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the selected members</td>
</tr>
</tbody>
</table>

Returned value

Array of Member. It contains the members that have been selected in the chart.

32.10.10 Get Style (getStyle)

Returns name of chart style.

Parameters

None.

Returned value

String. It contains the name of chart style.
32.10.11 Is Visible (isVisible)

Returns whether component is visible.

Parameters

None

Returned value

Boolean. True if component is shown or false if component is hidden.

32.10.12 On Select (onSelect)

Calls the script that is executed when the user selects an area in the chart.

Parameters

None

Returned value

None

32.10.13 Set Chart Type (setChartType)

Sets chart type.
**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chartType</td>
<td>ChartType</td>
<td>Chart type to set.</td>
</tr>
</tbody>
</table>

**Returned value**

None

**32.10.14  Set Legend Position (setLegendPosition)**

Sets the position of the chart legend.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>LegendPosition</td>
<td>Position of the legend.</td>
</tr>
</tbody>
</table>

**Returned Value**

None

**32.10.15  Set Style**

Sets chart style.

**Note**

As of release 1.1 this method has no effect in the application. You can still use this method but no error warning is displayed in the design studio.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>style</td>
<td>ChartStyle</td>
<td>Chart style to set</td>
</tr>
</tbody>
</table>

Returned value

None

32.10.16 Set Visible (setVisible)

Shows or hides component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isVisible</td>
<td>Boolean</td>
<td>Specifies whether to show or hide component</td>
</tr>
</tbody>
</table>

Returned value

None

32.10.17 Show Data Labels (showDataLabels)

Shows or hides the data labels.
Parameter

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isVisible</td>
<td>Boolean</td>
<td>Specifies whether to show or hide the data labels</td>
</tr>
</tbody>
</table>

Returned Value

None

32.10.18  Show Scaling Factors (showScalingFactors)

Shows the scaling factors on axis and tooltip.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isVisible</td>
<td>Boolean</td>
<td>Specifies whether to show or hide scaling factors on axis and tooltip.</td>
</tr>
</tbody>
</table>

Returned value

None

32.10.19  Show Totals (showTotals)

Shows or hides totals and subtotals.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isVisible</td>
<td>Boolean</td>
<td>Specifies whether to show or hide totals and subtotals</td>
</tr>
</tbody>
</table>

 Returned value

None

32.10.20 Swap Axes (swapAxes)

Swaps the result set axes of the assigned data source.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>swapAxes</td>
<td>Boolean</td>
<td>Specifies whether to swap axes</td>
</tr>
</tbody>
</table>

 Returned value

None

32.10.21 Set Data Selection (setDataSelection)

Sets the subset of the data of the source to be shown by the chart.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>selection</td>
<td>ResultSetSelection</td>
<td>Data selection to show by the chart</td>
</tr>
</tbody>
</table>
## Returned Value

None

### 32.11 Checkbox

### 32.11.1 Get Text (getText)

Returns checkbox text.

#### Parameters

None

#### Returned value

String. It contains the text of the checkbox.

### 32.11.2 Get Tooltip (getTooltip)

Returns the tooltip of the component.

#### Parameters

None.
Returned value

String. It contains the tooltip text.

32.11.3 Is Checked (isChecked)

Returns whether checkbox is selected. Returns true, if checkbox is selected, or false, if checkbox is unselected.

Parameters

None

Returned value

Boolean. True if checkbox is selected or false if checkbox is unselected.

32.11.4 Is Enabled (isEnabled)

Returns whether component is enabled.

Parameters

None

Returned value type

Boolean. True if component is enabled or false if component is disabled.

32.11.5 Is Visible (isVisible)

Returns whether component is visible.
Parameters

None

Returned value

Boolean. True if component is shown or false if component is hidden.

32.11.6 On Click (onClick)

Calls the script that is executed when the user clicks the checkbox.

Parameters

None

Returned value

None

32.11.7 Set Checked (setChecked)

Selects or unselects checkbox.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isChecked</td>
<td>Boolean</td>
<td>Specifies whether to select or unselect checkbox</td>
</tr>
</tbody>
</table>
32.11.8 Set Enabled (setEnabled)

Enables or disables component. Disabled components do not allow user interaction.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies whether to enable or disable the component</td>
</tr>
</tbody>
</table>

Returned value

None

32.11.9 Set Text (setText)

Sets checkbox text.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>String</td>
<td>Checkbox text</td>
</tr>
</tbody>
</table>

Returned value

None
32.11.10 Set Tooltip (setTooltip)

Sets the tooltip of the component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tooltip</td>
<td>String</td>
<td>Tooltip text</td>
</tr>
</tbody>
</table>

Returned value

None.

32.11.11 Set Visible (setVisible)

Shows or hides component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isVisible</td>
<td>Boolean</td>
<td>Specifies whether to show or hide component</td>
</tr>
</tbody>
</table>

Returned value

None
32.12 Checkbox Group

32.12.1 Add Item (addItem)

Adds an item to the component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String</td>
<td>Unique key (or technical value) of the item. The operation is ignored if an item with this key already exists in the component.</td>
</tr>
<tr>
<td>text</td>
<td>String</td>
<td>Display text of the item.</td>
</tr>
<tr>
<td>(optional) index</td>
<td>String</td>
<td>Index where to insert the item (default: 0). If the index is -1 or greater than the number of items then the item is added as the last item.</td>
</tr>
</tbody>
</table>

Returned value

None

32.12.2 Get Selected Texts (getSelectedTexts)

i Note

This method is only valid for the List Box component.

Returns the selected item's text. Can be used if multiple selection is enabled.

Parameters

None
Returned value

StringArray. It contains the selected texts as an array.

32.12.3 Get Selected Values (getSelectedValues)

**Note**

This method is only valid for the List Box component.

Returns the selected item's value. Can be used if multiple selection is enabled.

**Parameters**

None

Returned value

StringArray. It contains the selected values as an array.

32.12.4 Is Enabled (isEnabled)

Returns whether component is enabled.

**Parameters**

None

**Returned value type**

Boolean. True if component is enabled or false if component is disabled.
32.12.5 Remove All Items (removeAllItems)

Removes all items from the component.

Parameters

None

Returned value

None

32.12.6 Remove Item (removeItem)

Removes an item from the component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String</td>
<td>Unique key (or technical value) of the item. The operation is ignored if no such key exists in the component.</td>
</tr>
</tbody>
</table>

Returned value

None

32.12.7 Set Enabled (setEnabled)

Enables or disables component.

Disabled components do not allow user interaction.
### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies whether to enable or disable the component</td>
</tr>
</tbody>
</table>

### Returned value

None

### 32.12.8 Set Items (setItems)

Assigns a new list of items to the component.

The old content of the component will be deleted.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>ValueTextList</td>
<td>List of value-text pairs</td>
</tr>
</tbody>
</table>

#### Returned value

None

#### Example

In the following example, method `setItems` sets the first 100 members from the `Customer` dimension to the checkbox group / list box. Each list item's value is represented by the customer's key, each list item's text is represented by the customer's text and key. No item is added to represent all members.

```java
LISTBOX_1.setItems(DS_1.getMemberList("OD_CUSTOMER", MemberPresentation.KEY, MemberDisplay.TEXT_KEY, 100));
```

or

```java
CHECKBOXGROUP_1.setItems(DS_1.getMemberList("OD_CUSTOMER", MemberPresentation.KEY, MemberDisplay.TEXT_KEY, 100));
```
32.12.9 Set Selected Values (setSelectedValues)

Selects the items with the specified values.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>values</td>
<td>listValue Array</td>
<td>Values of the items to select</td>
</tr>
</tbody>
</table>

Returned value

None

32.12.10 Sort (sort)

Sorts values in alphabetical order.

Parameters

Table 48:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) isSortAscending</td>
<td>Boolean</td>
<td>Specifies whether values are sorted in ascending or descending alphabetical order</td>
</tr>
</tbody>
</table>

Returned value

None
32.13 Constants

32.13.1 Axis

A set of constants to specify an axis.

COLUMNS

Columns axis

FREE

Free axis

ROWS

Rows axis

32.13.2 ClickArea

A set of constants to specify the area of the click.

DATA

Data cell has been clicked

DIMENSION

Dimension cell has been clicked
MEMBER

Member cell has been clicked

32.13.3 DataSourceBrowseType

A set of constants to specify the type of a data source browser tree.

INFO_AREA_OR_FOLDER

Info Areas (BW) or Folders (HANA)

ROLES

SAP BW Roles

WORKSPACES

SAP BW Workspaces

32.13.4 DataSourceConnectionType

A set of constants to specify the type of a data source connection.

BW

SAP BW backend

HANA

SAP HANA backend
32.13.5 DataSourceExportType

A set of constants to specify the export type.

LUMIRA_lums

Export to Lumira (lums)

32.13.6 DataSourceType

A set of constants to specify the type of a data source.

FOLDER

Folder (not a data source)

INFOPROVIDER

Info Provider

QUERY

BEx query

VIEW

BEx query view

32.13.7 Layout

A set of constants to specify layout information.
AUTO

The value should be auto

32.13.8 LegendPosition

A set of constants to specify the position of the chart legend.

BOTTOM

Position the legend at the bottom of the chart

LEFT

Position the legend to the left of the chart

RIGHT

Position the legend to the right of the chart

TOP

Position the legend at the top of the chart

32.13.9 MemberDisplay

A set of constants to specify how members are displayed.

KEY

Member is displayed as external key
**KEY_TEXT**

Member is displayed as external key and text

**NO_DISPLAY**

Member is not displayed

**TEXT**

Member is displayed as text

**TEXT_KEY**

Member is displayed as text and external key

### 32.13.10 MemberPresentation

A set of constants to specify which aspect of a member to present.

**EXTERNAL_KEY**

Member is represented by its external key

**EXTERNAL_NONCOMPOUNDED_KEY**

Member is represented by its external non-compounded key

**INTERNAL_KEY**

Member is represented by its internal key
INTERNAL_NONCOMPOUNDED_KEY

Member is represented by its internal non-compounded key

TEXT

Member is represented by its text

32.13.11 NegativeNumberDisplay

A set of constants to specify how a negative number is displayed.

LEADING_MINUS

Negative values are displayed as \(-X\)

PARENTHESES

Negative values are displayed as \((X)\)

TRAILING_MINUS

Negative values are displayed as \(X-\)

32.13.12 Scaling

A set of constants to specify a scaling factor applied to numbers.

FACTOR_1

Scaling factor 1 (no scaling)
FACTOR_10
Scaling factor 10

FACTOR_100
Scaling factor 100

FACTOR_1000
Scaling factor 1000

FACTOR_10000
Scaling factor 10000

FACTOR_100000
Scaling factor 100000

FACTOR_1000000
Scaling factor 1000000

TEXT
Scaling factor 1000000

FACTOR_10000000
Scaling factor 10000000

FACTOR_100000000
Scaling factor 100000000
**FACTOR_1000000000**

Scaling factor 1000000000

**FACTOR_DEFAULT**

Default scaling factor

### 32.13.13 TotalsDisplay

A set of constants to specify whether totals are displayed in the result set.

**HIDE**

Totals are always hidden

**HIDE_IF_ONLY_ONE_MEMBER**

Totals are hidden if only one member is available

**SHOW**

Totals are always shown

### 32.13.14 TotalsPosition

A set of constants to specify where totals are displayed in the result set.

**AFTER**

Display totals after members
BEFORE
Display totals before members

DEFAULT
Use default display

32.13.15 UnitsAndScalingFactorsDisplay
A set of constants to specify how units and scaling factors are displayed in the Crosstab.

BOTH_IN_HEADER
Display both units and scaling factors in the header

NO_DISPLAY
Do not display units and scaling factors

UNITS_IN_DATA CELLS
Display units in the data cells

32.13.16 ZeroDisplay
A set of constants to specify how the number zero is displayed.

CUSTOM
Zero values are displayed as a custom text
DEFAULT

Default value

EMPTY_CELL

Zero values are displayed as empty strings

WITHOUT_CURRENCY_UNIT

Zero values are displayed without currency and unit information

32.14 Component

Base class for all visual components, for example, button and crosstab components. The methods in this class are available for all components.

32.14.1 Get Bottom Margin (getBottomMargin)

Returns the bottom margin of the component.

Parameters

None

Returned value

Integer. It contains the bottom margin of component in pixels. It is -1 if the current value is auto.
32.14.2 Get CSS Class (getCSSClass)

Returns the Cascading Style Sheet (CSS) class name of the component.

Parameters

None

Returned value

String. It contains the CSS class name of the component.

32.14.3 Get Height (getHeight)

Returns the height of the component.

Parameters

None

Returned value

Integer. It contains the height of component in pixels. It is -1 if the current value is auto.

32.14.4 Get Left Margin (getLeftMargin)

Returns the left margin of the component.

Parameters

None
Returned value

Integer. It contains the left margin of component in pixels. It is -1 if the current value is *auto*.

32.14.5 Get Right Margin (getRightMargin)

Returns the right margin of the component.

Parameters

None

Returned value

Integer. It contains the right margin of component in pixels. It is -1 if the current value is *auto*.

32.14.6 Get Top Margin (getTopMargin)

Returns the top margin of the component.

Parameters

None

Returned value

Integer. It contains the top margin of the component in pixels. It is -1 if the current value is *auto*.

32.14.7 Get Width (getWidth)

Returns the width of the component.
Parameters

None

Returned value

Integer. It contains the width of component in pixels. It is -1 if the current value is auto.

32.14.8 Hide Loading State (hideLoadingState)

Hides the loading indicator on the component.

Parameters

None

Returned value

None

32.14.9 Is Visible (isVisible)

Returns whether component is visible.

Parameters

None

Returned value

Boolean. True if component is shown or false if component is hidden.
32.14.10 Set Bottom Margin (setBottomMargin)

Sets the bottom margin of the component.

**Note**

This function has no effect, if the current value is *auto*.

**Parameters**

Table 49:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottomMargin</td>
<td>Integer</td>
<td>Bottom margin of component in pixels</td>
</tr>
</tbody>
</table>

**Returned value**

None

32.14.11 Set CSS Class (setCSSClass)

Sets the Cascading Style Sheet (CSS) class name of the component.

**Parameters**

Table 50:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>className</td>
<td>String</td>
<td>Name of the CSS class</td>
</tr>
</tbody>
</table>

**Returned value**

None
### 32.14.12 Set Height (setHeight)

Sets the height of the component.

**Note**

This function has no effect if the current value is `auto`.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>height</td>
<td>Integer</td>
<td>Height of the component in pixels</td>
</tr>
</tbody>
</table>

#### Returned value

None

### 32.14.13 Set Left Margin (setLeftMargin)

Sets left margin of the component.

**Note**

This function has no effect if the current value is `auto`.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>leftMargin</td>
<td>Integer</td>
<td>Left margin of component in pixels</td>
</tr>
</tbody>
</table>

#### Returned value

None
32.14.14 Set Right Margin (setRightMargin)

Sets right margin of the component.

**Note**
This function has no effect if the current value is `auto`.

**Parameters**

Table 53:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rightMargin</td>
<td>Integer</td>
<td>Right margin of component in pixels</td>
</tr>
</tbody>
</table>

**Returned value**

None

32.14.15 Set Top Margin (setTopMargin)

Sets top margin of the component.

**Note**
This function has no effect if the current value is `auto`.

**Parameters**

Table 54:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>topMargin</td>
<td>Integer</td>
<td>Top margin of component in pixels</td>
</tr>
</tbody>
</table>

**Returned value**

None
### 32.14.16 Set Visible (setVisible)

Shows or hides component.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isVisible</td>
<td>Boolean</td>
<td>Specifies whether to show or hide component</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.14.17 Set Width (setWidth)

Sets width of the component.

**Note**

This function has no effect if the current value is `auto`.

**Parameters**

Table 55:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td>Integer</td>
<td>Width of component in pixels</td>
</tr>
</tbody>
</table>

**Returned value**

None
32.14.18  Show Loading State (showLoadingState)

Shows a loading indicator on the component.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) text</td>
<td>String</td>
<td>Text to show in the loading indicator</td>
</tr>
</tbody>
</table>

**Returned value**

None

32.15  Connection

32.15.1  Get Children (getChildren)

Returns the child elements of data source descriptor.

The specific kind of child elements depends on the type of the data source descriptor. For example, a data source descriptor of type `DataSourceType.FOLDER` can contain anything, while a data source descriptor of type `DataSourceType.INFOProvider` usually contains queries, which can contain query views.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent</td>
<td>DataSourceDescriptor</td>
<td>Data source descriptor containing child elements</td>
</tr>
</tbody>
</table>
Returned value

Array of DataSourceDescriptor. It contains the child elements of a data source descriptor as an array of data source descriptors.

32.15.2 Get Connections (getConnections)

Returns all available connections for the currently running platform.

Parameters

None

Returned value

Array of DataSourceConnection. It contains an array of data source connections of all available connections.

32.15.3 Get Root Folders (getRootFolders)

Returns the folders of the first hierarchy level of the data source tree.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>browse type</td>
<td>DataSourceBrowseType</td>
<td>Browse Type</td>
</tr>
</tbody>
</table>

Returned value

Array of DataSourceDescriptor. It contains an array of data source descriptors that comprise the folders of the first hierarchy level of the data source tree. Each data source descriptor is of type DataSourceType.FOLDER. The child elements of a folder can be examined with method `getChildren`.
32.15.4 Get Selected Data Source (getSelectedDataSource)

Gets the selected data source of the data source browser.

**Note**

To ensure dialog processing is completed, call this method in the script of the On Data Source Browser Confirmed event property only.

**Parameters**

None

**Returned value**

DataSourceDescriptor. It contains the data source descriptor of the selected data source.

**Example**

In the following example, a data source that was selected in the data source browser is assigned to a data source:

```javascript
var ds = CONNECTION_1.getSelectedDataSource();
DS_1.assignDataSource(ds.connection, ds.type, ds.name, true);
```

32.15.5 Get System (getSystem)

Returns the backend system of the connection.

**Parameters**

None

**Returned value**

DataSourceConnection. It contains a data source connection of the backend system.
32.15.6 Search Data Sources (searchDataSources)

Searches the backend system for data sources that match the search pattern.

**Note**
Replaces method `APPLICATION.searchDataSources`.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchPattern</td>
<td>String</td>
<td>String pattern to search for</td>
</tr>
</tbody>
</table>

**Returned value**

Array of `DataSourceDescriptor`. It contains an array of data source descriptors that match the search pattern.

32.15.7 Set System (setSystem)

Sets the backend system for the connection.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td><code>DataSourceConnection</code></td>
<td>Backend system. Use either a <code>DataSourceConnection</code> object or a system name.</td>
</tr>
</tbody>
</table>

**Returned value**

None
32.15.8  Show Data Source Browser (showDataSourceBrowser)

Displays the data source browser for the backend system.

The backend system is configured in the Backend Connection component.

**Note**

This method opens the data source browser, but the script that called this method does not wait for the data source browser to be closed. See also method `getSelectedDataSource`.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) config.Json</td>
<td>String</td>
<td>Configuration JSON</td>
</tr>
</tbody>
</table>

### Returned value

None

32.16 Context Menu (ContextMenu)

32.16.1  Get Click Area (getClickArea)

Returns the click area of what has been clicked.

### Parameters

None

### Returned value

ClickArea. It contains the click area of what has been clicked.
32.16.2 Get Context (getContext)

Returns the data context of what has been clicked.

Parameters

None

Returned value

ResultSetSelectionByString. It contains the data context of what has been clicked.

32.17 Convert

A global object providing functions to convert data from one data type to another.

32.17.1 Float to String (floatToString)

Converts a floating-point number to a string by applying a formatting pattern.

The conversion uses the default locale (US English).

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>Float</td>
<td>Floating-point number</td>
</tr>
<tr>
<td>(optional) formatString</td>
<td>String</td>
<td>Formatting pattern, for example &quot;###.###.##0.00&quot;</td>
</tr>
</tbody>
</table>
Returned value

String. It contains a string representation of the given floating-point number.

Example

In the following example, the float number with the value 12345.678 is converted to the string "12.345.68 EUR":

```
Convert.floatToString(12345.678, "###,###,##0.00 EUR");
```

Note

Keep the following points in mind:

- values are rounded to two places after the separator
- the hash characters (number signs) in the formatting pattern are filled according to the length of the number

Example

In the following example, the floating-point with the value 0.6 is converted to the string "$ 0.60":

```
Convert.floatToString(0.6, "$ ###,###,##0.00");
```

32.17.2 Float to String Using Locale (floatToStringUsingLocale)

Converts a floating-point number to a string using the locale of the current user.

Parameters

Table 57:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>Float</td>
<td>Floating-point number</td>
</tr>
<tr>
<td>(optional) numberOfDecimals</td>
<td>Integer</td>
<td>Number of decimals used for rounding and formatting. If no number is specified, the default setting is used.</td>
</tr>
</tbody>
</table>

Returned value

String. It contains a string representation of the given floating-point number.
Example

In the following example, the floating-point number with the value 1234.56 is converted to the string "1.234.56" (the current user uses the US English locale):

```java
Convert.floatToStringUsingLocale(1234.56);
```

In the following example, the floating-point number with the value 1234.56 is converted to the string "1.234.56" (the current user uses the German locale):

```java
Convert.floatToStringUsingLocale(1234.56);
```

### 32.17.3 Index Of (indexOf)

Returns the index of the first occurrence of the specified substring within a string. Optionally, you can provide an index from where to start the search.

#### Parameters

| Table 58: |
|---|---|
| **Name** | **Type** | **Description** |
| string | String | String in which to search |
| searchFor | SearchExpression | Substring to search for |
| (optional) startIndex | Integer | Index from where to start searching (default:0) |

#### Returned value

Integer. It contains the index of the first occurrence of the specified substring within a string. The first character of that string has index 0. The returned value is -1 if the substring was not found.

Example

In the following example, the returned index is 6:

```java
Convert.indexOf("Hello world!", "world");
```

In the following example, the returned index is -1:

```java
Convert.indexOf("Hello world!", "sailor");
```
32.17.4 Replace All (replaceAll)

Replaces each substring within a string that matches the given search expression with the given replacement.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>String</td>
<td>String in which to search</td>
</tr>
<tr>
<td>searchFor</td>
<td>SearchExpression</td>
<td>Substring to search for</td>
</tr>
<tr>
<td>replaceWith</td>
<td>String</td>
<td>String to be replaced for each match</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the given string with the applied replacements.

Example

In the following example, in the string "We stayed three days and three nights, the substrings three are replaced with four. The returned string is We stayed four days and four nights."

```
Convert.replaceAll("We stayed three days and three nights","three", "four");
```

In the following example, the returned index is -1:

```
Convert.indexOf("Hello world!", "sailor");
```

32.17.5 String Length (stringLength)

Returns the length (number of characters) of the specified string.
### Parameters

Table 60:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>String</td>
<td>String to calculate the length of</td>
</tr>
</tbody>
</table>

### Returned value

Integer. It contains the number of characters in the specified string.

### 32.17.6 String to Float (stringToFloat)

Converts a string to a floating-point number. The conversion uses the default locale (US English).

### Parameters

Table 61:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>String</td>
<td>String representing a floating-point number</td>
</tr>
</tbody>
</table>

### Returned value

Float. It contains the specified string converted to a floating-point number. If the conversion is not possible, NaN is returned.

**Example**

In the following example, the string "1234.56" is converted to a floating-point number with the value 1234.56:

```java
Convert.stringToFloat("1234.56");
```
### 32.17.7 String to Float Using Locale (stringToFloatUsingLocale)

Converts a string to a floating-point number using the locale of the current user.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>floatLiteral</td>
<td>String</td>
<td>String representing a floating-point number in the locale of the current user</td>
</tr>
</tbody>
</table>

#### Returned value

Float. It contains the specified string converted to a floating-point number. If the conversion is not possible, NaN is returned.

#### Example

In the following example, the string "1,234.56" is converted to a floating-point number with the value 1234.56 (the current user is using the US English locale):

```java
Convert.stringToFloatUsingLocale("1,234.56");
```

In the following example, the string "1.234,56" is converted to a floating-point number with the value 1234.56 (the current user is using the German locale):

```java
Convert.stringToFloatUsingLocale("1.234,56");
```

### 32.17.8 String to Int (stringToInt)

Converts a string to an integer number.
### Parameters

Table 63:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>intLiteral</td>
<td>String</td>
<td>String representing an integer number</td>
</tr>
</tbody>
</table>

### Returned value

Integer. It contains the specified string converted to an integer number. If the conversion is not possible, NaN is returned.

**Example**

In the following example, the string "1234" is converted to an integer number with the value 1234:

```java
Convert.stringToInt("1234");
```

---

### 32.17.9 Substring (subString)

Returns a new string that is a substring of the specified string.

The substring is composed of the characters in the specified string between the specified startIndex and specified endIndex - 1. If endIndex is omitted, then the substring includes the characters up to the last character of the specified string.

### Parameters

Table 64:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>String</td>
<td>String from which a substring is extracted</td>
</tr>
<tr>
<td>startIndex</td>
<td>Integer</td>
<td>Start index of the substring, inclusive</td>
</tr>
<tr>
<td>(optional) endIndex</td>
<td>Integer</td>
<td>End index of the substring, exclusive</td>
</tr>
</tbody>
</table>

### Returned value

String. It contains the substring of the specified string.
32.17.10 URL Encode (urlEncode)

Encodes a string into x-www-form-urlencoded format. This is useful if the string is used as a parameter of an URL.

Parameters

Table 65:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>String</td>
<td>String to encode</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the x-www-form-urlencoded encoded string.

Example

In the following example, the returned string is Hello+world%21:

```javascript
Convert.urlEncode("Hello world!");
```

32.18 Crosstab

32.18.1 Get Selected Member (getSelectedMember)

Returns the member that is contained in the current selection of the crosstab.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the selected member.</td>
</tr>
</tbody>
</table>
Return value

Member. It is the member that has been selected in the crosstab.

Example

The following script statement sets all selected products in CROSSTAB_1 as a filter to the same dimension in another data source alias DS_2:

```javascript
DS_2.setFilter("0PRODUCT", CROSSTAB_1.getSelectedMembers("Product"));
```

32.18.2 Get Selected Members (getSelectedMembers)

Returns the members that are contained in the current selection of the crosstab.

Parameters

Table 66:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension</td>
</tr>
</tbody>
</table>

Returned value

Array of Member. It contains an array of members that have been selected in the crosstab.

32.18.3 Get Selection (getSelection)

Returns the selection.

Parameters

None
Returned value

ResultSetSelectionByString. It contains the selection

32.18.4 Is Column Resizing Enabled (isColumnResizingEnabled)

Returns whether column resizing is enabled.

Parameters

None

Returned value

Boolean

32.18.5 Is Conditional Formatting Visible (isConditionalFormattingVisible)

Returns whether the crosstab shows conditional formatting or not

Parameters

None

Returned value

Boolean
32.18.6 Is Hierarchy Navigation Enabled (isHierarchyNavigationEnabled)

Returns whether hierarchy navigation is enabled

Parameters

None

Returned value

Boolean

32.18.7 Is Sorting Enabled (isSortingEnabled)

Returns whether columns sorting is enabled.

Parameters

None

Returned value

Boolean

32.18.8 On Select (onSelect)

Calls the script that is executed when the user selects a row or column in the crosstab.
32.18.9 Remove Selection (removeSelection)

Removes the selection.

Parameters
None

Returned value
None

32.18.10 Reset All Column Widths (resetAllColumnWidths)

Resets all previously set column widths. All columns are rendered with automatic width calculation.

Parameters
None

Returned value
None
32.18.11 Reset Column Width (resetColumnWidth)

 Resets any previously set column width. If there is still a default width over a width for a specific column that is removed, the default setting will be applied. Calling this method again will then remove the default width for that specific column, thus rendering the column with auto size again.

 **Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>columnIndex</td>
<td>Integer</td>
<td>The zero-based index of the column</td>
</tr>
</tbody>
</table>

**Returned value**

None

32.18.12 Set Column Resizing Enabled (setColumnResizingEnabled)

 Specifies if columns can be resized in the crosstab

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies if columns can be resized in the crosstab.</td>
</tr>
</tbody>
</table>

**Returned value**

None
32.18.13  Set Column Width (setColumnWidth)

Sets the width for the specified column.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>columnIndex</td>
<td>Integer</td>
<td>The zero-based index of the column</td>
</tr>
<tr>
<td>width</td>
<td>Integer</td>
<td>The width in pixels</td>
</tr>
</tbody>
</table>

Returned value

None

Example

In the following example, the width of column 2 (zero-based) is set to 200 pixels.

```
setColumnWidth(2, 200);
```

32.18.14  SetConditionalFormattingVisible

(setConditionalFormattingVisible)

Sets if conditional formatting shall be displayed in the crosstab or not

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isVisible</td>
<td>Boolean</td>
<td>Specifies if conditional formatting shall be displayed in the crosstab or not.</td>
</tr>
</tbody>
</table>
Returned value
None

32.18.15 Set Default Column Width (setDefaultColumnWidth)

Sets the width for all columns that do not have a column-specific setting. This default width can be overwritten for each column by calling the setColumnWidth function.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td>Integer</td>
<td>The width in pixels</td>
</tr>
</tbody>
</table>

Returned value
None

Example

In the following example, the width of all columns is set to 50 pixels, except for column 2 that has a width of 200 pixels.

```javascript
setColumnWidth(2, 200);
setDefaultColumnWidth(50);
```

32.18.16 Set Hierarchy Navigation Enabled (setHierarchyNavigationEnabled)

Specifies if hierarchies can be expanded/collapsed in the crosstab
### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies if hierarchies can be expanded/collapsed in the crosstab.</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.18.17 Set Sorting Enabled (setSortingEnabled)

Specifies if columns can be sorted in the crosstab.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.18.18 Set Units and Scaling Factors Display (setUnitsAndScalingFactorsDisplay)

Specifies how units and scaling factors are displayed.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unitsAndScalingFactorsDisplay</td>
<td>unitsAndScalingFactorsDisplay</td>
<td>Display mode for units and scaling factors</td>
</tr>
</tbody>
</table>

Returned value

None

32.19 DataBoundComponent

32.19.1 Get Data Source (getDataSource)

Returns the assigned data source.

Parameters

None

Returned value

DataSourceAlias. It contains a data source alias.

32.19.2 Set Data Source (setDataSource)

Assigns a data source.
Parameters

Table 67:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataSourceAlias</td>
<td>DataSourceAlias</td>
<td>Data source alias</td>
</tr>
</tbody>
</table>

Returned value

None.

32.20 Data Cell (dataCell)

An object providing information about a data cell of a result set.

formattedValue

Formatted value of this cell. The scaling factor has been already applied to this value. This value also reflects the user locale and number settings of the BEx Query Designer.

scalingFactor

Scaling factor in powers of ten that has been applied to the value of this cell.

unitOfMeasure

Unit of measure of the value of this cell, for example, a currency.

value

Raw data value of this cell. The scaling factor has been already applied to this value.
32.21 Data Source Alias (DataSourceAlias)

32.21.1 Activate Hierarchy (activateHierarchy)

Activates the currently inactive hierarchy. If no hierarchy is inactive, no action will be performed.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension whose hierarchy is activated</td>
</tr>
</tbody>
</table>

**Returned value**

None

32.21.2 Assign Data Source (assignDataSource)

Assigns a data source.

This method can be used in combination with the data source alias property `Load in Script` to load data sources on demand.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataSourceConnection</td>
<td>DataSourceConnection</td>
<td>Connection alias</td>
</tr>
<tr>
<td>dataSourceType</td>
<td>DataSourceType</td>
<td>Type of data source</td>
</tr>
<tr>
<td>dataSourceName</td>
<td>DataSourceName</td>
<td>Name of data source</td>
</tr>
<tr>
<td>(optional) isLoadNow</td>
<td>Boolean</td>
<td>Specifies whether data source is loaded after assignment.</td>
</tr>
</tbody>
</table>
Returned value

None

Example

In the following example a new data source is assigned and the data is loaded immediately:

```javascript
DS_1.assignDataSource("Q99","QUERY","AM_SALES");
```

Example

In the following example a new data source is assigned and loading the data is deferred (see also method `loadDataSource`):

```javascript
DS_1.assignDataSource("Q99","QUERY","AM_SALES",false);
```

This prompts all components which are linked to it to signal, that the data was not loaded yet. The data source is "greyed out" as if no data source has been assigned.

### 32.21.3 Assign Hierarchy (assignHierarchy)

Assigns hierarchy to dimension.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be displayed as a hierarchy</td>
</tr>
<tr>
<td>hierarchy</td>
<td>Hierarchy</td>
<td>Hierarchy to be assigned</td>
</tr>
</tbody>
</table>

Returned value

None

### 32.21.4 Clear All Filters (clearAllFilters)

Removes filters for all dimensions.
**Parameters**

None

**Returned value**

None

### 32.21.5 Clear Filter (clearFilter)

Removes the filter for a dimension.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias with filter to be removed.</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.21.6 Collapse Node (collapseNode)

Collapses the specified node. The node has to be visible at the time of execution or no action will be performed.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the hierarchy whose node is collapsed</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>value</td>
<td>HierarchyNode</td>
<td>Node which is collapsed</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.21.7 Configure Input Readiness (configureInputReadiness)

Configures the input readiness (if supported) of a data source used for planning.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isInputReady</td>
<td>Boolean</td>
<td>If true then the data source is set to input ready if it has been loaded and no query locks are in place. If false then the data source is set to display mode.</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.21.8 Copy Filters (copyFilters)

Copies filter values of common dimensions from a data source.
32.21.9 Deactivate Hierarchy (deactivateHierarchy)

Deactivates the currently active hierarchy. If no hierarchy is active, no action will be performed.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataSourceAlias</td>
<td>DataSource Alias</td>
<td>Data source alias to copy filters from</td>
</tr>
<tr>
<td>isCopyMeasuresFilter (optional)</td>
<td>Boolean</td>
<td>Specifies whether the filter values for measures are copied as well (default: false).</td>
</tr>
</tbody>
</table>

 Returned value

None

32.21.10 Expand Node (expandNode)

Expands the specified node up to a defined level. The node has to be visible at the time of execution or no action will be performed.
### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the hierarchy whose node is expanded</td>
</tr>
<tr>
<td>value</td>
<td>HierarchyNode</td>
<td>Node that is expanded</td>
</tr>
<tr>
<td>(optional) levels</td>
<td>Integer</td>
<td>Number of levels by which the node is expanded</td>
</tr>
</tbody>
</table>

**Note**

Use the following syntax to specify a node: `HIER_NODE/NodeType/Node`

- for hierarchy nodes in BW systems
  
  You can find the relevant hierarchy node type in the hierarchy maintenance in the BW system. It is either `O_HIER_NODE` for text nodes or the technical name of the external InfoObject in the hierarchy. The node will be the internal fully compounded member key; for example, K42013001 for the fiscal year period 1 in 2013 (fiscal year variant K4). If the value consists of a `/`, remember to escape the `/` with a backslash (`\`).

- for hierarchy nodes in SAP HANA systems

  The node type for parent child hierarchies is the alias name of the child attribute. The node type for level-based hierarchies is the alias name of the corresponding level. If the value consists of a `/`, remember to escape the `/` with a backslash (`\`).

### Returned value

None

### 32.21.11 Export (export)

Exports the resultSet of the data source.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>exportType</td>
<td>DataSourceExportType</td>
<td>Type of export format</td>
</tr>
</tbody>
</table>
Returned Value

None

32.21.12 Get Assigned Hierarchy (getAssignedHierarchy)

Returns the assigned hierarchy.

**Note**
Before calling this method, check with method `isHierarchyAssigned()` if the hierarchy is assigned.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension if of the hierarchy</td>
</tr>
</tbody>
</table>

Returned value

Hierarchy. It contains the assigned hierarchy.

32.21.13 Get Conditional Formats (getConditionalFormats)

Returns the names of the conditional formats.

Parameters

None

Returned Value

Array of `ConditionalFormatId`. It contains the names of the conditional formats.
32.21.14 Get Conditional Format Name (getConditionalFormatName)

Returns the name of a conditional format.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ConditionalFormatId</td>
<td>id of the conditional format</td>
</tr>
</tbody>
</table>

Returned Value

String. It contains the name of the conditional format.

32.21.15 Get Conditional Format Value (getConditionalFormatValue)

Returns the conditional format value applied to a data cell at runtime

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure corresponding to returned value</td>
</tr>
<tr>
<td>selection</td>
<td>MultiDimFilter</td>
<td>Combination of dimension members in internal key format that specifies the data cell selection</td>
</tr>
</tbody>
</table>

Returned value

Integer. A value of 0 indicates that no conditional formatting is applied. Values of 1 to 9 indicate the priority level of the highest applied rule (see Conditional Formatting documentation).
32.21.16 Get Conditional Format Value Ext (getConditionalFormatValueExt)

Returns the conditional format value applied to a data cell at runtime using external member keys.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure corresponding to returned value</td>
</tr>
<tr>
<td>selection</td>
<td>MultiDimFilterExt</td>
<td>Combination of dimension members in external key format that specifies the data cell selection</td>
</tr>
</tbody>
</table>

Returned value

Integer. A value of 0 indicates that no conditional formatting is applied. Values of 1 â€“ 9 indicate the priority level of the highest applied rule (see Conditional Formatting documentation).

32.21.17 Get Data (getData)

Returns a single data cell from the result set.

Note

- This method specifies members in the internal format.
- For dimensions that are not specified, their aggregates will be used, provided that the corresponding aggregates exist in the result set.
- The requested data must be part of the retrieved result set. This is especially important when requesting hierarchy nodes. The hierarchy must be expanded in advance to show the required hierarchy node.
Parameters

Table 68:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure corresponding to returned value</td>
</tr>
<tr>
<td>selection</td>
<td>MultiDimFilter</td>
<td>Combination of dimension members in internal key format that specifies the data cell selection</td>
</tr>
</tbody>
</table>

Returned value

DataCell - an object providing information about a data cell of a result set.

Example

In the following example, the method returns the data cell containing the sales revenue for product P4711 of the first quarter of 2012:

```java
DS_1.getData("Sales",{"0MATERIAL":"P4711", "0CALYEAR":"2012", "0QUARTER":"01.2012"});
```

In the following example, the method returns the data cell containing the aggregate sales revenue for product P4711 in 2012:

```java
DS_1.getData("Sales",{"0MATERIAL":"P4711", "0CALYEAR":"2012"});
```

The value for dimension 0QUARTER was not specified. Therefore the aggregate value is returned, provided that it exists in the result set. You can only omit dimensions from right to left, but not in-between. For example, you can omit the value for dimension 0CALYEAR in the first example.

Related Information

Data Cell (dataCell) [page 427]

32.21.18 Get Data as String (getDataAsString)

Returns a single data cell value from the query result set.
Format of parameter selection

- To specify dimensions, use the JSON-format: You specify the dimension name and the value, separated by a colon, for example: {"DIM1": "Member1", "DIM2": "Member2"}.
- To specify hierarchy nodes, use the internal fully compounded format of the hierarchy node in the form `HIERARCHY_NODE/Node type/`, for example `HIERARCHY_NODE/OHIER_NODE/ROOT`.
- If you want to specify the members with the external format, use the method `getDataAsStringExt` instead.

Note

- For dimensions that are not specified, the result will be used - provided that the corresponding result line exists in the result set.
- The data needs to be part of the retrieved result set. This is especially important when requesting hierarchy nodes. The hierarchy needs to be expanded in advance to show the required hierarchy node.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure corresponding to returned value</td>
</tr>
<tr>
<td>selection</td>
<td>MultiDimFilter</td>
<td>Combination of dimension members in internal key format that specifies the data cell selection</td>
</tr>
</tbody>
</table>

Returned value type

String. It is formatted based on the user's regional settings, scaling factors and unit of measurement.

Example

In the following example the method returns the sales revenue for product P4711 of the first quarter of 2012:

```java
DS_1.getDataAsString("Sales",{"OMATERIAL":"P4711", "OCALYEAR":"2012", "OQUARTER":"01.2012"});
```

In the following example the method returns the aggregate sales revenue for product P4711 in 2012:

```java
DS_1.getDataAsString("Sales",{"OMATERIAL":"P4711", "OCALYEAR":"2012"});
```

The value for dimension OQUARTER was not specified, thus the aggregate value is returned, provided it exists in the result set. When retrieving values, make sure that the dimension values specify an existing value of the result set. For example, when you omit the value for dimension OCALYEAR (thus indicating an aggregate value for this dimension) in the first example then no value is returned, because there is no cell that matches the specified dimension values.
### 32.21.19 Get Data as String Ext (getDataAsStringExt)

Returns a single data cell value from the query result set using external member keys. The returned value is a string that is formatted based on the user’s regional settings, scaling factors and unit of measurement.

To specify dimensions, use the JSON-format: You specify the dimension name and the value, separated by a colon, for example `{"DIM1": "Member1", "DIM2": "Member2"}`.

#### Note
- For dimensions that are not specified, their aggregates will be used provided that the corresponding aggregates exist in the result set.
- The requested data needs to be part of the retrieved result set. This is especially important when requesting hierarchy nodes. The hierarchy needs to be expanded in advance to show the required hierarchy node.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure corresponding to returned value</td>
</tr>
<tr>
<td>selection</td>
<td>MultiDimFilterExt</td>
<td>Combination of dimension members in external key format that specifies the data cell selection</td>
</tr>
</tbody>
</table>

#### Returned value

String. It is formatted based on the user’s regional settings, scaling factors and unit of measurement.

#### Example

In the following example the method returns the sales revenue for product P4711 of the first quarter of 2012:

```
DS_1.getDataAsStringExt("Sales", {"OMATERIAL":"P4711", "OCALYEAR":"2012", "OQUARTER":"01.2012"});
```

In the following example the method returns the aggregate sales revenue for product P4711 in 2012:

```
DS_1.getDataAsStringExt("Sales", {"OMATERIAL":"P4711", "OCALYEAR":"2012"});
```

The value for dimension OQUARTER was not specified, thus the aggregate value is returned, provided it exists in the result set. When retrieving values, make sure that the dimension values specify an existing value of the result set. For example, when you omit the value for dimension OCALYEAR (thus indicating an aggregate value for this dimension) in the first example then no value is returned, because there is no cell that matches the specified dimension values.
32.21.20 Get Decimal Places (getDecimalPlaces)

Returns the number of decimal places displayed for a measure.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure for which the number of decimal places is returned</td>
</tr>
</tbody>
</table>

Returned value

Integer. It contains the number of decimal places of the measure.

32.21.21 Get Dimension Text (getDimensionText)

Returns the localized text of a dimension.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension with the text to be returned</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the localized text of the dimension.

32.21.22 Get Dimensions (getDimensions)

Returns an array of dimensions of an axis.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) axis</td>
<td>Axis</td>
<td>Axis for which to get the dimensions</td>
</tr>
</tbody>
</table>

Returned value

DimensionArray. It contains the dimensions of the axis. If no axis is specified, returns all dimensions.

32.21.23  Get Filter Ext (getFilterExt)

Returns the filter value of a data source dimension in external key format.

Note

Use this method to pass the filter value to other methods.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the filter value in external key format.

32.21.24  Get Filter Text (getFilterText)

Returns the filter value of a data source dimension.

Note

Use this method to display the filter value.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the filter value.

32.21.25 Get Hierarchies (getHierarchies)

Returns an array of available hierarchies for a dimension.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the hierarchies</td>
</tr>
</tbody>
</table>

Returned value

HierarchyArray. It contains the available hierarchies.

32.21.26 Get Info (getInfo)

Returns data source information, for example, the key date or the technical name.

Parameters

None
Returned value

DataSourceInfo. It contains information about the data source.

32.21.27 Get Measures Dimension (getMeasuresDimension)

Returns the measures dimension.

Parameters

None

Returned value

Dimension. It contains the measures dimension.

32.21.28 Get Measure Filters (getMeasureFilters)

Returns the names of the measures filters.

Parameters

None

Returned Value

Array of MeasureFilterId. It contains the names of the measures filters.

32.21.29 Get Measure Filter Name (getMeasureFilterName)

Returns the name of a measures filter.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>MeasureFilterId</td>
<td>id of the measures filter</td>
</tr>
</tbody>
</table>

Returned Value

String. It contains the name of the measures filter.

### 32.21.30 Get Members (getMembers)

Returns an array of members of a dimension.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension for which to get the members</td>
</tr>
<tr>
<td>maxNumber</td>
<td>Integer</td>
<td>Maximum number of members to return (should not be larger than 100)</td>
</tr>
<tr>
<td>(optional) attributes</td>
<td>Array of DimensionAttribute</td>
<td>Attributes to be fetched along with the members. Avoids backend access when calling <code>getAttributeMember</code> on the returned members - at the cost of more data being processed and retrieved during this call.</td>
</tr>
</tbody>
</table>

Returned value

MemberArray. It contains the members of the dimension.

### 32.21.31 Get Member Display (getMemberDisplay)

Returns the member display for the dimension of the data source.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source for which to retrieve the member display.</td>
</tr>
</tbody>
</table>

Returned value

MemberDisplay

32.21.32 Get Member List (getMemberList)

Retrieves a list of dimension members.

ℹ️ Note

If you define allMemberText for this method and use INTERNAL_KEY as the memberPresentation, and if you want to select allMemberText by using the method setSelected value, the value for setSelectedValue is "(ALL_MEMBERS)". If you choose a different presentation type, the value is an empty string.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias</td>
</tr>
<tr>
<td>memberPresentation</td>
<td>MemberPresentation</td>
<td>Presentation of member keys</td>
</tr>
<tr>
<td>memberDisplay</td>
<td>MemberDisplay</td>
<td>Textual display of members</td>
</tr>
</tbody>
</table>

⚠️ Caution

It is not supported to choose NO_DISPLAY as member display type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxNumber</td>
<td>Integer</td>
<td>Maximum number of members to be returned (should not be significantly larger than 100)</td>
</tr>
<tr>
<td>(optional) allMemberText</td>
<td>String</td>
<td>Text of the item that represents all members. If no text is specified, the item is not added to the list</td>
</tr>
</tbody>
</table>
Returned value

ValueTextList. It contains a list of dimension members that can be used to set the items list of a dropdown or another selection component.

**Example**

In the following example, a list box is populated with members from a data source:

```javascript
LISTBOX_1.setItems(DS_1.getMemberList("0D_CUSTOMER", MemberPresentation.INTERNAL_KEY, MemberDisplay.TEXT_KEY, 100));
```

### 32.21.33 Get Negative Number Display (getNegativeNumberDisplay)

Returns how negative numbers are displayed.

**Parameters**

None

**Returned value**

NegativeNumberDisplay. It contains the display mode for negative numbers of the data source.

### 32.21.34 Get Scaling Factor (getScalingFactor)

Returns the scaling factor for a measure.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure for which the scaling factor is returned</td>
</tr>
</tbody>
</table>
Returned value

Scaling. It contains the scaling factor of the measure.

32.21.35 Get Static Filter Ext (getStaticFilterExt)

Returns the static filter value of a data source dimension in external key format.

Note
Use this method to pass the filter value to other methods.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the static filter value in external key format.

32.21.36 Get Static Filter Text (getStaticFilterText)

Returns the static filter value of a data source dimension.

Note
Use this method to display the static filter value.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source</td>
</tr>
</tbody>
</table>
Returned value

String. It contains the static filter value.

32.21.37 Get Text (getText)

Returns the description of the data source as specified by the application developer.
This text will, for example, be displayed in the prompt dialog if the application runs in unmerged prompts mode.
This allows you to specify different descriptions for multiple instances of the same query: If DS_1 and DS_2 both refer to the same query, they can still have different texts assigned.

Note

Do not confuse this text with the backend text of the underlying query.

Parameters

None

Returned value

String.

32.21.38 Get Totals Display (getTotalsDisplay)

Returns how the totals of a dimension are displayed.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension for which the totals display is returned</td>
</tr>
</tbody>
</table>
Returned value

TotalsDisplay. It contains the totals display mode of the dimension.

32.21.39 Get Totals Position (getTotalsPosition)

Returns the position of the totals in rows or columns, relative to the members.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>axis</td>
<td>Axis</td>
<td>Axis for which the totals position is returned</td>
</tr>
</tbody>
</table>

Returned value

TotalsPosition. It contains the totals position of the axis.

32.21.40 Get Variables (getVariables)

Returns the available variables.

Parameters

None

Returned value

VariableArray. It contains the available variables.
32.21.41 Get Variable Value (getVariableValue)

Returns the variable value of a data source variable in internal key format.

**Note**

Use this method to pass the variable value to other methods.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable</td>
</tr>
</tbody>
</table>

**Returned value**

String. It contains the variable value in internal key format.

32.21.42 Get Variable Value Ext (getVariableValueExt)

Returns the variable value of a data source variable in external key format.

**Note**

Use this method to pass the variable value to other methods.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable</td>
</tr>
</tbody>
</table>

**Returned value**

String. It contains the variable value in external key format.
32.21.43 Get Variable Value Text (getVariableValueText)

Returns the variable value of a data source variable.

**Note**

Use this method to display the variable.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable</td>
</tr>
</tbody>
</table>

### Returned value

String. It contains the variable value.

32.21.44 Get Zero Display (getZeroDisplay)

Returns how zero values are displayed.

### Parameters

None

### Returned value

ZeroDisplay. It contains the display mode for zero values of the data source.
32.21.45 Get Zero Display Custom Text (getZeroDisplayCustomText)

Returns the custom text that represents zero values.

Parameters

None

Returned value

String. It contains the custom text that represents zero values. Is an empty string ("") if the display mode for zero values is not CUSTOM.

32.21.46 Is Conditional Format Active (isConditionalFormatActive)

Returns true if the conditional format is active and false if it is not active.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ConditionalFormatId</td>
<td>id of the conditional format</td>
</tr>
</tbody>
</table>

Returned Value

Boolean. True if the conditional format is active and false if it is not active.

32.21.47 Is Hierarchy Active (isHierarchyActive)

Returns true if a hierarchy is active and false if not.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the hierarchy</td>
</tr>
</tbody>
</table>

Returned value

Boolean. True if a hierarchy is active and false if it is not.

32.21.48  Is Initialized (isInitialized)

Returns true if a data source is assigned and loaded.

Parameters

None

Returned value

None

32.21.49  Is Input Ready (isInputReady)

Returns whether the data source is input ready.

Note

The returned value may differ from the configured input readiness if the data source has not been loaded or query locks are in place.

Parameters

None
Returned value

Boolean. True if the data source is input ready, false if it is not.

32.21.50 Is Measure Filter Active (isMeasureFilterActive)

Returns true if the measures filter is active and false if it is not active.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>MeasureFilterId</td>
<td>id of the measures filter</td>
</tr>
</tbody>
</table>

Returned Value

Boolean. True if the measures filter is active and false if it is not active.

32.21.51 Is Result Set Empty (isResultSetEmpty)

Returns true if a data source is not initialized or the result set is empty or too large (size restriction for result sets).

Parameters

None

Returned value

None
32.21.52 Load Data Source (loadDataSource)

Loads the assigned data source.

This method can be used in combination with the data source property Load in Script to load data sources on demand with the method assignDataSource.

Parameters

None

Returned value

None

Example

loadDataSource in combination with assignDataSource

In the following example the assignment of a data source and the loading of its data are separated: thus the loading of the data can be deferred. The data source is assigned with method assignDataSource(..., false), yet the data are actually loaded with method loadDataSource:

```javascript
DS_1.assignDataSource("Q99", DataSourceType.QUERY, "SAMPLE_QUERY", false);
DS_1.loadDataSource();
```

Example

loadDataSource in combination with the property Load in Script

In the following example the data of a data source are loaded with the property Load in Script. You use the data source alias DS_2. The property Load in Script of this data source has been set to true. To load DS_2 you use method loadDataSource:

```javascript
DS_2.loadDataSource();
```

32.21.53 Move Dimension After (moveDimensionAfter)

Adds a dimension after another dimension.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be added to the axis.</td>
</tr>
<tr>
<td>otherDimension</td>
<td>Dimension</td>
<td>The above dimension is added after this dimension on the rows or columns axis. If otherDimension is not on the rows or columns axis, the statement is ignored.</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.54 Move Dimension Before (moveDimensionBefore)

Adds a dimension in front of another dimension.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be added to the axis.</td>
</tr>
<tr>
<td>otherDimension</td>
<td>Dimension</td>
<td>The above dimension is added in front of this dimension on the rows or columns axis. If otherDimension is not on the rows or columns axis, the statement is ignored.</td>
</tr>
</tbody>
</table>

Returned value

None
32.21.55 Move Dimension to Columns (moveDimensionToColumns)

Moves a dimension to a specific position on the columns axis (drill-down in columns).

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be moved to the axis.</td>
</tr>
<tr>
<td>(optional) position</td>
<td>Integer</td>
<td>Position on the axis (position 0 is the first position on the axis). If there is already a dimension at this position, it is replaced. If the parameter is not specified, the dimension is moved to the last position on the axis. If the specified position index is higher than the number of dimensions on the axis, the dimension is also moved to the last position on the axis.</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.56 Move Dimension to Rows (moveDimensionToRows)

Moves a dimension to a specific position on the rows axis (drill-down in rows).

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be moved to the axis.</td>
</tr>
<tr>
<td>(optional) position</td>
<td>Integer</td>
<td>Position on the axis (position 0 is the first position on the axis). If there is already a dimension at this position, it is replaced. If the parameter is not specified, the dimension is moved to the last position on the axis. If the specified position index is higher than the number of dimensions on the axis, the dimension is also moved to the last position on the axis.</td>
</tr>
</tbody>
</table>
The dimension is moved to the last position on the axis if specified, and if the specified position index is higher than the number of dimensions on the axis, the dimension is also moved to the last position on the axis.

### Returned value

None

### 32.21.57 Open Prompt Dialog (openPromptDialog)

Opens Prompt dialog box.

### Parameters

Table 69:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td>Integer</td>
<td>Width of Prompt dialog in pixels</td>
</tr>
<tr>
<td>height</td>
<td>Integer</td>
<td>Height of Prompt dialog in pixels</td>
</tr>
</tbody>
</table>

### Returned value

None

#### Example

In the following example, the Prompt dialog is opened:

```javascript
DS_1.openPromptDialog(400, 500);
```
32.21.58  Reload Data (reloadData)

Reloads current data from data source

This method is useful with real-time data sources where the actual data is updated during the user’s session; the method enables the refresh of the data at runtime although nothing else besides the data has changed, that is, neither filters and drill-down have changed, nor the data source structure.

Parameters

None

Returned value

None

32.21.59  Remove Dimension (removeDimension)

Removes a dimension from the rows or columns axis.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be removed from the rows or columns axis. If the dimension is not on the rows or column axis, the statement is ignored.</td>
</tr>
</tbody>
</table>

Returned value

None
32.21.60  Set Conditional Format Active  
(setConditionalFormatActive)

Activates or deactivates a conditional format.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ConditionalFormatId</td>
<td>id of the conditional format</td>
</tr>
<tr>
<td>isActive</td>
<td>Boolean</td>
<td>Specifies whether the conditional format is active</td>
</tr>
</tbody>
</table>

Returned Value

None

32.21.61  Set Decimal Places (setDecimalPlaces)

Configures how many decimal places are displayed for a measure.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure to be configured</td>
</tr>
<tr>
<td>decimals</td>
<td>Integer</td>
<td>Number of decimal places displayed (possible values: -1 (use default), 0 - 9)</td>
</tr>
</tbody>
</table>

Returned value

None
32.21.62 Set Drill Level (setDrillLevel)

Sets the drill level for the hierarchy. All data is reloaded.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the hierarchy, whose drill level is set</td>
</tr>
<tr>
<td>level</td>
<td>Integer</td>
<td>Number of levels to drill down</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.63 Set Filter (setFilter)

Sets a filter for a dimension in the internal key format.

The existing filter is replaced.

You can specify filter values in the following ways:

- For a single member, specify the internal key, for example "US".
- For multiple single members, specify an array of internal keys, for example ["US", "DE"].
- For an interval, use the JSON-format with "low" and "high", for example {"low": "0", "high": "10"}.
- For an open range, use the JSON-format with either "low" or "high", for example {"high": "10"}.
- For a combination of multiple single members, intervals, and ranges, use for example ["0", {"low": "10", "high": "50"}, {"low": "100"}].
- For hierarchy nodes, prefix the internal, fully compounded format with HIERARCHY_NODE/Node Type/, for example HIERARCHY_NODE/0HIER_NODE/ROOT.

If you want to specify filter values with the external key format use the method setFilterExt instead.
### Set Filter Ext (setFilterExt)

Sets a filter for a dimension in the external key format.

The existing filter is replaced.

You can specify the input string as a filter value. The input string syntax allows you to specify complex selections, such as intervals and excluding selections. In contrast to method `setFilter`, the members are specified in the external fully compounded key member presentation format, for example **K4/001.2012**.

**Note**

External key values might be dependent on the user's regional settings. If the application is intended to be used by a multi-lingual group of users, or if you want to specify filter values with the internal key (for example, to avoid localization problems), use the method `setFilter` instead.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be filtered</td>
</tr>
<tr>
<td>value</td>
<td>Array of Filter</td>
<td>Filter to be set</td>
</tr>
</tbody>
</table>

You can specify the input string as a filter value. This syntax allows you to specify more complex selections (such as intervals and excluding selections). In contrast to the method `setFilter`, the members are specified in the external, fully compounded key member presentation format (for example, **K4/001.2012**).
Syntax for entering values

Follow these syntax rules when you enter members for filtering.

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>equal to</td>
<td>15</td>
</tr>
<tr>
<td>multiple</td>
<td>15;18;20</td>
</tr>
<tr>
<td>exclude value</td>
<td>!22</td>
</tr>
<tr>
<td>value range</td>
<td>1-5</td>
</tr>
<tr>
<td>exclude value range</td>
<td>!6-9</td>
</tr>
<tr>
<td>greater than</td>
<td>&gt;8</td>
</tr>
<tr>
<td>exclude values greater than &lt;value limit&gt;</td>
<td>!&gt;8</td>
</tr>
<tr>
<td>greater than or equal to</td>
<td>&gt;=8</td>
</tr>
<tr>
<td>less than</td>
<td>&lt;12</td>
</tr>
<tr>
<td>exclude values less than &lt;value limit&gt;</td>
<td>!&lt;12</td>
</tr>
<tr>
<td>less than or equal to</td>
<td>&lt;=12</td>
</tr>
<tr>
<td>exclude values less than or equal to &lt;value limit&gt;</td>
<td>!&lt;=12</td>
</tr>
<tr>
<td>complex combination</td>
<td>15:10-15;20-25;1-5;&gt;8; etc.</td>
</tr>
<tr>
<td>dimension hierarchy node</td>
<td>+&lt;Dimension Attribute&gt;(&lt;Technical Name of Dimension&gt;), e.g. +ELEMENT(WBS_ELEMENT)</td>
</tr>
<tr>
<td>text hierarchy node</td>
<td>+&lt;Technical Name of Hierarchy Node&gt;(&lt;HIER_NODE&gt;), e.g. +EUROPE(&lt;HIER_NODE&gt;)</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.65 Set Measure Filter Active (setMeasureFilterActive)

Activates or deactivates a measures filter.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>MeasureFilterId</td>
<td>id of the measures filter</td>
</tr>
</tbody>
</table>
### isActive

**Type:** Boolean

Specifies whether the measures filter is active.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isActive</td>
<td>Boolean</td>
<td>Specifies whether the measures filter is active</td>
</tr>
</tbody>
</table>

### Returned Value

None

### 32.21.66  Set Member Display (setMemberDisplay)

Sets the member display for the data source dimension.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source to which a new member display is set.</td>
</tr>
<tr>
<td>memberDisplay</td>
<td>MemberDisplay</td>
<td>Display mode of the members</td>
</tr>
</tbody>
</table>

### Returned value

None

### 32.21.67  Set Negative Number Display (setNegativeNumberDisplay)

Specifies how negative numbers are displayed.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>negativeNumberDisplay</td>
<td>NegativeNumberDisplay</td>
<td>Display mode for negative values</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.68 Set Scaling Factor (setScalingFactor)

Configures the scaling factor applied to a measure.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure to be configured</td>
</tr>
<tr>
<td>scaling</td>
<td>Scaling</td>
<td>Scaling factor applied</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.69 Set Totals Display (setTotalsDisplay)

Configures how the totals of a dimension are displayed.
### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension for which the totals display is configured</td>
</tr>
<tr>
<td>totalsDisplay</td>
<td>TotalsDisplay</td>
<td>Totals display</td>
</tr>
</tbody>
</table>

### Returned value

None

#### 32.21.70 Set Totals Position (setTotalsPosition)

Configures the position of the totals in rows or columns, relative to the members.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>axis</td>
<td>Axis</td>
<td>Axis for which the totals position is configured</td>
</tr>
<tr>
<td>totalsPosition</td>
<td>TotalsPosition</td>
<td>Position where the totals are displayed</td>
</tr>
</tbody>
</table>

### Returned value

None

#### 32.21.71 Set Variable Value (setVariableValue)

Sets data source variable values in internal key format, then executes the data source query again.

When the merge flag is set, variable values are set in the application for all data sources containing the variable, when merge is not set, variable values are set for the data source.

With this method you can set only single members or hierarchy nodes in the internal key format as variable values.
If you want to set variable values for the following cases, use `setVariableValueExt` instead:

- formula variables
- text variables
- hierarchy variables
- variables representing precalculated values sets (buckets)
- multiple single values
- intervals
- variables representing a selection option
- using external format

For hierarchy nodes, prefix the internal non-compounded presentation of the hierarchy node with `HIERARCHY_NODE/nodeType/`, for example `HIERARCHY_NODE/0HIER_NODE/ROOT`.

**Note**

It is not possible to use the Crosstab component and the Chart component method `getSelectedMember` for setting variables for compounded dimensions.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable</td>
</tr>
<tr>
<td>value</td>
<td>VariableValue</td>
<td>Variable value in internal key format</td>
</tr>
</tbody>
</table>

### Returned value

None

### Example

In the following example, a fixed string value is set to a variable:

```javascript
DS_1.setVariableValue("0VAR", "4711");
```

In the following example the value is set to a variable returned from a component selection:

```javascript
DS_1.setVariableValue("0VAR", DROPDOWN_1.getSelectedValue());
```
32.21.72 Set Variable Value Ext (setVariableValueExt)

Sets data source variable values in external key format, then executes the data source query again.

When the merge flag is set, variable values are set in the application for all data sources containing the variable, when merge is not set, variable values are set for the data source.

Use this method if other selections than single members or hierarchy nodes need to be set as variable value.

**Note**

External key values might be dependent on the user’s regional settings. If the application is intended to be used by a multi-lingual group of users, it is therefore recommended to use the `setVariableValue` method when possible.

**Parameters**

Table 71:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable</td>
</tr>
<tr>
<td>value</td>
<td>InputString</td>
<td>Variable value in external key format</td>
</tr>
</tbody>
</table>

**Returned value**

None

**Example**

In the following example, a fixed string value is set to a variable:

```
DS_1.setVariableValueExt("0VAR","4711");
```

In the following example the value returned from a component selection is set to a variable:

```
DS_1.setVariableValueExt("0VAR", DROPDOWN_1.getSelectedValue());
```

32.21.73 Set Zero Display (setZeroDisplay)

Specifies how zero values are displayed.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>zeroDisplay</td>
<td>ZeroDisplay</td>
<td>Display mode for zero values</td>
</tr>
<tr>
<td>(optional) userDefinedText</td>
<td>String</td>
<td>Specifies the text if the display mode for zero values is CUSTOM (default: &quot;&quot;)</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.74  Sort by Attribute (sortByAttribute)

Sorts the result set by a specified attribute of a specified dimension in ascending or descending order.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension whose attribute is used to sort the results</td>
</tr>
<tr>
<td>attribute</td>
<td>DimensionAttribute</td>
<td>Dimension attribute used to sort the results</td>
</tr>
<tr>
<td>isSortAscending</td>
<td>Boolean</td>
<td>Specifies the sort order, if true then sort in ascending order, if false then sort in descending order</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.75  Sort by Hierarchy (sortByHierarchy)

Sorts the result set by the natural sort order of the assigned hierarchy of a dimension.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension with the assigned hierarchy used to sort the results</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.76 Sort by Measure (sortByMeasure)

Sorts the result set by a specified measure in ascending or descending order.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>measure</td>
<td>Measure</td>
<td>Measure used to sort the results</td>
</tr>
<tr>
<td>isSortAscending</td>
<td>Boolean</td>
<td>Specifies the sort order. If true then sort in ascending order, if false then sort in descending order.</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.77 Sort By Member (sortByMember)

Sorts the result set by the members of a specified dimension in ascending or descending order.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension whose members are used to sort the results</td>
</tr>
<tr>
<td>isSortAscending</td>
<td>Boolean</td>
<td>Specifies the sort order. If true then sort in ascending order, if false then sort in descending order.</td>
</tr>
<tr>
<td>(optional) isSortByKey</td>
<td>Boolean</td>
<td>If the member display of the dimension is set to <code>KEY_TEXT</code> or <code>TEXT_KEY</code>, the sort order is refined: If <code>true</code> then sort by key, if <code>false</code> then sort by text (default: <code>false</code>).</td>
</tr>
</tbody>
</table>

Returned value

None

32.21.78 Swap Dimensions (swapDimensions)

Swaps two dimensions.

At least one of the dimensions must be on the rows or columns axis, otherwise the statement is ignored.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension1</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be swapped with dimension2.</td>
</tr>
<tr>
<td>dimension2</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be swapped with dimension1.</td>
</tr>
</tbody>
</table>

Returned value

None
32.21.79 Unassign Hierarchy (unassignHierarchy)

Unassigns hierarchy from dimension.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source alias to be not displayed as a hierarchy anymore</td>
</tr>
</tbody>
</table>

Returned value

None

32.22 Data Source Info (dataSourceInfo)

An object providing information about a data source.

createdBy

Name of the user who created the data source.

dataSourceName

Name of the data source.

infoProviderDescription

Description of the InfoProvider.
**infoProviderTechnicalName**

Technical name of the InfoProvider.

**keyDate**

Key date.

**lastChangedAt**

Date of the last change made to the data source.

**lastChangedBy**

Name of the user who made the last change to the data source.

**lastDataUpdate**

Date of the last data update.

**lastDataUpdateMaximum**

Date of the last data update of all InfoProviders in a MultiProvider. Example: A MultiProvider contains three InfoProviders. The last date when all three InfoProviders were updated is the LastDataUpdateMaximum.

**lastRefreshedAt**

Date of the last data source refresh.

**queryDescription**

Description of the query.
queryTechnicalName

Name of the query.

system

Name of the system.

user

Name of the current user.

32.23 DataSourceConnection

An object representing an available backend connection.

Fields

Table 72:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Name, alias, or key of the backend connection</td>
</tr>
<tr>
<td>text</td>
<td>String</td>
<td>Description text</td>
</tr>
<tr>
<td>type</td>
<td>DataSourceConnectionType</td>
<td>Connection type</td>
</tr>
</tbody>
</table>
32.24 DataSourceDescriptor

connection

The data source connection.

description

The description of the data source.

hasChildren

Specifies whether this data source descriptor has child elements.

name

The name of the data source.

type

The type of the data source (INFOPROVIDER, QUERY or VIEW).

32.25 DataSourceName

Name of an existing data source. A data source could be, for example, a query, a query view, or an InfoProvider.
32.26 Date Field

32.26.1 Get Date (getDate)

Returns the selected date.

Parameters

None

Returned value

String. It contains the date in the format "yyyymmdd".

32.26.2 Is Enabled (isEnabled)

Returns whether component is enabled.

Parameters

None

Returned value type

Boolean. True if component is enabled or false if component is disabled.

32.26.3 On Select (onSelect)

Calls the script that is executed when the user selects a date.
32.26.4 Set Date (setDate)

Sets selected date.
If the date is invalid then the Datem Field component is empty.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>String</td>
<td>Date to be set in the format &quot;yymmd&quot;</td>
</tr>
</tbody>
</table>

Returned value

None

32.26.5 Set Enabled (setEnabled)

Enables or disables component.
Disabled components do not allow user interaction.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies whether to enable or disable the component</td>
</tr>
</tbody>
</table>

Returned value

None

32.27 Dimension

A dimension groups data of a multi-dimensional cube in categories such as, for example, company code, product, customer group, fiscal year, period, or region.

isMeasuresDimension

Indicates if this is the measures dimension.

name

The dimension’s name.

text

The dimension’s text.
32.28 Dimension Filter

32.28.1 Cancel (cancel)

Discards the filter values that have not been applied yet.

Parameters

None

Returned value

None

32.28.2 Get Dimension Key (getDimensionKey)

Returns the technical key of the dimension.

Parameters

None

Returned value

String. It contains the technical key of the dimension.

32.28.3 Get Dimension Name (getDimensionName)

Returns the name of the dimension.
Parameters

None

Returned value

String. It contains the name of the dimension.

### 32.28.4 Set Dimension (setDimension)

Set the dimension.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the data source to be set</td>
</tr>
</tbody>
</table>

Returned value

None

### 32.28.5 Show Filter Dialog (showFilterDialog)

Shows the filter dialog.

#### Parameters

None.
Returned value

None.

32.28.6 Submit (submit)

Applies the filter values.

Parameters

Table 73:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) executeOnApply</td>
<td>Boolean</td>
<td>Specifies whether to execute the script of the On Apply property after the execution of submit (default: false)</td>
</tr>
</tbody>
</table>

Returned value

None

32.29 Dropdown Box, List Box, Radio Button Group

32.29.1 Add Item (addItem)

Adds an item to the component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String</td>
<td>Unique key (or technical value) of the item. The operation is ignored if an item</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>text</td>
<td>String</td>
<td>Display text of the item.</td>
</tr>
<tr>
<td>(optional) index</td>
<td>Integer</td>
<td>Index where to insert the item (default: 0). If the index is -1 or greater</td>
</tr>
<tr>
<td></td>
<td></td>
<td>than the number of items then the item is added as the last item.</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.29.2 Get Selected Text (getSelectedText)

Returns the selected item's text.

**Parameters**

None

**Returned value**

StringArray. It contains the selected text. If no text is defined, the selected item's value is returned instead.

### 32.29.3 Get Selected Texts (getSelectedTexts)

*Note*

This method is only valid for the List Box component.

Returns the selected item's text. Can be used if multiple selection is enabled.
Parameters
None

Returned value

StringArray. It contains the selected texts as an array.

32.29.4 Get Selected Value (getSelectedValue)

Returns the selected item's value.

Parameters
None

Returned value

String. It contains the selected value.

32.29.5 Get Selected Values (getSelectedValues)

Note
This method is only valid for the List Box component.

Returns the selected item's value. Can be used if multiple selection is enabled.

Parameters
None
Returned value

StringArray. It contains the selected values as an array.

32.29.6 Dropdown Box: Get Tooltip (getTooltip)

Returns the tooltip of the dropdown box.

Parameters

None.

Returned value

String. It contains the tooltip text.

32.29.7 Is Enabled (isEnabled)

Returns whether component is enabled.

Parameters

None

Returned value type

Boolean. True if component is enabled or false if component is disabled.

32.29.8 On select (onSelect)

Calls the script that is executed when the user selects an item.
Parameters

None

Returned value

Boolean. True if component is enabled or false if component is disabled.

32.29.9 Remove All Items (removeAllItems)

Removes all items from the component.

Parameters

None

Returned value

None

32.29.10 Remove Item (removeItem)

Removes an item from the component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String</td>
<td>Unique key (or technical value) of the item. The operation is ignored if no such key exists in the component.</td>
</tr>
</tbody>
</table>
32.29.11 Set Enabled (setEnabled)

Enables or disables component.
Disabled components do not allow user interaction.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies whether to enable or disable the component</td>
</tr>
</tbody>
</table>

Returned value
None

32.29.12 Set Items (setItems)

Assigns a new list of items to the component.
The old content of the component will be deleted.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>ValueTextList</td>
<td>List of value-text pairs</td>
</tr>
</tbody>
</table>

Returned value
None
Example

In the following example, method `setItems` sets the first 100 members from the `Customer` dimension to the checkbox group / list box. Each list item’s value is represented by the customer's key, each list item’s text is represented by the customer’s text and key. No item is added to represent all members.

```java
LISTBOX_1.setItems(DS_1.getMemberList("OD_CUSTOMER", MemberPresentation.KEY, MemberDisplay.TEXT_KEY, 100));
```

or

```java
CHECKBOXGROUP_1.setItems(DS_1.getMemberList("OD_CUSTOMER", MemberPresentation.KEY, MemberDisplay.TEXT_KEY, 100));
```

### 32.29.13 Set Selected Value (setSelectedValue)

Selects the item with the specified value.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>ListValue</td>
<td>Value of the item to select</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.29.14 Set Selected Values (setSelectedValues)

**Note**

This method is only valid for the `List Box` component.

Selects the items with the specified values.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>values</td>
<td>listValue Array</td>
<td>Values of the items to select</td>
</tr>
</tbody>
</table>

Returned value

None

32.29.15 Dropdown Box: Set Tooltip (setTooltip)

Sets the tooltip of the dropdown box.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tooltip</td>
<td>String</td>
<td>Tooltip text</td>
</tr>
</tbody>
</table>

Returned value

None.

32.29.16 Sort (sort)

Sorts values in alphabetical order.
Parameters

Table 74:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) isSortAscending</td>
<td>Boolean</td>
<td>Specifies whether values are sorted in ascending or descending alphabetical order</td>
</tr>
</tbody>
</table>

Returned value

None

32.30 Fragment Gallery

32.30.1 Add Item (addItem)

Add a PortableFragmentBookmark to the component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>info</td>
<td>PortableFragmentBookmarkInfo</td>
<td>An object representing an existing PortableFragmentBookmark. The operation is ignored if no such bookmark exists.</td>
</tr>
</tbody>
</table>

Returned Value

None

32.30.2 Add Items (addItems)

Add each PortableFragmentBookmark in a PortableFragmentArray to the component.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>array</td>
<td>Array of PortableFragmentBookmarkInfo</td>
<td>An object representing an existing PortableFragmentArray. The operation is ignored if no such collection exists.</td>
</tr>
</tbody>
</table>

Returned Value

None

32.31 Filter Panel

32.31.1 Cancel (cancel)

Discards the filter values that have not been applied yet.

Parameters

None

Returned value

None

32.31.2 Set Dimensions (setDimensions)

Sets the dimensions.
Parameters

Table 75:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimensionArray</td>
<td>Array of Dimension</td>
<td>Array of dimensions</td>
</tr>
</tbody>
</table>

Returned value

None

32.31.3 Submit (submit)

Applies the filter values.

Parameters

Table 76:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(optional) executeOnApply</td>
<td>Boolean</td>
<td>Specifies whether to execute the script of the On Apply property after the execution of submit (default: false)</td>
</tr>
</tbody>
</table>

Returned value

None

32.32 Formatted Text View

32.32.1 Get HTML Text (getHTMLText)

Returns the HTML text.
Parameters

None

Returned value

String

32.32.2 Set HTML Text (setHTMLText)

Sets the HTML text.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>htmlText</td>
<td>String</td>
<td>HTML text</td>
</tr>
</tbody>
</table>

Returned value

None

32.33 FragmentBookmarkInfo

A fragment bookmark info object which contains a description of the fragment bookmark.

32.34 Geo Map

32.34.1 Center Map (centerMap)

Centers the map on a layer.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>layerId</td>
<td>String</td>
<td>Layer id</td>
</tr>
</tbody>
</table>

Returned Value

None

32.34.2 Get Copyright Text (getCopyrightText)

Returns the basemap copyright text.

Parameters

None

Returned Value

String

32.34.3 Get Selected Layer (getSelectedLayer)

Returns the selected layer.

Parameters

None

Returned Value

String
32.34.4 Is Layer Visible (isLayerVisible)

Returns whether the map layer is visible.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>layerId</td>
<td>String</td>
<td>Layer id</td>
</tr>
</tbody>
</table>

Returned Value

Boolean. True if the layer is visible, false if the layer is not visible.

32.34.5 Set Copyright Text (setCopyrightText)

Sets the basemap copyright text.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>copyright</td>
<td>String</td>
<td>Copyright text</td>
</tr>
</tbody>
</table>

Returned Value

None

32.34.6 Set Layer Visible (setLayerVisible)

Shows or hides a map layer.
**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>layerId</td>
<td>String</td>
<td>Layer id</td>
</tr>
<tr>
<td>isVisible</td>
<td>Boolean</td>
<td>Specifies whether to show or hide layer.</td>
</tr>
</tbody>
</table>

**Returned Value**

None

### 32.34.7 Set Map Url (setMapUrl)

Sets the basemap URL.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>String</td>
<td>Basemap Url</td>
</tr>
</tbody>
</table>

**Returned Value**

None

### 32.34.8 Get Selected Member (getSelectedMember)

Returns the selected member.
### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>String</td>
<td>Dimension containing the selected member.</td>
</tr>
</tbody>
</table>

### Returned Value

Member. It contains the selected member.

### 32.35 Hierarchy

A hierarchy organizes members of a dimension into a tree structure such as, for example, a hierarchy for cost centers that are combined in cost center groups.

- **name**
  
  The hierarchy's name.

- **text**
  
  The hierarchy's text.

### 32.36 Image

#### 32.36.1 Get Click Image (getClickImage)

Returns path of the click image file.
Parameters

None

Returned value

String. It contains the path of the click image file in the format returned by the Open Image dialog of the Image property.

32.36.2 Get Hover Image (getHoverImage)

Returns path of the hover image file.

Parameters

None

Returned value

String. It contains the path of the hover image file in the format returned by the Open Image dialog of the Image property.

32.36.3 Get Image (getImage)

Returns path of the default image file.

Parameters

None
Returned value

String. It contains the path of the default image file in the format returned by the Open Image dialog of the Image property.

32.36.4 Get Opacity (getOpacity)

Returns opacity value.

Parameters

None

Returned value

Integer. A value between 0 (transparent) and 100 (opaque).

32.36.5 Get Tooltip (getTooltip)

Returns the tooltip of the component.

Parameters

None.

Returned value

String. It contains the tooltip text.

32.36.6 On Click (onClick)

Calls the script that is executed when the user clicks the image.
Parameters

None

Returned value

None

32.36.7 Set Click Image (setClickImage)

Sets image displayed on mouse click.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>imageURI</td>
<td>String</td>
<td>Path of image file in the format returned by the Open Image dialog of the Image property</td>
</tr>
</tbody>
</table>

Returned value

None

32.36.8 Set Hover Image (setHoverImage)

Sets image displayed on mouse hover.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>imageURI</td>
<td>String</td>
<td>Path of image file in the format returned by the Open Image dialog of the Image property</td>
</tr>
</tbody>
</table>

Returned value

None

32.36.9 Set Image (setImage)

Sets default image.

If the default image is not set, the Image component appears transparent.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>imageURI</td>
<td>String</td>
<td>Path of image file in the format returned by the Open Image dialog of the Image property</td>
</tr>
</tbody>
</table>

Returned value

None

32.36.10 Set Opacity (setOpacity)

Sets opacity value.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>opacity</td>
<td>Integer</td>
<td>Opacity value between 0 (transparent) and 100 (opaque)</td>
</tr>
</tbody>
</table>

Returned value

None

32.36.11 Set Tooltip (setTooltip)

Sets the tooltip of the component.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tooltip</td>
<td>String</td>
<td>Tooltip text</td>
</tr>
</tbody>
</table>

Returned value

None.

32.37 Input Field

32.37.1 Get Value (getValue)

Returns value of input field.
### Parameters

None

### Returned value

String. It contains the string entered into the input field.

---

### 32.37.2 Get Tooltip (getTooltip)

Returns the tooltip of the component.

#### Parameters

None.

#### Returned value

String. It contains the tooltip text.

---

### 32.37.3 Is Editable (isEditable)

Returns whether editing is enabled.

#### Parameters

None.

#### Returned value

Boolean. True if editing is enabled or false if editing is disabled.
### 32.37.4 Is Enabled (isEnabled)

Returns whether component is enabled.

**Parameters**

None

**Returned value type**

Boolean. True if component is enabled or false if component is disabled.

### 32.37.5 Set Editable (setEditable)

Enables or disables editing in the Inputfield.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEditable</td>
<td>Boolean</td>
<td>Specifies whether to enable editing</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.37.6 Set Enabled (setEnabled)

Enables or disables component.

Disabled components do not allow user interaction.
### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies whether to enable or disable the component</td>
</tr>
</tbody>
</table>

**Returned value**

None

### 32.37.7 Set Tooltip (setTooltip)

Sets the tooltip of the component.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tooltip</td>
<td>String</td>
<td>Tooltip text</td>
</tr>
</tbody>
</table>

**Returned value**

None.

### 32.37.8 Set Value (setValue)

Sets value of input field.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String</td>
<td>Input field value</td>
</tr>
</tbody>
</table>
Returned value

None

32.38 KeyValuePair

Generic key value pair structure

key

The key.

value

The value.

32.39 JSON

A generic untyped JSON.

32.39.1 For Each (forEach)

Iterates through the elements of a JSON.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callback</td>
<td>Function</td>
<td>Function that is called with each iteration</td>
</tr>
</tbody>
</table>
Returned value

None.

Example

In the following example, the final value of result is Key: key1 Value: 100 Key: key2 Value: 200.

```javascript
var result = "";
var sample = {"key1": 100, "key2": 200};
sample.forEach(function(value, key) {
    result = result + "Key: " + key + " Value: " + value;
});
```

32.40Math

**E**

Constant number value for e, the base of the natural logarithms, which is approximately 2.7182818284590452354.

**LN10**

Constant number value for the natural logarithm of 10, which is approximately 2.302585092994046.

**LN2**

Constant number value for the natural logarithm of 2, which is approximately 0.6931471805599453.

**LOG10E**

Constant number value for the base-10 logarithm of e, the base of the natural logarithms, which is approximately 0.4342944819032518.
LOG2E

Constant number value for the base-2 logarithm of e, the base of the natural logarithms, which is approximately 1.4426950408889634.

P1

Constant number value for pi, which is approximately 3.1415926535897932.

SQRT1_2

Constant number value for the square root of 1/2, which is approximately 0.7071067811865476.

SQRT2

Constant number value for the square root of 2, which is approximately 1.4142135623730951.

32.40.1 Abs (abs)

Returns the absolute value of x.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the absolute value of x.
32.40.2 Acos (acos)

Returns the arc cosine of x.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the arc cosine of x. The result is expressed in radians and ranges from +0 to +pi.

32.40.3 Asin (asin)

Returns the arc sine of x.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the arc sine of x. The result is expressed in radians and ranges from -pi/2 to +pi/2.

32.40.4 Atan (atan)

Returns the arc tangent of x.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the arc tangent of x. The result is expressed in radians and ranges from -pi/2 to +pi/2.

32.40.5 Atan2 (atan2)

Returns the arc tangent of the quotient y/x.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>Float</td>
<td>Number</td>
</tr>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the arc tangent of the quotient y/x, where the signs of y and x are used to determine the quadrant of the result. The result is expressed in radians and ranges from -pi to +pi.

32.40.6 Ceil (ceil)

Returns the smallest integer number that is not less than x.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Integer. It contains the smallest integer number that is not less than x.

32.40.7  Cos (cos)

Returns the cosine of x.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the cosine of x. The argument is expressed in radians.

32.40.8  Exp (exp)

Returns the result of e raised to the power of x.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains $e$ raised to the power of $x$.

32.40.9 Floor (floor)

Returns the greatest integer number that is not greater than $x$.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Integer. It contains the greatest integer number that is not greater than $x$.

32.40.10 Log (log)

Returns the natural logarithm of $x$. 
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the natural logarithm of x.

### 32.40.11 Max (max)

Returns the largest of the specified values.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Float</td>
<td>Value1</td>
</tr>
<tr>
<td>(optional) b</td>
<td>Float</td>
<td>Value2</td>
</tr>
<tr>
<td>(optional) c</td>
<td>Float</td>
<td>Value3</td>
</tr>
<tr>
<td>(optional) d</td>
<td>Float</td>
<td>Value4</td>
</tr>
<tr>
<td>(optional) e</td>
<td>Float</td>
<td>Value5</td>
</tr>
<tr>
<td>(optional) f</td>
<td>Float</td>
<td>Value6</td>
</tr>
<tr>
<td>(optional) g</td>
<td>Float</td>
<td>Value7</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the largest of the specified values.
32.40.12  Min (min)

Returns the smallest of the specified values.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Float</td>
<td>Value1</td>
</tr>
<tr>
<td>(optional) b</td>
<td>Float</td>
<td>Value2</td>
</tr>
<tr>
<td>(optional) c</td>
<td>Float</td>
<td>Value3</td>
</tr>
<tr>
<td>(optional) d</td>
<td>Float</td>
<td>Value4</td>
</tr>
<tr>
<td>(optional) e</td>
<td>Float</td>
<td>Value5</td>
</tr>
<tr>
<td>(optional) f</td>
<td>Float</td>
<td>Value6</td>
</tr>
<tr>
<td>(optional) g</td>
<td>Float</td>
<td>Value7</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the smallest of the specified values.

32.40.13  Pow (pow)

Returns the result of raising x to the power of y.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
<tr>
<td>y</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>
Returned value

Float. It contains x raised to the power of y.

### 32.40.14 Random (random)

Returns a random number.

**Parameters**

None

**Returned value**

Float. It contains a random number between 0 (included) and 1 (excluded).

### 32.40.15 Round (round)

Returns a number rounded to the nearest integer.

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

**Returned value**

Integer. It contains a number rounded to the nearest integer. If the number is equally close to two integer values, then the larger integer value is returned.
32.40.16  Sin (sin)

Returns the sine of x.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the sine of x. The argument is expressed in radians.

32.40.17  Sqrt (sqrt)

Returns the square root of x.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the square root of x.

32.40.18  Tan (tan)

Returns the tangent of x.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>Float</td>
<td>Number</td>
</tr>
</tbody>
</table>

Returned value

Float. It contains the tangent of x. The argument is expressed in radians.

32.41 Member

An object representing a single dimension member. If you use this object as string, specify the internal key format. The object also provides access to the representations of the member.

externalKey

The member’s representation as external key.

externalNoncompoundedKey

The member’s representation as external non-compounded key.

getAttributeMember

Returns an attribute member that belongs to this specific dimension member

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribute</td>
<td>DimensionAttribute</td>
<td>Returns an attribute member that belongs to this specific dimension member</td>
</tr>
</tbody>
</table>

Returned Value: AttributeMember. It contains an attribute member of the dimension member.
internalKey

The member’s representation as internal key.

internalNoncompoundedKey

The member’s representation as internal non-compounded key.

text

The member’s representation as text.

32.42 Navigation Panel

32.42.1 Set Dimensions (setDimensions)

Sets the dimensions.

Parameters

Table 78:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimensionArray</td>
<td>Array of Dimension</td>
<td>Array of dimensions</td>
</tr>
</tbody>
</table>

Returned value

None
32.43 Pagebook

32.43.1 Get Page Count (getPageCount)

Returns the number of pages.

Parameters

None.

Returned value

Integer. Returns the name of the selected page.

32.43.2 Get Selected Page (getSelectedPage)

Returns the name of the selected page.

Parameters

None

Returned value

String. It contains the name of the selected page.

32.43.3 Get Selected Page Index (getSelectedPageIndex)

Returns index of the selected page. The first page has the index 0.
Parameters

None

Returned value

Integer. It contains the index of the selected page. The first page has index 0.

32.43.4 Get Selected Page by Name (getSelectedPageByName)

Selects the page by its name.
The selected page is the visible page of the pagebook.

Parameters

Table 79:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageName</td>
<td>PageName</td>
<td>Name of page to select</td>
</tr>
</tbody>
</table>

Returned value

None

32.43.5 On Select (onSelect)

Calls the script that is executed when the user selects a page.

Parameters

None
Returned value

None

32.43.6 Set Selected Page by Name (setSelectedPageByName)

Selects the page by its name.
The selected page is the visible page of the pagebook.

Parameters

Table 80:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageName</td>
<td>PageName</td>
<td>Name of page to select</td>
</tr>
</tbody>
</table>

Returned value

None

32.43.7 Set Selected Page Index (setSelectedPageIndex)

Selects page with the specified index.
The selected page is the visible page of the pagebook.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>Integer</td>
<td>Index of the page to select. The first page has index 0.</td>
</tr>
</tbody>
</table>
32.44 Panel

32.44.1 On Click (onClick)

Calls the script that is executed when the user clicks the component.

Parameters

None

Returned value

None

32.45 PDF

A global object providing export to PDF functions.

32.45.1 Export Application (exportApplication)

Exports an application to PDF, report style.

Parameters

None
Returned Value

None

32.45.2 Export Panel Screen (exportPanelScreen)

Exports a panel to PDF, WYSIWYG.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>panel</td>
<td>Panel</td>
<td>Choose a panel component to print.</td>
</tr>
</tbody>
</table>

Returned Value

None

32.45.3 Export Application Screen (exportApplicationScreen)

Exports an application to PDF, WYSIWYG.

Parameters

None

Returned Value

None
32.46 Planning

A global object providing planning functionality.

32.46.1 Client Reset (clientReset)

Rolls back to the last successfully recalculated state.

Parameters

None

Returned value

None

32.46.2 Has Client Changes (hasClientChanges)

Returns whether there is unsaved planning data on the client.

Parameters

None

Returned value

Boolean. True if there is unsaved planning data on the client and false if there is not.
32.46.3 Has Unsaved Changes (hasUnsavedChanges)

Returns whether there is unsaved planning data.

Parameters

None

Returned value

Boolean. True if there is unsaved planning data and false if there is not.

32.46.4 Recalculate (recalculate)

Recalculates changed planning data. Returns a status indicating whether the recalculation executed correctly.

Parameters

None

Returned value

Boolean. True if the recalculation executed correctly and false if an error occurred during recalculation.

32.46.5 Reset (reset)

Rolls back to the last saved server state.

Parameters

None
Returned value

None

32.46.6 Save (save)

Saves changed planning data to the server. Returns a status indicating whether the save operation executed correctly.

Parameters

None

Returned value

Boolean. True if the save operation executed correctly and false if an error occurred during this operation.

32.47 PlanningFunction

An object representing a planning function.

32.47.1 Clear All Filters (clearAllFilters)

Removes filters for all dimensions.

Parameters

None
Returned value

None

32.47.2 Clear Filter (clearFilter)

Removes the filter for a dimension.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the planning function to remove the filter from</td>
</tr>
</tbody>
</table>

Returned value

None

32.47.3 Copy Filters (copyFilters)

Copies filter values of common dimensions from a data source.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataSourceAlias</td>
<td>DataSource Alias</td>
<td>Data source alias to copy filters from</td>
</tr>
<tr>
<td>isCopyMeasuresFilter (opt)</td>
<td>Boolean</td>
<td>Specifies whether the filter values for measures are copied as well (default: false).</td>
</tr>
</tbody>
</table>
Returned value

None

32.47.4 Execute (execute)

Performs the planning function. Returns a status indicating whether the planning function executed correctly.

Parameters

None

Returned value

Boolean. True if the planning function executed correctly and false if an error occurred while executing the planning function.

32.47.5 Get Dimension Text (getDimensionText)

Returns the localized text of a dimension.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the planning function</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the localized text of the dimension.
32.47.6 Get Dimensions (getDimensions)

Returns an array of dimensions of an axis.

Parameters

None

Returned value

DimensionArray. It contains all dimensions.

32.47.7 Get Filter Ext (getFilterExt)

Returns the filter value of a data source dimension in external key format.

Note

Use this method to pass the filter value to other methods.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the planning function</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the filter value in external key format.

32.47.8 Get Filter Text (getFilterText)

Returns the filter value of the specified dimension in external key format.
### Note

Use this method to display the filter value.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the planning function</td>
</tr>
</tbody>
</table>

### Returned value

String. It contains the filter value.

### 32.47.9 Get Member List (getMemberList)

Retrieves a list of dimension members.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the planning function</td>
</tr>
<tr>
<td>memberPresentation</td>
<td>MemberPresentation</td>
<td>Presentation of member keys</td>
</tr>
<tr>
<td>memberDisplay</td>
<td>MemberDisplay</td>
<td>Textual display of members</td>
</tr>
<tr>
<td>maxNumber</td>
<td>Integer</td>
<td>Maximum number of members to be returned</td>
</tr>
<tr>
<td>(optional) allMemberText</td>
<td>String</td>
<td>Text of the item that represents all members. If no text is specified, the item is not added to the list</td>
</tr>
</tbody>
</table>

### Returned value

ValueTextList. It contains a list of dimension members.
Example

In the following example, a list box is populated with members from a planning function:

```java
LISTBOX_1.setItems(DS_1.getMemberList("OD_CUSTOMER", MemberPresentation.INTERNAL_KEY, MemberDisplay.TEXT_KEY, 100));
```

32.47.10 Set Filter (setFilter)

Sets a filter for a dimension in the internal key format.
The existing filter is replaced.

**Note**

If you want to specify filter values in the external key format use the method `setFilterExt` instead.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the planning function to be filtered</td>
</tr>
<tr>
<td>value</td>
<td>Array of Filter</td>
<td>Filter to be set</td>
</tr>
</tbody>
</table>

Returned value

None.

32.47.11 Set Filter Ext (setFilterExt)

The existing filter is replaced.

Sets a filter for a dimension in the external key format.

You can specify the input string as a filter value. The input string syntax allows you to specify complex selections, such as intervals and excluding selections. In contrast to method `setFilter`, the members are specified in the external fully-compounded key member presentation format, for example `K4/001.2012`. 
Note

External key values might be dependent on the user’s regional settings. If the application is intended to be used by a multi-lingual group of users, or if you want to specify filter values with the internal key (for example, to avoid localization problems), use the method `setFilter` instead.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>Dimension</td>
<td>Dimension of the planning function to be filtered</td>
</tr>
<tr>
<td>value</td>
<td>InputString</td>
<td>Filter to be set</td>
</tr>
</tbody>
</table>

Returned value

None

32.48 PlanningObjectWithVariables

An object representing a planning object with variables.

32.48.1 Copy Variable Value (copyVariableValue)

Copies the value of a variable from a data source.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataSourceAlias</td>
<td>DataSourceAlias</td>
<td>Data source alias to copy the variable from</td>
</tr>
<tr>
<td>variableFrom</td>
<td>Variable</td>
<td>Data source variable to copy from</td>
</tr>
</tbody>
</table>
### Return value

None

### 32.48.2 Get Variable Value Ext (getVariableValueExt)

Returns the variable value of a planning object variable in external key format.

- **Note**
  Use this method to pass the variable value to other methods.

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable</td>
</tr>
</tbody>
</table>

**Returned value**

String. It contains the variable value in external key format.

### 32.48.3 Set Variable Value Text (setVariableValueText)

Returns the variable value of a planning function variable.

- **Note**
  Use this method to display the variable.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable</td>
</tr>
</tbody>
</table>

Returned value

String. It contains the variable value.

32.48.4 Set Variable Value (setVariableValue)

Sets planning object values in internal key format.

Variable values on planning objects are independent of values set on the application.

With this method you can set only single members or hierarchy nodes in the internal key format as variable values. If you want to set variable values for the following cases, use setVariableValueExt instead:

- formula variables
- text variables
- hierarchy variables
- variables representing precalculated values sets (buckets)
- multiple single values
- intervals
- variables representing a selection option
- using external format

For hierarchy nodes, prefix the internal non-compounded presentation of the hierarchy node with HIERARCHY_NODE/NodeType/, for example HIERARCHY_NODE/0HIER_NODE/ROOT.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Query variable to set</td>
</tr>
<tr>
<td>value</td>
<td>VariableValue</td>
<td>Variable value in internal key format</td>
</tr>
</tbody>
</table>

⚠️ Restriction

It is not possible to use the crosstab and chart method getSelectedMember for setting variables for compounded dimensions.
Returned value

None

Example

In the following example a fixed string value is set to a variable:

```java
PF_1.setVariableValue("0VAR", "4711");
```

In the following example the value is set to a variable returned from a component selection:

```java
PF_1.setVariableValue("0VAR", DROPDOWN_1.getSelectedValue());
```

32.48.5 Set Variable Value Ext (setVariableValueExt)

Sets query variable values in external key format.

Variable values on planning objects are independent of values set on the application. Use this method if other selections than single members or hierarchy nodes need to be set as variable value.

Note

External key values might be dependent on the user’s regional settings. If the application is intended to be used by a multi-lingual group of users, it is therefore recommended to use the `setVariableValue` method when possible.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Query variable to set</td>
</tr>
<tr>
<td>value</td>
<td>InputString</td>
<td>Variable value to set in external key format</td>
</tr>
</tbody>
</table>

Returned value

None
Example

In the following example a fixed string value is set to a variable:

```java
PF_1.setVariableValue("0VAR", "4711");
```

In the following example the value returned from a component selection is set to a variable:

```java
PF_1.setVariableValue("0VAR", DROPDOWN_1.getSelectedValue());
```

32.48.6 Set Variable Value Range (setVariableValueRange)

Sets the value range of a variable.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Variable</td>
<td>Data source variable</td>
</tr>
<tr>
<td>variableFrom</td>
<td>VariableValue</td>
<td>Variable value to start range, in internal key format</td>
</tr>
<tr>
<td>(optional) variableTo</td>
<td>VariableValue</td>
<td>Variable value to end range, in internal key format</td>
</tr>
</tbody>
</table>

Returned value

None

32.49 PlanningSequence

An object representing a planning sequence.
32.49.1 Execute (execute)

Performs the planning sequence. Returns a status indicating whether the planning sequence executed correctly.

Parameters

None

Returned value

Boolean. True if the planning sequence executed correctly and false if an error occurred while executing the planning sequence.

32.50 Popup

32.50.1 Hide (hide)

Hides popup.

Parameters

None

Returned value

None

32.50.2 Is Showing (isShowing)

Returns whether popup is shown.
Parameters

None

Returned value

Boolean. True if popup is shown or false if popup is hidden.

32.50.3 Show (show)

Shows popup.

Parameters

None

Returned value

None

32.51 SdkDataSource

An object representing an Sdk data source.

This object has the following methods:

- clearAllFilters Clear All Filters (clearAllFilters) [page 429]
- clearFilter Clear Filter (clearFilter) [page 430]
- getData Get Data (getData) [page 436]
- getDataAsString Get Data as String (getDataAsString) [page 437]
- getDimensionText Get Dimension Text (getDimensionText) [page 440]
- getDimensions Get Dimensions (getDimensions) [page 440]
- getFilterText Get Filter Text (getFilterText) [page 441]
- getMeasuresDimension Get Measures Dimension (getMeasuresDimension) [page 443]
- getMembers Get Members (getMembers) [page 444]
• setFilter Set Filter (setFilter) [page 461]

32.52 SearchExpression

A SearchExpression is used for search operations.

32.53 SingelMemberFilter

A SingleMemberFilter is a filter composed of a single dimension member. The dimension member is specified in internal key format.

32.54 Splitcell Container

32.54.1 Is Display Mode (isDisplayMode)

Returns whether the split cell container is in display mode.

Parameters

None

Returned value

Boolean. True if the split cell container is in display mode, false if the split cell container is not in display mode.

32.54.2 On Delete (onDelete)

Calls the script that is executed when the user deletes an object from the split cell container.
Parameters
None

Returned value
None

### 32.54.3 On Drop (onDrop)

Calls the script that is executed when the user drops an object on the split cell container.

Parameters
None

Returned value
None

### 32.54.4 Set Display Mode (setDisplayMode)

Enables or disables the display mode of the split cell container.

An enabled display mode of the split cell container does not allow user interaction.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
Returned value

None

32.55 State

32.55.1 Back One Step (backOneStep)

Back One Step. Reverts the last change in the analysis application.

Parameters

None

Returned Value

None

Note

Supported platforms - BI platform, SAP NetWeaver, Local. SAP HANA is not supported.

32.55.2 Back To Start (backToStart)

Back to Start. Reverts all changes in current analysis application.

Parameters

None

Returned Value

None
32.55.3  Is Back To Start Available (isBackToStartAvailable)

Indicates if the `backToStart()` method can be used to revert all changes in the analysis application.

**Parameters**

None

**Returned value**

Boolean. True if the method can be used at this time and false otherwise.

**Note**
Supported platforms - BI platform, SAP NetWeaver, Local. SAP HANA is not supported.

32.55.4  Is Back One Step Available (isBackOneStepAvailable)

Indicates if the `backOneStep()` method can be used to revert the last change in the analysis application.

**Parameters**

None

**Returned Value**

Boolean. True if the method can be used at this time and false otherwise.

**Note**
Supported platforms - BI platform, SAP NetWeaver, Local. SAP HANA is not supported.
32.55.5 Set Personalization (setPersonalization)

Saves the current state of this analysis application as the default state. The saved state will be applied to the application each time it is opened. Valid for the current version of the analysis application only.

Parameters

None

Returned value

None

i Note
Supported platforms - BI platform, SAP NetWeaver, Local

32.55.6 Delete Personalization (deletePersonalization)

Deletes the default state for this analysis application, if one exists.

Parameters

None

Returned value

None

i Note
Supported platforms - BI platform, SAP NetWeaver, Local
32.56 String

32.56.1 Index Of (indexOf)

Returns the index of the first occurrence of the specified substring within a string. Optionally, you can provide an index from where to start the search.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchFor</td>
<td>SearchExpression</td>
<td>Substring to search for</td>
</tr>
<tr>
<td>(optional) startIndex</td>
<td>Integer</td>
<td>Index from where to start searching (default: 0)</td>
</tr>
</tbody>
</table>

Returned value

Integer. It contains the index of the first occurrence of the specified substring within a string. The first character of that string has index 0. The returned value is -1 if the substring was not found.

Example

In the following example, the returned index is 6:

```
"Hello world!".indexOf("world");
```

In the following example, the returned index is -1:

```
"Hello world!".indexOf("sailor");
```

32.56.2 Length (length)

The string's number of characters.

32.56.3 Split (split)

Splits the string at a separator string into string fragments.
## Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>separator</td>
<td>SearchExpression</td>
<td>Separator string</td>
</tr>
<tr>
<td>(optional) limit</td>
<td>Integer</td>
<td>The maximum number of fragments to return</td>
</tr>
</tbody>
</table>

## Returned value

StringArray. It contains the string fragments.

### Example

In the following example, the array contains the string fragments `one`, `two`, and `three`:

```javascript
var array = "one|two|three".split("|");
```

### 32.56.4 Substring (substring)

Returns a new string that is a substring of the given string.

The substring is composed of the characters of the given string from the given `startIndex` up to the given `endIndex` - 1. If `endIndex` is omitted, then the substring includes the characters up to the last character of the given string.

## Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startIndex</td>
<td>Integer</td>
<td>Start index of the substring, inclusive</td>
</tr>
<tr>
<td>(optional) endIndex</td>
<td>Integer</td>
<td>End index of the substring, exclusive</td>
</tr>
</tbody>
</table>

## Returned value

String. It contains the substring of the given string.
32.57 Tab

32.57.1 Get Text (getText)

Returns text.

Parameters

None

Returned value

String. It contains the text of this component.

32.57.2 Is Enabled (isEnabled)

Returns whether component is enabled.

Parameters

None

Returned value type

Boolean. True if component is enabled or false if component is disabled.

32.57.3 Set Enabled (setEnabled)

Enables or disables component.

Disabled components do not allow user interaction.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabled</td>
<td>Boolean</td>
<td>Specifies whether to enable or disable the component</td>
</tr>
</tbody>
</table>

Returned value

None

32.57.4 Set Text (setText)

Sets text.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>String</td>
<td>Text to be set</td>
</tr>
</tbody>
</table>

Returned value

None

32.58 Tabstrip

32.58.1 Get Selected Tab (getSelectedTabIndex)

Returns the name of the selected tab.
**Parameters**

None

**Returned value**

String. It contains the name of the selected tab.

### 32.58.2 Get Selected Tab Index (getSelectedTabIndex)

Returns index of the selected text.

**Parameters**

None.

**Returned value type**

Integer. It contains the index of the selected tab. The first tab has index 0.

### 32.58.3 Get Tab

Returns a tab.

**Parameters**

Table 81:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tabIndex</td>
<td>Integer</td>
<td>Index of tab to return. The first tab has index 0.</td>
</tr>
</tbody>
</table>
Returned value type

Tab. It is the tab indicated by the given index.

32.58.4 On Select (onSelect)

Calls the script that is executed when the user selects a tab.

Parameters

None

Returned value

None

32.58.5 Set Selected Tab (setSelectedTab)

Selects the tab by its name.

The selected tab is the visible tab of the tabstrip.

Parameters

Table 82:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tabName</td>
<td>TabName</td>
<td>Name of the tab to select</td>
</tr>
</tbody>
</table>

32.58.6 Set Selected Tab Index

Selects tab with the specified index.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tabIndex</td>
<td>Integer</td>
<td>Index of tab to select. The first tab has index 0.</td>
</tr>
</tbody>
</table>

Returned value

None

32.59 Text

32.59.1 Get Text (getText)

Returns text.

Parameters

None

Returned value

String. It contains the text of this component.

32.59.2 Get Tooltip (getTooltip)

Returns the tooltip of the component.

Parameters

None.
Returned value

String. It contains the tooltip text.

### 32.59.3 Set Text (setText)

Sets text.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>String</td>
<td>Text</td>
</tr>
</tbody>
</table>

Returned value

None

### 32.59.4 Set Tooltip (setTooltip)

Sets the tooltip of the component.

#### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tooltip</td>
<td>String</td>
<td>Tooltip text</td>
</tr>
</tbody>
</table>

Returned value

None.
32.60 Variable

Variables are parameters of a query. They act as placeholders for, for example, members of dimensions, hierarchies, or hierarchy nodes.

**dimensionName**

Returns the name of the associated dimension.

**inputEnabled**

Returns true if the variable is input-enabled and false if it is not.

**name**

The variable’s name.

**text**

The variable’s text.
33 Working with the Local Mode of the Design Tool

From local mode to full mode of SAP BusinessObjects Design Studio

When starting the design tool of SAP BusinessObjects Design Studio after the installation, it is being launched in local mode by default. The local mode can be used to create first analysis applications for presenting and evaluating SAP BusinessObjects Design Studio with users at the customer site. You can save the analysis applications on your local system. Without the integration to the BI platform, the technical prerequisites are on a minimal level and an easy and fast evaluation can take place. The key users of a company - possibly together with consultants - can investigate if SAP BusinessObjects Design Studio is able to cover their requirements regarding data analysis and reporting in their company. After a successful evaluation phase and with the decision to implement SAP BusinessObjects Design Studio, the necessary requirements regarding the BI platform need to be provided.

As soon as the BI platform infrastructure has been set up by the administrator, you as an application designer who has been working in local mode can easily switch the default startup mode in the "Preferences" dialog box and connect to the BI platform. Next, you can upload your locally saved analysis applications to the BI platform.

**Restriction**

The local mode of the design tool is not designed for an offline usage of the design tool. You should not use it for downloading analysis applications from the BI platform, changing and saving them locally, and then uploading the applications again to the BI platform, repeating these steps several times.

33.1 Launching SAP BusinessObjects Design Studio

Context

To launch SAP BusinessObjects Analysis Design Studio, choose [Start] > [All Programs] > [SAP BusinessObjects] > [Design Studio]. The design tool is launched with the welcome page.

Related Information

Starting with the Welcome Page [page 25]
33.2 Creating New Analysis Applications

Context

When creating new analysis applications, you can select different templates that are optimized for desktop Web browser applications or mobile applications. You can choose between blank templates or predefined templates that correspond to various design and business needs. When you choose a predefined template, the system automatically creates a copy of this template. You can change this copy according to your needs.

Procedure

1. Click [Application > New...] in the menu of the design tool or click Create Analysis Application on the Welcome page. The New Application dialog box is displayed.
2. On the New Application dialog box, perform the following steps:
   a. In the Name box, type a unique name for the application.
   b. In the Description box, type a meaningful, easy-to-understand description for the application.
   c. In the dropdown box on the right of Target Device, specify whether you want to create a Desktop Browser application or an iPad or iPhone application.
      If you want to change the type of your application (desktop browser or iPad or iPhone application) later on, you can do this by editing the application property Theme.
3. Click Next to go to the template selection screen.
4. On the template selection screen, choose one of the following options:
   ○ Choose Blank if you want to create an empty application.
   ○ Choose one of the predefined templates if you want to create an application with specific contents for your business needs.
5. Click Finish. The editor is now ready for editing.

Results

You have created a new application. Now you can define the content of your application by adding components and data sources to the blank template or adjusting the predefined templates according to your needs. For more information, see the links in the Related Topics section.

Related Information

Adding Components to an Application [page 36]
Adding a Data Source [page 50]
Assigning a Data Source to a Component [page 79]
33.3  Maintaining Settings in the Design Tool in Local Mode

You can define settings for the design tool in the Preferences dialog box. To start the Preferences dialog box, choose Tools > Preferences > Application Design. Change the default values as required and choose Apply to make these values effective. If you want to reset the default values, choose Restore Defaults. To close the Preferences dialog box, press OK.

General

You can switch the startup mode. In the default setting, the design tool starts in local mode after installation. If you want to use SAP BusinessObjects Design Studio integrated into one of the supported platforms, select the required platform:

- SAP BusinessObjects BI Platform
- SAP HANA
- SAP NetWeaver

After restarting the design tool, you need to log on to the selected platform.

Note

In the Logon to <selected platform> dialog box, you can still decide to log on locally by clicking Skip, even if the startup mode is set to one of the supported platforms. The design tool starts in local mode.

In the Undo History Size field, you can change the default value 50 and enter a number between 20 and 999. This value defines how often application designers can undo their changes when working in the design tool.

Embedded Web Server

You can define the network port for the embedded Web server of the design tool. Enter a number between 1024 and 65535. When an application is executed, the port number is shown in the URL. If the port is set to 5555 for example, the URL is http://localhost:5555/aad/web.do?APPLICATION=MYAPP. If you leave the default value 0, the system automatically assigns a network port.

Tip

If users create bookmarks for their applications, the network port has to be set to a fixed number. Auto-assigning the network port does not guarantee that the bookmarks will work properly.
If you want to test your application on a mobile device, you need to allow external access to the embedded Web server. Select the **Allow external access to embedded Web server** checkbox. The design tool must be running on your PC.

⚠️ **Caution**

Note that allowing external access is a security risk. Everyone connected to the same network can potentially access all of your applications and can see the same data that you can (as the applied data authorizations are the same).

⚠️ **Restriction**

Before other users can open the analysis application on a mobile device, the **Logon to <BW or SAP HANA system>** dialog box always appears on the PC with the design tool that is running.

- SAP HANA data sources used in the analysis application: the application designer of the PC that the design tool is running on has to enter his/her credentials and click **Log on** in the **Logon to <SAP HANA system>** dialog box.

- BW data sources used in the analysis application: activate Single-Sign On. The application designer of the PC that the design tool is running on has to confirm the client and language (or has to change these entries if required) and then click **Log on** in the **Logon to <BW system>** dialog box.

### Application Recovery

In the default setting, the system automatically saves unsaved applications every minute. You can configure the auto-save time interval as required. Select the **Save application recovery information every <1> minute** checkbox and enter the required number for the auto-save time interval.

➤ **Tip**

The design tool needs to be restarted for the new interval to become active.

There is a background job that searches for unsaved applications in the given interval. If unsaved applications are found, the system extracts their XML code and stores it under `<user home directory>\Analysis-workspace\metadata\plugins\com.sap.ip.bi.zen`. The content of this file is encrypted using Eclipse secure store technology.

ℹ️ **Note**

The auto-save function does not replace saving an application. Saving an application or closing it and answering the **Save changes?** question with either Yes or No will delete the XML from the autosaves file.

In the event of a system crash, the auto-saved XML persists. When restarting the design tool and opening the affected application again, the designer is informed that an auto-saved version of the application exists.

- If the designer decides to restore the auto-saved version, the system uses the stored XML, saves the application and opens the application in the layout editor. The auto-saved XML is removed.
- If the designer decides to discard the auto-saved version, the auto-saved XML is removed as well.
Member Selection

Application designers need to pick single members of a dimension when using statements like `setFilter` for a dropdown box, for example. They can pick the members in the content assistance of the `Script Editor` dialog box or in the `Select Member` dialog box. The `Select Member` dialog box can list a small or large number of members, depending on the maximum threshold number of members. You can set the default threshold in the `Preferences` dialog box.

- You can define the maximum number of members that are displayed in the `Select Member` dialog box. Enter the required number in the `Maximum number of members to fetch from backend in dialog` checkbox. The default threshold is 1000.

- You can define the maximum number of members that are displayed in the content assistance of the `Script Editor` dialog box. Enter the required number in the `Maximum number of members to fetch from backend in content assistance` checkbox. The default threshold is 20. If the number of available members exceeds this threshold number, the content assistance does not list single members. Instead it offers the `Select Member...` entry, which opens the `Select Member` dialog box.

  **Tip**

  Designers can access the content assistance in the script editor by pressing `CTRL` + `SPACE`.

- You can also decide whether the system should display warnings in the script editor whenever designers manually enter non-existent values. To activate the warnings, select the `Display warnings for manually entered invalid values` checkbox.

Prompt Handling

In the `Prompts` dialog box, application designers and application users set values for prompts. For SAP NetWeaver BW data sources, prompts are defined as variables. When working with data sources with defined variables, application designers might be prompted to set the required values before continuing their work in the design tool (if there are mandatory variables without default values, or if variables have invalid default values, for example). SAP BusinessObjects Design Studio stores all valid prompt values of each analysis application in the user’s cache file in `<user home directory>\Analysis-workspace\metadata\plugins\com.sap.ip.bi.zen\cache`. This provides application designers with a smooth workflow in the design tool. If this was not the case, the `Prompts` dialog would appear an analysis application is reloaded, or when the initial state of the data source is modified in the `Initial View` dialog box, for example.

In the `Preferences` dialog box, you can specify whether the `Prompts` dialog box appears when an analysis application is executed locally:

- If you want to simulate how an application user opens the analysis application, leave the checkbox deselected (This is the default setting). When you execute an analysis application locally, the `Prompts` dialog box appears and you can set the prompt values like an end user would do.

- If you want to use the prompt values from the cache file, select the `Use cached prompt values for local execution` checkbox. The `Prompts` dialog box does not appear and application designers can test their analysis applications quickly.
Note

If the *Prompts* dialog box appears despite this setting, check if the *Force Prompts On Startup* property of the analysis application is set to true. This property forces the *Prompts* dialog box to appear, regardless of your choice for the *Use cached prompt values for local execution* checkbox.

If you want to clear the prompt values of an analysis application in the cache file, press *Clear Prompt Value Cache...* You can select the required analysis application(s) for this cache deletion. When the application is reloaded, the *Prompts* dialog box will appear and you can set new values, for example.

Report-Report Interface

With the report-report interface application designers can jump to predefined jump targets in analysis applications.

If the jump target configured for the data source query is itself a query, the target query is launched as a BEx Web application by default. To avoid this and to ensure that the jump is handled by SAP BusinessObjects Design Studio and that the target query is displayed as an analysis application, you need to specify a generic analysis template.

Under *Application to use for query jumps*, enter the name of the locally saved generic analysis template.

Application Templates

SAP BusinessObjects Design Studio includes a set of templates that offer designers an easy way to get started. When creating new applications, designers can choose between different templates that are optimized for desktop Web browser applications or mobile applications in the *New Application* dialog box (Application > New...). For more information, see “Creating a new analysis application” in the Application Designer Guide: Designing Analysis Applications under Help > Help Contents in the design tool.

Application designers can also create analysis applications and provide them as templates for other application designers. Under Tools > Preferences > Application Design > Application Templates, you can add the path to the folder where these templates are stored. Here you can also define your own template categories that indicate the target device types recommended for a specific template. The template categories are then listed in the *New Application* dialog box (Application > New...), and also in the Export Application As Template dialog box (Application > Export as Template...). For more information, see “Exporting Applications As Templates” in the Application Designer Guide: Designing Analysis Applications under Help > Help Contents in the design tool.

Network Connections

Under Tools > Preferences > Application Design > Network Connections, you can specify the proxy settings to be used when opening connections. For more information, see “Network Connection Preferences” in the online help. You can access this chapter by pressing F1 or the Help button in the Preferences dialog box.
Support Settings

Under **Tools > Preferences > Application Design > Support Settings**, you can specify the amount of information stored in a log file and activate functions to record traces or collect statistics data.

Related Information

Activating Runtime Traces [page 251]
Activating SAP JCo Traces [page 252]
Configuring the Report-Report Interface for Analysis Applications in Local Mode [page 565]

33.4 Storage of Applications and Images

To work efficiently with the design tool, you need to know where your applications are stored and where to store the images and icons that you want to insert in your applications.

Analysis applications

The applications you create are automatically stored in the folder `<userhome>\Analysis-workspace\com.sap.ip.bi.zen\repository`.

Analysis applications history

The analysis applications history in the Application menu is persisted in your cache file `<home directory>\.sap\com.sap.ip.bi\cache`.

Images and Icons

There are several possible locations where you can store the images and icons that you want to use in your application:

- **in the applications directory** `<userhome>\Analysis-workspace\com.sap.ip.bi.zen\repository\MyApplication\MyImage.jpg`
- **in a subfolder of the relevant application directory**, for example, `<userhome>\Analysis-workspace\com.sap.ip.bi.zen\repository\MyApplication\MySubFolder\MyImage.jpg`

© 2015 SAP SE or an SAP affiliate company. All rights reserved.
33.5 Selecting a Connection

Prerequisites

Before you can choose a connection, you have to create connections to the BI backend systems containing the business data. SAP BusinessObjects Design Studio can access SAP HANA systems or SAP NetWeaver BW systems as BI backend systems.

Context

Connections represent BW or SAP HANA systems. They are defined and configured by your administrator. Connections have to be active if you want to select a data source and use it immediately. However, you can also work with inactive connections when designing and validate the data sources for this connection later when the connection is active. The connection is automatically active when the backend system (SAP HANA or SAP NetWeaver BW) is up and running.

Procedure

In the Connection box, click Browse.... The Select Connection dialog opens where you can choose one of the created connections.

Note

You can see all created connections in the design tool under Tools > Preferences > Backend Connections. To display newly defined connections in the corresponding table, click Reload All Connections.

Results

You have selected a connection and can now select a data source based on this connection.
You can see all created connections in the design tool under Tools » Preferences » Backend Connections. To display newly defined connections in the corresponding table, click Reload All Connections.

Related Information

Selecting a Data Source [page 53]

33.5.1 Defining Connections to BI Backend Systems

Context

Before you can add data sources to your analysis applications, you have to create connections to the BI backend systems containing the business data. SAP BusinessObjects Design Studio can access SAP HANA systems or SAP NetWeaver BW systems as BI backend systems.

Procedure

1. Open the design tool and choose Tools » Preferences » Application Design » Backend Connections.
2. Decide whether you want to create a SAP NetWeaver BW connection or a SAP HANA connection.
   - For SAP NetWeaver BW connections, click Launch SAP Logon in the upper right corner of the Connections maintained in SAP Logon area. In the SAP Logon dialog box, click New. Add the required system parameters.
   - For SAP HANA connections, click ODBC Data Source Administrator in the upper right corner of the SAP HANA Connections maintained in ODBC Data Source Administrator area. Click Add.... Select the driver HDBODBC32.
     - This driver is only available if the SAP HANA client tools are installed on your local machine.
3. Click Finish.
4. Enter the SAP HANA system in the Data Source Name field.
5. Enter the corresponding server and port in the Server:Port field.
6. Click OK and again OK.
3. To display newly defined connections in the corresponding table, click *Reload All Connections*.

4. To close the dialog box, click *OK*.

**Results**

The defined connections will be listed in the Select Connection dialog box, which appears after browsing for a different connection than the preselected one in the Add Data Source dialog box.

### 33.6 Saving an Application Using a Different Name

**Prerequisites**

You have made changes to an existing application.

**Context**

You want to save the changed application using a different name. Perform one of the following steps:

**Procedure**

Click ![Application](Application) ![Save as...](Save as...). The Save as dialog box opens. The system suggests the current name for the application.

- In the Name box, type in a new name for the application and click Save. The new application name is displayed in the outline view of the editor.
- In the list of existing applications, select the one that you want to use for your application and click Save. The system asks you to confirm that you want to overwrite an existing application. Click Overwrite. The chosen application name is displayed in the relevant tab in the editor.

⚠️ Caution

Once you have confirmed that you want to overwrite an existing application, you cannot undo the changes.

You cannot select the name of an application that is open in another editor. Select another application name.
Results

You have now saved the application using a new or different name.

33.7 Executing an Application

Context

You can always execute an application locally in your Web browser while you are working on it, and you don not need to save it first. This enables you to easily check your design steps in the application.

Procedure

Click \textit{Application} \textit{Execute Locally} in the menu or toolbar. The application is displayed in your Web browser.

33.8 Executing an Application on a Mobile Device

Prerequisites

- To execute a mobile application, make sure that you have enabled external access to the embedded Web server. Activate this setting under \textit{Tools} \textit{Preferences}.
- Execute the application on your desktop Web browser.

Context

To execute the application on a mobile device:

Procedure

1. Click \textit{Send to Mobile Device (using QR code \textsuperscript{*})}. The dialog box \textit{QR Code} is displayed with the URL of the current application encoded.
2. If there is more than one suitable network adapter/interface with at least one IP address assigned, choose the required option in the dropdown box next to \textit{IP address to use}. The system generates a new QR code.
3. Scan the QR code with a mobile device, for example, an iPad (iPad 2 or higher), using one of the various QR code scanner apps.

Results

The application opens in the mobile Safari browser on the iPad or iPhone and navigates to the application URL.

33.9 Collecting Support Information in Local Mode

Context

If you encounter problems in the design tool, you can collect the relevant information to send to SAP in a zip file.

Procedure

1. In the design tool, choose Help Support Collect Support Information...
2. Select the target folder for the zip file.
3. Click OK.
4. Choose the analysis applications you want to add.
5. Click Package.
   The support information is saved in the file DS_Support.zip.
6. To view the content of the zip file, click View....
7. Click OK.

Results

You can attach the zip file to a customer message and send it to SAP.
33.10 Coordinating the Translation of Translatable Texts in Local Mode

Context

The texts in analysis applications that are created by the application designers (for example, button texts) are translatable. To collect these texts for translation, the texts are stored in the localization.properties file that is located in the directory of your analysis application (content.biapp file).

The localization.properties file contains all translation-relevant property values and all manual entries in the Text Pool component.

Note

Every time the analysis application is saved, the system updates the localization.properties file. Do not modify the file manually.

Procedure

1. To prepare the translation of the localization.properties file, copy the file and append the required ISO language code as required.
   For example: localization_de.properties for the translation into German
2. Send the files to the people responsible for your translations.
3. After you receive the translated files, save the files in the same directory on your machine.

Results

When executing the analysis application locally, you see the translated texts displayed as specified in your language settings in the Web browser.

Tip

You can override the browser setting using the URL parameters LANGUAGE and COUNTRY. For example, you can specify American English by appending &LANGUAGE=en&COUNTRY=US to the URL of your analysis application.

In the design tool, the original texts entered by the application designer are always displayed.

Example

The localization.properties file can contain the following texts:

```javascript
# XMSG
BUTTON_1.TEXT=Filter
```
33.11 Configuring the Report-Report Interface for Analysis Applications in Local Mode

Context

Before application users can use the report-report interface (RRI) in analysis applications, you need to configure the jump targets for the query that is used as data source in the analysis application. For more information, see

- “BEx Query as Recipient” on SAP Help Portal at http://help.sap.com

If the jump target, which is configured for the data source query, is also a query, the target query is launched as a BEx Web application by default. In order to avoid this and ensure that the jump is handled by Design Studio and the target query is displayed as an analysis application, you need to specify a generic analysis template.

Procedure

1. In the design tool, create an analysis application that can be used as the generic analysis template.
   As this analysis application will be used for all query jumps, it should be a very generic application, which can handle basic analysis of an arbitrary query. In general, it must conform to the following constraints:
   ○ Contains only one data source, which is loaded in script.
   ○ Accepts query ID and system ID as URL parameters XQUERY and XSYSTEM respectively.
   ○ Loads the single data source (usually in the On Startup script) by calling assignDataSource() and passing the XQUERY and XSYSTEM parameters to the corresponding parameters of assignDataSource().

   ➤ Tip
   A good starting point, and an example of a valid generic analysis template, would be to select Generic Analysis Template when creating an application in the design tool. Note that you do not need to assign a data source to this Generic Analysis Template. When application users jump to the target query, the necessary query information is automatically added to the Generic Analysis Template as URL parameters (XQUERY, XSYSTEM).

2. Save your generic analysis template on your local machine.
3. To configure the runtime in order to use this analysis application as the generic analysis template for query jumps, proceed as follows:
a. Choose **Tools > Preferences > Application Design**

b. In the **Report-Report Interface** section at the bottom of the page, under **Application to use for query jumps**, enter the name of the locally saved generic analysis template.

## Results

In addition to the **XQUERY** and **XSYSTEM** URL parameters, which are passed to the receiving application, the filters and selection context are also passed, which are needed for dimensions relevant to the target query from the sending application. The mappings from source to target query are processed on the BW system, and the application of the appropriate filter values are processed by the Design Studio runtime.

⚠️ **Restriction**

Only query targets that are on the same system as the source query can be launched as analysis applications. All other targets are processed as BEx Web applications.
34  Terminology Essentials

Before you read any further, it might be useful to understand some basic OLAP and analysis design tool terminology.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(analysis) application</td>
<td>An entity created in the design tool of SAP BusinessObjects Design Studio. An application usually consists of user interface components like charts, crosstabs and buttons and data source aliases.</td>
</tr>
<tr>
<td>connection</td>
<td>A representation of BW or SAP HANA systems (in the design tool) that is used to add data sources.</td>
</tr>
<tr>
<td>(user interface) component</td>
<td>A design element and entity in the design tool. Components are used to visualize data (such as chart, crosstab) or to enable application users to interact with the data (for example, change filters or select drill-down dimensions). The appearance and behavior of components can be changed by editing their properties.</td>
</tr>
<tr>
<td>data source</td>
<td>A BW query or query view, a SAP HANA analytic or calculation view, used in the design tool.</td>
</tr>
<tr>
<td>data source alias</td>
<td>An instance of a data source at runtime of an application and an entity in the design tool.</td>
</tr>
<tr>
<td>data binding</td>
<td>A reference to a data source alias that provides the data for a component. Data binding is defined in the design tool and describes the relationship between components and data source aliases. These relationships are displayed in the outline view of the design tool.</td>
</tr>
<tr>
<td>dimension</td>
<td>A collection of related data members, which represents one aspect of a business; for example, products or sales.</td>
</tr>
<tr>
<td>event</td>
<td>A system notification about a specific user interaction, for example, a click on a button.</td>
</tr>
<tr>
<td>measure</td>
<td>A number or quantity that records a directly observable value or performance. Examples of measures include: sales, revenue, fixed costs, sales quantity, or number of employees.</td>
</tr>
<tr>
<td>script</td>
<td>Series of statements which are created by the user of the design tool (using the script editor or the statement wizard). By adding a script to a component, you can influence the behavior of this component and thus enable user interaction, also referred to as events, at runtime. A script typically consists of several statements.</td>
</tr>
<tr>
<td>script editor</td>
<td>A tool within the design tool to specify the actions that should take place when an event is triggered by an application user.</td>
</tr>
<tr>
<td>statement</td>
<td>A programmatic instruction within a script. The execution of a statement is typically triggered by user interaction with the component.</td>
</tr>
<tr>
<td>statement wizard</td>
<td>Dialog within the script editor that guides the application designer through the necessary steps and finally creates a script statement.</td>
</tr>
</tbody>
</table>
Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP’s gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP’s current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP’s gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).