



PUBLIC

SAP Lumira

Document Version: 2.3 SP01 – 2019-05-25

Application Designer Guide: Designing Analysis Applications

Content

1	About This Guide.	13
1.1	Who Should Read This Guide?.	13
1.2	Document History.	13
2	Getting Started.	14
2.1	What is SAP Lumira ?.	14
2.2	Basic Concepts.	15
2.3	Launching SAP Lumira Designer in Different Modes.	17
2.4	Lumira Documents.	20
2.5	Lumira Documents Mode.	21
	Context Menu Functions on Applications and Composites.	25
2.6	Feature Dependencies on Start Up Mode.	27
2.7	Starting with the Welcome Page.	28
2.8	Working with the Design Tool.	28
	Components View.	29
	Outline View.	32
	Properties View.	33
	Error Log View.	33
	Problems View.	33
	Layout Editor.	33
	Maintaining Settings in Lumira Designer.	34
	Data Storage Security.	40
3	Creating Lumira Documents.	43
4	Creating New Analysis Applications (Lumira Documents Mode).	44
4.1	Creating New Analysis Applications (BI Platform Legacy Mode).	44
	The SAPUI5 m Library as Rendering Mode.	45
4.2	Adding Components to an Application.	46
	Specifying the Properties of a Component.	47
	Deleting Components in an Application.	47
	Selecting Multiple Components in the Editor.	48
4.3	Working with Lumira Documents.	48
	Working with Lumira Documents Stored on the BI Platform.	49
	Working with Lumira Documents Stored on the Local Machine.	52
	Working with MIME Objects in SAP Lumira Documents Mode.	54
4.4	Working with Composites.	56
	Creating a Composite.	57

	Editing a Composite.	57
	Nesting Composites.	64
	Using a Composite.	64
	Lumira Stories.	64
4.5	Creating Analysis Applications for Mobile Devices.	65
5	Creating the Layout of an Analysis Application.	66
5.1	Changing the Layout Properties and Docking Behavior.	66
5.2	Using Container Components.	68
5.3	Working with Rectangle Selection.	68
6	Working with the Outline View.	69
6.1	Technical Components.	70
	Action Sheet.	70
	Application List.	70
	Authoring.	71
	Backend Connection.	72
	Bookmarks.	73
	Comments.	73
	Components.	74
	Conditional Formatting Settings.	76
	Context Menu.	85
	Document.	93
	Export AO.	94
	Export CSV.	94
	Export Excel.	95
	Export PDF.	95
	Global Scripts Objects.	96
	Keyboard Shortcuts.	96
	Text Pool.	96
	Timer.	97
	Value Help Settings.	97
6.2	Using the Context Menu in the Outline View.	101
7	Adding a Data Source.	113
7.1	Selecting a Connection.	114
	Mapping Connections.	116
	Deleting Unused Connections.	117
7.2	Selecting a Data Source.	117
	Creating a Query.	119
	Working with the Member Selector.	120
	Filtering Data in the Query Panel.	130

7.3	Universe Access in Lumira Designer.	135
7.4	Selecting Data from a Data Source for Charts and SDK Extensions.	137
7.5	Deleting a Data Source.	139
8	Inserting a Copied Data source from SAP BusinessObjects Analysis, Edition for Microsoft Office.	140
9	Working with the Initial View Dialog Box for Data Source Aliases.	141
10	Assigning a Data Source to a Component.	144
10.1	Binding the Properties of Standard Components to Data Sources.	145
	Displaying Cell Values.	146
	Configuring a List Box for Filtering Data	147
	Configuring an SDK Component to Use Multiple Data Sources.	148
11	Saving an Application	149
12	Executing an Application.	151
13	Executing an Application on a Mobile Device.	152
14	Working with Templates.	153
14.1	Using the Basic Analysis Layout Template.	153
14.2	Using the Generic Analysis Ready-to-Run Template.	155
14.3	Using the Planning Layout Template.	158
14.4	Using the Tabstrip Layout Template.	160
14.5	Using the Samples	160
15	Exporting to CSV.	162
15.1	Properties of the Export CSV Technical Component.	163
16	Exporting a Data Source View to SAP Analysis for Microsoft Office.	164
16.1	Properties of the Export AO Technical Component.	165
17	Exporting to Microsoft Excel.	166
17.1	Properties of the Export Excel Technical Component.	167
18	Exporting to PDF.	169
18.1	Languages and Export to PDF.	170
18.2	Properties of the Export PDF Technical Component.	171
18.3	View Export to PDF.	176
18.4	Report Style Export to PDF.	176
	Using the Property to Select Components for Report Style Export to PDF.	177
	Using Scripting to Select Components for Report Style Export to PDF	177
	Configuring the Properties of a Crosstab for Report Style Export to PDF.	178
18.5	Panel View Export to PDF.	181

18.6	Exporting Comments.	182
18.7	Selecting Content for the Appendix.	182
18.8	Selecting Content for Headers.	183
18.9	Exporting Components within Composites to PDF.	184
18.10	Scheduling an Export to PDF using Variants.	185
19	Changing an Analysis Application.	187
19.1	Saving an Application Using a Different Name (Lumira Documents Mode).	188
19.2	Saving an Application Using a Different Name (BI Platform Legacy Mode).	188
20	Deleting an Analysis Application.	190
21	Searching in Applications.	191
22	Assigning Analysis Applications to the Mobile Category.	192
23	Creating Planning Applications.	193
23.1	Selecting a Planning Connection.	194
23.2	Entering Data in the Crosstab (Manual Planning).	195
23.3	Editing Short Texts in Queries.	195
23.4	Using Planning Functions and Sequences (Automated Planning).	196
	Properties of Planning Functions and Planning Sequences.	198
23.5	Cell Locking.	199
	Front-End Cell Locks.	199
	Back-End Cell Locks.	200
	Locking Cells.	201
24	Working with an Application List.	204
24.1	Returning a Bookmark Url.	204
24.2	Returning a Bookmark List	205
25	Working with the Adaptive Layout.	206
25.1	Defining a Viewport.	206
25.2	Adding Blocks to the Adaptive Layout.	207
25.3	Adding Components to Blocks	208
25.4	Moving Blocks.	209
25.5	Properties of an Adaptive Layout Block.	209
25.6	Resizing Blocks.	210
26	Working with Bookmarks.	211
26.1	Security Considerations with Bookmarks.	212
26.2	Bookmarks Technical Component.	212
26.3	Multiple Bookmarks Technical Components.	213
26.4	Properties of the Bookmarks Technical Component	214
26.5	Defining Bookmarks.	215

	Using the Context Menu to Define Bookmarks.	216
	Using the Definition Property to Define Bookmarks.	217
	Using the Context Menu to Remove Item from Bookmark Definition.	218
	Using the Definition Property to Remove Component from Bookmark Definition.	218
26.6	Filtering Bookmarks.	219
26.7	Assigning Bookmarks to a Folder.	220
26.8	Scripting Bookmarks to Display in a Tree Component.	221
26.9	Setting Bookmarks as Items in a Selection Component.	222
26.10	Accessing Bookmarks.	223
26.11	Versioning a Bookmark.	223
26.12	Deleting a Bookmark.	223
26.13	Set and Get Bookmarks at Runtime.	224
26.14	Personalization.	224
	Personalization Technical Component.	225
26.15	Compatibility with Legacy Bookmark Types.	225
26.16	Bookmarking Composites.	226
	Adding a Composite to a Parent Bookmark Definition.	227
27	Working with Comments.	229
27.1	Comment Context Types.	231
27.2	Model Entity Relationships.	231
27.3	Creating a Context.	232
	Context from Crosstab Selection.	232
	Context from Crosstab with Background Filters.	234
	Context as a JSON object.	234
	Context as a String.	235
27.4	Creating a Comment.	235
	Create Comments with Different Contexts.	236
27.5	Displaying Comments.	237
27.6	Querying a Comment.	238
27.7	Updating a Comment.	238
27.8	Deleting a Comment.	239
27.9	Displaying Comments in the Crosstab.	239
28	Working with Charts.	241
28.1	Adding a Chart to your Application.	242
28.2	Configuring Initial Settings for Charts.	242
28.3	Using the Configure Chart Dialog.	243
28.4	Chart Types.	245
28.5	Chart Configuration Context Menu.	247
28.6	Working with the Chart Color Palette.	249
28.7	Sharing a Chart Color Palette across Multiple Charts.	250

28.8	Using Ad-hoc Data Comparison.	251
28.9	Number Formatting in Charts.	252
	Customized Number Formatting in Charts.	252
29	Working with Conditional Formatting in Crosstabs.	254
30	Working with Variants.	255
30.1	Using Query Variants.	255
31	Working with Drag and Drop in Applications and Crosstabs.	257
31.1	Dragging a Dimension or Dimension Member.	258
31.2	Removing a Dimension or Dimension Member.	261
31.3	Dropping Dimensions or Dimension Members.	262
31.4	Drag and Drop Constraints.	265
31.5	Dragging and Dropping Between Crosstab and Navigation Panel.	266
	Dragging and Dropping a Dimension to a Data Cell.	267
31.6	Drag and Drop in Crosstabs with Property "Display Repeated Texts".	270
32	Working With Copy and Paste in Crosstabs.	273
33	Working with the Crosstab Sorting.	278
34	Working with the Undo/Redo Functionality.	282
35	Working with the Feed List.	284
36	Working with the Grouped List.	286
36.1	Component Types Supported in the Grouped List.	287
36.2	Using the Grouped List with the Authoring Technical Component.	288
36.3	Adding Items to the Grouped List.	290
37	Working with Keyboard Shortcuts.	292
37.1	Defining Keyboard Shortcuts.	292
37.2	Properties of the Keyboard Shortcuts Technical Component.	293
38	Working with Maps.	295
38.1	Map Settings.	296
38.2	Configuring Initial Map Settings.	296
38.3	Binding Data Sources to a Map.	297
38.4	Working with GeoJSON.	297
38.5	Configuring Map Layers.	298
	Configuring a Choropleth Layer.	299
	Configuring a Bubble Chart Layer.	301
	Configuring a Marker Layer.	304
	Configuring a Pie Chart Layer.	306

38.6	Interacting with Maps.	309
39	Working with Real-Time Dashboards.	311
39.1	How to Create Real-Time Dashboards with Streaming Data Sources.	312
39.2	How to Create Real-Time Dashboards with a Single Pull-Based Data Source.	313
39.3	How to Create Real-Time Dashboards with Multiple Pull-Based Data Sources.	314
39.4	Streaming Data Source Additional Properties.	316
40	Working with Scorecards.	319
40.1	Scorecard Configuration.	320
	Defining the Row Scope.	320
	Generating the Initial Scorecard.	321
40.2	Basic Scorecard Concepts	322
40.3	Configuring Initial Scorecard Settings.	322
	Configuring Scorecard Columns.	322
	Configuring Scorecard Headers and Group Headers.	325
	Configuring the Scorecard Cell Content.	325
	Defining the Scorecard Column Content.	326
	Defining the Textual Content.	326
	Configuring Trend Charts.	326
	Configuring Comparison Charts.	327
40.4	Interacting with Scorecards.	327
40.5	Binding the Properties of Scorecards to Data Sources	327
	Properties with Single Values.	328
	Properties with Multiple Values.	329
40.6	Binding Type Settings.	329
41	Working with the Scroll Container.	333
41.1	Enabling and Disabling Scrolling.	334
41.2	Adding Items to the Scroll Container.	335
41.3	Moving Items Within the Scroll Container.	335
42	Working with the Story Component.	337
42.1	Adding Content to the Story.	337
43	Using Header Scrolling in Crosstabs.	339
43.1	"Horizontal Scrolling Enabled" Set to "False".	340
43.2	"Horizontal Scrolling Enabled" Set to "True".	341
	General Strategy if "Maximum Width of Header Area" Set to "Auto".	341
	Behavior after Manual Changing of Header With.	345
	Crosstabs With a Dimension Area or a Measure Area Only.	350
	"Maximum Width of Header Area" Set To A Pixel Value.	351
44	Using the Value Help Settings for the Prompts Dialog.	353

45	Using the Compact Layout Mode for Prompts Dialog.	354
46	Working with 1.6 Applications.	357
46.1	Transforming 1.6 Applications to 2.x Applications	358
46.2	Exchanging SAP HANA JDBC Connections in 1.6 Analysis Applications.	359
46.3	Using Automated Migration to Migrate 1.x Applications From SAP HANA JDBC to SAP HANA HTTP.	360
47	Advanced Design Tasks.	362
47.1	Using the Script Editor.	362
	Content Assistance in the Script Editor.	363
	Error Analysis in Scripting.	365
	Navigating Between Scripts.	366
	Using Offline Data Sources in Scripting.	367
47.2	Selecting Members of a Dimension.	367
47.3	Script Processing Flow in Applications.	369
47.4	Process Flow at Application Start.	371
47.5	Business Cases.	373
	Setting a Crosstab Member as a Filter on Another Component	373
	Using Two Buttons with Toggle Function	374
	Switching Pages by Clicking Images	375
	getDataAsString for BW Queries with Structures.	376
	Using the Browser Close Warning Message.	378
47.6	Working with Global Scripts Objects and Global Script Functions.	378
47.7	Enabling Text Translation in Analysis Applications.	380
	Using Scripts for Translatable Dynamic Texts.	381
	Working With Translatable Texts in Analysis Applications.	382
47.8	Deploying SDK Extensions.	383
	Installing Component SDK Extensions to SAP Lumira Designer.	384
	Installing Visualization SDK Extensions to SAP Lumira Designer.	385
	Removing Extensions from SAP Lumira Designer.	386
47.9	Creating a Generic Analysis Template for RRI Jump Targets.	386
47.10	Exporting Analysis Applications.	387
47.11	Importing Analysis Applications	389
47.12	Exporting Applications As Templates.	391
47.13	Using the Backend Connection Component for Data Source Browsing.	393
47.14	Using Processing Groups for Parallel Query Execution.	395
47.15	Unmerging Prompts (Variables).	397
47.16	Working with Background Processing.	399
	Using Simple Background Processing.	401
	Using Background Processing with Multiple Processing Steps.	402
	Using Background Processing Beyond Application Startup.	404

47.17	Using Roundtrip Optimization.	406
	Configuring Roundtrip Optimization for Analysis Applications in Lumira Designer.	409
47.18	Optimizing Application Startup.	410
	Using Optimized Components.	410
	Using the Fast Startup Application Property.	411
48	Troubleshooting.	412
48.1	Auto-Recovering Applications.	412
48.2	Using the Script Problems View.	412
48.3	Setting Network Connections If Logon Problems Occur.	413
48.4	Activating Runtime Traces.	414
48.5	Activating SAP JCo Traces.	414
48.6	Viewing And Collecting Statistics Data At Runtime.	415
48.7	Collecting Support Information.	416
48.8	Managing Logs in Lumira Designer.	417
48.9	Best Practices.	418
	Improving Performance of Analysis Applications.	418
49	User Interface Reference.	419
49.1	Properties of the Application.	419
49.2	Properties of the Data Source Alias.	429
49.3	General Properties for All Components.	432
49.4	Display Properties for All Components.	433
49.5	Layout Properties for All Components.	434
49.6	Table Components.	435
	Crosstab.	435
	Scorecard.	446
	Spreadsheet.	447
49.7	Chart Components.	451
	Chart.	451
	Chart Feeding Panel.	457
	Chart Property Editor.	458
	Chart Type Picker.	458
	Map.	460
49.8	Filter Components.	467
	Dimension Filter.	467
	Filter Bar.	469
	Filter Line.	470
	Filter Panel.	471
	Navigation Panel.	473
49.9	Basic Components.	474
	Button.	474

Checkbox.	475
Checkbox Group.	475
Date Field.	476
Dropdown Box.	477
Feed List.	478
Formatted Text View.	479
Grouped List.	481
Icon.	483
Image.	484
Input Field.	485
Link.	486
List Box.	487
Progress Indicator.	488
Radio Button Group.	488
Segmented Button.	490
Shape.	490
Switch.	491
Text.	492
Text Area.	493
Toggle Button.	494
Tree.	495
Visual Timer.	496
49.10 Container Components.	497
Adaptive Layout.	497
Dialog.	498
Grid Layout.	499
Pagebook.	499
Panel.	500
Popup.	501
Scroll Container.	503
Story.	503
Tabstrip.	505
50 Component API Reference.	506
51 Working with the Legacy Local Mode of Lumira Designer.	507
51.1 Launching SAP Lumira Designer.	507
51.2 Creating New Analysis Applications.	508
51.3 Maintaining Settings in Lumira Designer in Local Mode.	509
51.4 Storage of Applications and Images.	514
51.5 Selecting a Connection.	515
Defining Connections to BI Backend Systems.	516

51.6	Saving an Application Using a Different Name.	517
51.7	Executing an Application.	518
51.8	Executing an Application on a Mobile Device.	518
51.9	Collecting Support Information in Local Mode.	519
51.10	Coordinating the Translation of Translatable Texts in Legacy Local Mode.	520
51.11	Configuring the Report-Report Interface for Analysis Applications in Local Mode.	521
52	Terminology Essentials.	523

1 About This Guide

1.1 Who Should Read This Guide?

This guide is intended for application designers and users interested in building analysis applications using SAP Lumira Designer.

1.2 Document History

This table provides an overview of the most important document changes.

With SAP Lumira **2.3 SP01**, the following changes have been made:

Document History

Description	Related Links
A new chapter has been added in the area of working with charts, to further describe how to share a chart color palette across multiple charts.	Sharing a Chart Color Palette across Multiple Charts [page 250]
A new chapter has been added about working with the chart color palette to describe defining standard, and user defined color palettes, and customized dimension palettes.	Working with the Chart Color Palette [page 249]
The chapter about working with scheduling an export to PDF using variants has been enhanced.	Scheduling an Export to PDF using Variants [page 185]
The section about working with the application list has been enhanced around returning a bookmark url and returning a bookmark list.	Returning a Bookmark Url [page 204] Returning a Bookmark List [page 205]

2 Getting Started

2.1 What is SAP Lumira ?

SAP Lumira enables customers to gain insights from trusted enterprise data sources and personal data, and to share those insights through interactive visualizations, stories, and tailored analysis applications with other users, on desktop browsers and mobile devices.

SAP Lumira provides connectivity to SAP enterprise data models in SAP BW, SAP HANA, and SAP BusinessObjects universes (UNX) as well as to a wide variety of 3rd party databases and file data, for data acquisition, cleansing and manipulation. In addition, online connectivity with full support of SAP HANA and SAP BW (including BW Integrated Planning and SAP Business Planning and Consolidation 10.1 Embedded) data models is provided to enable drill-down into the lowest granularity of trusted data without data replication. Integration with SAP HANA smart data streaming is also provided to enable true real-time applications.

To visualize data and enable interactivity, SAP Lumira provides powerful UI elements such as charts, crosstabs, geo maps and filter components out of the box, along with a rich set of ready-to-run analysis applications, templates, and samples. Both self-service dashboards and tailored, centrally (IT) managed applications are built from the same libraries of UI elements and the same data connections and sources, to provide a uniform and consistent experience for business end users. The UI element library is based on the latest SAPUI5 HTML5 library, and ensures alignment with SAP's Fiori UI strategy while leaving full flexibility to customize the look and feel according to customer-specific corporate standards. Rich APIs and SDKs are provided to create customer-specific visualizations and data connectors, thanks to a considerable ecosystem of partner extensions.

SAP Lumira provides dedicated tools and deployment units to serve the particular needs of business key users, corporate analysis application designers, administrators, and business end users:

SAP Lumira Discovery

SAP Lumira Discovery is the rich client for business key users who need a flexible tool to connect to data sources, acquire, manipulate, and merge data, to work offline with data, to explore and analyze data online, and to create stories with visualizations from all types of data in an ad-hoc fashion.

SAP Lumira Designer

SAP Lumira Designer is the rich client for professional analysis application designers (typically working in IT departments) to create corporate analysis applications and reports. Lumira Designer provides the same UI elements as Lumira Discovery and more, and allows full control of the application look and feel and user interaction through scripting and corporate CSS style sheets. Lumira Designer also enables the creation of shared UI elements (such as headers, footers, toolbars) and the breaking down of complex applications into smaller, more manageable parts, which enable cost-effective management of large corporate BI deployments.

SAP Lumira Server

SAP Lumira Server is installed on the SAP BusinessObjects BI platform and hosts the execution runtime of Lumira documents that are created from Lumira Discovery and Lumira Designer, as well as analysis applications created with Lumira Designer or Design Studio. Administrators manage and secure Lumira documents and analysis applications with standard tools and mechanisms of the BI platform. Business users can view Lumira documents and analysis applications from the BI Launchpad in desktop browsers or on mobile devices through SAP BusinessObjects Mobile. They can interact with data (filtering, sorting, ranking, for example), adapt visualizations, create and share personal views (bookmarks), export data, and print Lumira documents.

Interoperability

All three deployment units share the same technology stack for data connectivity, UI elements, personalization, and collaboration, enabling a coherent user experience and smooth interoperability between the Lumira deployment units:

WYSIWYG layout and design

Stories (created with Lumira Discovery) and analysis applications (created with Lumira Designer) use the same data and look the same for authors and for consumers.

Extension of Lumira Discovery stories

Lumira Discovery stories can be opened and enhanced in Lumira Designer, to evolve an ad-hoc story into a corporate application, for example.

Open extended stories in Lumira Discovery

Lumira Discovery stories that have been extended in Lumira Designer can be viewed in Lumira Discovery, and further enhanced with functionalities such as adding new datasets, creating new stories, and others.

Data Manipulation for Lumira Designer

Prepared data sets from Lumira Discovery can be used as data sources in Lumira Designer, to meet information needs that cannot be served with online data connectivity.

2.2 Basic Concepts

Different Start Up Modes for the Design Tool

SAP Lumira Designer can be used and started in the following modes:

- **SAP Lumira Documents mode**
This is the default mode that supports the creation of Lumira documents, which may contain any number of analysis applications, composites, MIME objects and/or offline datasets created with Lumira Discovery. This mode can handle both local documents and documents stored on the BI platform.
- **SAP BusinessObjects BI Platform (Legacy)**
This mode supports the creation of analysis applications in the legacy format of Design Studio 1.6, which are stored on the BI platform. You can also open, edit and save analysis applications created with Design Studio 1.6 on the BI platform in this mode.
- **Local Mode (Legacy)**
This mode supports the creation of analysis applications in the legacy format of Design Studio 1.6, which are stored on your local machine. You can also open, edit and save analysis applications created with Design Studio 1.6 on your local machine in this mode.

Entities

The design tool of SAP Lumira Designer 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.

Rendering Mode for Components

The “Main” part of the SAPUI5 library (also referred to as m-mode in the design tool) is the default rendering mode for analysis applications created with SAP Lumira Designer 2.0. The Main (m) part was developed having the mobile use case in focus; those components are therefore specialized for mobile devices. However, the usage of the SAP UI5 m library is not restricted to mobile scenarios; it also supports desktop applications. In order to adjust the visualization accordingly, there are two form factors for the m mode: the compact for the desktop and the cozy form factor with more spacing and padding for mobile use cases.

2.3 Launching SAP Lumira Designer in Different Modes

Context

You can work with SAP Lumira Designer in different modes.

Procedure

1. To launch SAP Lumira Designer, choose ► **Start** ► **All Programs** ► **SAP Business Intelligence** ► **SAP Lumira** ► **Lumira Designer** ►.
2. Depending on the mode in which you log on, perform the following steps:

Option	Description
SAP Lumira Documents mode	<p>This is the default mode that lets you create Lumira documents, which may contain any number of analysis applications, composites, MIME objects, and/or offline datasets created with Lumira Discovery. This mode can handle both local documents and documents stored on the BI platform.</p> <ol style="list-style-type: none"> 1. When you launch Lumira Designer for the first time, type the host and port of the BI platform in the Web Service URL: http(s) : //<host>:<port>/dswsboobje/services/Session For further information, contact your system administrator. <div data-bbox="448 792 539 828" data-label="Section-Header"> <h3>i Note</h3> </div> <p>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.</p> <ol style="list-style-type: none"> 2. If you work with several BI platform servers, select the one you need in the Web Service URL drop-down box. 3. In the User Name box and the Password box, enter your logon credentials for the BI platform. 4. Under Authentication, select an authentication type. <div data-bbox="448 1059 539 1093" data-label="Section-Header"> <h3>i Note</h3> </div> <p>Enterprise authentication is the default authentication method. LDAP, Windows AD, Windows NT, and other third-party authentication types require a special configuration. If you don't know which authentication type to use, contact your system administrator. For information about authentication types, see the <i>SAP BusinessObjects Business Intelligence Platform Administrator Guide</i>.</p> <p>With X.509 Certificate as authentication method, select your X.509 certificate instead of providing your user credentials. For more information about X.509 authentication, see "X.509 Certificate-Based Logon to BI Platform" in the <i>Administrator Guide: SAP Lumira on SAP Help Portal</i> at https://help.sap.com.</p> <ol style="list-style-type: none"> 5. Click Connect.
SAP BusinessObjects BI Platform (Legacy)	<p>This mode lets you create analysis applications in the legacy format of Design Studio 1.6, which are stored on the BI platform. You can also open, edit, and save analysis applications created with Design Studio 1.6 on the BI platform in this mode.</p> <ol style="list-style-type: none"> 1. When you launch Lumira Designer for the first time, type the host and port of the BI platform in the Web Service URL: http(s) : //<host>:<port>/dswsboobje/services/Session For further information, contact your system administrator. <div data-bbox="448 1740 539 1774" data-label="Section-Header"> <h3>i Note</h3> </div> <p>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.</p> <ol style="list-style-type: none"> 2. If you work with several BI platform servers, select the one you need in the Web Service URL drop-down box. 3. In the User Name box and the Password box, enter your logon credentials for the BI platform.

Option	Description
	<p>4. Under Authentication, select an authentication type.</p> <div> <p>i Note</p> <p>Enterprise authentication is the default authentication method. LDAP, Windows AD, Windows NT, and other third-party authentication types require a special configuration. If you don't know which authentication type to use, contact your system administrator. For information about authentication types, see the <i>SAP BusinessObjects Business Intelligence Platform Administrator Guide</i>.</p> <p>With X.509 Certificate as authentication method, select your X.509 certificate instead of providing your user credentials. For more information about X.509 authentication, see “X.509 Certificate-Based Logon to BI Platform” in the <i>Administrator Guide: SAP Lumira on SAP Help Portal</i> at https://help.sap.com.</p> </div> <p>5. Click Connect.</p>
Local Mode (Legacy)	<p>This mode lets you create analysis applications in the legacy format of Design Studio 1.6, which are stored on your local machine. You can also open, edit, and save analysis applications created with Design Studio 1.6 on your local machine in this mode.</p> <p>The Lumira Designer is launched immediately, and you do not have to enter any user credentials.</p>

You can switch the startup mode in the [Preferences](#) dialog box under [Tools](#) > [Preferences](#) > [Application Design](#) > [General](#) .

Results

The design tool is launched with the welcome page.

i Note

In the logon dialog box, you can still decide to log on locally by clicking [Work Offline](#).

- If the startup mode is set to [SAP Lumira Documents](#) SAP Lumira Designer starts in the local mode of the [SAP Lumira Documents](#) mode.
- If the startup mode is set to [SAP BusinessObjects BI Platform \(Legacy\)](#) SAP Lumira Designer starts in [Local Mode \(Legacy\)](#).

Related Information

[Starting with the Welcome Page \[page 28\]](#)

2.4 Lumira Documents

Lumira documents are files produced with either SAP Lumira Discovery or SAP Lumira Designer that are stored on the BI platform.

In both tools, SAP Lumira Designer and SAP Lumira Discovery, it is also possible to store Lumira documents on the user's local machine in a common folder shared between SAP Lumira Designer and SAP Lumira Discovery, called `SAP Lumira Documents`. The file extension of locally stored Lumira documents is `.lumx`.

End users access Lumira documents from the BI Launchpad.

Differences with regard to the document creation tool

Depending on the tool Lumira documents are created with, there are the following differences:

- Lumira Discovery documents can contain the following:
 - One or more datasets
 - Charts built from datasets
 - Stories that describe data using charts, text, navigation points, and images.
- Lumira Designer documents can contain the following:
 - Any number of analysis applications
 - Any number of composites
 - Any number of MIME objects
 - Any number of offline datasets created in Lumira Discovery

Defining the displayed content of Lumira documents for end user consumption

On the BI platform, the stored Lumira documents can serve as containers for Lumira Discovery stories, analysis applications, composites, MIME objects (MIMEs) or offline datasets.

With Lumira Designer documents, the application designer defines which content of the document is displayed by setting one analysis application of the document as default application.

In BI Launchpad, end users access Lumira Designer documents, and not analysis applications as with Design Studio 1.x. The analysis application that is tagged as default application is launched when they open the document in BI Launchpad.

With Lumira Discovery documents, the contained stories are embedded into a Lumira Discovery application that will be launched when the document is double-clicked in BI Launchpad.

i Note

From a technical point of view, stories are composites. As composites cannot be displayed on their own, but need an analysis application as a container, the Lumira Discovery stories need the Lumira Discovery application. This Lumira Discovery application is not visible as such in the Lumira document file.

In Lumira Designer, application designers can either define any analysis application of the document as default application, or define the Lumira Discovery application as default application.

2.5 Lumira Documents Mode

This is the default mode that supports the creation of Lumira documents, which may contain any number of analysis applications, composites, MIME objects and/or offline datasets created with Lumira Discovery.

The SAP Lumira Documents Mode (Lumira Documents mode) is a combined local and BI platform-connected mode. It is always possible to edit or create documents in the SAP Lumira Documents folder, whether you are connected to a BI platform or not. If you are connected to the BI platform, any document stored on the platform can also be edited. Technically, if your work in the Lumira Documents mode, you always edit a document stored locally on your PC/laptop. This means, that you first have to download the documents that should be edited, if you are connected to the BI platform.

i Note

When working disconnected from the BI platform, it is possible to edit the documents downloaded from the BI platform when you were connected. This special edit mode is however quite limited as you cannot create new documents, work with (online) data sources, etc.

Lumira Documents can contain the following:

- any number of applications (one application must be tagged as the main application that will be launched when the document is double-clicked in BI Launchpad)
- any number of composites
- any number of MIME objects
- any number of offline data sets created in Lumira Discovery

If you work in the Lumira Documents mode and you are connected to the BI platform and you save changes to an application or composite, the document will only be saved locally. Then, at any point in time, you can decide to upload your local changes back to the platform.

For technical and performance reasons, all documents are initially closed and you can only interact with an application/composites of a document if you have opened a document.

i Note

If multiple users modify the same document either in Lumira Discovery or Lumira Designer after downloading it from the BI Platform, each user can save his/her changes and publish the document back to the BI Platform. In Lumira Designer, the user gets informed that the document has been changed on the BI platform, when the user tries to upload the changed document to the BI platform. If the user decides to overwrite the local version with the BI Platform version, but does not want to lose the local changes, the user can also choose *Cancel* or use *Save As* to save the locally modified document to the BI Platform with a different name, and then restart the synchronization.

Logon

When starting up in Lumira Documents mode, a logon dialog box is displayed. It allows you to either connect to a BI platform, to work offline (this means disconnected from the BI platform) or to exit the design tool. After successful logon, the Documents view is displayed among the usual views (Components view, Outline view, ...). If you are connected to a BI platform, two root nodes are displayed in the Documents view: the [Local Documents](#) representing the SAP Lumira Documents folder and the [BI Platform Documents](#), representing the BI platform. If you choose to [Work Offline](#) on the logon dialog, the [BI Platform Documents](#) node is only displayed if you have already been connected to a BI platform previously, and if you have downloaded at least one document. Otherwise the node will be hidden in the disconnected mode.

The Documents View

The [Documents](#) view is introduced to Lumira Designer in 2.0 and is only available in the SAP Lumira Documents mode. It is the central view for changing and creating documents, and although it can be closed by the user, the design tool will re-open it on the next launch. As already mentioned above, this view has either one or two root nodes, one of which is always representing the SAP Lumira Documents folder and the other the BI Platform. Initially, all documents are closed (indicated by the grayed-out look of the document name). Be aware that even though the documents are closed, the design tool knows which applications and composites are contained in each document. This allows you to expand the nodes in the Documents view (you can even click on [Expand All](#) in the toolbar of the view), and to filter the tree by entering some filter text into the field at the top, for quickly finding an application or composite.

Context Menu Functions on Local Documents and BI Platform Documents Nodes

The [Local Documents](#) node allows you by using the context menu to create a new document, close all currently open documents, refresh all documents (after external file system changes for example, by Lumira Discovery), and to open a Windows Explorer window for the SAP Lumira Documents folder. The [BI Platform Documents](#) node allows you to create a new document, download existing documents from the BI platform, close all currently open documents, and to synchronize all documents.

Context Menu Functions on Documents

Documents listed under the [Local Documents](#) node are from now on referred to as local documents, and documents under the [BI Platform Documents](#) node as remote documents. On any document (local and remote documents), the following context menu functions are available:

- Open / Close
- Create Application
- Save As

- Import Application
- Find References

On local documents you find additionally the following functions:

- Refresh
- Delete

On remote documents you find additionally the following functions:

- Synchronize
- Remove (from client)
- Delete (from BI platform)

For a detailed explanation of these document actions and how they behave, see the info in the *Related Links* section.

Context Menu Functions on Applications and Composites

For applications and composites the following functions are available:

- Open (can also be called via double click)
- Save As (creates a copy inside the same document)
- Find References (only valid for composites)
- Delete

Opening an Application or Composite

When opening an application that contains instances of composites, all documents containing those composites must be available locally and be opened. If missing documents are detected in BI platform-connected Lumira Documents mode, a popup comes up asking you if the missing documents should be downloaded. Afterwards the system downloads the documents and opens them.

Read-Only Documents

Both local and remote documents can be read-only. If this is the case, these documents are visualized in the *Documents* view by appending *read-only* to the document name. If a document is read-only, there are no context menu functions for the document available which would modify the document's content. Furthermore, if an application/composite of the document is opened, the editor will notify you about its read-only status. The application/composite can be modified (and executed locally which will pick up the in-memory changes), but cannot be saved.

Data Sources

When adding a new data source to an application or composite, local documents can use:

- SAP BW systems from the SAP Logon dialog
- SAP HANA HTTP systems configured in the preference page of the design tool
- Offline data sources created by Lumira Discovery and stored inside of the document

i Note

- Documents created or modified by Lumira Discovery can contain arbitrary mixtures of SAP Logon systems, OLAP connections from multiple BI platforms, universes from multiple BI platforms and offline data sets. However, in Lumira Designer the above mentioned limitations apply. An OLAP connection from BI platform 2 in a remote document stored on BI platform 1 will not work in Lumira Designer, which means the data sources will fail to initialize.
- It is not possible to copy and paste data sources from applications/composites of a local document to an application/composite of a (and vice versa). Generally, offline data sources cannot be copied across documents, even if both documents are local or both are remote.

MIME Objects Handling

Every document contains internally a MIMES folder. The images and CSS files below this folder can be organized in sub folders, and are shared between all applications and composites in the document. Remote documents additionally can use shared MIME objects stored on the BI platform. The design tool provides a MIME browser, which for local documents displays the folder tree below the MIMES folder, and for remote documents additionally displays the BI platform folder structure and two root nodes (for example, "Document[<document name>]" and "BI Platform"). To upload a MIME object from the local file system to either the current document or to some BI platform folder, use the [Upload](#) button in the MIME browser.

Execution

Applications from the local documents folder can be executed locally. Applications from remote documents can additionally be executed on the BI platform. Composites cannot be executed.

Whole documents can also be executed locally or on the BI Platform (in the context menu of the document). This allows the application designer to simulate the launching of the document in BI Launchpad.

If there is more than one application contained in the document, it is important to specify which application should be launched by default ([Default Application](#) in the context menu of the document).

i Note

Note that executing the currently edited application via toolbar or [File](#) menu always launches this application, overriding the configured default application of the document.

Dependencies Between Documents

An application in one document can embed composites from a second, third, etc. document, thereby creating a dependency between the document containing the application to the document(s) containing the composite(s).

Handling Lumira documents from untrusted sources

We recommend that you to handle Lumira documents from untrusted sources (received by E-Mail, for example) with caution like any other files from untrusted sources (executables or Microsoft Excel files, for example). The dynamic scripting capabilities of Lumira documents could harm the confidentiality of your data or the integrity of your system.

Related Information

[Context Menu Functions on Applications and Composites \[page 25\]](#)

2.5.1 Context Menu Functions on Applications and Composites

Open Function on Documents

Opens the selected documents (multi-selection possible). Updates the metamodel of the design tool, which will add all contained composites to the components palette.

Close Function on Documents

Checks if applications of other documents are currently open which use composites from the document to be closed and closes them, giving the application designer the possibility to save unsaved changes.

Delete Function on Local Documents

Checks if applications/composites of other documents are currently open which use composites from the document to be closed and closes them giving the application designer the possibility to save unsaved

changes. All these editors must be closed first (only the editors, not the documents!). If such editors are dirty, the application designer must be given the chance to save them. Then it closes the selected documents and updates the designer's metamodel, which will delete all contained composites from the components palette.

Remove From Client Function on Remote Documents

Checks if the document has unsynchronized local modifications. Since removing the document from the client would lose all changes, a warning dialog box explaining this is displayed. The dialog also explains that the removed document can be downloaded again at any time. If confirmed, all currently open editors of applications from other documents, which use any of the composites in the document to be removed, are closed, giving the possibility to save changes. Then all open editors for applications / composites / CSS files of the document to be deleted are closed, even if they have unsaved changes. Finally, the document is removed and the designer's meta model is updated, removing all contained composites from the components palette.

Delete from Server Function on Remote Documents

This function is very similar to the [Remove from Client](#) function, but additionally deletes the LUMX InfoObject from the BI platform, and also checks all documents on the BI platform if they are still referencing the document to be deleted (in addition to all downloaded documents).

Refresh Function on Local Documents

Local documents can be modified externally (outside Lumira Designer), for example by Lumira Discovery, or by copy and paste operations in the file system. The [Refresh](#) function on one or more local documents re-scans the selected documents for such changes. If changes are detected, the affected documents and all their open editors are closed first (without the possibility to save changes, since the document was already changed in the file system anyway).

Synchronize function on Remote Documents

The action performed when synchronizing a document depends on three different timestamps:

- the current last-modified timestamp of the InfoObject on the BI Platform
- the last-modified timestamp of the InfoObject on the BI platform at the time the document was downloaded
- the last-modified timestamp of the document in the file system

First, the design tool fetches the current remote timestamp. This might fail if the remote document has been deleted in the meantime. In this case, the application designer is asked to remove his downloaded copy. If he does not want to lose his local changes, he can choose [Cancel](#), use [Save As](#) to save the document under a new name, and restart the synchronization.

Now the current and downloaded remote timestamps are compared. If they are equal but the document has been modified locally in the meantime, the local version of the document will be uploaded to the BI Platform:

- If they differ and the document has not been modified locally, the new version is downloaded from the BI Platform, overwriting the local version.
- If they differ but the document has been modified locally, there is a conflict (both client-side and server-side modifications). A popup is displayed to the application designer explaining the situation, and he resolves the conflict by either overwriting the BI Platform version with his local version or vice versa. Again, if the application designer does not want to lose his local changes, he can cancel the synchronization, use [Save As](#) to save his local version under a new name on the BI Platform, and restart the synchronization process to finally resolve the conflict.

2.6 Feature Dependencies on Start Up Mode

The following table shows you an overview of the main Lumira Designer features and their availability in different deployments of the design tool.

Features delivered with Release 2.0

Feature	Lumira Documents Mode (connected to the BI Platform)	Lumira Documents Mode (as work offline mode)	Local Mode (Legacy)	BI Platform Mode (Legacy)
Lumira Designer Templates	yes	yes	yes	yes
Composites	yes	yes	no	no
Create BI Platform Documents (Remote Documents)	yes	no	no	no
Export Analysis Applications	no	no	yes	yes
Import Analysis Applications Designed in SAP UI5 Commons Mode	no	no	yes	yes
Offline Data Sources	yes	yes	no	no
Open Lumira Discovery Documents in Lumira Designer	no	yes	no	no
Right-to-Left-Support	yes	yes	yes	yes

2.7 Starting with the Welcome Page

When you start SAP Lumira Designer 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 Lumira Designer 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 Lumira Document (for Lumira Documents start up mode) or Create New (for BI platform legacy start up mode):** Enables you to create a new Lumira document / a new analysis application. You can select different templates. 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:** Enables you to open recently-used applications and composites (only valid for Lumira Documents mode).
- **Useful Links:** Enables you to find further useful information about the solution, for example, SAP Community Network and SAP Help Portal.

i 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 \[page 508\]](#)

2.8 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 \[page 29\]](#)

[Outline View \[page 32\]](#)

[Properties View \[page 33\]](#)

[Layout Editor \[page 33\]](#)

[Managing Logs in Lumira Designer \[page 417\]](#)

[Maintaining Settings in Lumira Designer \[page 34\]](#)

[Storage of Applications and Images \[page 514\]](#)

[Activating Runtime Traces \[page 414\]](#)

2.8.1 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:

Table Components

You use *Table Components* to visualize your data. After you have dragged and dropped for example, a crosstab into the layout editor, the component initially displays no data (in the case of crosstab and spreadsheet) or dummy data (in the case of scorecard). As soon as you assign a data source to the table component, it displays the data of the data source. This group contains the following components:

- **Crosstab**
The crosstab displays multi-dimensional data in a grid with analytic functions.
- **Scorecard**
Scorecards can help you to make complex result sets browsable by using user friendly visualizations (like charts) and highly customizable layout properties.
- **Spreadsheet**
Spreadsheets display multi-dimensional data in a grid with analytic functions. In addition, the Spreadsheet component offers an easy-to-use copy function for copying cells and sheet areas. The spreadsheet is mainly intended for use with planning applications with input-ready queries.

Chart Components

You use chart components like charts and maps to visualize your data. After you have dragged and dropped a chart or a map into the layout editor, the component initially displays dummy data. As soon as you assign a data source to the chart or map, it displays the data of the data source.

This group contains the following components:

- **Chart**
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.

- Chart Feeding Panel
Use the [Chart Feeding Panel](#) to display all the measures and dimensions bound to a chart.
- Chart Property Editor
Use the [Chart Property Editor](#) to allow the application user to change the properties of a data bound chart when running the application.
- Chart Type Picker
Use the [Chart Type Picker](#) to allow the application user to select an alternative chart type to display their data, when running the application.
- Map
Use the [Map](#) component to display different layers of geographical information on a map and allow users to drilldown through the different layers to reveal data in a variety of ways.

Filter Components

You can add filter components to your application to enable detailed data views for the application user by adding filters on dimensions and measures.

This group contains the following components:

- Dimension Filter
Using the [Dimension Filter](#), you can add a generic filter for one dimension to an application, without the need to use scripting.
- Filter Bar
The [Filter Bar](#) is the horizontal filter element for applications. Initially, the filter area of the filter bar is not displayed, even if you have chosen a dimension in the [Dimensions](#) properties. Instead the application user has to click on [Show Filter Bar](#) at runtime.
- 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.
- Filter Panel
With the [Filter Panel](#), you can add a generic filter to an application without the need to use 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.
- 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.

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
- Checkbox Group
- Date Field
- Dropdown Box
- Feed List
- Formatted Text View
- Icon
- Image
- Input Field
- Link
- List Box
- Progress Indicator
- Radio Button Group
- Segmented Button
- Shape
- Switch
- Text
- Text Area
- Toggle Button
- Tree
- Visual Timer

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:

- Adaptive Layout
The [Adaptive Layout](#) allows you to create an adaptive application that can be viewed on multiple devices. It allows you to preview different layouts at design time, so you can develop one application that will work on large or small devices.
- Dialog
The [Dialog](#) component allows you to create a dialog box with or without buttons
- Grid Layout
Use the [Grid Layout](#) component to group and order the content of your application in a grid that is not displayed at runtime.
- Pagebook
Use the [Pagebook](#) component to enable the user to switch between the different views of an application. You group the components or contents of a view on a separate page.
- Panel
Use the [Panel](#) component to group other components in your application.
- Popup

The [Popup](#) component allows you to help 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.

- **Scroll Container**
Use the [Scroll Container](#) to enable or disable horizontal or vertical scrolling on supported container components.
- **Story**
Use the [Story](#) component to create a presentation-style application that uses visualizations, text, graphics, illustrations, shapes, and other customizations to describe data. You can also use it to apply a filter across multiple data bound components within the same story.
- **Tabstrip**
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.

2.8.2 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 \[page 69\]](#)

[Using the Context Menu in the Outline View \[page 101\]](#)

2.8.3 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.

i 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.

2.8.4 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 \[page 365\]](#)

2.8.5 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.

2.8.6 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.

2.8.7 Maintaining Settings in Lumira Designer

You can define settings for Lumira Designer 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, Lumira Designer starts in [SAP Lumira Documents](#) mode after installation. If you want to use SAP Lumira Designer in a different mode, select the required platform mode:

- [SAP Lumira Documents](#)
This is the default mode that supports the creation of Lumira documents, which may contain any number of analysis applications, composites, MIME objects and/or offline datasets created with Lumira Discovery. This mode can handle both local documents and documents stored on the BI platform.
- [SAP BusinessObjects BI Platform \(Legacy\)](#)
Supports the creation of analysis applications in the legacy format of Design Studio 1.6 that are stored on the BI platform. You can also open, edit, and save analysis applications created with Design Studio 1.6 on the BI platform in this mode.
- [Local Mode \(Legacy\)](#)
Supports the creation of analysis applications in the legacy format of Design Studio 1.6 that are stored on your local machine. You can also open, edit, and save analysis applications created with Design Studio 1.6 on your local machine in this mode.

After restarting Lumira Designer, you need to log on to the selected platform.

i Note

In the [Logon to <selected platform>](#) dialog box, you can still decide to log on locally by clicking [Work Offline](#).

- If the startup mode is set to [SAP Lumira Documents](#) SAP Lumira Designer starts in the local mode of the SAP Lumira Designer mode.
- If the startup mode is set to [SAP BusinessObjects BI Platform \(Legacy\)](#) SAP Lumira Designer starts in [Local Mode \(Legacy\)](#).

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 Lumira Designer.

Embedded Web Server

You can define the network port for the embedded Web server of Lumira Designer. 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 at 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. Lumira Designer must be running on your PC.

i Note

Allowing external access is supported in [SAP Lumira Documents](#) mode and in [Local Mode](#).

⚠ 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 Lumira Designer that is running.

- SAP HANA data sources used in the analysis application: the application designer of the PC that Lumira Designer 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 Lumira Designer 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 Preview Language (remote documents only)

The language of message texts and tooltips is defined in 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).

i Note

These language settings, which also affect the formatting of numbers, dates and times, are not available in the offline Lumira documents mode and in the legacy local mode.

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

Lumira Designer 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 will extract their XML code and store it under `<user home directory>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen`. The content of this file is encrypted using Eclipse secure store technology.

i 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 Lumira Designer 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 will not list single members. Instead it will offer 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.

Script Editing

If you run into performance issues when editing complex scripts, you can disable the backend validation of your scripts by selecting [Disable backend validation](#). When backend validation is enabled, the script editor's automatic syntax checks involve calls to a backend server, for example to validate the name of a dimension or a hierarchy. This can slow down working with larger scripts significantly, especially if you are working over a WAN connection. For this reason, you can explicitly disable backend validation (leaving all other checks in place).

In addition, you can disable the inplace help of the script editor by selecting [Disable inplace help](#).

i Note

If you experience performance issues on operating systems with limited memory and large applications with complex scripts, SAP recommends to disable the inplace help.

These script editing settings are also available in the script editor. For more information, see “Content Assistance in the Script Editor” in the *Application Designer Guide: Designing Analysis Applications* under [Help > Help Contents](#) in Lumira Designer.

Prompt Handling

In the [Prompts](#) dialog box, application designers and application users set values for prompts. For SAP 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 Lumira Designer (if there are mandatory variables without default values, or if variables have invalid default values, for example). SAP Lumira Designer stores all valid prompt values of each analysis application in the user's cache file in `<user home directory>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen\cache`. This provides application designers with a smooth workflow in Lumira Designer. 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.

i 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 Lumira Designer 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 ► Help Contents ►](#) in Lumira Designer.

In the local and BI platform legacy modes, 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 Lumira Designer.

Backend Connections

Under [► Tools ► Preferences ► Application Design ► Backend Connections ►](#), you can add new SAP HANA HTTP connections to Lumira Designer. You use these connections in local Lumira documents. If you need SAP BW connections in your local documents, Lumira Designer uses the SAP BW systems that are available in the *SAP Logon* dialog box.

i Note

The *Backend Connections* page is available in the Lumira documents mode. It is also available in the Legacy Local mode to add and reload SAP BW and SAP HANA connections.

Network Connections

Under **Tools** > **Preferences** > **Application Design** > **Network Connections**, you can specify the proxy settings to be used when opening connections.

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. For more information, see the links in the *Related Information* section below.

Under **SAP HANA Data Sources (HTTP)**, you can specify that you use an HTTP proxy for communication with the SAP HANA system. If you have configured your SAP HANA system so that it can only be accessed using a proxy, select the **Use HTTP Proxy** checkbox.

- Under **Proxy Host**, enter the name of the system hosting the proxy service used by the HTTP destination.
- Under **Proxy Port**, enter the port to connect on the system hosting the proxy service.

You can also use this setting for support purposes: Using Fiddler as a proxy, you can record Fiddler traces, which helps SAP to diagnose your issues with SAP HANA HTTP data sources. To set Fiddler as a proxy, select the **Use HTTP Proxy** checkbox.

- Under **Proxy Host**, enter **localhost**.
- Under **Proxy Port**, enter **8888**.

For more information, see SAP Note [2166049](#).

In the status bar of Lumira Designer, the indicator **HTTP Proxy: Off/On** shows you if this setting is switched on or off. By double-clicking on the indicator, you can quickly access the **Preferences** page to change your current setting.

Scripting

Under **Tools** > **Preferences** > **Scripting**, you can specify settings for syntax coloring and maintain templates for scripting.

Related Information

[Setting Network Connections If Logon Problems Occur \[page 413\]](#)

[Activating Runtime Traces \[page 414\]](#)

[Activating SAP JCo Traces \[page 414\]](#)

2.8.8 Data Storage Security

Information about the storage of Lumira documents, analysis applications, analysis applications history, data source history, prompt values and bookmarks.

Locally stored documents (Lumira documents or analysis applications) are not protected by SAP Lumira . Protection needs to be provided by the respective device management (access control or encryption, for example).

Lumira Documents

Application designers working with Lumira Designer and business users working with Lumira Discovery store the Lumira documents in folders on the BI platform. They can also store images and icons, which are not part of the Lumira documents, but are referred to in their Lumira documents, on the BI platform.

→ Tip

We recommend using a common root folder for all images and icons that are referred to in the Lumira documents on all BI platform instances. This will make it simpler to transport Lumira documents and the referenced images or icons between different BI platform systems.

You can restrict access to the Lumira document files by assigning access rights to the folders. For more information on configuring the authorizations for the folders containing Lumira documents, see "Setting Rights", in the *Business Intelligence Platform Administrator Guide*, at <https://help.sap.com>.

Application designers can also use images and icons in their Lumira documents from the Internet or intranet. Your application designers should be aware that using icons and images from the Internet could be a security risk.

Lumira documents that are created in the *Local Documents* folder of Lumira Designer are stored under %USERPROFILE%\Documents\SAP Lumira Documents by default. Lumira documents that are created in Lumira Discovery locally are also stored in this SAP Lumira Documents folder.

If you want to place the SAP Lumira Documents folder in a different location, proceed as described under *Related Information*.

Analysis applications

Application designers store the analysis application files that are created in the BI platform legacy mode in folders on the BI platform. They can also store images and icons they use in their analysis applications on the BI platform.

→ Tip

We recommend using a common root folder for all images and icons used in analysis applications on all BI platform instances. This will make it simpler to transport analysis applications and the referenced images or icons between different BI platform systems.

You can restrict access to the analysis application files by assigning access rights to the folders. For more information on configuring the authorizations for the folders containing analysis applications, see “Setting Rights”, in the *Business Intelligence Platform Administrator Guide*, at <https://help.sap.com>.

Application designers can also use images and icons in their analysis applications from the Internet or intranet. Your application designers should be aware that using icons and images from the Internet could be a security risk.

Analysis applications history

The analysis applications history that is available in the *Application* menu of Lumira Designer in the local and BI platform legacy modes is persisted in the user's cache file in `<user home directory>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen\cache`. The content of this file is encrypted using Eclipse secure store technology.

Analysis applications and composites history

The analysis applications and composites history that is available under **File > Open Recent** in the Lumira Documents mode, is persisted in the user's cache file in `<user home directory>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen\cache`. The content of this file is encrypted using Eclipse secure store technology.

Data source history

The data source history in the *Add Data Source* dialog box available in all modes of Lumira Designer is persisted in the user's cache file in `<user home directory>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen\cache`. The content of this file is encrypted using Eclipse secure store technology.

Prompt values

All valid prompt values the user entered in the *Prompts* dialog for any analysis application (when working in any of the available modes in Lumira Designer) are persisted in the user's cache file in `<user home directory>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen\cache`. The content of this file is encrypted using Eclipse secure store technology.

Related Information

[Lumira Designer: Changing the location of the SAP Lumira Documents folder \[page 42\]](#)



2.8.8.1 Lumira Designer: Changing the location of the SAP Lumira Documents folder

With Lumira Designer, you can place the `SAP Lumira Documents` folder in a different location.

Context

For Lumira documents created with Lumira Designer, add a JAVA VM argument to the `SapLumiraDesigner.ini` file:

Procedure

1. Go to the directory on your machine, where you have installed Lumira Designer.
2. Open the `SapLumiraDesigner.ini` file.
3. Enter the following JAVA VM argument with the required value at the end:–
`Dhilo.document.dir=<document folder>`
4. Choose  **File** .
5. Restart Lumira Designer.

3 Creating Lumira Documents

You create Lumira documents that can obtain any number of analysis applications, composites, MIME objects and offline datasets created in Lumira Discovery.

Context

To create analysis applications or composites, you first have to create documents in Lumira Designer. Following the template you have chosen (blank, pre-defined or read-to-run), the system automatically creates the corresponding application in your document's folder listed in the [Documents](#) view.

Procedure

1. Choose [Create Lumira Document](#) in one of the following locations:
 - on the [Welcome](#) page
 - in the toolbar of Lumira Designer
 - in the [File](#) menu of Lumira Designer
2. (optional) You can now create applications and composites for this document.

Results

The system creates a documents folder with the name of your document in the [Documents](#) view. Under your documents folder you will find your applications and composites and a [Connections](#) and [MIMEs](#) folder.

Related Information

[Creating New Analysis Applications \(Lumira Documents Mode\) \[page 44\]](#)

[Working with Composites \[page 56\]](#)

[Creating New Analysis Applications \(BI Platform Legacy Mode\) \[page 44\]](#)

4 Creating New Analysis Applications (Lumira Documents Mode)

If you work in Lumira documents mode, you create new applications within Lumira documents.

Context

When creating new analysis applications in the Lumira Documents mode, you can choose between standard templates (*Blank* or predefined ones like *Basic Analysis Layout*, *Basic Layout*) or ready-to-run predefined templates (such as *Generic Analysis*) that correspond to various design and business needs. When you choose a template, the system automatically creates a copy of it. You can change this copy according to your needs.

You can only create applications if you have created a local or BI Platform document.

Procedure

1. Right-click the document in the *BI Platform Documents* or the *Local Documents* folder of the *Documents* view and choose *Create Application*. The *New Application* dialog box is displayed.
2. In the *New Application* dialog box, perform the following steps:
 - a. Enter a unique name for your application in the field under *Application Name*.
 - b. If your Lumira Designer is deployed on a BI platform, the system suggests under *Folder* a folder on the platform to store your application. However, you can also browse for a different folder.
 - c. Choose the template you want to use for your application. A short description of the template is displayed at the right side of the dialog box.
 - d. Choose *Create*. The template is opened in the editor where you can change and edit it.

4.1 Creating New Analysis Applications (BI Platform Legacy Mode)

Context

When creating new analysis applications in the BI Platform legacy mode, you can select different templates. You can choose between standard templates (*Blank* or predefined ones like *Basic Analysis Layout*, *Basic Layout*) or ready-to-run predefined templates (such as *Generic Analysis*) that correspond to various design and business needs. When you choose a template, the system automatically creates a copy of it. 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. In the **New Application** dialog box, perform the following steps:
 - a. Enter a unique name for your application in the field under **Application Name**.
 - b. If your Lumira Designer is deployed on a BI, the system suggests under **Folder** a folder on the platform to store your application. However, you can also browse for a different folder.
 - c. Choose the template you want to use for your application. A short description of the template is displayed at the right side of the dialog box.
 - d. Choose **Create**. The template is opened in the editor where you can change and edit it.

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 or ready-to-run templates according to your needs.

Related Information

[Working with Templates \[page 153\]](#)

4.1.1 The SAPUI5 m Library as Rendering Mode

The “Main” part of the SAPUI5 library (also referred to as m-mode in the design tool) is the default rendering mode for analysis applications created with SAP Lumira Designer 2.0. The Main (m) part was developed having the mobile use case in focus; those components are therefore specialized for mobile devices. However, the usage of the SAP UI5 m library is not restricted to mobile scenarios; it also supports desktop applications. In order to adjust the visualization accordingly, there are two form factors for the m mode: the compact for the desktop and the cozy form factor with more spacing and padding for mobile use cases.

Analysis applications of SAP BusinessObjects Design Studio 1.6 that have been created with components from the commons library of SAP UI5 can still be opened, edited and saved but only in the BI Platform mode of Lumira Designer. It is not possible to create new applications in the commons mode (by choosing **New**, **Save As** or **Import** from the menu of the design tool).

For further information about the commons library of SAP UI5 used in SAP BusinessObjects Design Studio 1.6, please refer to the documentation chapter “Using the SAPUI5 m Library” on SAP Help Portal at <https://help.sap.com>.

4.2 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, [Table 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 \[page 47\]](#)

[Deleting Components in an Application \[page 47\]](#)

4.2.1 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):

Option	Description
numeric entry	Enter a number and click <input type="text" value="Enter"/> . This number represents either a numeric pixel description (such as for the properties <i>Top Margin</i> , <i>Left Margin</i> , <i>Bottom Margin</i> , <i>Right Margin</i> , <i>Width</i> , <i>Height</i>), or an absolute number (such as for the properties <i>Grid Row</i> and <i>Grid Column</i>).
string entry	Enter a text and click <input type="text" value="Enter"/> . Properties of this type include <i>Caption</i> and <i>Tooltip</i> .
Boolean choice	Click the dropdown box in the <i>Value</i> column of the relevant property. Click <i>false</i> or <i>true</i> and then <input type="text" value="Enter"/> . Properties of this type include <i>Enabled</i> and <i>Style</i> .
interaction using dialog boxes	With more complex properties you open and work in special dialogs. Click the <i>Value</i> 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.

4.2.2 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.

4.2.3 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).

4.3 Working with Lumira Documents

Lumira documents are files which may contain any number of analysis applications, composites, MIME objects and/or offline datasets created with Lumira Discovery.

Lumira documents are only created in the Lumira Documents mode, the default mode for Lumira Designer.

As an application designer you can work with Lumira documents that are stored on the BI platform. In addition, you can work with Lumira documents stored on your local machine. In order to provide a broader access to a local Lumira document, you can upload the local document to the BI platform.

4.3.1 Working with Lumira Documents Stored on the BI Platform

You can create, access, change, synchronize, delete, and execute Lumira documents that are stored on the BI platform in Lumira Designer.

Creating Lumira Documents

You can create Lumira documents to be stored on the BI platform by choosing [Create Document...](#) in the context menu of the [BI Platform Documents](#) folder in the [Documents](#) view. A new Lumira document containing an empty application, is created on the BI platform. The system also creates a [Connections](#) folder and a MIME object ([MIMEs](#)) folder beneath the [Documents](#) node.

i Note

When application users open this Lumira document on the BI Launchpad, it is launched with its empty application as default application.

Accessing Lumira Documents

You can access existing Lumira documents on the BI platform for the following purposes:

- to edit the Lumira document
- to use the composites of the Lumira document in another Lumira document

To download the documents, choose [Download Documents...](#) in the context menu of the [BI Platform Documents](#) folder in the [Documents](#) view. Select the required Lumira document from the BI Launchpad folders. Downloaded documents appear under the [BI Platform Documents](#) folder.

i Note

Lumira Designer keeps local copies of the downloaded documents (these local copies are not to be confused with the documents under the Local Documents folder), remembers the original location - the BI Launchpad folder - on the BI platform, and synchronizes with the BI platform.

Changing Lumira Documents

Before you can edit Lumira documents stored on the BI platform, you need to open them by choosing [Open](#) in the context menu of the document. Open documents are displayed in dark font, while documents that are not open are displayed in grey font color. Technically, as an application designer you do your changes in local copies of the remote document.

i Note

When opening a document, Lumira Designer reads all metadata, detects dependencies, and - if applicable - loads acquired data from documents created with Lumira Discovery. These background processes can be time-consuming, and therefore Lumira Designer does not open any document under BI Platform Documents or Local Documents on startup.

Depending on the content of the Lumira document, you can then open an analysis application or composite by double-clicking the analysis application or the composite, or by choosing [Open](#) in the context menu of the respective analysis application or composite. As soon as you have modified and saved an application or composite, an asterix (*) is displayed in front of the Lumira document name in the Documents view, indicating that the local copy is out-of-sync with the version on the BI platform.

i Note

It is also possible to work disconnected from the BI platform and edit the documents that were downloaded from the BI platform when you were connected. However, this special edit mode is quite limited as you cannot create new documents, work with (online) data sources, etc.

Synchronizing Lumira Documents

If you work in the Lumira Documents mode connected to the BI platform and save changes to an application or composite, the document is only saved locally. At any point in time, you can upload their local changes back to the BI platform by choosing [Synchronize](#) in the context menu of the document.

i Note

If multiple users modify the same document either in Lumira Discovery or Lumira Designer after downloading it from the BI Platform, each user can save changes and publish the document back to the BI Platform. In Lumira Designer, you are informed that the document has been changed on the BI platform, when trying to upload the changed document to the BI platform. If you decide to overwrite the local version with the BI platform version without losing the local changes, you can choose [Cancel](#) or use [Save As](#) to save the locally modified document to the BI Platform with a different name, and then restart the synchronization.

Executing Lumira Documents

Application users have the following options for running Lumira documents:

- In the BI launch pad, application users run the documents by choosing [Open](#) in the context menu of a document, or by double-clicking the document. Each document is displayed in a new tab in the BI launch pad. The document displays the analysis application that is set as the default analysis application in Lumira Designer.
- Application users can open a document directly without navigating in the BI launch pad. In the [Documents](#) tab in the BI launch pad, users can choose [Document Link](#) in the context menu of a Lumira document. The URL of the document is displayed. Users can copy this URL, send it in an e-mail, and run the URL on their

mobile device. If they want to add this URL to their bookmarks, they have to add it manually instead of pressing the corresponding button.



- Application users can open a document in the SAP BusinessObjects Mobile application by tapping on the document thumbnail. For more information, see the *Mobile Application User Guide* at <https://help.sap.com>.





As an application designer you have the following additional options for running Lumira documents:

- You can choose *Execute Locally* in the context menu of a document under *BI Platform Documents*. The document is displayed with its default analysis application in a separate Web browser window, using a local Web server embedded in Lumira Designer. You can use this function for checking you local changes before uploading the document.
- You can choose *Execute on BI Platform* in the context menu of a document under *BI Platform Documents*. The document is displayed with its default analysis application in a separate Web browser window, using the document link of the Lumira document and the current session of the BI platform. You do not have to log on to the BI platform when executing the Lumira document on the BI platform. The document is run with the credentials that the application designer provided when logging on to Lumira Designer in Lumira Documents mode.

i Note

With this function, the Lumira document version of the BI platform is launched, not the local copy with the local changes. This allows you to simulate the launching of the document in the BI launch pad directly in Lumira Designer.

In order to compare the BI platform version with the local version, you can additionally use the   *Execute Locally* icon of the toolbar in Lumira Designer. This function launches the currently open analysis application of the editor in a separate Web browser window, regardless of the configured default application of the corresponding Lumira document. The *Execute Locally* function is available for analysis applications, but not for composites.

- In Lumira Designer, you can test your local document versions for usage on mobile devices by choosing   *Send Local Version to Mobile Device* in the toolbar.
- In Lumira Designer, you can test the BI platform document versions for usage on mobile devices by choosing   *Send BI Platform Version to Mobile Device* in the toolbar.

Both of the functions *Send Local Version to Mobile Device* and *Send BI Platform Version to Mobile Device* open the QR Code dialog box with the following possibilities:

- You can click the QR code and a new Web browser window opens. You are prompted to log on to the BI platform.
- You can click *Copy URL to clipboard*, paste the URL and send it in an e-mail, and execute the URL on your mobile device. If you want to add this URL to your bookmarks, you have to add it manually instead of pressing the corresponding button.

i Note

This is necessary because the original URL is immediately forwarded to a generic URL, keeping the document reference in the session. When bookmarking and opening this generic URL that lacks the document reference, the Web browser shows an error page only.

Removing Local Copies of Lumira Documents from Local Machine

You can remove your local copies of Lumira documents in the *BI Platform Documents* folder by choosing *Remove from Local Machine* in the context menu of the document.

i Note

Lumira Designer checks if the document has unsynchronized local modifications. Since removing the document from the local machine would lose all changes, a warning dialog box explaining this is displayed. The dialog also explains that the removed document can be downloaded again at any time. If confirmed, all currently open editors of applications from other documents which use any of the composites in the document to be removed are closed, making it possible to save changes. Then all open editors for applications / composites / CSS files of the document to be removed are closed, even if they have unsaved changes. Finally, the document is removed and the meta model of Lumira Designer is updated, removing all contained composites from the components palette.

Deleting Lumira Documents

In Lumira Designer, you can delete Lumira documents by choosing *Delete from BI Platform* in the context menu of the document beneath the *BI Platform Documents* folder. The Lumira documents are permanently deleted, both the remote documents on the BI platform and also the local copies on the local machine. They cannot be restored.

i Note

Similar to the *Remove from Local Machine* function, Lumira Designer checks dependencies of the Lumira document to be deleted, both on the local machine and on the BI platform, and warns the application designer by listing dependent applications and composites still used in other Lumira documents, for example.

4.3.2 Working with Lumira Documents Stored on the Local Machine

You can create, access, change, delete, and execute Lumira documents that are stored on their local machine in Lumira Designer.

Creating Local Lumira Documents

In Lumira Designer, you create local Lumira documents by choosing *Create Document...* in the context menu of the *Local Documents* folder in the *Documents* view. A new empty Lumira document containing an empty application is created in the *SAP Lumira Documents* folder of the local machine. The system also automatically creates a *Connections* folder and a MIME object (*MIMEs*) folder beneath the *Documents* node.

Accessing Local Lumira Documents

The local documents that are stored in the `SAP Lumira Documents` folder of the local machine are available under [Local Documents](#) in the [Documents](#) view of Lumira Designer.

If the content of the `SAP Lumira Documents` folder has been enhanced while you are working in Lumira Designer - you have added a Lumira document that was received in an e-mail or on a file share, for example - you can access the newly added documents in Lumira Designer by choosing [Refresh](#) in the context menu of the [Local Documents](#) folder.

If you start Lumira Designer after enhancing the `SAP Lumira Documents` folder, the newly added documents are displayed automatically under the [Local Documents](#).

Changing Local Lumira Documents

Before you can edit local Lumira documents, you need to open them by choosing [Open](#) in the context menu of the document. Open documents are displayed in dark font, while documents that are not open are displayed in grey font color.

i Note

When opening a document, Lumira Designer reads all metadata, detects dependencies and - if applicable - loads acquired data from documents created with Lumira Discovery. These background processes can be time-consuming, and therefore Lumira Designer does not open any document under [Local Documents](#) on startup.

Depending on the content of the Lumira document, you can then open an analysis application or composite by double-clicking the analysis application or the composite, or by choosing [Open](#) in the context menu of the respective analysis application or composite.

Uploading Local Lumira Documents to BI Platform

In order to provide a broader access to a local Lumira document, you can upload the document to the BI platform by choosing [Upload to BI Platform](#) in the context menu of the local document.



i Note

The Lumira document needs to be self-contained. It is not possible to upload local documents that use composites of other local documents.

After the upload, you need to check if the used data sources are available on the BI platform and change them accordingly in the analysis application of the document.

Executing Local Lumira Documents

You have the following options for running Lumira documents:

- In Lumira Designer, you can choose [Execute Locally](#) in the context menu of a document under [Local Documents](#).
The document is displayed with its default analysis application in a separate Web browser window, using a local Web server embedded in Lumira Designer. You can use this function for checking your local document before uploading it to the BI platform, for example.
- In addition, you can use the  [Execute Locally](#) icon of the toolbar in Lumira Designer. This function launches the currently open analysis application of the editor in a separate Web browser window, regardless of the configured default application of the corresponding Lumira document. The [Execute Locally](#) function is available for analysis applications, but not for composites.
- In Lumira Designer, you can test your local document for use on mobile devices by choosing  [Send to Mobile Device](#) in the toolbar.
The QR Code dialog box appears with the following possibilities:
 - You can click the QR code and a new Web browser window opens.
 - You can click [Copy URL to clipboard](#), paste the URL and send it in an e-mail, and execute the URL on your mobile device. If you want to add this URL to your bookmarks, you have to add it manually instead of pressing the corresponding button.

Deleting Local Lumira Documents

In Lumira Designer, you can delete locally stored documents by choosing [Delete](#) in the context menu of the document beneath the [Local Documents](#) folder. The local documents are permanently deleted from the `SAP Lumira Documents` folder on the local machine and cannot be restored.

Note

Lumira Designer checks dependencies of the Lumira document to be deleted, and warns you by listing dependent applications and composites still used in other Lumira documents, for example.

4.3.3 Working with MIME Objects in SAP Lumira Documents Mode

You can work with MIME objects in SAP Lumira documents mode.

When you create a Lumira document on the BI platform or your local machine, a new empty Lumira document is created on the BI platform or in the `SAP Lumira Documents` folder of the local machine. At the same time, the system also creates a MIME object ([MIMEs](#)) folder under the documents node.

Whenever you add MIME objects to your applications, the objects are displayed in a folder hierarchy below the [MIMEs](#) node, reflecting the folder hierarchy in the Lumira document.. There are two types of nodes below the [MIMEs](#) node: folders and files. The [MIMEs](#) node itself acts as a special read-only folder.

Context Menu Functions of the MIMEs Folder

From the context menu of the MIMEs folder, you can do the following:

- paste MIMEs that were previously copied to the clipboard.
- create new sub-folders
- upload MIMEs from the local file system

Context Menu Functions of a MIME Object Folder

In the context menu of a MIME object folder, you can do the following:

- copy the folder to the clipboard
- paste MIMEs that were previously copied to the clipboard
- create new MIME object folders
- upload MIMEs from the local file system
- rename the folder
- delete the folder

Context Menu Functions of a MIME File

In the context menu of a MIME object file, you can

- copy the file to the clipboard.
- paste MIME objects that were previously copied to the clipboard.
- rename the file.
- delete the file.
- find all references to the file in all applications and composites of the document.

Working with Drag and Drop, Copy and Paste and Double-Clicking

You can use drag and drop in the [Documents](#) view to easily re-arrange your MIMEs in folders.

i Note

Note that you can use drag and drop only within the document but not between different documents. However you can work with copy and paste across documents.

When you rename, delete or move MIME objects, all affected applications and composites that are currently using those MIME objects are listed in the confirmation dialog box, and will automatically adapt the MIME references in the affected apps/composites.

i Note

Note that for this purpose, it is necessary to close any affected open editor.

If you double-click a MIME object representing an image, an image preview dialog box is opened that allows you to preview the image (either scaled to the dialog window or in its original size) and see the image's width, height and filesystem size. If you double-click a MIME representing a textual MIME file (CSS file, JSON file or CSV file), a different dialog box is opened where the textual content can be viewed (read-only).

You can drag images onto components in the design canvas/editor, if the following conditions are fulfilled:

- The target application or composite has the same root document as the dragged MIME object.
- You can drag only one image file (multi-selection is not possible)
- The target component must be one of the following:
 - an *Image* component
In this case the *Image* property of the *Image* component will be set to the dragged image (but not the *Hover Image* or *Click Image* properties).
 - a *Button* component
In this case the *Icon* property of the *Button* component will be set to the dragged image.

4.4 Working with Composites

Composites are components, which can be created by an application designer - similar to SDK components but the application designer does not have to be a software developer.

Composites behave like standard components as they can have the following elements:

- properties (like the *Text* property of *Text* component)
- script methods (like the `setText` method of the *Button* component)
- events (like the `onClick` event of the *Button* component)

In addition, composites - like applications - can contain other components and data sources. This allows you to group user interface elements and corresponding business logic into a dedicated composite. This can be used for refactoring purposes to group a subset of an application (for example, a header or repeating elements in your applications) or make the subset reusable so it can be shared with other application designers.

i Note

- Composites are only supported in the Lumira Documents mode of the Lumira Designer.
- Applications can use composites from the same document and other documents. In order to use composites from another document, the referenced document must already exist. Therefore dependent documents need to be considered during transportation of objects (promotion), for example.
- One composite can be part of multiple applications. One application can contain multiple composites even if they are of the same type. Composites cannot contain other composites.

4.4.1 Creating a Composite

You create composites in the SAP Lumira Documents mode to group a subset of an application (for example, a header or repeating elements in your applications) or make the subset reusable so it can be shared with other application designers.

Prerequisites



You need to fulfill the following prerequisites before you can create composites:

- You work in the Lumira Documents mode of your Lumira Designer.
- You have created a local or remote document for which you want to create a composite.

Procedure

1. In the *Documents* view of the design tool, right-click a document in the *Local* or *Remote Documents* folder and choose *Create Composite...*. The *New Composite* dialog box is displayed.
2. Type in a name for the composite you want to create and choose *Create*. A new tab with the name of your composite is displayed in the editor.

Now you can start editing your composite.

3. After editing your composite, save your work by choosing  *File*  *Save* in the menu or by clicking on the *Save* symbol.

4.4.2 Editing a Composite

Editing the content of a composite works in the same way as for applications. Components can be added by dragging them from the *Components* view or by using the context menu in *Outline* View. Data sources can be added by using the *Add Data Source* button in the toolbar or by using the context menu in *Outline* view. All components and data sources are available within this composite only.

Context

To edit a composite, open it by double clicking it or choosing *Open* in the context menu of the composite in the *Documents* view.

Procedure

1. You can start editing your composite by
 - adding components from the *Components* view
 - adding data sources in the *Outline* view or from the menu bar
 - setting properties in the *Properties* view
 - adding an interface with properties, events and functions in the *Outline* view
 - adding global variables in the *Outline* view
 - adding technical components in the *Outline* view
2. After editing your composite, save your work by choosing ► *File* ► *Save* ► in the menu or by clicking on the *Save* symbol.

4.4.2.1 About Composites

When you use composites, note the following.

Specific behavior of some components when used in composites

- Context Menu
Defined *Context Menu* (technical) components only affect the containing application or composite. This means if there are context menu components defined in the application and in a composite of the application, then a different context menu may appear at runtime, depending on the component where the context menu was opened.
- Message View
There is only one *Message* view, which is configured by the application.
- Global Variables
Composites can contain *Global Variables* like applications. These variables are only available within this composite. At runtime they do not interfere with variables that have the same name in the application, which uses these variables or other instances of this composite.

Settings inherited from the Application

All the properties set for the application are global and cannot be configured or influenced inside a composite, especially not:

- *Theme*, *SAP UI 5 m Mode* and *Compact Form Factor*
- all *Message* view related properties like *Displayed Message Type*
- all prompt-specific settings like *Merge Prompts*
- all planning-specific settings like *Planning Connection*

However, it is possible to define an additional *Custom CSS* or a specific *On Initialization* script for a composite. See section "General Composite Properties".

4.4.2.2 Properties of the Composite

Like any other component, composites have the following properties, which can be edited in the [Properties](#) view:

General Properties

Property	Value	Description
Name	String	Displays the composite name that you entered when creating the composite. This name will be the type of the composite instance in the application where it is used.
Type	String	Specifies the object type, which the properties belong to. For a composite, the type is Composite .
Title	String	Specifies the text which should be displayed for this composite, for example, in the Components view.
Description	String	Specifies more information on this composite and will be displayed for example as tool tip in Components view.
Content Version	numeric value	Specifies the content number of the composite. This number relates to the release of the Lumira Designer and cannot be changed
Creation Time	numeric value	Displays the time stamp when the composite was created.
Created By	UserName	Displays the user who created the composite.
Last Modification Time	numeric value	Displays the time stamp of the last change.
Last Modified By	UserName	Displays the user who made the last change.
Last Edited With	numeric value	Displays the software tool, which was last used to edit the composite. If the changes have been made with Lumira Designer, the composite cannot be edited any more in Lumira Discovery.

Display Properties

Property	Value	Description
Custom CSS	CSS style	<p>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.</p> <div> <p>! Restriction</p> <p>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.</p> </div> <div> <p>i Note</p> <p>When executing an application with a composite in the browser, all components will be in the same browser DOM. Therefore it is recommended to define the CSS selectors specific to the composite, in order to not disturb content of the outer application and other composites.</p> </div>
Component Group	String	Specifies the name of the group where the composite is listed in the Components view and context menu in the Outline view.
New Instance Prefix	String	<p>Defines the name pattern to be used for composite instances added to an application. Composite instances added to an application are assigned a name like:</p> <p><New Instance Prefix>_<number>. By default, the prefix is the name of the composite.</p>

Events

Event	Description
On Initialization	<p>This event is executed when the composite is initially loaded (similar to the On Startup event of an application).</p> <p>Opens the script editor. With this property/event, you can enable user interaction with the application by writing scripts. Click <code>CTRL</code> + <code>Space</code> to see the list of available methods for the application, data source alias and the components. Choose one of these methods.</p>
On Property Change	<p>This event is executed whenever an Interface Composite Property changes (see Interface Composite Properties under Interface of the Composite [page 61]). Changing the value of a custom composite property in scripting will not trigger this event.</p> <p>Opens the script editor. With this property/event, you can enable user interaction with the application by writing scripts. Click <code>CTRL</code> + <code>Space</code> to see the list of available methods for the application, data source alias and the components. Choose one of these methods.</p>

Layout Properties

Property	Value	Description
Width	numeric value or auto	Specifies the width of a composite. Enter the numeric value in pixels, or set the value to <code>auto</code>
Height	numeric value or auto	Specifies the height of a composite. Enter the numeric value in pixels, or set the value to <code>auto</code>
Fixed Height	false, true	Specifies if the height is fixed so that it cannot be modified in the application using the composite.
Fixed Width	false, true	Specifies if the width is fixed so that it cannot be modified in the application using the composite.

4.4.2.3 Interface of the Composite

The composite interface specifies the specific behavior of your composite.

You can define properties, events and functions for the interface. To do this, proceed as follows:

1. In the [Outline](#) view of your composite, navigate to the Interface folder.

2. Select via the context menu of the corresponding folder (*Properties*, *Events* or *Functions*) *Create Property*, *Create Event* or *Create Function*. The element you have created is displayed in the corresponding folder.
3. Click on the new property, event or function. The property sheet for this element is updated and you can edit it.

Properties

Interface properties allow the users of your composite to configure their instances. These properties will be displayed in the *Property* sheet for a selected composite instance in an application. For each defined property, you can set the following:

- The *Name* property defines the technical name of the property. The property will be exposed with this name as a variable in scripting like a global variable.
- The *Title* property defines the readable name. This will be used as name in the *Property* Sheet.
- In the *Description* property more details can be described. This text is used as a tooltip in *Property* sheet, for example.
- The *Type* property defines the expected type of the property. This is used for type checking in scripting (for example, a property of type "boolean" should not be assigned by "String" values) or for using a meaningful input help in *Property* sheet, for example a color picker for properties of the type "Color".
- The *Default Value* property specifies the value to be used if there is no value set for this property of this composite instance.
- The *Bindable* property can be used to specify if a property can be bound by a property binding.

Properties of the type "DataSourceAlias"

Properties of the type "DataSourceAlias" have an additional behavior. They can be used like a data source by data bound components. This means you can choose any interface property of type "DataSourceAlias" as the data source property of a data bound component. This allows you to forward a data source, which exists in the application, to a composite, without the need to define it inside the composite again.

Functions

Interface functions define the script functions of your composite. These can be called in an BIAL script on an instance of this composite. For each function, you can set the following properties:

- The *Name* property specifies the technical name of this function. The function will be exposed with this name on a variable of the type of this composite instance.
- The *Title* property defines the readable name. This will be used as name in the *Property* sheet.
- In the *Description* property, more details can be described. Your input is displayed as text when hovering over this function in the script editor.

To edit an interface function, open the *Edit Script Function* dialog box, by double-clicking or choosing *Edit...* from the context menu of the function. In the *Edit Script Function* dialog box you can define the *Return Type*,

[Input Parameters](#) and the [Script](#). The [Return Type](#) defines the type of the value, which is returned by this function.

In the [Input Parameters](#) section you can define the arguments required when calling this function. The order of these parameters also defines the required order to specify these arguments when calling this function. An [Input Parameter](#) has a name which is the technical name of this parameter. The parameter is exposed with this name as variable in the script. The description for this parameter is displayed in the documentation of the function, for example, when hovering over it in the [Script Editor](#) dialog box. The type specifies the expected type of the parameter. When this function is called, the system runs a check to determine whether this given value matches the expected type. The business logic of the function is defined in the [Script](#) section.

Events

Composite events allow the application designer to react to defined actions of this composite. This is the same as as an event for standard components like the [On Click](#) event of the [Button](#) component, for example. These events are exposed as an event property in the [Property](#) view of an instance of this composite. The value help for this property/event opens the script editor to implement the script that is executed when this event occurs.

The [Name](#) property specifies the technical name of this event. More details can be described in the [Description](#) property. This will be used as a tooltip when hovering over this event type in a script editor.

The event can be triggered by calling the `fireEvent` method on the `COMPOSITE` variable, for example: `COMPOSITE.fireEvent("MyEventName")`. Triggering the event will execute the event handler script, which is defined in the instance of this composite for this event.

Example: You have a composite containing a [Button](#) component and you want to allow the user of this composite to react to clicks on this button.

1. Create an event named `CompositeButtonClicked`.
2. Trigger the `CompositeButtonClicked` event on the [Button](#) component by implementing the event handler of the [Button](#) component as follows: `COMPOSITE.fireEvent("CompositeButtonClicked")`;

Users of the composite can implement the `CompositeButtonClicked` event handler in their instances as needed.

Interface Bindings

You can use interface properties without writing scripts, thanks to an additional type of property binding named [Interface Property Binding](#), which is available inside composites. Any bindable property inside a composite can be bound to an [Interface Property Binding](#) directly. It is possible to define a formatter function (like with data cell bindings). This can be used if the types of the interface property and targeted component property do not match.

Example: You have a [Text](#) component inside a composite, and you want it to display the value of a specific interface property.

1. Open the [Property](#) view for the [Text](#) component
2. Create a property binding by pressing the [Plus](#) symbol in the [Binding](#) column for the [Text](#) property.

3. Choose *Interface Property Binding* as *Type*.

4.4.3 Nesting Composites

You can nest composites one in another.

There is no restriction on the nesting level. However, you cannot drag a composite into itself, either directly or indirectly via some intermediate composite, as the run time will fail if such a circular reference exists.

The system tries to prevent the creation of circular references by showing the error message *Composites cannot be nested recursively*. However, circular references cannot be prevented in all cases.

Example: You drag composite B into composite A and the next day your colleague edits composite B and adds an instance of A to it. In this case all applications containing instances of composite A will fail to run since they now contain a circular reference.

4.4.4 Using a Composite

You can add composites to an application by dragging them from the *Composites* folder, *Components* view or using the context menu in the *Outline* view as you would do for any other component. In addition, you can also drag composites from the *Documents* view. Afterwards you can configure the composite like any other component in the *Property* sheet.

Common Properties

Some properties are common for all composites and are available on each composite instance in addition to the custom Interface Properties. The *Text* property specifies the readable name of the composite instance. As an example, this text is displayed in the *Prompts* dialog box when you are using unmerged prompts to identify variables/prompts of data sources. These are defined in the composite because there might be multiple data sources with the same name, but from different composites or composite instances of the same type.

Layout properties are the same as for all non-technical components. These can be configured for a composite type. See the *Layout Properties* section in the *Properties of the Composite [page 59]* chapter.

4.4.5 Lumira Stories

Lumira Stories are stored as composites inside a document. Such composites behave like any other composite. However when you edit a Lumira story in Lumira Designer it becomes read-only in Lumira Discovery and cannot be edited there anymore.

4.5 Creating Analysis Applications for Mobile Devices

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

- Keep the applications simple and do not use too many components.
- Do not use crosstab components for smartphone applications.
- To improve performance and user experience for tablet applications that contain a crosstab component, SAP recommends the following:
 - limit to 500 the total number of cells in your crosstab.
 - your crosstab should contain, for example, a maximum of 50 rows and 10 columns for pixel-based scrolling.
- When running an application in the SAP BusinessObjects Mobile application, design smartphone applications in portrait format and design tablet applications in landscape format. You could also use the *Adaptive Layout* container component when designing an application. It allows you to create an adaptive application that can be viewed on multiple devices. For more information, you can refer to the chapter called *Working with the Adaptive Layout*.
Lumira Designer uses the **Main** part of the SAPUI5 library. The Main (m) part was developed with the mobile use case in mind. Those components are therefore specialized for mobile devices. When you select to create an analysis application, you also have to set the application property *Compact Form Factor true* or *false*. This allows you to adjust your visualization between two form factors: the **compact** form factor for the desktop and the **cozy** form factor with more spacing and padding for mobile use cases.

i Note

- For more information about Android version support, see the SAP Product Availability Matrix: <https://support.sap.com/release-upgrade-maintenance/pam.html>
- Refer to the following SAP Note:

SAP Note Number	Description
2461669	Considerations when viewing SAP Lumira Designer Applications on SAP BusinessObjects Mobile client application (MOBI)

Related Information

[Working with the Adaptive Layout \[page 206\]](#)

5 Creating the Layout of an Analysis Application

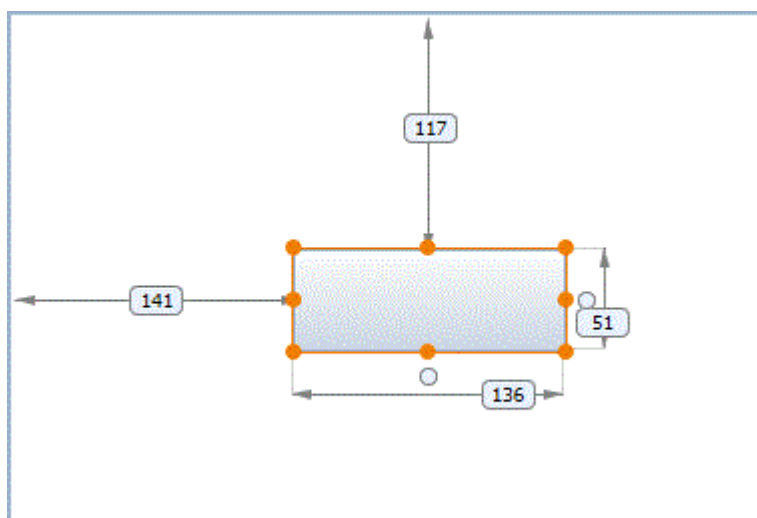
You create the layout of an application by inserting components in the editor, changing the properties of the components and arranging the components within an application (by using container components). Most of the work is performed in the layout editor of the design tool. The editor provides a What-You-See-Is-What-You-Get 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). The 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 \[page 69\]](#)

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 (adaptive layout, dialog, grid layout, pagebook, panel, popup, scroll container, story, tabstrip, pagebook) in the editor, you can place analytic components, basic components and other container components into the (first) container component.

5.3 Working with Rectangle Selection

Rectangle selection allows you to select multiple components in the design tool using a combination of the [Alt](#) key and a mouse drag.

The rectangle selection works with every component that can be selected in the [Layout](#) area with a mouse selection. All components with at least one edge included in the selection are selected. When you select the [Alt](#) key, the mouse cursor changes to indicate that the rectangle selection mode is enabled.

6 Working with the Outline View

In the *Outline* view of the Lumira Designer, you see all available components and elements of an application such as 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. Keep the following principle in mind when designing an application and working within the *Layout* folder: When you place components in the *Layout* folder at the same level as siblings, the one at the lowest position of this level is in foreground of the application, whereas the sibling component above is in the background of the application. This principle is also valid for components that are listed as lower levels (children) or higher levels (parents) in the folder hierarchy. The children are in the foreground of the application, the parents in the background. This principle is important when you add transparent components to the front and thus hide the components behind it. In this case the user cannot interact with the background components at runtime and you cannot click on them in the editor at design time. To be able to work with a background component, click on it in the *Layout* folder.
- 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* SAP BW Integrated Planning, which enable you to enter data automatically.
- In the *Technical Components* folder, you will find the following elements:
 - Action Sheet
 - Application List
 - Authoring
 - Backend Connection
 - Bookmarks
 - Comments
 - Components
 - Conditional Formatting Settings
 - Context Menu
 - Document
 - Export AO
 - Export CSV
 - Export Excel
 - Export PDF
 - Global Scripts Object
 - Keyboard Shortcuts
 - Personalization
 - Text Pool
 - Timer
 - Value Help Settings

Related Information

[Using the Context Menu in the Outline View \[page 101\]](#)

[Using Planning Functions and Sequences \(Automated Planning\) \[page 196\]](#)

[Working with Global Scripts Objects and Global Script Functions \[page 378\]](#)

[Exporting to PDF \[page 169\]](#)

[Using the Backend Connection Component for Data Source Browsing \[page 393\]](#)

[Working with Real-Time Dashboards \[page 311\]](#)

[Modifying Value Help Settings for Dimensions \[page 98\]](#)

[Exporting to Microsoft Excel \[page 166\]](#)

6.1 Technical Components

6.1.1 Action Sheet

Action Sheet is a technical component.

The *Action Sheet* folder, you can add planning functions and can be scripted to contain an action list that opens next to another component. To activate the *Action Sheet*, call a scripting method from within the *onClick* event of a component, such as a button. This function takes the component it is rendered beside as a non-optional parameter. The *Action Sheet* is modal. It is closed when the application user clicks away from it. You can configure multiple instances of the *Action Sheet* technical component within an application. The *Action Sheet Items* property allows you to set the Value, Text and Icon to appear in your *Action Sheet*. The *Placement* property allows you to position the *Action Sheet* relative to another component. The placement options include the following:

- auto
- left
- right
- top
- bottom
- horizontal
- vertical

6.1.2 Application List

Application List is a technical component.

Use the *Application List* technical component to return a list of bookmarks created in other documents and applications. These bookmarks can be launched from within the current application, by selecting them from the list. When you select a bookmark from the list, the application that contains that bookmark is launched.

Related Information

[Working with an Application List \[page 204\]](#)

[Returning a Bookmark Url \[page 204\]](#)

[Returning a Bookmark List \[page 205\]](#)

6.1.3 Authoring

Authoring is a technical component.

The *Authoring* technical component allows you to offer application users the ability to change the layout of a dashboard created for them by you. Use the *Authoring Area* property of the *Authoring* technical component to select the *Composite* component in the application you want to make editable by the end user at runtime.

i Note

- The *Composite* component is the only component supported as an area in which authoring can be enabled.
- *Container Components* are not supported in the area enabled for authoring at runtime.
- Authoring is supported in SAP Lumira Documents mode only.
- The *Bookmarks* technical component is not supported in the area enabled for authoring.
- Before working with the *Authoring* technical component, refer to the following SAP Note:

SAP Note Number	Description
2654202 	Considerations when working with the Authoring technical component.

Related Information

[Properties of the Authoring Technical Component \[page 72\]](#)

[Using the Grouped List with the Authoring Technical Component \[page 288\]](#)

6.1.3.1 Properties of the Authoring Technical Component

The [Authoring](#) technical component has the following specific property:

Properties of the Authoring Technical Component

Property	Property Value	Property Description
authoring area	dropdown	Select the area in the application that is enabled for authoring. When the authoring area is set using the Authoring Area property, the application user can then interact with this authoring area at runtime. The area enabled for authoring can also be set through scripting, using the <code>getAuthoringArea()</code> and <code>setAuthoringArea()</code> scripting methods.

i Note

The [Composite](#) component is the only component supported as an area in which authoring can be enabled.

Related Information

[Authoring \[page 71\]](#)

[Using the Grouped List with the Authoring Technical Component \[page 288\]](#)

6.1.4 Backend Connection

[Backend Connection](#) is a technical component.

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.

Related Information

[Using the Backend Connection Component for Data Source Browsing \[page 393\]](#)

6.1.5 Bookmarks

Bookmarks is a technical component.

The technical component *Bookmarks* enables the application user to apply bookmark functionality to an application. This bookmark functionality serializes the state of an application at a certain point in time. Using scripting methods, or the properties of the technical component, you can define the elements of an application that are included in a bookmark definition.

Related Information

[Bookmarks Technical Component \[page 212\]](#)

6.1.6 Comments

Comments is a technical component.

The *Comments* technical component allows users to create and view comments in Lumira documents. To allow users to avail of this functionality, you must add the *Comments* technical component to the *Outline* view of the design tool. When the *Comments* technical component is added to the design tool, the scripting methods associated with the comments functionality are available in the script editor. Comments can be applied to a Lumira document in the following ways:

- on the entire document
- on a data cell
- on a dimension
- on a dimension member
- on a context

There are different contexts in which the comments apply. The different types of contexts include the following:

- DATA
- MEMBER
- DIMENSION
- CONTEXT
- NONE

Using the properties `context` and `contextType` on the *Comment* object, you can allow end users to create comments with different contexts, and then filter them, so the comments can be rendered separately by context. For more information you can refer to the topics called *Create Comments with Different Contexts*.

Related Information

[Properties of the Export PDF Technical Component \[page 171\]](#)

[Working with the Feed List \[page 284\]](#)

[Feed List \[page 478\]](#)

[Working with Comments \[page 229\]](#)

6.1.7 Components

Components is a technical component.

You can use the *Components* technical component to allow users to dynamically create components such as charts, basic components etc. at runtime. Users can also dynamically connect these components to a data source. To avail of this functionality you must add the *Components* technical component to the application.

6.1.7.1 Copy Properties

You can copy properties from one component to another.

Through scripting, you can copy properties from a source component to a target component. Using an optional parameter, you can allow users to have more granular control over which properties are not copied.

Sample Code

```
COMPONENTS.copyProperties(CHART_1, CHART_2, [KEEP_CURRENT_VALUES,  
KEEP_CHART_TYPE, KEEP_SCRIPTING])
```

The following table lists the optional parameters and their description:

Optional Parameter	Description
CopyPropertiesOptionsEnumfield KEEP_LAYOUT	The location, width and height of the component will be maintained.
CopyPropertiesOptionsEnumfield KEEP_CURRENT_VALUES	Any values that have been set on the component are maintained.
CopyPropertiesOptionsEnumfield KEEP_CHART_TYPE	The chart type will be maintained.
CopyPropertiesOptionsEnumfield KEEP_SCRIPTING	Any scripting that is on the component will be maintained.

For more information you can refer to the *Component API Reference* on the help portal at <https://help.sap.com/viewer/index>.

Note

- When using KEEP_CURRENT_VALUES, it is important to remember that if the destination chart has been manipulated, the result may not be as expected, especially when data changes.
- Colors are bound to the measure, and not contained within an individual chart type.

- With third party SDK components, `copyProperties()` may not work as expected.
- The following components are not supported:
 - Technical Components
 - Filter Bar

Related Information

[Sharing a Chart Color Palette across Multiple Charts \[page 250\]](#)

6.1.7.2 Get Components

You can use the [Components](#) technical component to get all the children within the scope of the application or composite.

Through scripting, you can return an array of all the components within the scope of an application or composite. For more information, you can refer to the *Component API Reference* on the SAP Help Portal at <http://help.sap.com>.

To give you an idea of how to apply this functionality, you can refer to these examples:

Application example

❖ Example

≡ Sample Code

```
// get all Icons in application
var icons = COMPONENTS.getComponents(ComponentType.com_sap_ip_bi_Icon);
//This is an array that can be looped over
icons.forEach(function(component, index) {
    // Cast to an ICON
    // Set the background color to green
    component.as(ComponentType.com_sap_ip_bi_Icon).setBackgroundColor("green");
});
```

Composite example

❖ Example

≡ Sample Code

```
// get all components in a the composite KPI_TILE
var componentArray = COMPONENTS.getComponents(ComponentType.ComponentBase,
KPI_TILE_1);
//This is an array that can be looped over
componentArray.forEach(function(component, index) {
    if (component.getComponentType() == ComponentType.com_sap_ip_bi_Icon)
    { //if it is an icon
        var icon = component.as(ComponentType.com_sap_ip_bi_Icon); // cast
to Icon
        // set the background color to green
        icon.setBackgroundColor("green");
    }
    else if (component.getComponentType() ==
ComponentType.com_sap_ip_bi_VizFrame){ // if it is a Chart (VizFrame)
        var vizFrame =
component.as(ComponentType.com_sap_ip_bi_VizFrame); // cast to VizFrame
        // set the chart type to be bar
        vizFrame.setType(ChartDisplayType.BAR);
    }
});
```

6.1.8 Conditional Formatting Settings

Conditional Formatting Settings is a technical component.

The technical component *Conditional Formatting Settings* allows you to apply a set of conditional formatting rules, including BEx levels exceptions, to components.

6.1.8.1 Working with Conditional Formatting Settings

The *Conditional Formatting Settings* technical component allows you to set conditional formatting rules on a chart or crosstab.

Using the *Conditional Formatting Rule Manager*, you can create a collection of conditional formatting rules that can be applied to an application. You can assign the rule name and the formatting to be applied to specific measures and dimensions. You can also apply multiple rules to one ruleset based on the measure or dimension you have selected. You can also select to include BEx level exceptions to the list of conditional formatting settings that can be applied to charts and crosstabs.

Using the *Show Conditional Formatting Items* and *Keep Original Legend Items* chart properties, you can decide which conditional formatting settings to display in the legend of a chart. These properties are in the *Legend* area of the *Properties* tab, within the *Configure Chart* dialog.

Using the *Conditional Formatting Visible* crosstab property, you can decide to display conditional formatting in the crosstab.

i Note

The settings selected in the *Conditional Formatting Settings* components applied to a crosstab or a chart, overrides the conditional formatting coming from the BEx query.

Related Information

[Conditional Formatting Settings \[page 76\]](#)

[Working with Conditional Formatting in Crosstabs \[page 254\]](#)

6.1.8.1.1 Properties of the Conditional Formatting Settings

The *Conditional Formatting Settings* technical component has the following properties:

Properties of the Conditional Formatting Settings Technical Component

Property	Property Value	Property Description
enable BEx exceptions	boolean	<ul style="list-style-type: none">Specifies whether conditional formats (exceptions) that have been enabled in the BEX Query Designer or in SAP Analysis for Microsoft Office (Analysis), are shown in the crosstab and chart.Specifies whether the option to include <i>BEx Levels</i> exceptions appears within the <i>Conditional Formatting Rule Manager</i> dialog. The <i>Conditional Formatting Rule Manager</i> dialog is called by selecting the ellipsis button of the <i>Rules</i> property.
rules	none	<p>Select the <i>Rules</i> property ellipsis button to call the <i>Conditional Formatting Rule Manager</i> dialog. This dialog allows you to do the following:</p> <ul style="list-style-type: none">create a new measure rule

Property	Property Value	Property Description
		<ul style="list-style-type: none"> • create a new dimension rule • allow BEx Levels exceptions to be displayed • edit an existing rule • duplicate a rule • delete a rule

Related Information

[Conditional Formatting Settings \[page 76\]](#)

6.1.8.1.2 Chart Reset Color Property Setting and Conditional Formatting

The *Color* property setting of the *Chart* takes higher priority over *Conditional Formatting Settings*.

The *Color* property setting of the *Chart* component can customize color settings down to the level of specific data points. The *Conditional Formatting Settings* technical component sets conditional formatting colors in more general way, in a chart or crosstab. The *Color* property setting takes higher priority over the *Conditional Formatting Settings*. We recommend that you use either one color setting, or the other.

If you want to set a default color on all bars of a chart, for example, we recommend that you create a conditional formatting rule that sets the color on is any value.

The *Reset Color* button in the *Chart* property tab, allows you to clear the *Color* property setting. Clearing the property setting allows the *Conditional Formatting Settings* color to take effect.

The types of *Chart* that have this *Reset Color* property are as follows:

- bar
- column
- stacked bar
- stacked column
- dual axis
- line
- line chart for time series
- area
- pie
- donut

6.1.8.1.3 Creating Rules for Conditional Formatting

Based on measures, dimensions and BEx exceptions, you can create different types of conditional formatting rules.

You can apply three different types of rules to your crosstab or chart:

- new measure rules
- new dimension rules
- BEx levels rules

Related Information

[Conditional Formatting Settings \[page 76\]](#)

[Properties of the Conditional Formatting Settings \[page 77\]](#)

[Creating a New Measure Rule \[page 79\]](#)

6.1.8.1.3.1 Creating a New Measure Rule

You can create a conditional formatting rule based on a measure in a chart or crosstab.

Context

You have added the [Conditional Formatting Settings](#) technical component to an application and the [Conditional Formatting Visible](#) property is set to true for the crosstab or chart.

Procedure

1. Select the ellipsis button beside the [Rules](#) property.
The [Conditional Formatting Rule Manager](#) dialog appears.
2. Select the [New Measure Rule](#) button to call the [New Measure Rule](#) dialog.
3. Give the measure rule a name by populating the [Name](#) free text field.
4. Select the measure on which the rule is to be based from the [Based On](#) field.
5. Select from one of the following options to define the rule to apply to the selected measure: greater than, greater than or equal to, less than, less than or equal to, equal to, not equal to, between, not between, in list, not in list, is null, is not null, or is any value.
6. Select a fill color to define the color to apply to the rule you are creating.

7. Select a text formatting style from the list of options provided, including font, bold, italics, and color. These settings can be applied to crosstabs or charts.
8. Select to display a symbol in a crosstab only. The options available include: none, status symbol, trend ascending, trend descending, or trend gray.
9. You can also apply multiple filters to one rule based on the measure you have selected. To create another filter within the same rule, simply select the + symbol, and repeat steps 5 to 9.
10. Select *OK*.

The new measure rule is listed in the *Conditional Formatting Rule Manager* dialog.

11. Select *OK* again.
12. Select the ellipsis button within the *Conditional Formatting Settings* property of the chart or crosstab, to call the *Conditional Formatting Rule Manager* dialog again.
13. Select the checkbox called *Applied* and select *OK*.

The conditional formatting rule is applied to the crosstab or chart. Also the rule applied is listed in the value field of the *Conditional Formatting Settings* property of the crosstab or chart.

Related Information

[Creating a New Dimension Rule \[page 80\]](#)

[Creating a BEx Level Exceptions Rule \[page 81\]](#)

6.1.8.1.3.2 Creating a New Dimension Rule

You can create a conditional formatting rule based on a dimension in a chart or crosstab.

Context

You have added the *Conditional Formatting Settings* technical component to an application and the *Conditional Formatting Visible* property is set to true for the crosstab or chart.

Procedure

1. Select the ellipsis button beside the *Rules* property.

The *Conditional Formatting Rule Manager* dialog appears.
2. Select the *New Dimension Rule* button to call the *New Dimension Rule* dialog.
3. Give the dimension rule a name by populating the *Name* free text field.
4. Select the dimension on which the rule is to be based from the *Based On* field.

5. Select from one of the following options to define the rule to apply to the selected dimension: greater than, greater than or equal to, less than, less than or equal to, equal to, not equal to, between, not between, in list, not in list, is null, is not null, or is any value.
6. Select the field marked with the text [Click to Open List of Values](#).
A dialog appears offering you the list of items from which to choose, based on text and key.
7. Select the appropriate checkboxes.
The section called [Selected Items](#) is populated with the selections you make.
8. Select [OK](#).
9. Select a text formatting style from the list of options provided, including font, bold, italics, and color. These settings can be applied to crosstabs or charts.
10. Select a fill color to define the color to apply to the rule you are creating.
11. You can also apply multiple filters to one rule based on the dimension you have selected. To create another filter within the same rule, simply select the + symbol, and repeat steps 5 to 10.
12. Select [OK](#).
The new dimension rule is listed in the [Conditional Formatting Rule Manager](#) dialog.
13. Select the checkbox called [Applied](#) and select [OK](#).
The conditional formatting rule is applied to the crosstab or chart. Also the rule applied is listed in the value field of the [Conditional Formatting Settings](#) property of the crosstab or chart.

Related Information

[Creating a New Measure Rule \[page 79\]](#)

[Creating a BEx Level Exceptions Rule \[page 81\]](#)

6.1.8.1.3.3 Creating a BEx Level Exceptions Rule

You can create a conditional formatting rule based on BEx level exceptions.

Context

You have added the [Conditional Formatting Settings](#) technical component to an application. The property [Conditional Formatting Visible](#) is set to true for your chart or crosstab.

Procedure

1. Set the [Enable BEx Exceptions](#) property of the [Conditional Formatting Settings](#) technical component to true.

If the [Enable BEx Exceptions](#) property is set to false, the option to select BEx level exceptions is not available in the [Conditional Formatting Settings](#) dialog.

2. Select the ellipsis button beside the [Rules](#) property.

The [Conditional Formatting Rule Manager](#) dialog appears, and the [BEx Levels](#) button is available.

3. Select the [BEx Levels](#) button.

The [BEx Exception Formatting Rule](#) dialog appears.

4. In the [Defined Rules](#) section, select the + icon to expose the other properties.
5. Select the BEx level exception level you want to use from the options in the drop down.
6. Select a text formatting style from the list of options provided, including font, bold, italics, and color. Select a fill color to define the color to apply to the rule you are creating.
7. Select to display a symbol in a crosstab only. The options available include: none, status symbol, trend ascending, trend descending, or trend gray.
8. You can apply multiple conditional formatting settings to different BEx exception levels. Simply select the + symbol in the [Defined Rules](#) section, and repeat steps 5 to 7.
9. Select [OK](#).

The new BEx level exceptions rule is not listed in the [Conditional Formatting Rule Manager](#) dialog. The BEx level exceptions rules are applied.

Related Information

[Creating a New Dimension Rule \[page 80\]](#)

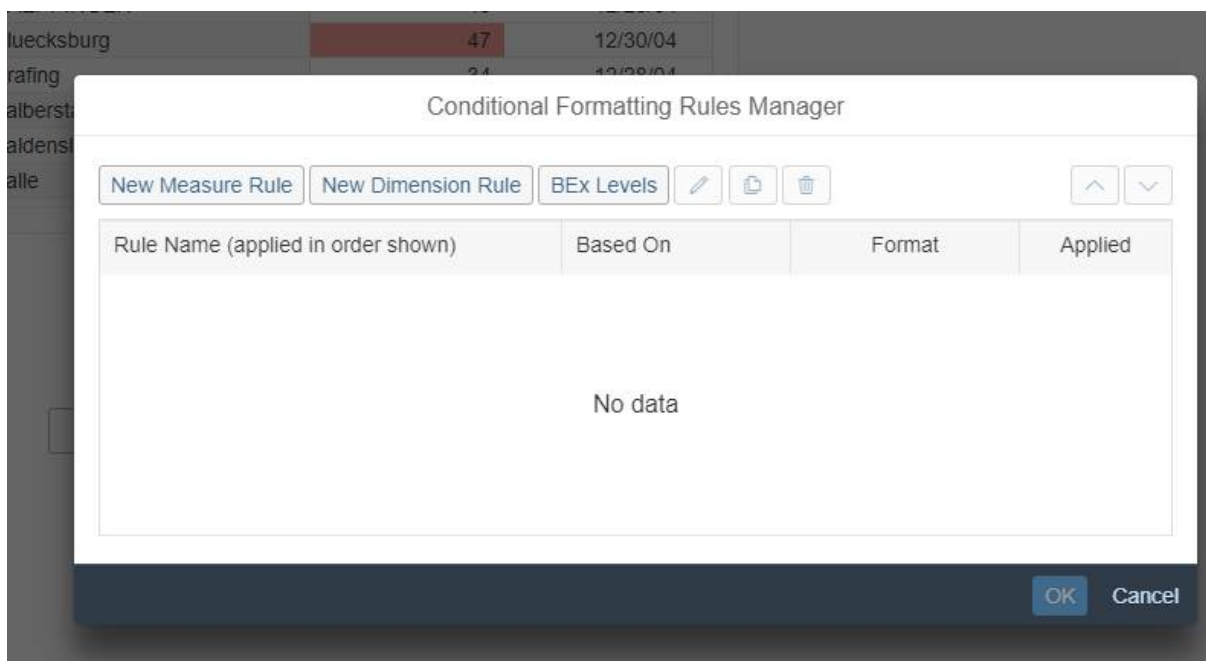
[Creating a New Measure Rule \[page 79\]](#)

6.1.8.1.4 Using BEx Exception Manager in Conditional Format Settings Dialog

If an application contains a [Conditional Formatting Settings](#) technical component and you have activated the [Enable BEx Exceptions](#) property of this component and your data source contains BEx exceptions, you can

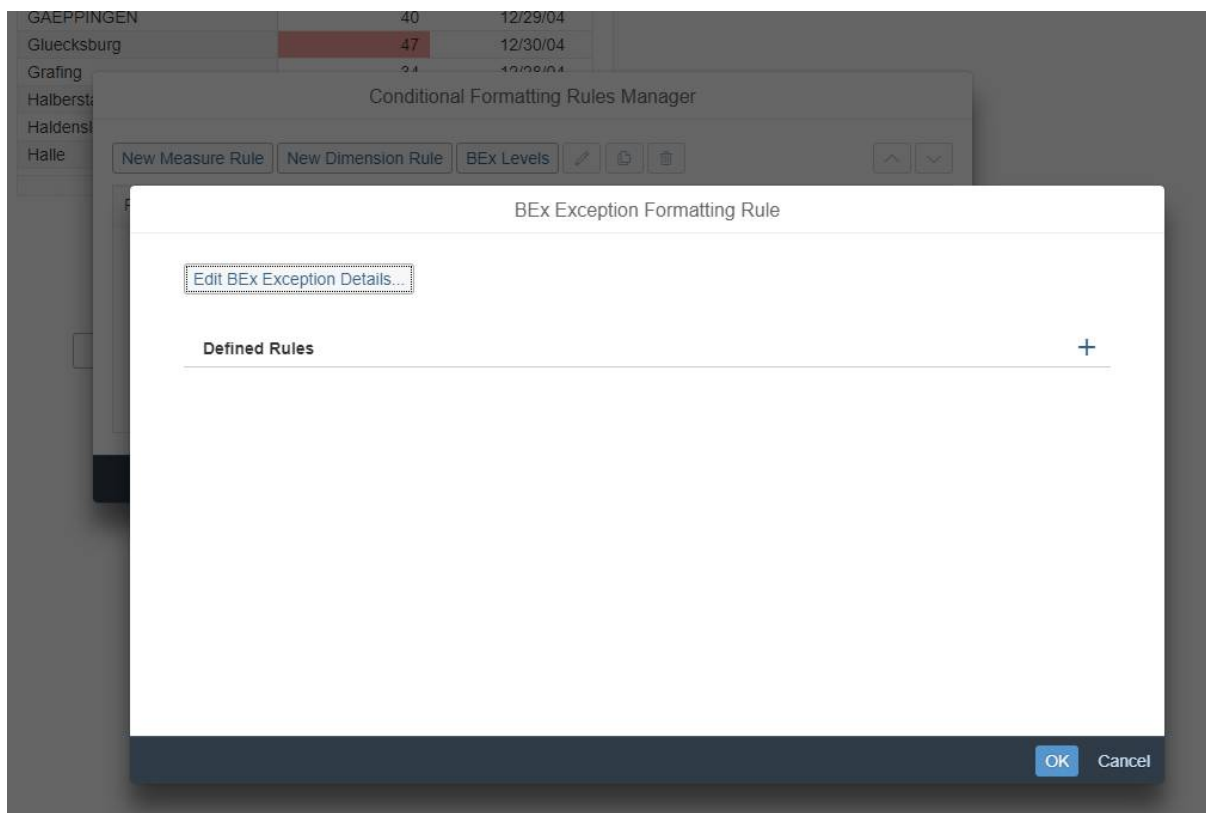
open the Business Explorer (BEx) Exception Manager via the crosstab context menu entry [Conditional Formatting](#).

If you have added a [Conditional Formatting Settings](#) technical component to your application, the context menu entry [Conditional Formatting](#) will show the [Conditional Formatting Rules Manager](#).

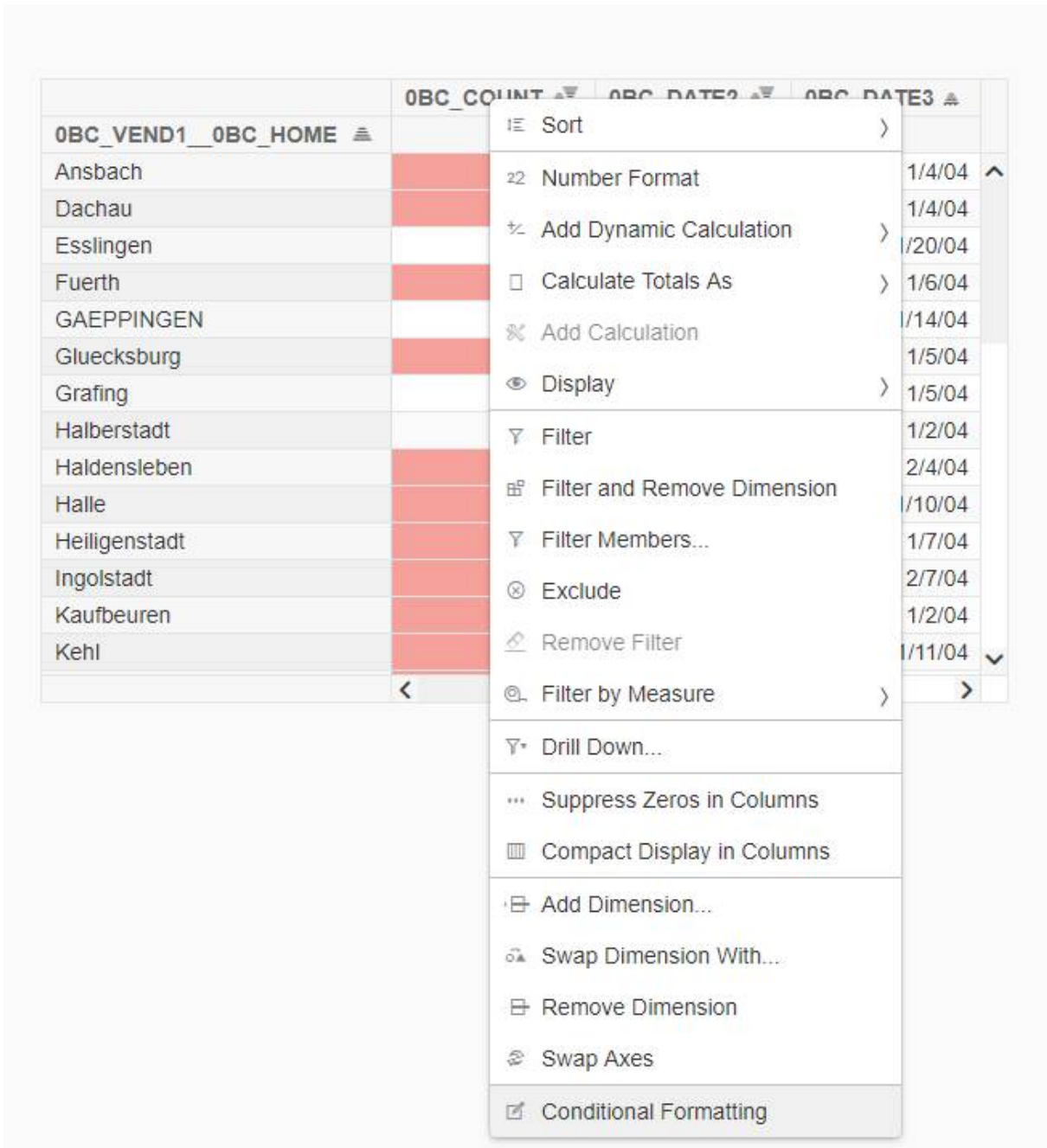


To configure BEx Exceptions, make sure that the property [Enable BEx Exceptions](#) of the [Conditional Formatting Settings](#) technical component is set to [true](#).

Clicking the [BEx Levels](#) button in the [Conditional Formatting Rules Manager](#) shows the dialog [BEx Exception Formatting Rule](#). In this dialog you can overwrite the visual formatting for certain BEx Exception levels. And you can bring up the [BEx Exception Manager](#) by clicking the [Edit BEx Exception Details](#) button.



If you have not added a [Conditional Formatting Settings](#) technical component to your application and the data source contains BEx exceptions, the [BEx Exception Rule Manager](#) opens directly when you click [Conditional Formatting](#) in the context menu of the crosstab.



6.1.9 Context Menu

Context Menu is a technical component.

The *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.

6.1.9.1 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](#).

The technical component [Context Menu](#) has the following properties:

Property	Value	Description
custom menu options	your entries	Specifies own menu options. In the Edit Menu Entries dialog box, you can enter own menu entries and sub menus, change their order and by using the script editor assign adequate script methods to the menus.
maximum member of members	numeric value (default: 100)	Specifies the maximum number of members displayed in a value help for non-hierarchical dimensions.
value help settings	technical name of Value Help Settings technical component	Specifies the technical component Value Help Settings which you can use to define the value help settings for dimensions.

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](#), [▶ Select Multiple Totals... ▶ Result ▶ Average/Sum/Count/Minimum/Maximum](#)).
 - filter dimension members
 - open the filter to filter the dimension ([Filter Members](#))
 - clear the current filter ([Select All Members](#))

- define filters to get the TOP N or Bottom N values of a specified dimension based on the measure values (*Filter my Measure*)
 - 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
 - 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*)
 - lock/unlock row and lock/unlock column for input-enabled crosstabs (*Lock Row/Unlock Row* and *Lock Column/Unlock Column*)
- 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*)
 - add a calculation or change the totals calculation mode
 - add a new calculation based on one available measure (*Add Dynamic Calculation* > *[Operator]*) or a calculation dialog where the user can specify the calculation).
 - change the totals calculation mode (*Calculate Totals As* > *[Operator]*)
 - add a calculation by using two or more available measures as operands and an operator (*Add Calculation* > *[Operator]*)
- 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 \[page 386\]](#)

[Calculating New Measures at Runtime \[page 88\]](#)

6.1.9.1.1 Calculating New Measures at Runtime

Based on measures that are available in your crosstab, you can calculate new measures. There are two types of calculations.

For simple calculations, you use two or more available measures as operands and an operator to create a new measure. The new measure is the sum of two available measures, for example.

For dynamic calculations, you only use one available measure as an operand and you create a new measure based on this operand. For example, if you use sales volume per region as the operand, you can add a new measure that displays the rank of each region according to sales volume. If you now filter out or add new regions to your analysis, the rank numbers are changed dynamically. These calculations are therefore called dynamic calculations. The newly created measures are added to the crosstab. You can edit the name and delete measures by using the context menu for these measures.

Depending on the data source you can also use calculations that are based on dialog boxes where you can specify your calculation.

Related Information

[Adding \(Simple\) Calculations \[page 88\]](#)

[Adding Dynamic Calculations \[page 89\]](#)

6.1.9.1.1.1 Adding (Simple) Calculations

Context

By adding a (simple) calculation, you can create a new measure based on two or more available measures in your crosstab. The menu entry *Add Calculation* in the context menu of the crosstab is only available if the *Selection Type* property of the crosstab component has been set to *Multi* at design time.

Procedure

1. Select the measure headings that you want to use.

The first measure that you select is the first operand. Use the **CTRL** key to select the next measure that is the second operand. You can also select more than two operands for your calculation.

2. In the context menu, choose **Add Calculation** > **[Operator]**.

The following operators are available:

- Add
- Subtract
- Multiply
- Divide
- Percentage Difference

The new measure calculates the difference between operand 1 and operand 2 as a percentage. For example, operand 1 has the value 80 and operand 2 has the value 20. The new measure displays the value 300%, as the difference is 60 and 60 is 300% of 20. If operand 2 has a higher value than operand 1, the result is negative. For example, operand 1 has value 20 and operand 2 has value 80. The new measure displays the value -75%, as the difference is -60 and -60 is -75% of 80.

- Percentage Share

The new measure calculates the share of operand 1 in comparison with operand 2 as a percentage. For example, operand 1 has the value 20 and operand 2 has the value 80. The new measure displays the value 25%, as 20 is 25% of 80.

Results

The new calculated measure is added to the crosstab.

A measure value displayed with sign reversal is used in calculations as displayed.

❖ Example

A measure value is 200. If you have selected the sign reversal property in the query definition, the value is displayed as -200 in the crosstab. For calculations in Lumira Designer, the value 200 is used. If you select this checkbox, the value -200 is used for calculations.

6.1.9.1.1.2 Adding Dynamic Calculations

Context

By adding a dynamic calculation, you can create a new measure based on one measure in your crosstab that works as operand. Or depending on the data source and context you can use calculations that are based on dialogs.

Procedure

1. Select the measure heading that you want to use.
2. Choose ► *Add Dynamic Calculation* ► [Operator] ►.

The following operators are available:

Operator	Description
<i>Moving Minimum Value</i>	<p>The new measure displays the lowest value available up to this point.</p> <p>For example, there are ten rows with values in your cross-tab. The new measure starts in the first row with the same value as the original measure. If the second row in the original measure has a higher value than the first, the value of the first row is repeated in the second row of the new measure and so on.</p>
<i>Moving Maximum Value</i>	<p>The new measure displays the highest value available up to this point.</p>
<i>Accumulative Sum</i>	<p>The new measure displays the sum of all values up to this point.</p> <p>For example, the original measure has eight rows with values. In the fourth row of the new measure, the sum of first four rows is displayed.</p>
<i>Accumulative Sum of Rounded Values</i>	<p>The new measure displays the sum of all rounded values up to this point.</p>
<i>Accumulative Count of All Values</i>	<p>The values per measure are counted and numbered starting with 1 for the first value. If there are 8 rows with values, including zeros, which contribute to the total, the new measure displays the numbers 1 to 8 for the 8 values.</p>
<i>Accumulative Count of All Values that are Not Zero, Null or Error</i>	<p>The values per measure are counted and numbered starting with 1 for the first value, excluding values that are equal to zero.</p> <p>For values that are equal to zero, the last number is repeated.</p>
<i>Moving Average</i>	<p>The new measure calculates the average of all values up to this point.</p> <p>For example, if there are five rows with values in the cross-tab, the new measure calculates in row 2 the average of the values in row one and two, and so on.</p>

Operator	Description
<i>Moving Average that is Not Zero, Null or Error</i>	The new measure calculates the average of all values up to this point, excluding values that are equal to zero.
<i>Rank Number</i>	<p>The new measure displays a rank number for each value of the original measure. The highest value in the original measure has the rank number 1.</p> <p>If a value occurs more than once, the values are assigned to the same rank number. If there are two identical values with rank number 4, the next smallest value has rank number 5.</p>
<i>Olympic Rank Number</i>	<p>The Olympic ranked list differs from the basic ranked list as follows: In the Olympic ranked list, when a value occurs more than once, the next lowest value is not assigned the rank incremented by one, but the rank that corresponds to the number of previous values (including the current value).</p> <p>For example, if the rank 4 occurs twice, the new measure displays number 6 for the next lowest value rank.</p>
<i>Percentage Contribution</i>	The new measure calculates the percentage contribution of a value in the original measure to the overall result of the original measure.

Depending on the data source and context you can also use the following calculation dialogs:

- Counter
- Running Calculations
 - Average
 - Count
 - Minimum
 - Maximum
 - Sum
- Moving Average
- Percentage Of
- Difference From
- Custom Calculation

Results

The new measure is added to the crosstab.

6.1.9.1.2 Creating Filters by Measure

By using the *Filter by Measure* entry in the context menu on a dimension in the crosstab, you can define filters to get the Top N or Bottom N values of a specified dimension, based on their measure values.

This means the filter is applied to the members of the selected dimension and does not affect totals or subtotals in your crosstab.

Filtering measures is a dynamic action. Whenever you change the view of your data, the filter is applied again. For example, if you add a Top 5 filter, five members are shown in your analysis. If you then add members that you previously removed from the crosstab, some of these newly added members could match the Top 5 criteria and replace some of the previously displayed members.

Prerequisites

This function is only available for SAP BW data sources or data sources based on SAP HANA HTTP connections via SAP HANA Info Access Service (InA).

SAP BW data sources	SAP HANA data sources
<p>The menu entry <i>Filter by Measure</i> is only available in the context menu if there is only a measure structure contained in the query.</p> <p>Therefore Filter by Measure is not available in the following scenarios:</p> <ul style="list-style-type: none">queries with two structuresqueries with no structurequeries with no structure and a measure restriction in the fixed filter (selection space)queries with one non-measure structure (and a measure restriction in the fixed filter)	<p>SAP HANA data sources (via InA / HTTP connection) only offer a restricted feature set:</p> <ul style="list-style-type: none">Only one measure based filter is supported per data source. Therefore the menu entry is automatically deactivated as soon as the data source has one measure-based filter and will only be activated again if this filter is removedIf one dimension in the drilldown (rows or columns) contains an active hierarchy, measure-based filters are not supported. Therefore this menu entry is deactivated.

Creating a measure-based filter

1. Right-click on a dimension header or dimension member in your crosstab and choose *Filter by Measure*. A new dialog box opens.
2. In the dialog box, perform the following steps:
 - Choose the measure that you want to use as the basis for filtering.
 - Choose the operator (*Top N* or *Bottom N*).
 - Enter a value for the operator (filter criteria), for example, 5 for Top 5.
 - Choose *OK*.

A new measure-based filter is created for the dimension you have chosen. This means that the filter only filters the values for this dimension, for example, you filter the Top 5 customers based on their revenue by choosing the customer dimension.

If you have additional dimensions in the drilldown, they are not filtered.

Editing a measure-based filter

1. Right-click on a dimension header or dimension member in your crosstab.
2. Chooses ► [Filter by Measure](#) ► [Edit](#) ► to edit an existing measure-based filter. A new dialog box opens.
3. Choose the measure, the operator and value for filter criteria according to your needs.
4. Choose OK. The measure-based filter is changed according to your input.

Deleting a measure-based filter

1. Right-click on a dimension header or dimension member for which you have created a measure-based filter.
2. Choose ► [Filter by Measure](#) ► [Clear](#) ► to remove the measure-based filter that you have created for this dimension.

If you have multiple filters defined for multiple dimensions, you need to repeat this procedure for every dimension that contains a measure-based filter.


(De)activating predefined conditions

Measure-based filters that were created in another tool (for example, BEx conditions created in BEx Query Designer or filters created in SAP BusinessObjects Analysis, edition for Microsoft Office) can only be (de)activated in Lumira Designer.

1. Right-click on a dimension header or dimension member in your crosstab.
2. Choose [Predefined Measure Filters](#) and click on any of the listed filters to activate or deactivate them.

6.1.10 Document

Use the [Document](#) technical component to save a Lumira document to the BI platform.

We recommend you use the [Document](#) technical component with the [Authoring](#) and [Components](#) technical components only. For more information, you can refer to the following SAP Note: [2654202](#) .

6.1.10.1 Properties of the Document Technical Component

[Document](#) is a technical component.

Properties of the Document Technical Component

Property	Property Value	Property Description
on dataset refresh complete	ellipsis	Select the ellipsis button to call the Script Editor dialog. You can script for an event that is triggered when the dataset refresh is completed.

6.1.11 Export AO

[Export AO](#) is a technical component.

You can select a data source view from within an application and open that same data source view in the Analysis Plug-in for SAP Analysis for Microsoft Office. This is done through the [Export AO](#) technical component. The [Export AO](#) technical component generates an SAP Analysis for Microsoft Office launcher file. The launcher file is then opened in the Analysis desktop client. The data source connection details and data source state are applied to the Analysis workbook.

6.1.12 Export CSV

[Export CSV](#) is a technical component.

All Lumira Designer export to CSV file format functionality is consolidated into the technical component called [Export CSV](#). To access export to CSV functionality simply add the [Export CSV](#) technical component to an application. To add this technical component, select [Export CSV](#) from the context menu of the [Technical Components](#) area of the [Outline](#) view of an application.

The [Export CSV](#) technical component contains:

- A set of scripting methods that allows you to enable the application user to configure how the [Crosstab](#) is exported to a CSV file format.
- A set of [Properties](#) which define all characteristics of your export to CSV. For more information about the properties available, you can refer to the chapter called *Properties of the Export CSV Technical Component*.

6.1.13 Export Excel

[Export Excel](#) is a technical component.

All Lumira Designer export to Microsoft Excel functionality is consolidated into the technical component called [Export Excel](#). To access export to Microsoft Excel functionality simply add the [Export Excel](#) technical component to an application. To add this technical component, select [Export Excel](#) from the context menu of the [Technical Components](#) area of the [Outline](#) view of an application.

The [Export Excel](#) technical component contains:

- A set of scripting methods that allows you to enable the application user to configure how a crosstab or a data source is exported to Microsoft Excel.
- A set of [Properties](#) which define all characteristics of your export to Microsoft Excel. For more information about the properties available, you can refer to the chapter called *Properties of the Export Excel Technical Component*.

Related Information

[Properties of the Export Excel Technical Component \[page 167\]](#)

[Exporting to Microsoft Excel \[page 166\]](#)

6.1.14 Export PDF

[Export PDF](#) is a technical component.

All Lumira Designer export to PDF functionality is consolidated into a technical component called [Export PDF](#). To access export to PDF functionality simply add the [Export PDF](#) technical component to an application. To add this technical component, select [Export PDF](#) from the context menu of the [Technical Components](#) area of the [Outline](#) view of an application.

The [Export PDF](#) technical component contains:

- A set of scripting methods that allows you to enable the user to configure the export to PDF properties. Through scripting you can to export to PDF in the following output formats:
 - View
 - Report
 - Panel View
- A set of [Properties](#) which define all characteristics of the exported PDF file. For more information about the [Export PDF](#) technical component , you can refer to the chapters called [Properties of the Export PDF Technical Component \[page 171\]](#) and [Exporting to PDF \[page 169\]](#).

6.1.15 Global Scripts Objects

Global Scripts Objects is a technical component type.

The *Global Scripts Objects* 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.

Related Information

[Working with Global Scripts Objects and Global Script Functions \[page 378\]](#)

6.1.16 Keyboard Shortcuts

Keyboard Shortcuts is a type of technical component.

The *Keyboard Shortcuts* technical component allows you to create your own set of keyboard shortcuts for use in your applications. You can create your own set of key code combinations to execute scripting methods associated with your application.

Related Information

[Working with Keyboard Shortcuts \[page 292\]](#)

6.1.17 Text Pool

The *Text Pool* is a technical component.

The *Text Pool* enables text translation in analysis applications. In addition to the texts of the data from the back-end system (SAP 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.

Related Information

[Enabling Text Translation in Analysis Applications \[page 380\]](#)

6.1.18 Timer

The *Timer* is a technical component.

The *Timer* technical component allows you to call an *On Timer* event, which executes a script periodically. The time interval of the *Timer* is set with the *Interval in Milliseconds* property of the component. The default setting is *1000*. The *Timer* could be used to periodically pull updates from a data source. It can be used with any OOTB *Chart* component to simulate real-time dashboards with single or multiple SAP HANA or SAP BW data sources.

6.1.19 Value Help Settings

Value Help Settings is a technical component.

The technical component *Value Help Settings* serves two purposes:

- It allows you to share value help settings among several components (that support the technical component *Value Help Settings* in their properties) instead of defining all these settings multiple times with each component.
- It allows you to define the value help settings of dimensions. Also, the order of appearance of these dimensions in the component at runtime can be configured.

i Note

- The settings of a *Value Help Settings* technical component are applied to a target component's value help settings, but not to the target component's data source. For example, the display of a *Crosstab*'s data is not affected by the settings of a *Value Help Settings* technical component, but the value help of a planning-enabled *Crosstab* only.
- Keep in mind that the settings of a *Value Help Settings* technical component are not necessarily valid for all target components value help settings.

The *Value Help Settings* technical component works only as soon as you assign it to components that support the *Value Help Settings* technical component, for example:

- Crosstab
- Dimension Filter
- Filter Bar
- Filter Panel
- Context Menu

You can create one or more *Value Help Settings* technical components. This enables you to assign different settings to different sets of components.

Some components share specific settings with the *Value Help Settings* component. This allows to set these values of a component without the need of a *Value Help Settings* component. If such a component references a *Value Help Settings* component, then the settings in the *Value Help Settings* component are applied - and the specific settings of the component are ignored.

6.1.19.1 Modifying Value Help Settings for Dimensions

Use the *Value Help Settings* technical component to share value help settings among several components and to define the value help settings for dimensions.

Context

The technical component *Value Help Settings* serves two purposes:

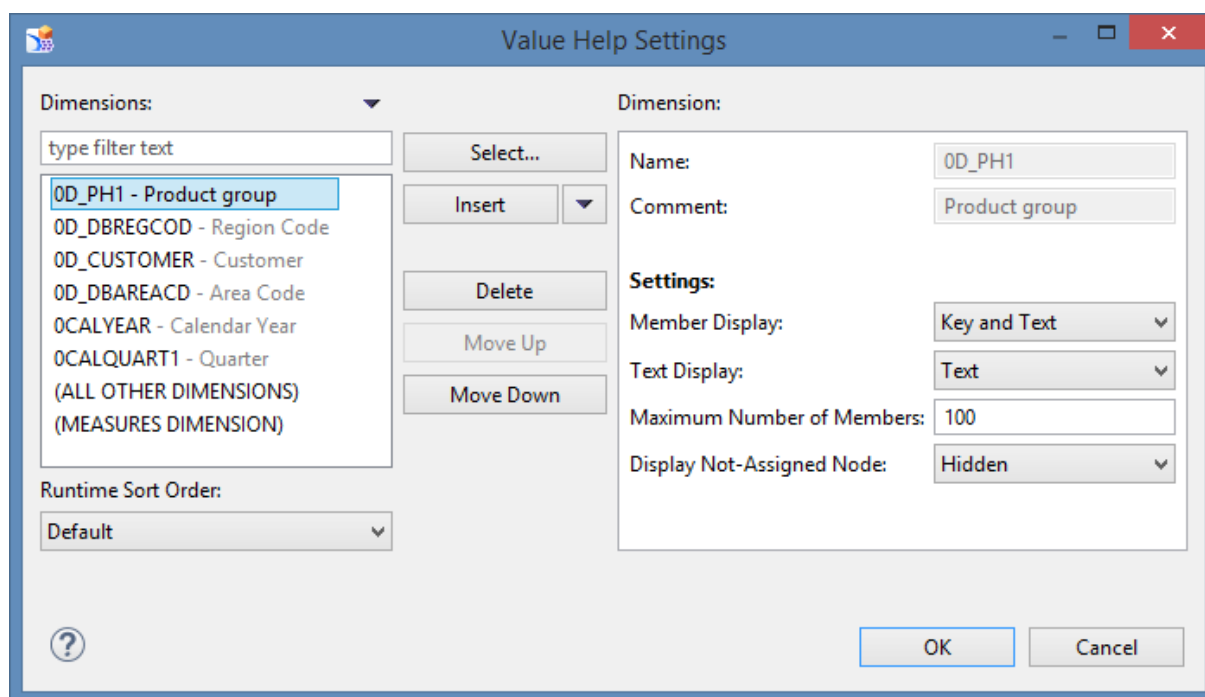
- It allows you to share value help settings among several components (that support the technical component *Value Help Settings* in their properties) instead of defining all these settings multiple times for each component.
- It allows you to define the value help settings of dimensions. Also, the order of appearance of these dimensions in the component at runtime can be configured.

You create the *Value Help Settings* technical component in the *Outline* view by selecting the context menu on **Technical Components** > **Create** > *Value Help Settings*.

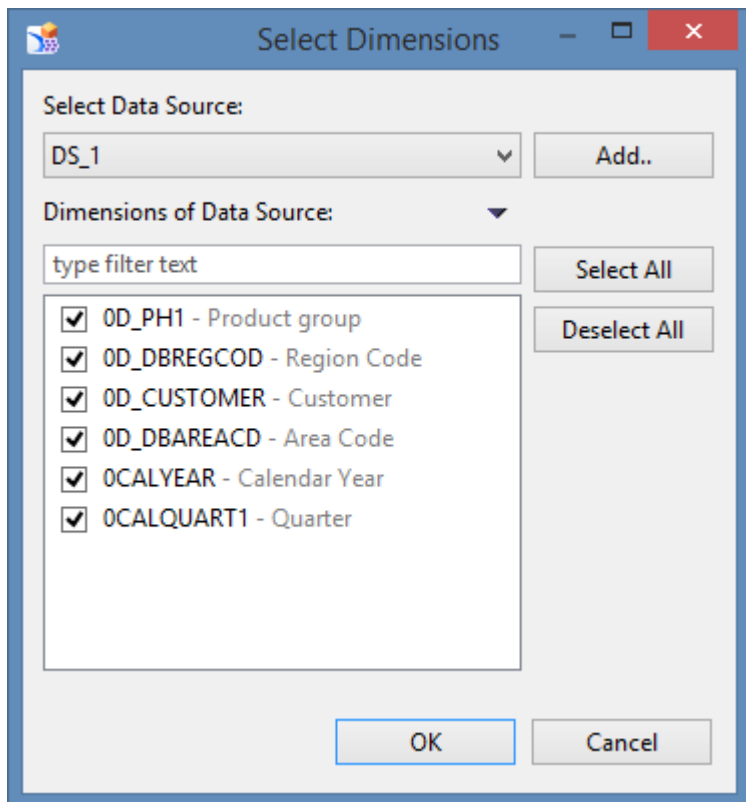
To modify value help settings, select a dimension from a data source in a *Value Help Settings* component in the following way:

Procedure

1. Click the *Edit the Value Help Settings* button right of the *Value Help Settings Items* property of your *Value Help Settings* technical component. The *Value Help Settings* dialog box opens.



2. Click the [Select...](#) button. The [Select Dimension](#) dialog box opens.



3. Select a data source of the application from the [Select Data Source](#) dropdown box or add a new one by clicking the [Add...](#) button. The dimensions of the data source are displayed.
4. Select the checkbox of one or more dimensions whose value help settings you want to modify, then click [OK](#). The selected dimensions are added to the list of dimensions in the [Value Help Settings](#) dialog box.
5. Select one dimension and modify its value help settings on the right part of the dialog box under [Settings](#). You can modify the following settings for each dimension:
 - [Member Display](#)
Specifies whether a dimension member's text, key, or both are displayed.
 - [Text Display](#)
If you choose to display text with the [Member Display](#) setting, you can choose to display a dimension member's short, medium, long text – or the first text found when looking at the short, medium, or long text, in that order.
 - [Maximum Number Of Members](#)
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.
 - [Display Not-Assigned Node](#)
Specifies whether to display the not-assigned node. If you choose to display the not-assigned node, you can also select to make the not-assigned node expandable. This setting applies only to a dimension with an active hierarchy.

i Note

Besides specific dimensions of a data source, you can also select two pseudo-dimensions, which represent the following:

- **(ALL OTHER DIMENSIONS)**: All dimensions of the component's data source that have not been selected in the assigned *Value Help Settings* target component.
- **(MEASURES DIMENSION)**: The measures dimension of the component's data source.

You can add these pseudo-dimensions by clicking the triangle button adjacent to the *Insert* button and choosing the appropriate item from the menu in the *Value Help Settings* dialog box (*Add "All other dimension"* and/or *Add "Measures"*).

i Note

Keep in mind that the settings of a *Value Help Settings* technical component are not necessarily valid for all target components value help settings.

6. (optional) You can define the order in which the dimensions are displayed at runtime with the *Runtime Sort Order* dropdown box. Note that the runtime sort order does not apply to components that display only one dimension, for example, the *Dimension Filter* component. You can choose between the following order possibilities:
 - *Default*
Displays the dimensions as defined in the query/query view.
 - *As In List Above*
Displays the dimensions as listed in the dimension list in the *Value Help Settings* dialog box.
 - *Ascending/Descending by Name/Text*
Displays the dimensions sorted by the selected criterion name or text.

6.1.19.2 Value Help Settings and Applications with Unassigned Data Sources

You can use the technical component *Value Help Settings* for applications with unassigned data sources.

Applications with unassigned data sources are usually used to assign a query or query view with frequently-used dimensions at a later point in runtime.

For these applications, you cannot select a concrete dimension from a data source in the *Value Help Settings* dialog box, as by definition, a query or query view with concrete dimensions has not been assigned to the data source yet. However, you can define value help settings by specifying the dimension manually.

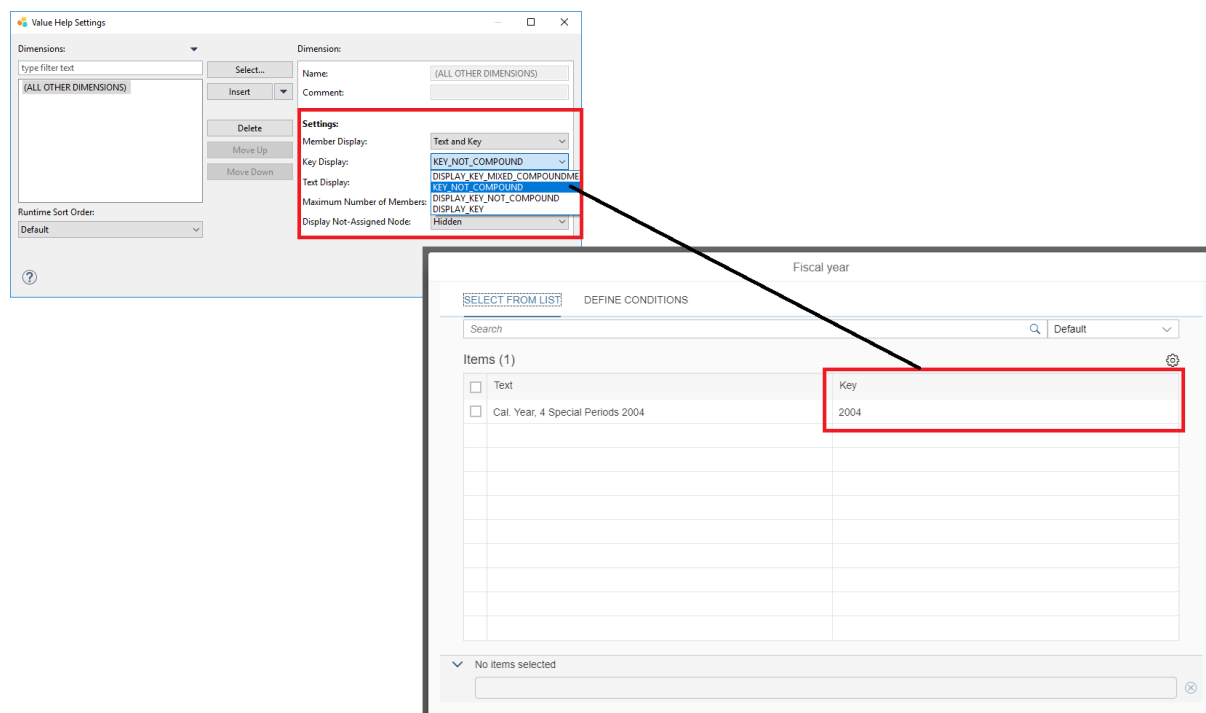
To specify a dimension manually, proceed as follows:

1. Click the *Insert* button in the *Value Help Settings* dialog box.
2. Enter a name and a comment (optional) in the fields *Name* and *Comment* on the right side of the dialog box. Specify the value help settings for this dimension.
As dimension names are often very technical, the comment field allows you to enter a user-friendly text to identify the dimension.

6.1.19.3 Using the Key Presentation for Dimension Members in the Value Help Settings Dialog

You can change the key presentation for dimension members (for example compounded keys, display key) in the value help settings dialog.

You can choose the display you want in the *Key Display* list in the settings area of the *Value Help Settings* dialog.



The key will be displayed in the corresponding dimension if the dimension supports the display presentation you chose.

If the dimension doesn't support the display presentation you chose, it will be displayed with the key presentation display key (default configuration).

6.2 Using the Context Menu in the Outline View

The context menu for the respective elements in the *Outline* view offers you various 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.
- add to bookmark definition
Use [Add to Bookmark Definition](#), if you want to include the selected data source when serializing the state of your application with a bookmark. The data source is then listed in the [Definition](#) property of the [Bookmarks](#) technical component.
- remove from bookmark definition
Use [Remove from Bookmark Definition](#), if you no longer want to include the selected data source in your bookmark definition. The data source no longer appears in the [Definition](#) property of the [Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for data sources that have already been added to the bookmark definition.

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 components

For single components in the [Outline](#) view you find the following functions:

- create
Use [Create](#) to add a [Block](#) to the [Adaptive Layout](#) component only.
- preview of viewport
Use [Preview of Viewport](#) to select to preview the different viewports at design time in the [Adaptive Layout](#) component only.
- 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 Lumira Designer to arrange them as required in the application.

i Note

Depending on the docking behavior 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 behavior.
- 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.

i Note

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. The results are displayed in the [Search Results](#) view.
- add to bookmark definition
Use [Add to Bookmark Definition](#), if you want to include the selected component when serializing the state of your application with a bookmark. The component is then listed in the [Definition](#) property of the [Bookmarks](#) technical component.
- remove from bookmark definition
Use [Remove from Bookmark Definition](#), if you no longer want to include the selected component in your bookmark definition. The component no longer appears in the [Definition](#) property of the [Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for components that have already been added to the bookmark definition.

Context menu functions for action sheet

The following functions are available for the technical component [Action Sheet](#):

- copy
Use [Copy](#) to copy the selected [Action Sheet](#) technical component.
- paste
Use [Paste](#) to paste a copied [Action Sheet](#). The system displays the pasted [Action Sheet](#) immediately in the [Technical Components](#) folder and automatically generates a new name (for example Action_Sheet_2).
- rename
Use [Rename](#) if you want to rename an existing [Action Sheet](#).
- delete
Use [Delete](#) to delete the [Action Sheet](#) technical component from the application.
- find references
Use [Find References](#) if you want to know which elements of the application reference the [Action Sheet](#) technical component. The results are displayed in the [Search Results](#) view.
- add to bookmark definition
Use [Add to Bookmark Definition](#), if you want to include the selected [Action Sheet](#) technical component when serializing the state of your application with a bookmark. The [Action Sheet](#) is then listed in the [Definition](#) property of the [Bookmarks](#) technical component.
- remove from bookmark definition
Use [Remove from Bookmark Definition](#), if you no longer want to include the selected [Action Sheet](#) technical component in your bookmark definition. The [Action Sheet](#) no longer appears in the [Definition](#) property of the [Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for [Action Sheet](#) technical components that have already been added to the bookmark definition.

Context menu functions for application list

The following functions are available for the technical component *Application List*:

- copy
Use *Copy* to copy the selected *Application List* technical component.
- delete
Use *Delete* to delete the *Application List* technical component from the application.
- find references
Use *Find References* if you want to know which elements of the application reference the *Application List* technical component. The results are displayed in the *Search Results* view.

Context menu functions for authoring

The following functions are available for the technical component *Authoring*:

- copy
Use *Copy* to copy the selected *Authoring* technical component.
- delete
Use *Delete* to delete the *Authoring* technical component from the application.
- find references
Use *Find References* if you want to know which elements of the application reference the *Authoring* technical component. 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
Use *Rename* if you want to rename an existing backend connection.
- 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.
- add to bookmark definition
Use *Add to Bookmark Definition*, if you want to include the selected backend connection when serializing the state of your application with a bookmark. The backend connection is then listed in the *Definition* property of the *Bookmarks* technical component.
- remove from bookmark definition

Use [Remove from Bookmark Definition](#), if you no longer want to include the selected backend connection in your bookmark definition. The backend connection no longer appears in the [Definition](#) property of the [Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for backend connections that have already been added to the bookmark definition.

Context menu functions for bookmarks

The following functions are available for the technical component [Bookmarks](#):

- copy
Use [Copy](#) to copy the selected [Bookmarks](#) technical components.
- paste
Use [Paste](#) to paste a copied [Bookmarks](#) technical component.
- rename
Use [Rename](#) if you want to rename an existing [Bookmarks](#) technical component.
- delete
Use [Delete](#) to delete the [Bookmarks](#) technical component from the application.
- find references
Use [Find References](#) if you want to know which elements of the application reference the [Bookmarks](#) technical component. The results are displayed in the [Search Results](#) view.

Context menu functions for comments

The following functions are available for the technical component [Comments](#):

- copy
Use [Copy](#) to copy the selected [Comments](#) technical component.
- delete
Use [Delete](#) to delete the [Comments](#) technical component from the application.
- find references
Use [Find References](#) if you want to know which elements of the application reference the [Comments](#) technical component. The results are displayed in the [Search Results](#) view.

Context menu functions for component

The following functions are available for the technical component [Component](#):

- copy
Use [Copy](#) to copy the selected [Component](#) technical component.
- delete
Use [Delete](#) to delete the [Component](#) technical component from the application.
- find references
Use [Find References](#) if you want to know which elements of the application reference the [Component](#) technical component. The results are displayed in the [Search Results](#) view.

- add to bookmark definition
Use [Add to Bookmark Definition](#), if you want to include the selected [Component](#) technical component when serializing the state of your application with a bookmark. The [Component](#) technical component is then listed in the [Definition](#) property of the [Bookmarks](#) technical component.
- remove from bookmark definition
Use [Remove from Bookmark Definition](#), if you no longer want to include the selected [Component](#) technical component in your bookmark definition. The [Component](#) no longer appears in the [Definition](#) property of the [Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for [Component](#) technical components that have already been added to the bookmark definition.

Context menu functions for conditional formatting settings

The following functions are available for the technical component [Conditional Formatting Settings](#):

- copy
Use [Copy](#) to copy the selected [Conditional Formatting Settings](#) technical component.
- delete
Use [Delete](#) to delete the [Conditional Formatting Settings](#) technical component from the application.
- find references
Use [Find References](#) if you want to know which elements of the application reference the [Conditional Formatting Settings](#) technical component. The results are displayed in the [Search Results](#) view.
- add to bookmark definition
Use [Add to Bookmark Definition](#), if you want to include the selected [Conditional Formatting Settings](#) technical component when serializing the state of your application with a bookmark. The [Conditional Formatting Settings](#) technical component is then listed in the [Definition](#) property of the [Bookmarks](#) technical component.
- remove from bookmark definition
Use [Remove from Bookmark Definition](#), if you no longer want to include the selected [Conditional Formatting Settings](#) technical component in your bookmark definition. The [Conditional Formatting Settings](#) no longer appears in the [Definition](#) property of the [Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for [Conditional Formatting Settings](#) technical components that have already been added to the bookmark definition.

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.
- add to bookmark definition
Use [Add to Bookmark Definition](#), if you want to include the selected [Context Menu](#) when serializing the state of your application with a bookmark. The [Context Menu](#) is then listed in the [Definition](#) property of the [Bookmarks](#) technical component.

- remove from bookmark definition
Use [Remove from Bookmark Definition](#), if you no longer want to include the selected [Context Menu](#) in your bookmark definition. The [Context Menu](#) no longer appears in the [Definition](#) property of the [Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for [Context Menu](#) technical components that have already been added to the bookmark definition.

Context menu functions for document

The following functions are available for the technical component [Document](#):

- copy
Use [Copy](#) to copy the selected [Document](#) technical component.
- delete
Use [Delete](#) to delete the [Document](#) technical component from the application.
- find references
Use [Find References](#) if you want to know which elements of the application reference the [Document](#) technical component. The results are displayed in the [Search Results](#) view.

Context menu functions for export ao

The following functions are available for the technical component [Export AO](#):

- copy
Use [Copy](#) to copy the selected [Export AO](#) technical component.
- delete
Use [Delete](#) to delete the [Export AO](#) technical component from the application.
- find references
Use [Find References](#) if you want to know which elements of the application reference the [Export AO](#) technical component. The results are displayed in the [Search Results](#) view.
- add to bookmark definition
Use [Add to Bookmark Definition](#), if you want to include the selected [Export AO](#) technical component when serializing the state of your application with a bookmark. The [Export AO](#) technical component is then listed in the [Definition](#) property of the [Bookmarks](#) technical component.
- remove from bookmark definition
Use [Remove from Bookmark Definition](#), if you no longer want to include the selected [Export AO](#) technical component in your bookmark definition. The [Export AO](#) technical component no longer appears in the [Definition](#) property of the [Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for [Export AO](#) technical components that have already been added to the bookmark definition.

Context menu functions for export csv

The following functions are available for the technical component *Export CSV*:

- copy
Use *Copy* to copy the selected *Export CSV* technical component.
- delete
Use *Delete* to delete the *Export CSV* technical component from the application.
- find references
Use *Find References* if you want to know which elements of the application reference the *Export CSV* technical component. The results are displayed in the *Search Results* view.
- add to bookmark definition
Use *Add to Bookmark Definition*, if you want to include the selected *Export CSV* technical component when serializing the state of your application with a bookmark. The *Export CSV* technical component is then listed in the *Definition* property of the *Bookmarks* technical component.
- remove from bookmark definition
Use *Remove from Bookmark Definition*, if you no longer want to include the selected *Export CSV* technical component in your bookmark definition. The *Export CSV* technical component no longer appears in the *Definition* property of the *Bookmarks* technical component. The *Remove from Bookmark Definition* is only displayed for *Export CSV* technical components that have already been added to the bookmark definition.

Context menu functions for export excel

The following functions are available for the technical component *Export Excel*:

- copy
Use *Copy* to copy the selected *Export Excel* technical component.
- delete
Use *Delete* to delete the *Export Excel* technical component from the application.
- find references
Use *Find References* if you want to know which elements of the application reference the *Export Excel* technical component. The results are displayed in the *Search Results* view.
- add to bookmark definition
Use *Add to Bookmark Definition*, if you want to include the selected *Export Excel* technical component when serializing the state of your application with a bookmark. The *Export Excel* technical component is then listed in the *Definition* property of the *Bookmarks* technical component.
- remove from bookmark definition
Use *Remove from Bookmark Definition*, if you no longer want to include the selected *Export Excel* technical component in your bookmark definition. The *Export Excel* technical component no longer appears in the *Definition* property of the *Bookmarks* technical component. The *Remove from Bookmark Definition* is only displayed for *Export Excel* technical components that have already been added to the bookmark definition.

Context menu functions for export pdf

The following functions are available for the technical component *Export PDF*:

- copy
Use *Copy* to copy the selected *Export PDF* technical component.
- delete
Use *Delete* to delete the *Export PDF* technical component from the application.
- find references
Use *Find References* if you want to know which elements of the application reference the *Export PDF* technical component. The results are displayed in the *Search Results* view.
- add to bookmark definition
Use *Add to Bookmark Definition*, if you want to include the selected *Export PDF* technical component when serializing the state of your application with a bookmark. The *Export PDF* is then listed in the *Definition* property of the *Bookmarks* technical component.
- remove from bookmark definition
Use *Remove from Bookmark Definition*, if you no longer want to include the selected *Export PDF* technical component in your bookmark definition. The *Export PDF* no longer appears in the *Definition* property of the *Bookmarks* technical component. The *Remove from Bookmark Definition* is only displayed for *Export PDF* technical components that have already been added to the bookmark definition.

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.
- add to bookmark definition
Use *Add to Bookmark Definition*, if you want to include the selected global scripts object when serializing the state of your application with a bookmark. The global scripts object is then listed in the *Definition* property of the *Bookmarks* technical component.
- remove from bookmark definition
Use *Remove from Bookmark Definition*, if you no longer want to include the selected global scripts object in your bookmark definition. The global scripts object no longer appears in the *Definition* property of the

[Bookmarks](#) technical component. The [Remove from Bookmark Definition](#) is only displayed for context menus that have already been added to the bookmark definition.

i Note

If you want to change an existing global script function, use the context menu of this function and choose [Edit](#). The [Edit Script Function](#) dialog box opens where you can make your changes.

Context menu functions for keyboard shortcuts

The following functions are available for the technical component [Keyboard Shortcuts](#):

- copy
Use [Copy](#) to copy the selected [Keyboard Shortcuts](#) technical component.
- delete
Use [Delete](#) to delete an existing [Keyboard Shortcuts](#) technical component.
- find references
Use [Find References](#) if you want to know which applications reference the selected [Keyboard Shortcuts](#) technical component. The results are displayed in the [Search Results](#) view.

Context menu functions for personalization

The following functions are available for the technical component [Personalization](#):

- copy
Use [Copy](#) to copy the selected [Personalization](#) technical component.
- delete
Use [Delete](#) to delete an existing [Personalization](#) technical component.
- find references
Use [Find References](#) if you want to know which applications reference the selected [Personalization](#) 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.

Context menu functions for timer

The following functions are available for the technical component *Timer*:

- copy
Use *Copy* to copy the selected *Timer* technical component.
- paste
Use *Paste* to paste a copied *Timer* technical component.
- rename
Use *Rename* if you want to rename an existing *Timer* technical component..
- delete
Use *Delete* to delete the *Timer* from the application.
- find references
Use *Find References* if you want to know which applications reference the selected *Timer*. The results are displayed in the *Search Results* view.

Context menu funktion for value help setting

The following functions are available for the technical component *Value Help Settings*:

- copy
Use *Copy* to copy the selected *Value Help Settings* technical component.
- paste
Use *Paste* to paste a copied *Value Help Settings* technical component.
- rename
Use *Rename* if you want to rename an existing *Value Help Settings* technical component..
- delete
Use *Delete* to delete the *Value Help Settings* technical component from the application.
- find references
Use *Find References* if you want to know which elements of the application reference the *Value Help Settings* technical component. The results are displayed in the *Search Results* view.

Related Information

[Working with the Initial View Dialog Box for Data Source Aliases \[page 141\]](#)

[Inserting a Copied Data source from SAP BusinessObjects Analysis, Edition for Microsoft Office \[page 140\]](#)

[Working with Global Scripts Objects and Global Script Functions \[page 378\]](#)

[Using the Context Menu \(Technical Component\) \[page 86\]](#)

[Using Planning Functions and Sequences \(Automated Planning\) \[page 196\]](#)

[Exporting to PDF \[page 169\]](#)

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 Lumira Designer can access SAP HANA systems or SAP 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 <https://help.sap.com>), 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 - also relating to the start up mode of the design tool - for adding a data source to the application. Choose the method that suits the way you like to work in the design tool:
 - (only relevant for the Lumira Documents mode) Click ► *File* ► *Add Data Source...* ► in the menu of the design tool.
 - (only relevant for the legacy BI platform mode) 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.

- c. 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 \[page 515\]](#)

[Defining Connections to BI Backend Systems \[page 516\]](#)

[Selecting a Data Source \[page 117\]](#)

[Assigning a Data Source to a Component \[page 144\]](#)

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, SAP 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.

i Note

The BI platform server needs to be SAP BusinessObjects Business Intelligence 4.1. This is also valid, if you want to use universe queries as data sources in your analysis applications.

In Lumira Designer you can use the following types of universes:

- relational
- single-source
- multi-source relational

For general information about universe access in SAP Lumira Designer, see [Universe Access in Lumira Designer \[page 135\]](#)

If you want to create a planning application, you have to select a planning connection. For more information see [Selecting a Planning Connection \[page 194\]](#)

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 BW, universe) is up and running.

You choose the connection when you add a data source to your application and open the [Add Data Source](#) dialog box.

Procedure

1. In the [Connection](#) box (that is part of the [Add Data Source](#) dialog 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 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.
- Note that for OLAP connections of type SAP BW or SAP HANA the authentication type [Prompt](#) is not supported by Lumira Designer. 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.

i Note

You can see all created connections in the design tool under **Tools > Preferences > Backend Connections**. To display recently defined connections in the corresponding table, click **Reload All Connections**.

7.1.1 Mapping Connections

In Lumira Documents mode, you can exchange the connections for the data sources used in your Lumira document.

Context

You can map connections for each Lumira document in Lumira Documents mode, that is for documents stored locally and for documents stored on the BI platform. For documents stored on the BI platform, you need to work online with an active BI Platform connection.

Procedure

1. In Lumira Designer, select the Lumira document you want to assign to different connections and choose **Map Connections...**

You can access the **Map Connections** dialog box on both open and closed Lumira documents. With a closed Lumira document, the system opens the dialog box automatically.

The system scans your document for SAP BW and SAP HANA connections and lists any it finds in the lefthand column of the **Map Connections** dialog box.

2. To map a connection, choose a connection from the list in the righthand column. If the connection you need isn't listed, choose **Browse Connections** and select the connection from there.
3. **Optional:** If you want to delete the original connections while mapping your connections to the new ones you selected, check **Delete unused original connections from the document**.
4. Click **OK**.

Results

The system searches for references to the connections you mapped in all analysis applications and composites of your Lumira document and replaces all occurrences accordingly. If you checked **Delete unused original**

[connections from the document](#), the system deletes the original connections. Otherwise, they are still contained in your Lumira document.

7.1.2 Deleting Unused Connections

In Lumira Documents mode, you can clean up your Lumira documents by deleting unused connections.

Context

You can delete unused connections for each Lumira document in Lumira Documents mode, that is for documents stored locally and for documents stored on the BI platform. For documents stored on the BI platform, you need to work online with an active BI Platform connection.

Procedure

1. In Lumira Designer, select the Lumira document you want to clean up and choose [Map Connections....](#)

You can access the [Map Connections](#) dialog box on both open and closed Lumira documents. With a closed Lumira document, the system opens the dialog box automatically.

2. Check [Delete unused original connections from the document](#).
3. Click [OK](#).

The system scans all SAP BW and SAP HANA connections that are contained in the Lumira document and checks if they are still referenced in analysis applications and composites of the document. If this is not the case, the system lists the unused connections.

4. If there are unused connections in the document, confirm the deletion of the listed connections.

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 [Add Data Source](#) dialog box, use one of the following methods:

- In the [Data Source](#) field type the name of a data source that you know.

→ Tip

To see the history of your last ten choices, press `CTRL` + `Space` on your keyboard.

- In the [Data Source](#) field 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](#) in the [Data Source](#) field. 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.

Option	Description
connection to BW system	<p>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.</p> <p>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.</p>
connection to SAP HANA system	<p>Folder tab: All available data sources are displayed in a hierarchical structure. Select the one that suits your needs.</p> <p>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.</p>

→ 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 [Working with the Member Selector \[page 120\]](#).
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 \[page 141\]](#).

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 \[page 144\]](#)

7.2.1 Creating a Query

Prerequisites

This procedure assumes that you have opened the Query Panel (also referred to in the Lumira Designer 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.

To change the layout of hierarchical data, click the [Result set display options](#) icon in the [Data Preview](#) pane and select an option from the list:

Option	Description
Flat layout	Displays repeated values for a level in every row.
Hierarchical layout	Displays repeated values once for a level.

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.

Related Information

[Working with the Initial View Dialog Box for Data Source Aliases \[page 141\]](#)

[Working with the Member Selector \[page 120\]](#)

[Creating a Business Filter \[page 130\]](#)

[Filtering Data Using Prompts \[page 132\]](#)

7.2.2 Working with 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 \[page 122\]](#)

[About selecting hierarchy members \[page 121\]](#)

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:

Tab	Description
<i>Members</i>	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.
<i>Metadata</i>	Shows the hierarchy levels (if the hierarchy supports named levels), named sets, and calculated members.
<i>Prompts</i>	Lets you define and modify prompts.

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

Related Information

[Selecting hierarchy members \[page 122\]](#)

[Selecting members by hierarchy relationship \[page 123\]](#)

[Selecting hierarchy members by level \[page 124\]](#)

[Selecting named sets \[page 125\]](#)

[Selecting calculated members \[page 125\]](#)

[Searching for hierarchy members \[page 126\]](#)

[Excluding hierarchy members \[page 126\]](#)


[Defining a prompt to select members \[page 127\]](#)

[Showing selected members in the Member Selector \[page 128\]](#)

[Sorting hierarchy members \[page 128\]](#)
[Setting display options \[page 129\]](#)
[Showing estimated child count \[page 129\]](#)

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 \[page 121\]](#)

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:

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

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 \[page 122\]](#)

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.

Relationship Function	Description
Self	Includes only the selected member. This is the default setting.
Children	Includes members one level below the selected member that have the selected member as their parent. The selected member is not included.
Descendants	Includes all members at all levels below the selected member. The selected member is not included.
Descendants until Named Level...	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.
Descendants until...	Includes the members at levels below the selected member until the number of levels you select.
Parent	Includes the member that is one level above the selected member. The selected member is not included.
Ancestors	Includes all members at all levels above the selected member. The selected member is not included.

Relationship Function	Description
Siblings	Includes members at the same level that have the same parent as the selected member. The selected member is not included.
Exclude	Excludes members according to the relationship function (Self/Children/Descendants/Parent/Ancestors/Siblings).

Related Information

[Opening the Member Selector in the Query Panel \[page 122\]](#)

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.

i 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 \[page 122\]](#)

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.

i 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 \[page 122\]](#)

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.

i 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 \[page 122\]](#)

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	Description
*	Matches any string of characters
?	Matches any one character

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 \[page 122\]](#)

[Selecting hierarchy members \[page 122\]](#)

[Selecting members by hierarchy relationship \[page 123\]](#)

[Selecting hierarchy members by level \[page 124\]](#)

[Selecting named sets \[page 125\]](#)

[Selecting calculated members \[page 125\]](#)

[Searching for hierarchy members \[page 126\]](#)

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 \[page 122\]](#)

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 \[page 122\]](#)

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  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 \[page 122\]](#)

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  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:

Filter Component	Description
Filtered object	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.
Operator	The operator is used to compare the filtered object with the operand. For example, the <i>Equal To</i> operator retains only those values of the filtered object that correspond exactly to the value of the operand.
Operand	The operand supplies the value or values used to filter the filtered object.

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:

Operand type	Description
<i>Constant</i>	<p>Use the <i>Constant</i> operand to enter values directly into the filter. For example, you can use a constant to enter France into the filter:</p> <pre>[Country] Equal To France</pre> <p>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.</p>
<i>List of Values</i>	<p>Use the <i>List of Values</i> 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.</p>
<i>Object</i>	<p>Use the <i>Object</i> operand to specify an object in the business layer. Drag and drop a business layer object to the operand position when defining the filter.</p> <div> <p>i Note</p> <p>You cannot select an object as an operand on some OLAP data sources or if the filtered object is a hierarchy.</p> </div>
<i>Prompt</i>	<p>Use the <i>Prompt</i> operand when you want to be prompted for a value when the query is refreshed. See the related topic about filtering using prompts.</p> <div> <p>i Note</p> <p>Prompt operands are not available if the business filter is defined for a Business Security Profile.</p> </div>

- 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.

i Note

The OR operator is not supported for OLAP data sources.

- 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.

i 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 \[page 133\]](#)

[Using an Existing Prompt to Filter Data \[page 134\]](#)

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.

Related Information

[Creating a Business Filter \[page 130\]](#)

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.

i Note

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

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.

Related Information

[Creating a Business Filter \[page 130\]](#)

7.3 Universe Access in Lumira Designer

Supported Platform Release

SAP Lumira Designer supports universe access for BI Platform 4.1 and 4.2.

Supported Universe Types

With SAP Lumira Designer, you can use the following types of universes:

- relational
- single-source
- multi-source relational

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 Lumira Designer 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:

Average	Data is aggregated with average
Count	Number of occurrences is calculated
Max	Data is aggregated with maximum
Min	Data is aggregated with minimum
None	No aggregation available; if data needs to be aggregated, no value is displayed
Sum	Data is aggregated with summation

- Value help (list of values)
Value helps defined in the universe are only used in the prompt context in Lumira Designer. 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 Lumira Designer:
 - 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 Lumira Designer 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 [Select only from list](#), values that do not exist in the list of values are not allowed. The property [Keep last value](#) has no effect; the last value is always kept - as in the Lumira Designer 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 20000 rows or 200000 data cells. If either limit is exceeded, no data is displayed.

- Filter display

In Lumira Designer 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 assigned the data source to the charts or SDK extensions.

i Note

The data selection for one data source is only available for charts and SDK extensions.

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 components 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 [Show](#). 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).

i 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:

Property Type	Data Values
ResultCell	A single data value
ResultCellList	A single row or column of data values
ResultCellSet	A complex selection of data values from rows and columns (a Cartesian selection)
ResultSet	All data values of the result set

i Note

For the ResultCellSet property type, there are limitations of selection options in the [Select Data from...](#) dialog box.

For further information, see “SDK Extensions and Data Binding” in the *Developer Guide: Component SDK* at <http://help.sap.com/>.

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.

Related Information

[Chart \[page 451\]](#)

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.

8 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 Lumira Designer 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 Lumira Designer 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 Lumira Designer.

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 Lumira Designer, 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.

9 Working with the Initial View Dialog Box for Data Source Aliases

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

i 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)

i 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.

10 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.

i 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.

i Note

Switch to the [Data Binding](#) view in the Outline view to see which components have been assigned to a specific data source.

10.1 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 in the [Properties](#) view of this component.

The [Binding](#) column in the [Properties](#) view displays a plus symbol for every property which can be bound (but currently is not bound). Properties without a plus symbol in the binding column cannot be bound at all. Pressing the plus symbol creates a new [Source](#) binding (and expands it automatically in the property sheet). In this way, you can create analysis applications without using scripts.

Note

In the special case of the [Items](#) property of a selection component (for example, List Box, Dropdown Box, Radio Button Group, Checkbox Group), you can also create a [Target](#) binding that in turn has an effect in the other direction: It filters the data source.

The Target binding for selection components can also be performed by dragging and dropping data sources (excluding SDK data sources) from the [Outline](#) view onto a selection component (in both the [Outline](#) view and the editor area). The dialog box [Create Filter Binding](#) is displayed where you have to select the dimension to bind. The source and target binding are configured automatically. In this way, the selection component can now act as a filter component for that data source and dimension).

Usually, when using drag and drop of data sources on selection components, only small adaptations are necessary in the property sheet (for example, setting the [All Members Text](#) property). Alternatively, you can set every property for the target binding manually in the [Properties](#) sheet.

After creating a binding, the plus symbol turns into a trashcan symbol that allows you to remove the binding again.

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 \[page 146\]](#)

10.1.1 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 plus symbol behind the [Text](#) property (in the [Binding](#) column of the [Properties](#) view). Clicking the plus symbol creates a new [Source](#) binding and expands it automatically in the property sheet).
4. Set the following properties for the [Source](#) binding:
 - [Type](#) (value [Data Cell Binding](#))
This property is set automatically.
 - [Data Source](#)
(optional) If there is more than one data source in your application, choose the data source that you want to bind from the [Data Source](#) dropdown box.
 - [Selected Data Cell](#)
In the [Select Data](#) dialog box, select a single data cell that you want to bind from the table displayed or enter a selection expression into the input field. You can either select a data cell or toggle members to change the effective selection. And click [OK](#).
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.
 - [Formatter Function](#)
Click on the dropdown box of the [Formatter Function](#) and choose [Add](#). The [Create Script Function](#) dialog box is displayed.
5. 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](#).
6. 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.

i Note

You can only use those global script functions as formatter functions which were created in the property view as described above. 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.

10.1.2 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 plus symbol in the *Binding* column of the *Items* property. The property sheet is expanded and the properties for *Source* and *Target* binding are displayed.
4. In the *Source* binding section 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.
5. Navigate to the *Target* binding section and click on the plus symbol.
6. Choose at least one target data source (*Target Data Source*). It can be the same as the data source you used to fill the list box.
7. Choose a *Dimension* to filter. Usually this will be the same dimension as you used to fill the list box.
8. Execute the application. Choosing one or more entries in the list box will automatically filter the visualization component (for example, the crosstab).

i Note

If you want several dimension members to be selectable in the list box, set the *Selection Mode* property of the list box to *Multiple Selection*.

10.1.3 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 plus symbol in the *Binding* column of the property. 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

Depending on the start up mode of your design tool, you have the following options to save your application.

- If you work in Lumira Documents mode, choose ► [File](#) ► [Save](#) ►  in the menu or choose the [Save](#) button in the tool bar of the design tool.
- If you in the BI platform legacy mode, choose ► [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 ► [File](#) ► [Save All](#)  (if you work in Lumira Documents mode) or ► [Application](#) ► [Save all.](#)  (if you work in BI platform legacy mode).

Next Steps

If you want to save the application using a different name, use the save as function. For more information, see the links 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 \[page 518\]](#)

[Saving an Application Using a Different Name \(Lumira Documents Mode\) \[page 188\]](#)

[Saving an Application Using a Different Name \(BI Platform Legacy Mode\) \[page 188\]](#)

12 Executing an Application

Context

Depending on the start up mode of your design tool, there are several ways to execute an application:

- If you work in Lumira Documents mode, you have the following options:
 - Choose **File > Execute Locally** if you work local
 - Choose **File > Execute on BI Platform** if you are connected to the BI platform.
- If you work in the BI platform legacy mode, you have the following options:
 - 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.

i 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.

13 Executing an Application on a Mobile Device

Context

Depending on the start up mode of our design tool, you can execute an application on a mobile device.

Procedure

1. If you work in Lumira Documents mode, click [Send to Mobile Device](#) - or - if you work in the BI platform legacy mode, click [Send to Mobile Device \(using QR code ®\)](#) in the toolbox of the design tool. 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).

14 Working with Templates

SAP Lumira Designer offers you a set of standard (blank and predefined) and ready-to-run templates as well as some samples that serve various design and business needs. When you choose a template, the system automatically creates a copy of it. You can change the copy according to your needs.

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 \[page 391\]](#).

The following *Standard* templates are available:

- Blank
- Basic Analysis Layout
- Basic Layout
- Planning Layout
- Tabstrip Layout

The following *Ready-To-Run* templates are available:

- Generic Analysis

The following *Samples* are available:

- Dashboard Sample
- Feature Samples

14.1 Using the Basic Analysis Layout Template

The Basic Analysis Layout template is a predefined Standard template used for ad-hoc slicing and dicing data sources. The template contains a table view (crosstab or spreadsheet), a chart view, a table/chart view and a filter area (filter line or filter bar). You can drag and drop between the navigation panel and the crosstab and chart. A context menu is also available on the crosstab, which allows you to easily navigate and analyze your data.

You can easily adapt and extend the template by using a data source of your choice. To do so, open the Basic Analysis Layout template in the design tool and add a data source to the template: In the *Outline* view choose the *unsigned data source* in the *Data Source* folder and set the property *Data Source*. You can now run the template or adapt and change it.

When you run the template as it is, you can use the following functions:

Switching the data visualization

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

i Note

In the generic analysis template the crosstab component is set by default to display data in a table view.. However, if you want to use the spreadsheet component, set the *Default Value* property of the global variable *g_crosstab_on* to *false* at design time. The same applies for the filter line. If you want to replace the default filter line by a filter bar, set the *Default Value* property of the global variable *g_filterline_on* to *false* at design time.

In the *Chart/Table* view, you can also switch the arrangement of the table and chart on your desktop browser from vertical to horizontal, and switch the position of the table and chart from right to left or from above to bottom.

Using the Show Design Panel icon

By choosing the *Show Design Panel* arrow icon, you open the navigation area with *Drilldown* options for the table, chart and table/chart view where 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 by using drag and drop within the navigation panel. After you have rearranged the display of your data within the panel, the changes are automatically displayed. You can also see dimensions in the navigation panel that are not in the drilldown (*FREE*).

If you use the *Chart* or *Table/Chart* view, the *Chart Types* and *Chart Properties* options are enabled.

If you display the data in the table, you will find further drill-down and display options in the context menu of the table. The same is true for the spreadsheet.

Using the filter line

If you want to set a filter, click on the *Add Filters* icon in the left corner of the filter line and choose one of the displayed dimensions. The filter dialog box opens where you can choose the appropriate dimension members. The display of your data will change according to the chosen filters, which are displayed in the filter line.

Using prompts

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

Displaying information

Click on the *Application Information* icon in the top right corner of the application to display information about your application. Here you will see for example, the technical name of your query and the prompts and filters you have set.

Using the Export icon

In the upper right corner of the application, you will find the [Export](#) icon, where you can choose the functions [Export to Microsoft Excel](#), [Export to PDF](#) or [Export to Analysis for Microsoft Office](#).

In the [Export to PDF](#) dialog box, you can choose to export the data as [Report](#), which prints the data visualized in table, chart or chart/table or as [Screenshot](#) (WYSIWYG). You can also enter a name for your PDF, decide on the page size and orientation, add customized headers and footers with or without page numbers and add an appendix to your PDF.

i Note

Printing PDF files is not supported for the spreadsheet.

14.2 Using the Generic Analysis Ready-to-Run Template

The [Generic Analysis](#) template is a predefined ready-to-run template based on the SAPUI5 standard theme [Belize](#) used for slicing and dicing data sources. It contains a table view (crosstab or spreadsheet), a chart view, a table/chart view and a filter area (filter line or filter bar). You can drag and drop between the navigation panel and the crosstab and chart. A context menu is also available on the crosstab, which allows you to easily navigate and analyze your data.

i Note

The template is optimized for the theme [Belize](#). You can change the theme in the Lumira Designer. If you change the theme, you have to make various adjustments to the template (CSS style, size of components, etc...)

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 choosing a data source in the [Data Source Browser](#) dialog box or 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 Lumira Designer:

- for SAP BusinessObjects BI Platform:
`&XSYSTEM=<cuid:YourUniqueCUID>&XQUERY=<TechnicalNameOfYourQuery>`
`http://<applicationURL>&XSYSTEM=cuid:<sysID>&XQUERY=<queryID>`
For further information on the CUID, see the *Administrator Guide: SAP Lumira* on the SAP Help Portal at <http://help.sap.com/>.

i Note

There is also the URL parameter XTYPE. This parameter is initialized with the value "BW". However, it is also possible to use the value "HANA" and in this case the data source is a SAP HANA view.

You can also assign a data source to the template at design time when you open the template in your Lumira Designer. In this case, the [Load in Script](#) data binding property of the data source has to be set to [false](#).

Switching the data visualization

To switch the visualization of your data, choose [Table](#), [Chart](#) or [Chart/Table](#) in the top right corner of the application.

i Note

In the generic analysis template, the crosstab component is set by default to display data in a table view.. However, if you want to use the spreadsheet component, set the [Default Value](#) property of the global variable [g_crosstab_on](#) to [false](#) at design time. The same applies for the filter line. If you want to replace the default filter line by a filter bar, set the [Default Value](#) property of the global variable [g_filterline_on](#) to [false](#) at design time.

In the [Chart/Table](#) view, you can also switch the arrangement of the table and chart on your desktop browser from vertical to horizontal, and switch the position of the table and chart from right to left or from top to bottom.

Using the Show Design Panel icon

By choosing the [Show Design Panel](#) arrow icon, you open the navigation area with [Drilldown](#) options for the table, chart and table/chart view where 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 by using drag and drop within the navigation panel. After you have rearranged the display of your data within the panel, the changes are automatically displayed. You can also see dimensions in the navigation panel that are not in the drilldown ([FREE](#)).

If you use the [Chart](#) or [Table/Chart](#) view, the [Chart Types](#) and [Chart Properties](#) options are enabled.

If you display the data in the table, you will find further drill-down and display options in the context menu of the table. The same is true for the spreadsheet.

Using the filter line

If you want to set a filter, click on the [Add Filters](#) icon in the left corner of the filter line and choose one of the displayed dimensions. The filter dialog box opens where you can choose the appropriate dimension members. The display of your data will change according to the chosen filters, which are displayed in the filter line.

Adding and managing bookmarks

If you have analyzed your data for a particular navigation status, you can set a bookmark to reopen this navigation status of the query (with filter settings and prompts) at a later date, or to share it with others. To do so, select [Add Bookmark](#) in the right area of the application. The [Add Bookmark](#) dialog box opens where you can add a title to your bookmark and choose the folder where you want to save it. You can also decide whether your bookmark should be [Personal](#) or [Global](#):

- Personal bookmarks can only be viewed, edited and deleted by their author.
- Global bookmarks can be viewed by all users. However, they can only be deleted, overwritten, moved and added to a folder by their author.

When you save bookmarks, the system automatically saves all relevant query information (such as system, query name and query type) to guarantee the uniqueness of the bookmark.

Select [Manage Bookmarks](#) in the upper right area of the running template, if you want to share, rearrange or delete your bookmarks. In the [Manage Bookmarks](#) dialog box, your personal and global bookmarks are displayed either in a list or in a tree. For each bookmark you can display specific information (for example, ID, title, folder, access type), send and share it per e-mail, change the sorting order and the display art (in a list or in a tree). Moreover, you can add your bookmarks to the favorites list. If your Lumira Designer is deployed on a BI platform - you can add it to a folder. In this dialog box, it is also possible to add new bookmarks or delete existing ones.

In the [Manage Bookmarks](#) dialog box you can also set a bookmark as default. Select a bookmark and choose [Set as default](#). The [Set as default](#) symbol is displayed in blue. You can change your default bookmark at any time. Just select another bookmark and choose [Set as default](#) again. The system automatically exchanges the bookmarks. If you do not want to have a default bookmark at all, select your default bookmark and choose [Clear default](#).

Using the Favorites list

In the [Favorites](#) list, you will find all the bookmarks you have marked as favorites. By clicking the star icon in the toolbar, you can remove this bookmark from the favorites list.

Using the Data Source Browser

In the [Data Source Browser](#) dialog box, you can change your data source at runtime. You can either select a data source from the recently used data source list or from the system list. The system list varies in its appearance depending on the deployment of your Lumira Designer:

- If you are logged on locally to your Lumira Designer, all systems available in the SAP Log On dialog box are also displayed in the system list of the [Data Source](#) dialog box, together with SAP HANA connections.
- If your Lumira Designer is deployed on the BI platform, you will see all the available connections and folders, which you are authorized for.

Using prompts

If your data source is designed for setting prompts, click on [Prompts](#) in the top right corner of the application to set the prompts.

Displaying information

Click on the [Application Information](#) icon in the top right corner of the application to display information about your application. Here you will see for example, the technical name of your query and the prompts and filters you have set.

Using the Export icon

In the upper right corner of the application, you will find the [Export](#) icon, where you can choose the functions [Export to Microsoft Excel](#), [Export to PDF](#) or [Export to Analysis for Microsoft Office](#).

In the [Export to PDF](#) dialog box, you can choose to export the data as [Report](#), which prints the data visualized in table, chart or chart/table or as [Screenshot](#) (WYSIWYG). You can also enter a name for your PDF, decide on the page size and orientation, add customized headers and footers with or without page numbers and add an appendix to your PDF.

i Note

Printing PDF files is not supported for the spreadsheet.

Related Information

[Creating a Generic Analysis Template for RRI Jump Targets \[page 386\]](#)

14.3 Using the Planning Layout Template

Context

The [Planning Layout](#) template is a predefined ready-to-run template based on the SAPUI5 standard theme [Belize](#) used for planning business data. Besides the functions of the Basic Analysis template, the Planning Layout template also has the following functions designed for planning applications:

- Switch to Display/Edit Mode (Lock/Unlock Data)
- 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 [Create 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 as required. 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 Lumira Designer 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 element ([Planning Functions](#)). In the planning template, the planning functions are bound to one button on the UI. This means that the planning functions you add are executed when the application user clicks on the button and chooses one of the planning functions from the list. In the design tool, you will find the corresponding buttons under the [ACTION_SHEET_PLANNING_FUNCTIONS](#) section in the [Technical Components](#) folder in the [Outline](#) view:
 - a. Change the names of the planning functions by double-clicking [ACTION_SHEET_PLANNING_FUNCTIONS](#) and editing the respective [Text](#) property, for example **Copy Initial Data** for the Copy Planning function you have chosen in step 3.
 - b. Change the [On Select](#) event of the Action Sheet component as required by using the relevant method for the planning function. For example, for the **Copy Initial Data** function and the Delete Data button, you can add scripts like these:

```
if (ACTION_SHEET_PLANNING_FUNCTIONS.getSelectedValue() == "COPY") {  
    PF_COPY.copyFilters(DS_1)  
    PF_COPY.execute();  
}  
  
if (ACTION_SHEET_PLANNING_FUNCTIONS.getSelectedValue() == "DELETE") {  
    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 the planning function buttons, remove them from the items list.

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 edit mode by default. Click on the icon [Lock data](#) if you want to view the data in display mode. If you want to plan data, switch back to the edit mode by clicking the icon [Unlock data](#).
7. Optional: In the edit 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.

i Note

Keep the following points in mind:

- in the display mode, the planning template behaves like the basic 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.

14.4 Using the Tabstrip Layout Template

The *Tabstrip Layout* template is a predefined standard template based on the SAPUI5 standard theme Belize.

The template provides two level nested tabstrip container. An additional pagebook container within the second level tabstrip acts as a spinner, allowing spinner style tab advancement. The Tabstrip Layout template does not contain any analytic or basic components and just provides a skeletal application.

Template Structure

The Tabstrip Layout Template consists of two nested Tabstrip components. The first Tabstrip component contains the first level tabs TAB 1, TAB 2 and TAB 3. Each first level tab contains a pagebook component with pages that correspond to the respective second level tabs for example TAB 2.1, TAB 2.2, TAB 2.3. The use of the pagebook component enables the navigation at run time either by swiping the pages and thus selecting the tabs or by selecting the tabs and thus moving to the corresponding page.

Adding content to the Pages of the Pagebook

As each second level tab correlates to the respective page, you have to add the content into the respective page of the pagebook component.

14.5 Using the Samples

Samples are intended to demonstrate features and functions with offline data sets.

You can edit, copy, or modify the *Samples* to suit your own applications.

i Note

Note that SAP cannot provide OSS support for these samples.

The following samples are available:

- Dashboard Sample City Analysis

This sample shows you a typical dashboard example with KPI tiles, map, charts, and how these components interact with one another. You can use this dashboard to demonstrate the following functionalities:

- responsive layout
- composites
- KPI tiles
- bookmarks
- CSS capability
- map
- interaction between map and other components
- chart capabilities

- Feature Samples

Within the feature samples you can find the following composites and their related applications:

- bookmark dialog
- export to PDF dialog
- comments dialog
- comments panel
- application header
- card

You can open the composites in Lumira Designer to see how they are designed, and execute the related applications to see how these functions work at runtime.

15 Exporting to CSV

Use the [Export CSV](#) technical component to export data sources from Lumira Designer to CSV file format.

This feature enables the application user to specify and export a number of data sources from Lumira Designer into CSV file format. You can use the [Export CSV](#) technical component property [Separator](#), to set at design time the type of field delimiter you want to use. The following options are available:

- comma
- tab
- semicolon
- space

You can also use the scripting methods `getSeparator()` and `setSeparator(CsvSeparator)` to allow an application user to choose a field delimiter at runtime of the application.

Related Information

[Properties of the Export CSV Technical Component \[page 163\]](#)

15.1 Properties of the Export CSV Technical Component

The properties of the [Export CSV](#) technical component specify all the settings associated with the export to CSV output.

Export CSV Properties

Export CSV Properties

Property Area	Property Value	Property Description
data sources	dialog	Select the ellipsis button to call the Edit CSV Order dialog. The Edit CSV Order dialog allows you to select which data sources are included in the exported output to CSV. Use the Insert and Remove buttons to add or remove data sources from your export output. Use the Move Up and Move Down buttons to change the order in which your data sources are exported to CSV.
file name	none	Define a file name for the CSV file to be exported.
separator	comma (default), tab, semicolon, space	Select from the dropdown list the type of field delimiter you want to use.

Related Information

[Exporting to CSV \[page 162\]](#)

16 Exporting a Data Source View to SAP Analysis for Microsoft Office

A data source view can be exported to SAP Analysis for Microsoft Office.

You can select a data source view from within an analysis application and open that same data source view in the Analysis Plug-in for SAP Analysis for Microsoft Office. This is done through the [Export AO](#) technical component, which generates an SAP Analysis for Microsoft Office launcher file. Selecting to activate the launcher file opens the Analysis desktop client. The data source connection details are required. When username and password are entered, the data source state is applied to the Analysis workbook. In local mode, username and password data source connection details are required. In BI platform mode, where the backend connection is not set up with single sign on, logon credentials are required. Where single sign on is setup on the BI platform, logon credentials are not required.

One data source can be selected at a time.

A data source view represents a data source connection and a data source state. The data source state comprises the following:

- system
- query
- filters
- variables
- connection details
- authentication ticket

i Note

The authentication ticket is included only when using BI platform and the backend OLAP connection is configured to use single sign on.

16.1 Properties of the Export AO Technical Component

The properties of the [Export AO](#) technical component specify all the settings associated with the export to SAP Analysis for Microsoft Office output.

Export AO Properties

Export AO Properties

Property Area	Property Value	Property Description
data source	dropdown	Select the data source to be included in the exported output to SAP Analysis for Microsoft Office.

Related Information

[Exporting a Data Source View to SAP Analysis for Microsoft Office \[page 164\]](#)

17 Exporting to Microsoft Excel

Use the [Export Excel](#) technical component to export a crosstab or a data source from SAP Lumira Designer to Microsoft Excel.






This feature enables the application user to select and export crosstabs or data sources from Lumira Designer into Microsoft Excel. Data exports the same way as it is displayed in the initial view of the crosstab or data source. Expanded hierarchies or expanded multiple hierarchies in either columns or rows, or both, export to Microsoft Excel in the same expanded state as shown in the crosstab or data source. A separate tab for each crosstab or data source is created in the Microsoft Excel spreadsheet. The order in which the tabs are created matches the order of the selection made in the [Components](#) property of the [Export Excel](#) technical component.

The following file format is supported:

- Microsoft Excel 2007 (.xlsx)

i Note

- In 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 sheet 1. Sheet 2 contains the metadata including runtime changes, applied filters, or variables. Columns can also be resized depending on the length of the exported columns.
- In Microsoft Excel 2007, the data source metadata is not exported.
- Export to Microsoft Excel is not supported with SAP BusinessObjects Mobile.
- Export to .xls file format is not supported.
- Export to CSV file format is done using the [Export CSV](#) technical component.
- Read the following SAP Notes when exporting to Microsoft Excel.

SAP Note Number	Description
2460561 	Listing Microsoft Excel export feature as last statement SAP Lumira Designer.
2460992 	Increasing jvm memory for large data export for SAP Lumira Designer.
2460973 	SAP Lumira Designer Microsoft Excel export cell size.
2460987 	SAP Lumira Designer Microsoft Excel hierarchy support.
2523266 	SAP Lumira Designer Microsoft Excel export RTL support.

Related Information

[Properties of the Export Excel Technical Component \[page 167\]](#)

[Exporting to CSV \[page 162\]](#)

17.1 Properties of the Export Excel Technical Component

The properties of the [Export Excel](#) technical component specify all the settings associated with the export to Microsoft Excel output.

Export to Excel Properties

Export Excel Properties

Property Area	Property Value	Property Description
components	dialog	Select the ellipsis button to call the Edit Component Order dialog. Use the Insert and Remove buttons to add or remove crosstabs or data sources from the list. The Edit Component Order dialog allows you to select which crosstabs or data sources are included in the exported output to Microsoft Excel. Use the Move Up and Move Down buttons to change the order in which your crosstabs or data sources are exported to Microsoft Excel.
<div><div>i Note</div><div>The default component selection value is null. You should ensure that you have selected at least one crosstab or data source within the Edit Component Order dialog.</div></div>		
file name	none	Set the name of the exported Microsoft Excel file. If a value is not entered, the application name is used as the file name.

Property Area	Property Value	Property Description
selection	<ul style="list-style-type: none"> time of export system id query technical name static filters dynamic filters variables 	<p>Select the ellipsis button to call the Select Metadata dialog. This allows you to use checkboxes to select which types of metadata is used in your Microsoft Excel output.</p> <div> <p>i Note</p> <p>Only metadata associated with crosstabs is exported. Data source metadata is not exported.</p> </div>
location	<ul style="list-style-type: none"> none (default) separate sheet 	<p>Select None to exclude crosstab metadata from the Microsoft Excel export.</p> <p>Select Separate Sheet to export the crosstab metadata to a separate tab in the Microsoft Excel spreadsheet.</p>

Related Information

[Export Excel \[page 95\]](#)

18 Exporting to PDF

The export to PDF functionality allows you to export all, or part of an application to PDF.

Add the [Export PDF](#) technical component in the [Outline](#) view of an application to avail of the export to PDF functionality. Using the [Export PDF](#) technical component properties, you can configure the exact design and layout of all output exported to PDF. You can set properties that export to PDF in a report style, where components of your choice are export to PDF, one component per page. [Crosstab](#) components can be exported to one PDF page, or exported across multiple pages. How it exports depends on the size of the [Crosstab](#) when compared to the PDF page dimensions. Page orientation, paper size, location of the page number and the name of the file you are exporting are some other properties you can set. You can also decide which image and text you would like to see exported in your header and footer. The order in which the components in your application are exported can also be configured. You can choose whether to display an appendix, and whether to include in the appendix, some comments added in the document. In addition, you can choose which metadata should be displayed, and where it should appear in your output. If a crosstab is very wide, you can choose to auto size the crosstab column width, and/or fit the crosstab to the width of the PDF page. If the crosstab header text is very wide, you can choose to wrap the header text.

The following metadata can be selected for display in your export to PDF output:

- system ID
- query technical name
- static filters (filters that cannot be changed by the user)
- dynamic filters (filters that can be changed by the user, or applied at runtime)
- variables
- map layers

i Note

Map layer metadata appears in the appendix only.

The following components can be included in your export to PDF output:

- chart
- crosstab
- map
- sdk (Most SDKs are exported. For more information see the SAP Note [2460298](#).)

A set of scripting methods allows you to enable an application user to configure the export to PDF properties.

You can decide to design an application with a customized dialog box to allow the user to change their export to PDF property settings before exporting. The designer property values are set by default. Any selections made by the user overwrite any default values or values selected by the application designer in the [Export PDF](#) technical component designer properties.





Through scripting you can decide to export to PDF in the following output formats:

- View
- Report
- Panel View

For more information on the [Export PDF](#) technical component, you can refer to the chapters called *Export PDF Technical Component* and *Properties of the Export PDF Technical Component*.

i Note

- If you migrate a 1.6 application that contains legacy export to PDF functionality to Lumira Designer 2.0, add the [Export PDF](#) technical component to avail of the 2.0 functionality.
- If you choose to display a header, footer, appendix or comments in the appendix in the export to PDF, ensure that you have also set the [Header Visible](#), [Footer Visible](#) or [Appendix Visible](#) properties to true.
- If you choose to export to PDF in an export report style, ensure that you have also added the components to be included in the PDF report output by using the [Components](#) property in the [Report](#) area of the [PDF](#) technical component.
- Refer to the “SAP Lumira Sizing Guide” on the SAP Help Portal at <http://help.sap.com>.
- Before working with the export PDF feature, refer to the following SAP Notes:

SAP Note Number	Description
2460407 	CSS limitations with the Export PDF technical component in SAP Lumira Designer.
2460298 	Considerations when working with export PDF in SAP Lumira Designer.
2232036 	Export to PDF - browser issue when exporting very large crosstabs
2614784 	Chart text is not as sharp as the rest of the chart when exporting to PDF

Related Information

[Properties of the Export PDF Technical Component \[page 171\]](#)

[View Export to PDF \[page 176\]](#)

[Report Style Export to PDF \[page 176\]](#)

[Panel View Export to PDF \[page 181\]](#)

18.1 Languages and Export to PDF

The locale of the user's browser session determines the language in which text content is exported to PDF.

When configuring an application, it is important to be aware that the locale of the user's browser session determines the language in which text content is exported to PDF. Only one language can be exported to PDF, per application.

i Note

If your application contains text that does not match the user locale, this text may not export correctly.

To find out the latest information about the languages supported, you can refer to the SAP Lumira Product Availability Matrix available on the SAP Support page at <https://support.sap.com/en/index.html>. From the menu, select ► *Maintenance* ► *Overview* ► *Product Availability Matrix* ► *Display All Product Versions* , and select the latest version of SAP Lumira .

For more details on language settings in general, you can refer to the chapter called *Language of Master Data and Metadata* in the *Administrator Guide: SAP Lumira* on the SAP Help Portal at <https://help.sap.com/viewer/index>.

18.2 Properties of the Export PDF Technical Component

The properties of the *Export PDF* technical component specify all the settings associated with the export to PDF output.

Export to PDF Properties

Export PDF Properties

Property Area	Property Value	Property Description
settings	orientation <ul style="list-style-type: none">landscape (default)portrait	Sets how the export output is displayed. This property is set to Landscape by default.
	paper size <ul style="list-style-type: none">a2a3a4 (default)a5letterlegalledger/tabloid	Sets the paper size of the export output. This property is set to A4 by default.

Property Area	Property Value	Property Description
	page number location <ul style="list-style-type: none"> • none (default) • header • footer 	Sets the location of the page number in the export output. This property is set to None by default. <div> i Note If you choose to display the page number in the header or footer in the export to PDF, you must ensure that you have also set the Header Visible or Footer Visible property to true. </div>
	date location <ul style="list-style-type: none"> • none (default) • header and appendix • header • appendix 	Sets the location of the date in the export output. This property is set to None by default. <div> i Note If you choose to display the date in the header or appendix in the export to PDF, you must ensure that you have also set the Header Visible or Appendix Visible property to true. </div>
	file name	Allows you to enter a file name of your choice for the export output.
report	components	Ellipsis button that calls the Edit Report Order dialog. The Edit Report Order dialog allows you to set the order in which the components are displayed in the export output. <div> i Note If you choose to export to PDF in a report style, you must ensure that you have added the components to be included in the output by using the Insert option within this dialog. </div>

Property Area	Property Value	Property Description
	components to fit page width	<p>Specifies the components that are to fit the PDF page width. The PDF page width is determined by the Page Size and Orientation property values.</p> <div> i Note <p>Currently this property applies to the Crosstab component only. The components will only scale down to fit to page width. Components will not scale up.</p> </div>
	components with auto size column width	<p>Specifies the components that have auto size column width enabled when configuring a report style export to PDF.</p> <div> i Note <p>Currently this property applies to the Crosstab component only.</p> </div>
	wrap header text <ul style="list-style-type: none"> • none (default) • columns • rows • columns and rows 	<p>Wraps the column or row header text in a native crosstab when the header text is too long to fit by default. The header text is wrapped over a maximum of two lines.</p> <p>If the default value None is selected, the text is truncated and an ellipsis is added.</p>
appendix	appendix visible <ul style="list-style-type: none"> • true • false (default) 	<p>Sets whether the appendix is visible in the export output. This property is set to False by default.</p>

Property Area	Property Value	Property Description
	comments visible <ul style="list-style-type: none"> • true • false (default) 	<p>When the property <i>Comments Visible</i> is set to true, some comments that were applied to the application, are exported to a specific comments section of the PDF appendix. The types of comments exported are comments with the context types Context or None. For more information of the <i>Comments</i> technical component, you can refer to the chapter in this guide called <i>Working with Comments</i>.</p> <div> <p>i Note</p> <p>The <i>Appendix Visible</i> property must be set to the value true, to allow comments to appear in the appendix.</p> </div>
header	header visible <ul style="list-style-type: none"> • false (default) • true 	Sets whether the header is visible in the export output. This property is set to False by default.
	text	Sets the text that is displayed by default in the header of the export output.
	image	<p>Ellipsis button to allow you import a header image that is displayed by default in the header of the export output.</p> <div> <p>i Note</p> <p>This is always specified in the <i>Export PDF</i> technical component designer properties only.</p> </div>
footer	footer visible <ul style="list-style-type: none"> • false (default) • true 	Sets whether the footer is visible. This property is set to False by default.
	text	Sets the text that is displayed by default in the footer of the export output.
	image	<p>Ellipsis button to allow you import a footer image that is displayed by default in the footer of the export output.</p>

Property Area	Property Value	Property Description
metadata	selection	<p>Ellipsis button that calls the Select Metadata dialog that allows you to select the metadata types to be included in the export report output. The types of metadata you can select from include the following:</p> <ul style="list-style-type: none"> • system ID • query technical name • static filters • dynamic filters • variables • map layers <div> <p>i Note</p> <ul style="list-style-type: none"> • To export metadata successfully, you must also ensure to set the appropriate value in the Metadata > Location property. • Map layer metadata can only appear in the appendix. </div>
	location	<p>Sets the location of the metadata in the export output. This property is set to None by default.</p> <div> <p>i Note</p> <ul style="list-style-type: none"> • If you choose to display your metadata in the header or appendix in the export to PDF, you must ensure that you have also set the Header Visible or Appendix Visible property to true. </div>

Related Information

[Exporting to PDF \[page 169\]](#)

[Comments \[page 73\]](#)

[Working with Comments \[page 229\]](#)

18.3 View Export to PDF

A [View](#) export to PDF exports all elements within the application as displayed in the browser.

To export all elements within the application as displayed in the browser (or within the scrollable area) to PDF, you must apply the scripting method `exportView()`. By default, the optional parameter `isFullView` is set to true. The `isFullView` parameter ensures that the application is exported to PDF, even if some of the application is not visible in the viewport. If you want to export only the portion of the application visible within the viewport, you can change the value of this parameter to false.

The value of false is discouraged as it may hide relevant parts of the application.

18.4 Report Style Export to PDF

An [Export Report](#) style export to PDF exports the selected components in an application in a specific way.

You can use the [Export PDF](#) technical component properties to configure what components to include in a report style export to PDF of an application. You can then allow an application user to call this functionality at runtime by applying the scripting method `exportReport()`. In an [Export Report](#) style export to PDF, specific components contained within in the application are exported in the following ways:

- [Chart](#) and [Map](#) components are exported, one component per page.

i Note

When exporting a [Chart](#) component with the [Appendix Visible](#) property set to true, by default the [Chart](#) is identified in the appendix list by the query used in the chart. If you edit the [Chart](#) title at design time or runtime, the chart is identified in the appendix by the edited chart title. If you remove the chart title and leave the title blank, the [Chart](#) is identified in the appendix by the chart type, for example Column Chart. [Map](#), [SDK](#) and [Crosstab](#) components are identified in the appendix by their title, by default.

- Most [SDK](#) components export to PDF. For more information you should refer to the SAP Note [2460298](#).
- [Crosstab](#) components can be exported to one PDF page, or exported across multiple pages, depending on the size of the [Crosstab](#) when compared to the PDF page dimensions.

Related Information

[Configuring the Properties of a Crosstab for Report Style Export to PDF \[page 178\]](#)

[Using the Property to Select Components for Report Style Export to PDF \[page 177\]](#)

18.4.1 Using the Property to Select Components for Report Style Export to PDF

Use the [Components](#) property to select the components you want to include in a report style export to PDF.

Context

You have added the [Export PDF](#) technical component to the application, and the application contains components you wish to export to PDF in a report style. The analytical components to be exported to PDF are determined by setting properties in the design tool, or by scripting. For more information on scripting, you can refer to the chapter called *Using Scripting to Select Components for Report Style Export to PDF*

i Note

Ensure you select at least one component to export in a report style export to PDF.

Procedure

1. Select the [Export PDF](#) technical component from the [Outline](#) view of the design tool to activate the [Export PDF](#) properties tab.
2. Select the ellipsis button beside the [Components](#) property within the [Report](#) section.
The [Edit Report Order](#) dialog appears.
3. Select the [Insert](#) button.

The [Edit Report Selection](#) dialog appears.

4. Select the components you wish to include in the report style export to PDF.
5. Select [OK](#).

The selected components are listed in the [Edit Report Order](#) dialog.

6. Set the order in which the components are export to PDF, using the [Move Up](#) or [Move Down](#) buttons.
7. Select [OK](#).

The selected components are listed in the [Value](#) column of the [Components](#) property, in the order selected.

18.4.2 Using Scripting to Select Components for Report Style Export to PDF

Use the `setReportComponents()` ; scripting method to select the components you want to include in a report style export to PDF.

The analytical components to be exported to PDF can be determined by scripting, or by setting a property in the design tool. For more information on setting the property, you can refer to the chapter called *Using the*

Property to Select Components for Report Style Export to PDF. Ensure you select at least one component to export in a report style export to PDF. The types of components that can be exported are [Chart](#), [Map](#), [Crosstab](#), and most [SDK](#) components.

Sample Code

```
EXPORT_PDF.setReportComponents(<arguments>)
```

where arguments can be a single exportable component id:

Sample Code

```
EXPORT_PDF.setReportComponents(CHART_1);
```

or if it is intended to export more than one component, an array of exportable component ids

Sample Code

```
EXPORT_PDF.setReportComponents([CHART_1, CROSSTAB_1]);
```

Ensure that `setReportComponents()` is called before `exportReport()`, where interaction with the analytic component for export also leads to the report export being created. Ensure that you position `EXPORT_PDF.setReportComponents(<arguments>)` before `EXPORT_PDF.exportReport()` in the components [On Click](#) or [On Select](#) property.

18.4.3 Configuring the Properties of a Crosstab for Report Style Export to PDF

You can configure the [Export PDF](#) properties of the [Crosstab](#) component, to achieve different output results when exporting to PDF in a report style.

The following [Export PDF](#) properties are available to you, to configure the way in which the [Crosstab](#) is exported:

- Auto Sizing Column Width
- Wrapping Headers and Auto Sizing Column Width
- Components to Fit Page Width

Related Information

[Auto Sizing Column Width \[page 179\]](#)

[Using the Property to Set Auto Size Column Width \[page 179\]](#)

[Using Scripting to Set Auto Size Column Width \[page 180\]](#)

[Wrapping Headers and Auto Sizing Column Width \[page 180\]](#)

[Components to Fit Page Width \[page 180\]](#)

18.4.3.1 Auto Sizing Column Width

You can decide which crosstabs are exported with auto sized column widths.

You can use the *Components with Auto Size Column Width* property of the *Export PDF* technical component at design time, to select which crosstabs in an application are exported with column widths that are auto sized. When a crosstab has auto sized column widths, it means that the column widths are based on the values or texts within the columns. The columns adjust to fit the longest text.

Header titles are considered in determining the column width, as well as cell content. Columns within the PDF report style output are then not of a fixed and equal width. Using the auto size column width feature means that columns do not truncate the values or text, and unnecessarily wide columns do not waste valuable space. However, if the maximum column width, which is automatically set by the system by default, is reached, the text in the column will truncate. Also, if wrapping is enabled, the text in the column will wrap.

Related Information

[Wrapping Headers and Auto Sizing Column Width \[page 180\]](#)

18.4.3.1.1 Using the Property to Set Auto Size Column Width

Optimize how the content of crosstab content is displayed when exported to PDF in a report style.

Context

You have at least one *Crosstab* in an application, and you want to adapt the column widths to accommodate the longest text in the column when exporting in a report style to PDF.

Procedure

1. Select the *Export PDF* technical component from the *Outline* view of the design tool to activate the *Export PDF* properties tab.
2. Select the ellipsis button beside the *Components with Auto Size Column Width*.
The *Select components* dialog appears.
3. Using the checkboxes beside each component, select the crosstabs for which you want to enable column auto sizing.
4. Use the *Select All* option if you want to select all crosstabs.
5. Select *OK*.

The selected crosstab components are listed in the [Value](#) column of the [Components with Auto Size Column Width](#) property, in the order selected.

Next Steps

You can use scripting to allow the crosstabs selected to display dynamic, or auto sized column widths when the application user selects to export to PDF a report style version of the application.

18.4.3.1.2 Using Scripting to Set Auto Size Column Width

Select the components to which you want to apply auto sized column widths in a report style export to PDF.

Use the `setComponentsToAutoSizeColumnWidth()` ; scripting method to select the components to which you want to apply auto sized column widths in a report style export to PDF.

Sample Code

```
EXPORT_PDF.setComponentsToAutoSizeColumnWidth();  
EXPORT_PDF.setComponentsToAutoSizeColumnWidth([CROSSTAB_1,CROSSTAB_2]);  
EXPORT_PDF.exportReport();
```

18.4.3.2 Wrapping Headers and Auto Sizing Column Width

You can define the behavior of header wrapping and column widths.

A maximum size for the column width is automatically set by the system by default. When the maximum column width limit is reached, the column header text will wrap according to the settings in the [Wrap Header Text](#) property of the [Export PDF](#) technical component. If wrapping and auto sized column widths functionality is enabled, the title will wrap over two lines only, once the maximum column width has been reached. If after that there is still remaining text, it will truncate.

18.4.3.3 Components to Fit Page Width

[Crosstab](#) components can be scaled to fit the PDF page width.

The size and width of the PDF are based on the settings in the [Page Size](#) and [Orientation](#) properties. If there are no components selected to fit to page width, and if the [Crosstab](#) does not fit the width of the page, the [Crosstab](#) is split across multiple pages.

If you set a crosstab to have auto sized columns widths, using the [Components with Auto Size Column Width](#) property, this could create the requirement to split columns across multiple pages in the report style export to PDF. The page split may not be desirable, as you lose the overview of the crosstab content on a single page. To achieve an overview of the entire crosstab on one page, you use the [Components to Fit Page Width](#) property.

If the *Export PDF* properties *Components to Fit Page Width* and *Components with Auto Size Column Width* are set on the same *Crosstab* simultaneously, the exported PDF will fit the whole *Crosstab* on one page, and allow the column widths to auto size.

To achieve an overview of the entire *Crosstab* on one page, you use the *Components to Fit Page Width* property. The *Crosstab* may still be split horizontally if there are sufficient amount of rows, so even with *Components to Fit Page Width*, the crosstab may occupy multiple pages.

18.4.3.4 Exporting Hierarchies within a Crosstab Component for Report Style Export to PDF

You can export a crosstab with collapsed or expanded hierarchies on the column headers and row headers.

If your crosstab contains hierarchies on the column headers and row headers, these hierarchies are also shown when exporting to PDF in a report style. When the hierarchy is collapsed in your crosstab at the time of export to PDF, the output displays the indentation and expansion symbols. When the hierarchy is expanded at the time of export, this too is reflected in the export to PDF.

→ Tip

When exporting expanded hierarchies, the total height required by column headers, or the total width required by row headers increases. We recommend that you configure the properties of the *Export PDF* technical component in such a way as to provide as much capacity as needed to accommodate the expanded crosstab hierarchies.

18.5 Panel View Export to PDF

A *Panel View* export to PDF exports a panel or multiple panels to a PDF. The exported content looks like a screenshot.

To export the contents of a *Panel* component, or an array of *Panel* components as they are displayed in the browser, you must apply the scripting method `exportPanelView(PANEL_1)`; or `exportPanelView([PANEL_1, PANEL_2])`. Each exported *Panel* component is exported on a page of its own in the PDF output.

The scripting method `exportPanelView()` includes an optional parameter called `isFullPanel`, that allows the application designer to export the full view of the panel to PDF. This optional parameter is set to true by default. The application designer can opt to change the optional parameter value to false. Changing the parameter to false means that only the section of the application that is visible in the *Panel* is exported to PDF. When you open an application that is already scripted with the previous version of `exportPanelView()`, the parameter `isFullPanel` is set to true by default. With the `isFullPanel` set to true, the entire contents of the panel is exported to PDF.

18.6 Exporting Comments

You can export to a PDF appendix, some comments added to a Lumira documents (.lumx files).

To export some comments set in a document, you must set the *Comments Visible* property of the *Export PDF* technical component to the value true. When you export the application, certain types of comments will then be included in the PDF appendix under a heading of Comments. When creating comments in a document, there are different contexts in which the comments apply. The types of comments exported to PDF from the document, are ones with the comment context type Context or *None*. For more information of the different comment context types, you can refer to the chapter in this guide called *Comment Context Types*.

Related Information

[Comment Context Types \[page 231\]](#)

[Working with Comments \[page 229\]](#)

[Exporting to PDF \[page 169\]](#)

18.7 Selecting Content for the Appendix

You can choose whether to display an appendix, and what should be included in it.

Context

You have created an application with the *Export PDF* technical component included in the *Outline* view.

Procedure

1. Select the *Export PDF* technical component from the *Outline* view.
The *Export PDF* properties appear in the design tool.
2. Set the *Appendix Visible* property to true.
3. Set the *Comments Visible* property to true, if you want to display some comments that were applied to the application. The types of comments exported are comments with the context types Context or None.
The comments are exported to a specific comments section of the PDF appendix.
4. To choose the metadata you want to appear in the appendix, use the *Selection* property ellipsis button to call the *Select Metadata* dialog box.

5. Select the metadata types for your export to PDF. The options available include: System ID, Data Sources, Static Filters, Dynamic Filters, Variables, or Map Layers.
6. Select *OK*.

The *Selection* property value lists the selections you have made.

7. Use the *Location* property to select where the metadata you have selected in step 5 should appear in your PDF export. In this instance, select Appendix or Header and Appendix.

For more information on adding content to the headers, you can refer to the topic called *Selecting Content for Headers*.

Related Information

[Selecting Content for Headers \[page 183\]](#)

18.8 Selecting Content for Headers

You can choose whether to display a header, and what should be included in it.

Context

You have created a document with the *Export PDF* technical component included in the *Outline* view.

Procedure

1. Select the *Export PDF* technical component from the *Outline* view.
The *Export PDF* properties appear in the design tool.
2. Set the *Header Visible* property to true.
3. Enter the text you want to appear in header of your export to PDF, using the *Text* property value free text field.
4. You can also import an image of your choice, using the ellipsis button of the *Image* property.
5. To choose the metadata you want to appear in the header, use the *Selection* property ellipsis button to call the *Select Metadata* dialog box.
6. Select the metadata types for your export to PDF. The options available include: System ID, Data Sources, Static Filters, Dynamic Filters, Variables, or Map Layers.
Map layer metadata can only appear in the appendix.
7. Select *OK*.

The [Selection](#) property value lists the selections you have made.

8. Use the [Location](#) property to select where the metadata you have selected in step 6 should appear in your export to PDF. In this instance, select Header or Header and Appendix.

For more information on adding content to the appendix, you can refer to the topic called *Selecting Content for the Appendix*.

Related Information

[Selecting Content for the Appendix \[page 182\]](#)

18.9 Exporting Components within Composites to PDF

Components within composites can be exported to PDF.

You can include components within composites when exporting an application to PDF. To export composite components to PDF, you must add just one [Export PDF](#) technical component on any level within the application. Components within composites for export to PDF can be selected in two ways:

- The first way is to use the [Edit Report Order](#) dialog of the [Components](#) property of the [Export PDF](#) technical component.
- The second way is to select components within composites from any scope in the application by using scripting.

Using the [Edit Report Order](#) dialog to select components

The [Edit Report Order](#) dialog displays for selection only the components that are saved within the same composite scope. For more information on using the [Edit Report Order](#) dialog, you can refer to the chapter called *Using the Property to Select Components for Report Style Export to PDF*.

Using scripting to select components

With scripting, you can create a communication channel between the composites in the application. Create a function to retrieve your chosen components from the composite. This function can pass a reference from one level of the application to another, and returns the components you want to export to PDF. The content assist will show you the function that you created to return the components from the composite. For an example of how scripting can be used to export composite components to PDF, you can see below:

Sample Code

First you need to create a function in the composite. This returns the components you wish to export from the composite.

```
Function name - getChosenComponents  
Return type "Component", array  
Code: var myComponents = [CHART_1, CROSSTAB_1];  
return myComponents;
```

Sample Code

Now you can call the function on the composite from the main application.

```
var compositeComponents = MY_COMPOSITE.getChosenComponents();  
EXPORT_PDF.setReportComponents(compositeComponents);  
EXPORT_PDF.exportReport();
```

Related Information

[Using the Property to Select Components for Report Style Export to PDF \[page 177\]](#)

[Exporting to PDF \[page 169\]](#)

18.10 Scheduling an Export to PDF using Variants

You can schedule a Lumira Designer application to export to PDF using variants.

Context

Note

To schedule an export to PDF in a Linux OS environment, you must install the Google Chrome web browser. The certification of the OS is based on the latest available version of the Google Chrome browser. For SUSE Linux Enterprise Server, you might require extra openSUSE packages to install the Google Chrome web browser. So, we recommend that you pay particular attention when adding the openSUSE package dependencies.

Procedure

1. Add the [Export PDF](#) technical component to your application.
2. To edit the script in the [On Scheduling](#) event, use the [On Scheduling](#) property of the application. Enter `EXPORT_PDF.exportView();` in the script editor.
3. Set the [Variant Mode](#) property to the value Document variants, within the [Prompts](#) application property area.
4. Set the [Force Prompts on Startup](#) property to the value false, within the same property area.
5. Add the global variable **XBIVariantName**.
6. Set the [Expose as URL Parameter](#) property to the value true, within the [Global Variable](#) property area.
7. To edit the script in the [On Variable Initialization](#) event, use the [On Variable Initialization](#) property of the application. Enter the following in the script editor:

```
APPLICATION.applyDocumentVariant(XBIVariantName);
```

8. Edit the script in the [On Startup](#) event script editor as follows:

```
//APPLICATION.applyDocumentVariant(XBIVariantName);  
if (XBIVariantName) {  
}else {  
    APPLICATION.openPromptDialog();  
}
```

Related Information

[Working with Variants \[page 255\]](#)

19 Changing an Analysis Application

Context

Depending on the start up mode of your design tool, follow the relevant steps to open, change and save an existing application

Procedure

1. Open an existing application:
 - If your work in the Lumira Documents mode, click **File > Open Application or Composite...** in your design tool or open the appropriate document in your **Documents** view where your application is listed. Right-click the application and choose **Open** or double-click the application to open it.
 - if you work in the BI platform legacy mode, 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. If you work
3. Save the changed application or click **Application > Save As** to save the changed application using a different name. if you work in Lumira Documents mode, the new application (with changed name) is listed within the document's tree structure.

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 tot the type of modification.

Related Information

[Saving an Application Using a Different Name \[page 517\]](#)

19.1 Saving an Application Using a Different Name (Lumira Documents Mode)

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

Choose ► **File** ► **Save As...** or right-click the application and choose **Save As...**. The **Save Application As** dialog box opens. The system suggests the current document to which your application belongs and the current name for the application.

⚠ 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 Saving an Application Using a Different Name (BI Platform Legacy Mode)




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

If you work in the BI platform legacy mode, 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 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.

20 Deleting an Analysis Application

Prerequisites

You have opened the application you want to delete.

Procedure

1. Depending on the start up mode of your design tool, follow the appropriate steps:
 - If you work in Lumira Documents mode, click ► [File](#) ► [Delete](#) ► or right-click the application you want to delete and choose [Delete](#)
 - If you work in BI platform legacy mode, 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 search.
2. Select in the menu of the Lumira Designer ► *Search* ► *Search Application...* . The *Search Application* dialog box is displayed.
3. Enter the 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.

22 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 [Categories](#).
The [Categories](#) dialog box appears.
3. Choose the mobile category.
4. Click [OK](#).

23 Creating Planning Applications

With SAP Lumira Designer 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 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 Lumira Designer with BI Platform 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

i Note

Input-enabled queries are not supported for mobile solutions.

For further information about administration tasks for planning applications, see “Administration tasks for Planning Applications” in the *Administrator Guide: SAP Lumira* on the SAP help portal at <https://help.sap.com>.

Related Information

[Entering Data in the Crosstab \(Manual Planning\) \[page 195\]](#)

[Using Planning Functions and Sequences \(Automated Planning\) \[page 196\]](#)

[Selecting a Planning Connection \[page 194\]](#)

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 Lumira Designer'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 at <https://help.sap.com>.

Related Information

[Selecting a Data Source \[page 117\]](#)

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 <https://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 Links

See the method `configureInputReadiness` in the *Component API Reference* of SAP Lumira on the Help Portal at <https://help.sap.com>.

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.

		Amount	Comment (Amount)	Quantity	Comment (Quantity)
Cost Center	Cost Element	EUR		KG	
CCT_000001	1/CE00000001	250.00	A comment		
	1/CE00000002	250.00	Test		
	Result	500.00	NOT_EXIST		
CCT_000002	1/CE00000001	0.00	Asd	500.000	good
	1/CE00000002			500.000	good
	Result	0.00	Asd	1,000.000	good
Overall Result		500.00	NOT_EXIST	1,000.000	good
<input type="text"/>	<input type="text"/>		<input type="text"/>		<input type="text"/>
<input type="text"/>	<input type="text"/>		<input type="text"/>		<input type="text"/>

Example

Note

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

- Editing short texts in queries in Lumira Designer 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 <https://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 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 in the SAP NetWeaver library on SAP Help Portal at <https://help.sap.com>:

- “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 Objects* *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 edit the properties of the planning object or sequence and 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

Related Information

[Properties of Planning Functions and Planning Sequences \[page 198\]](#)

23.4.1 Properties of Planning Functions and Planning Sequences

Planning functions and planning sequences have several properties.

Planning Function Properties

Property	Property Value	Property Description
load in script	false, true	Specifies whether the planning function is initialized in the script, instead of automatically when the application is started.
process changed data	false, true	Specifies whether only data that was changed by the user in the current session since the last save will be processed.
aggregation level		Specifies which aggregation level is used as a filter to define the changed data. If no value is specified, the function's aggregation level defined in the backend is used.

Planning Sequence Properties

Property	Property Value	Property Description
load in script	false, true	Specifies whether the planning function is initialized in the script, instead of automatically when the application is started.
process changed data	false, true	Specifies whether only data that was changed by the user in the current session since the last save will be processed.

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. The user can choose whether a single cell or multiple cells belonging to a row or column header should be locked.
- 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 (Lumira Designer).

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 <https://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 <https://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 (Lumira Designer).

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

i 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 <https://help.sap.com>.

23.5.3 Locking Cells

Context

You can lock input-ready crosstab cells and total cells in planning applications. In addition, you can also lock all data cells within a selected row or column header in the crosstab. 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. Decide if you want to lock a single data cell or several data cells within a selected tuple.
 - Select the crosstab cell you want to lock.
 - Select the header row or header column containing the data cells that you want to lock. If you click on a cell that spans multiple rows, all contained rows will be locked for entry.

The selected cells must be input-enabled and must not have been changed since the last recalculation.

2. Open the context menu of the selected cell or row or column header.
 - Choose Lock Cell for single data cells.
 - Choose Lock Row/Lock Column for data cells within the row or column header.
 - If the drilldown state of the crosstab is changed (for example, by swapping the axis) the locks will be kept on the cells and - in the case of row/header locking - not kept on the rows/headers. For example, if row 1 (column1) was locked before the drilldown was changed, there is no guarantee that row 1 (column 1) will still be locked after the change, as the cells that were previously in that row (column) could now be in a new position.
 - If you want to unlock the cell/row header/column header, select it and open the context menu. You can now deselect the menu entry *Lock cell* or choose *Unlock Row/Unlock Column*.

0BCPL_PRO	0BCPL_PLANT	Calendar Year/Month	0BCPL_DADEC	0BCPL_INT	0BCPL_NUM	0BCI
		01.2003	10,000,040	15	1.000	66.000
PROD00001	PLANT001	Sort	0/00	0	0.000	33.000
		Display	NOT_EXIST	15	1.000	
PROD00002	PLANT001	Attributes	,000,040	1	1.000	1.500
		Expand All	,000,040	1	1.000	1.500
PROD00003	PLANT001	Collapse All	NOT_EXIST	2	2.000	
		Expand to Level	,000,040	1	1.000	1.500
Result		Select Hierarchy	,000,040	1	1.000	1.500
	PLANT002	Totals	NOT_EXIST	2	2.000	
PROD00012	PLANT001	Keep Member	NOT_EXIST	19	5.000	
		Keep Member on Axis	,000,040	1	1.000	1.500
		Filter Member and Swap With...	,000,040	1	1.000	1.500
		Filter Members	,000,040	1	1.000	1.500
		Select All Members	NOT_EXIST	2	2.000	
		Filter by Measure				
		Suppress Zeros in Rows				
		Compact Display in Rows				
		Drilldown by...				
		Swap With...				
		Remove Drilldown				
		Swap Axes				
		Lock Row				
		Unlock Row				
		Conditional Formatting				

Results

The selected cells are locked for changes and are displayed with a lock icon .

			0BCPL_DADEC	0BCPL_INT	0BCPL_NUM	0BCPL_NUM
0BCPL_PRO	0BCPL_PLANT	Calendar Year/Month				
PROD00001	PLANT001	01.2003	10,000,040	15	1.000	
		02.2003	0/0/00	0	0.000	
		Result	NOT_EXIST	15	1.000	
PROD00002	PLANT001	01.2003	10,000,040	1	1.000	1.500
		02.2003	10,000,040	1	1.000	1.500
		Result	NOT_EXIST	2	2.000	
PROD00003	PLANT001	01.2003	10,000,040	1	1.000	1.500
		02.2003	10,000,040	1	1.000	1.500
		Result	NOT_EXIST	2	2.000	
Result			NOT_EXIST	19	5.000	
PROD00012	PLANT002	01.2003	10,000,040	1	1.000	1.500
	PLANT001	01.2003	10,000,040	1	1.000	1.500
		02.2003	10,000,040	1	1.000	1.500
		Result	NOT_EXIST	2	2.000	

24 Working with an Application List

Use the [Application List](#) technical component to return a list of bookmarks created in other documents and applications.

You can apply the [Application List](#) technical component to an application to avail of the functionality that allows the user to return a list of bookmarks. The bookmarks that are returned by the [Application List](#) can come from multiple documents and applications. Using this component, you can create an application from where the user can to access and launch their list of bookmarks. When they select a bookmark from the list, the application that contains that bookmark is launched.

You can apply filters to refine the list. The filter type Access Type is a mandatory filter. This filter determines if the listed bookmarks are Personal or Global. You can also filter the list of bookmarks returned by Folder or Owner. Both Folder and Owner are optional filters.

For each bookmark listed in the [Application List](#) component, a list of bookmark properties can also be returned. Bookmark properties are based on the [BookmarkProperties](#) constant, and include the following:

- id
- title
- description
- group
- folder
- accessType
- readOnly
- creationTime
- lastModifiedTime
- documentTitle
- owner

Related Information

[Returning a Bookmark Url \[page 204\]](#)

[Returning a Bookmark List \[page 205\]](#)

24.1 Returning a Bookmark Url

You can apply scripting to return a bookmark Url in the [Application List](#).

Using the [Application List](#) technical component, the `getBookmarkUrl()` scripting method can be used to return a Url, which will open the bookmark in the application in which it was originally saved.

Sample Code

```
APPLICATION_LIST.getBookmarkUrl(id);
```

Related Information

[Returning a Bookmark List \[page 205\]](#)

[Working with an Application List \[page 204\]](#)

24.2 Returning a Bookmark List

You can apply scripting to return a list of bookmarks in the *Application List*.

The `getBookmarkProperties()` scripting method can be used to get a list of bookmarks on the system. You can filter bookmarks by *Access Type*, *Owner Id*, and *Folder Id*.

Access Type is a mandatory filter, while *Owner Id* and *Folder Id* are optional. To get a list of all global bookmarks on the system, and return an array of *BookmarkProperties*, you could apply the following code sample:

Sample Code

Get all global bookmarks

```
var array = APPLICATION_LIST.getBookmarksProperties(AccessType.GLOBAL);
```

To get a list of global bookmarks, created by a certain user, saved to a certain folder, you could apply the following sample code:

Sample Code

Get bookmarks with filters

```
var array = APPLICATION_LIST.getBookmarksList(AccessType.GLOBAL, {"ownerId":  
"AWdlVsMna9lMkV6EwrpdHYC", "folderId": "ASmaexDK4CBEnmDkV6u6U6Y"})
```

Extra optional filters can be added as part of a JSON object.

Related Information

[Working with an Application List \[page 204\]](#)

[Returning a Bookmark Url \[page 204\]](#)

25 Working with the Adaptive Layout

Use the [Adaptive Layout](#) to create an adaptive application that can be viewed on multiple devices.

The [Adaptive Layout](#) contains a series of blocks, which have a defined colspan per viewport. You can add components into these blocks to build up an application. While designing your application, define the colspan of each block for each viewport. The [Adaptive Layout](#) rearranges blocks, according to the colspan of the blocks and the current viewport. The viewport that is currently active depends on the current width of the [Adaptive Layout](#) component. You can select to preview the different viewports at design time. This offers you the flexibility to create one application that can then be used across multiple devices.

Using the context menu of the [Adaptive Layout](#) you can select the [Preview of Viewport](#) context menu item and select one of the following options:

- auto
- small
- medium
- large
- extra large

Related Information

[Defining a Viewport \[page 206\]](#)

25.1 Defining a Viewport

A viewport is the visible area of an [Adaptive Layout](#) container. The viewport will change size depending on the width of the [Adaptive Layout](#) container.

Context

You have created your application within an [Adaptive Layout](#) component.

Procedure

1. Select the [Adaptive Layout](#) from within the [Outline](#) view.

2. Using the *Minimum Width of Medium Viewport* property of the *Adaptive Layout* component, define the pixel width at which your application layout transitions from a small to a medium viewport.
3. Using the *Minimum Width of Large Viewport* property of the *Adaptive Layout* component, define the pixel width at which your application layout transitions from a medium to a large viewport.
4. Using the *Minimum Width of Extra Large Viewport* property of the *Adaptive Layout* component, define the pixel width at which your application layout transitions from a large to an extra large viewport.
5. Using the context menu, select *Preview of Viewport*.
6. Select one of the options available:

auto	dynamically updates the layout of your application depending on the size of the canvas at design time
small	provides a preview of how your application will look on a small viewport
medium	provides a preview of how your application will look on a medium viewport
large	provides a preview of how your application will look on a large viewport
extra large	provides a preview of how your application will look on an extra large viewport

Related Information

[Working with the Adaptive Layout \[page 206\]](#)

25.2 Adding Blocks to the Adaptive Layout

You can add blocks to the *Adaptive Layout* component.

Context

You have added the *Adaptive Layout* component to your application. No blocks are present in the *Adaptive Layout* in the initial state.

Procedure

1. Select the *Adaptive Layout* component from the *Layout* area.
2. Using the *Context Menu* select ► *Create* ► *Block* ►.

A block is added to the *Adaptive Layout* in the top lefthand corner of the component.

3. Repeat step 2 until to have the number of blocks you require for your application. Blocks are added to the *Adaptive Layout* in a linear pattern.

Related Information

[Adding Components to Blocks \[page 208\]](#)

[Properties of an Adaptive Layout Block \[page 209\]](#)

25.3 Adding Components to Blocks

You can add components to the blocks in the *Adaptive Layout*.

Context

You have the *Adaptive Layout* component added to the application, and you have added at least one block.

Procedure

1. Add a component to the *Adaptive Layout* in one of the following ways:
 - From the *Components* view, drag and drop a component near a block of your choice.
 - From the *Outline* view, use the *Block* context menu item *Create* and then select a component type.

The component is added to the block.

2. To make sure that the component fills the entire block, use the *Maximize Selected Component* menu option. The component fills the entire block in the case where you drop only one component. If you drop several components in a single block, the components do not extend beyond the block.

Related Information

[Adding Blocks to the Adaptive Layout \[page 207\]](#)

[Properties of an Adaptive Layout Block \[page 209\]](#)

25.4 Moving Blocks

Blocks can be moved around within the [Outline](#) view.

The order of blocks can be changed only in the [Outline](#) view, by dragging blocks up and down. The order of blocks displayed in the [Adaptive Layout](#) reflects directly the order in the [Outline](#) view. This applies for all viewports.

Related Information

[Resizing Blocks \[page 210\]](#)

[Adding Blocks to the Adaptive Layout \[page 207\]](#)

25.5 Properties of an Adaptive Layout Block

The blocks of the [Adaptive Layout](#) can be modified using colspan properties.

The blocks of the [Adaptive Layout](#) have the following specific properties:

Property Type	Property Value	Property Description
small colspan	12	Sets the column span of the block for the small viewport.
medium colspan	6	Sets the column span of the block for the medium viewport.
large colspan	3	Sets the column span of the block for the large viewport.
extra large colspan	3	Sets the column span of the block for the extra large viewport.

Related Information

[Adding Blocks to the Adaptive Layout \[page 207\]](#)

[Working with the Adaptive Layout \[page 206\]](#)

25.6 Resizing Blocks

You can resize blocks in the [Adaptive Layout](#).

The component you drop onto the block takes the height of existing blocks on that row. If the 12 blocks are full in a row, the [Adaptive Layout](#) component will push blocks underneath. The [Adaptive Layout](#) is like a notepad with 12 characters per line. When 12 characters are taken, it goes to the next line. If you resize a block, only that block is resized. The container will look at the blocks in order and add them next to each other. If a block requires a 13th column, a new row will be added.

Related Information

[Adding Blocks to the Adaptive Layout \[page 207\]](#)

26 Working with Bookmarks

Bookmarks serialize the state of an analysis application at a certain point in time. You can define exactly which elements of an application to be included in a bookmark.

To serialize the state of a Lumira Designer analysis application at a certain point in time, you can save a bookmark to the application. When you save a bookmark, the state of the application persists on the platform backend.


Bookmarks only persist the state of the application. Content from the BiApp file or scripts are not included in the bookmark. Bookmark content is versioned. The version of the bookmark is maintained by the application designer. You can search, rename, group, move, and delete bookmarks. Bookmarks have a unique ID. Any additions made to the bookmark definition is included in the [Find References](#) and application global [Search](#).

You can specify which parts of an application are included in a bookmark. You can configure a bookmark to contain any of the components used in an application. Components that are saved as part of a composite can also be selected for inclusion in an application bookmark. If you change a component property that is configured to be included in a bookmark, that change is saved as part of the bookmark. You can save changes down to component level in an application.

i Note

1. It is important to consider security when working with bookmarks. For more information, you can refer to the chapter in this guide called *Security Considerations with Bookmarks*.
2. Bookmarks can be saved in local mode for testing and development purposes.
3. When working in local mode, bookmark URLs can only be loaded within a browser, which is launched from the user's own analysis application, in the following circumstances:
 - While the designer is running
 - In a different tab within the same browser
 - In a different browser as long as the designer is running and the designer session ID is manually added to the URL generated by the script method. Bookmark URLs generated in local mode do not include this transient session ID as the ID changes between designer sessions.
4. When working in local mode, bookmarks only load within the same Lumira document. You cannot load bookmarks from a different Lumira document. In BI platform mode, bookmarks can be loaded from one Lumira document to another.
5. On Startup scripts and On Variable initialization do not run when loading bookmarks in an analysis application. This prevents startup scripts and variable initialization settings from overriding standard bookmark settings. This applies also to all bookmark types when they are being loaded via URL.
6. Planning functionality is not supported by 2.x bookmarks.
7. Before working with bookmarks, refer to the following SAP Notes:

SAP Note Number	Description
2168984 	Considerations when working with Standard/Fragment/Portable Fragment Bookmarks in Design Studio or Lumira Designer.

SAP Note Number	Description
2157239 	Performing more than one free text search from your bookmark filter list will return zero items.

Related Information

[Bookmarks Technical Component \[page 212\]](#)

[Security Considerations with Bookmarks \[page 212\]](#)

26.1 Security Considerations with Bookmarks

Avoid saving sensitive data when creating a bookmark definition.

In the interest of data protection and security, we recommend that you always pay particular attention to the type of data being saved as part of a bookmark definition. No personal or sensitive data should be included when you are defining the contents of a bookmark. It is important to bear in mind where bookmarks are saved. When saving a bookmark definition, we suggest that you specify the folder on the platform into which the bookmark is saved. By default, the *Bookmarks* technical component property *Folder* saves the bookmark to the root folder on the platform. Contents of the root folder are visible to everyone. When saving bookmarks to a folder on the platform, also be aware that folders and folder permissions can change over time.

We recommend that administrators create a well-defined folder tree structure, with clear titles and descriptions. A well-defined folder structure will help you decide where best to save the bookmarks you create.

Consistent awareness of content sensitivity, and access to folders will help you avoid security concerns when working with bookmarks.

26.2 Bookmarks Technical Component

The *Bookmarks* technical component provides all the functionality required to bookmark an analysis applications.

All Lumira Designer bookmark functionality is consolidated into a technical component called *Bookmarks*. To access bookmark functionality simply add the *Bookmarks* technical component to an application. To add this technical component, select *Bookmarks* from the context menu of the *Technical Components* section. The *Technical Components* section is in the *Outline* view of an application.

The *Bookmarks* technical component contains:

- A set of *Properties* which define all characteristics of a bookmark. For more information about the properties available, you can refer to the chapter called *Properties of the Bookmarks Technical Component*.

- [On Load](#) event, which is triggered on loading a bookmark.
- Scripting methods to define all the actions associated with working with bookmarks.

i Note

- If you rename one of the [Bookmarks](#) technical components, you can still access the bookmarks stored within that technical component.
- When you delete a [Bookmarks](#) technical component, the technical component, and all the bookmarks associated with it, are deleted from the application. However, these bookmarks are still stored on the platform.

Related Information




[Working with the Outline View \[page 69\]](#)

[Properties of the Bookmarks Technical Component \[page 214\]](#)

[Multiple Bookmarks Technical Components \[page 213\]](#)

26.3 Multiple Bookmarks Technical Components

You can bookmark different sections of an application by having multiple bookmark technical components.

To add another bookmark technical component, simply select  [Create](#)  [Bookmarks](#)  from the [Technical Components](#) context menu. The ability to create multiple technical components leads to more flexibility and the ability to create more complex applications.

Identical bookmark definitions can be saved and loaded in different [Bookmarks](#) technical components. For example, you can create a [Global Variable](#) definition and add it to two or more bookmark definitions, from different [Bookmarks](#) technical components. Saving a bookmark definition in one [Bookmarks](#) technical component does not exclude it from other [Bookmarks](#) technical components.

Related Information

[Bookmarks Technical Component \[page 212\]](#)

26.4 Properties of the Bookmarks Technical Component

The properties of the *Bookmarks* technical component specify all the characteristics of the bookmarks created within a particular *Bookmarks* technical component.

General Properties of the Bookmarks Technical Component

Property	Property Value	Property Description
group	none	Allows you to specify the group, to which all bookmarks created in the current bookmark technical component are assigned. Alternatively, you can allow the end user to assign these bookmarks to a group using scripting.
definition	none	Specifies the components, data sources, technical components, and global variables to be included when defining bookmarks. Selecting the ellipsis button, calls the <i>Edit Bookmark Definition</i> dialog.
folder	none	Specifies the folder on the platform in which bookmarks are stored. Selecting the ellipsis button, calls the <i>Select Folder</i> dialog. <div>i Note<p>By default, the <i>Folder</i> property saves the bookmark to the root folder on the platform. For security reasons, we recommend that you specify exactly to which folder you want the bookmark to be saved.</p></div>
access type	personal, global	Specifies whether a bookmark is visible to all users, or only visible to, and editable by, the user that created it.
sort order	ascending, descending	Specifies whether bookmarks are sorted in an ascending or descending order. The default value is Descending.
sort by	title, created on, last modified	Specifies whether bookmarks are sorted by the date they were created, title or the date they were last modified. The default value is Created On.

Property	Property Value	Property Description
version	number	Specifies the version of the current bookmark. Change the version number when a change is made to a bookmark that renders it obsolete.
save with parent bookmark	boolean	The Save with Parent Bookmark property enables or disables the saving to the bookmark definition, of components within composites.

Related Information

[Bookmarks Technical Component \[page 212\]](#)

[Working with Bookmarks \[page 211\]](#)

[Defining Bookmarks \[page 215\]](#)

26.5 Defining Bookmarks

Select the parts of the application to be included in a bookmark definition.

You can select the parts of the application to be included in the bookmark from the context menu of the [Bookmarks](#) technical component in the [Outline](#) view, or by using the [Definition](#) property of the [Bookmarks](#) technical component. For more information, you can refer to the chapters [Using the Context Menu to Define Bookmarks](#) and [Using the Definition Property to Define Bookmarks](#).

⚠ Caution

It is important to carefully consider which elements of the application should be included in the bookmark. In the interest of data protection and security, no personal or sensitive data should be included when you are defining the contents of a bookmark.

The bookmark metadata reflects what is contained in the bookmark. If there is no change to a bookmarked component, the bookmark metadata will appear to be empty. Bookmark metadata does not include a reference to elements of the bookmark merely because they are configured to be included in the bookmark.

🎯 Example

If you select only to include an application title in a bookmark definition and proceed to make some changes to other parts of the application, no bookmark metadata is recorded. If, however, you make a change to the title of the application, bookmark metadata is recorded.

If you remove an element of the application from the bookmark definition, changes to the element will not be reflected, and it will no longer be recorded in the metadata.

The following components can be saved, and then loaded as part of your bookmark definition:

- global variables
- data sources
- standard components
- sdk
- sdk data sources
- technical components:
 - action sheet
 - conditional formatting settings
 - backend connection
 - context menu
 - component
 - export ao
 - export csv
 - export excel
 - export pdf

i Note

- It is not possible to copy and paste bookmark definitions from one bookmark to another.
- When including a *Panel* within a container in a bookmark definition, be sure to include the components within that *Panel* if you also wish to have them included in the bookmark definition.
- For more information on how to improve bookmarking performance, you can refer to the section on the SAP Community website at <https://wiki.scn.sap.com/wiki/display/BOBJ/Performance+Infos> called *Generic Analysis Template Bookmarking Performance Improvements*.

Related Information

[Working with Bookmarks \[page 211\]](#)

26.5.1 Using the Context Menu to Define Bookmarks

Use the context menu of the *Outline* view to select the components to be included in a bookmark.

Context

You have added the *Bookmarks* technical component to an application, and you want to define the contents of the bookmarks.

Procedure

1. In the *Outline* view of the application, identify which items you want to include in the bookmark definition.
2. Use the context menu to select *Add to Bookmark Definition* for each item you want to include.
3. Repeat step 2 until all items are added to the bookmark definition.

The items selected are listed beside the *Definition* property.

26.5.2 Using the Definition Property to Define Bookmarks

Use the *Definition* property of the bookmark technical component to select the components to be included in a bookmark.

Context

You have added the *Bookmarks* technical component to an application, and you have defined the contents on the bookmarks.

Procedure

1. Select the *Bookmarks* technical component from the *Outline* view.

The *Bookmarks* properties appear in the *Properties* tab.

2. Select the ellipsis icon beside the *Definition* property.

The *Edit Bookmark Definition* dialog appears. The *Edit Bookmark Definition* dialog lists of all the components, data sources, global variables, and technical components included in the application. The list will include only the items eligible for inclusion in the bookmark definition.

3. Select the checkboxes that correspond with the items you want to include in the bookmark. If you want to include everything, simply select the checkbox at the top of each group of items.
4. Select OK.

The items selected are listed beside the *Definition* property.

26.5.3 Using the Context Menu to Remove Item from Bookmark Definition

Use the context menu of the [Outline](#) view to select the components to be removed from a bookmark.

Context

You have added the [Bookmarks](#) technical component to an application, and you have defined the contents on the bookmarks.

i Note

- A bookmark definition persists even after you delete an item previously defined as part of the bookmark definition. The bookmark definition simply no longer contains that particular item.

Procedure

1. In the [Outline](#) view of your application, identify which items you want to remove from the bookmark definition.
2. Use the context menu to select [Remove from Bookmark Definition](#) for each item you want to remove.
3. Repeat step 2 until all items are removed the bookmark definition.

The items selected are removed from the list beside the [Definition](#) property.

26.5.4 Using the Definition Property to Remove Component from Bookmark Definition

Use the [Definition](#) property of the bookmark technical component to select the components to be removed from a bookmark.

Context

You have added the [Bookmarks](#) technical component to an application, and you have defined the contents on the bookmarks.

i Note

- A bookmark definition persists even after you delete an item previously defined as part of the bookmark definition. The bookmark definition simply no longer contains that particular item.

Procedure

1. Select the [Bookmarks](#) technical component from the [Outline](#) view.

The [Bookmarks](#) properties appear in the [Properties](#) tab.

2. Select the ellipsis icon beside the [Definition](#) property.

The [Edit Bookmark Definition](#) dialog appears. The [Edit Bookmark Definition](#) dialog lists of all the components, data sources, global variables, and technical components included in an application. The list will include only the items eligible for inclusion in the bookmark definition.

3. Unselect the checkboxes that correspond with the items you want to remove from the bookmark.
4. If you want to unselect all selected items, choose the [Clear Selection](#) button.
5. Select OK.

The unselected items are no longer listed beside the [Definition](#) property.

26.6 Filtering Bookmarks

You can filter bookmarks based on folder, group and access type.

The [Bookmarks](#) technical component properties, or the scripting editor can be used to filter bookmarks. Bookmarks can be filtered by a combination of the following characteristics:

- the currently selected folder
- the group to which they are assigned
- the access type

This combination of characteristics, or filters, allows users running the application to distinguish between bookmarks based on how these properties are set, or how the application is scripted. For example, when you then use the `getAll` scripting method, you get all bookmarks created by the current bookmark technical component, and filtered by the currently selected folder, group and access type.

The optional parameters `startIndex` and `endIndex` offer you the flexibility to allow users to return a batch of bookmarks at one time. To start your list of filtered bookmarks from the first bookmark, you should set the [startIndex](#) parameter of the [getAll](#) scripting method to 0. 1000 bookmarks or less are returned, if you choose not to use these optional parameters.

To return the total number of bookmarks for a [Bookmarks](#) technical component, you can use the scripting method `getCount`. The user only sees the total number of bookmarks for that technical component, to which they have access.

Folder

You can decide to assign bookmarks to a default bookmark folder structure. All bookmarks created within this [Bookmarks](#) technical component will be automatically assigned to the folder selected. Alternatively, you can apply scripting methods to give the application user the freedom to select the folder into which they can save their own bookmarks.

Group

You can apply scripting method `setGroup` to create a group based on a string to act as an identifier, or filter when defining a bookmark. Through scripting, you can also allow the application user to decide what string they use to define the group by which their bookmark is defined.

Access Type

You can specify whether a bookmark is visible to all users, or only visible to, and editable by, the user that created it. This is defined using the [Access Type](#) property of the [Bookmarks](#) technical component. Bookmarks can be categorized as either Personal or Global.

Related Information

[Working with Bookmarks \[page 211\]](#)

26.7 Assigning Bookmarks to a Folder

Use the [Bookmarks](#) technical component [Folder](#) property to assign bookmarks to a default bookmark folder structure. All bookmarks created within this [Bookmarks](#) technical component will be automatically assigned to the folder selected.

Prerequisites

You have added a [Bookmarks](#) technical component to an application.

Context

We recommend that administrators create a well-defined folder tree structure, with clear titles and descriptions. A well-defined folder structure will help you decide where best to save the bookmarks you create.

The folder structure is created by the administrator on the BI platform in the Central Management Console. User security is applied on a folder level. Access to bookmark folders is restricted according to the rights of individual users.

Note

- By default, the *Folder* property saves the bookmark to the root folder on the platform. For security reasons, it is recommended that you specify exactly to which folder you want the bookmark to be saved.
- You can also assign bookmarks to a folder using the scripting method `assignToFolder()`.

Procedure

1. In the *Properties* section of the *Bookmarks* technical component, select the ellipsis button in *Folder* property .

The *Select Folder* dialog appears.

2. Select the folder into which you want to save your bookmarks created within the current *Bookmarks* technical component.
3. Select OK.

The folder selected is listed in the *Folder* property.

Related Information

[Security Considerations with Bookmarks \[page 212\]](#)

26.8 Scripting Bookmarks to Display in a Tree Component

Script a *Tree* component to display bookmark folders.

You can script a *Tree* component to display to the application user, a list of all bookmark folders available to them. You can also script a *Tree* component to allow the application user to save one of their own bookmarks to a selected folder. It is also possible to display the root folder within the *Tree* component.

Users can select a folder from a *Tree* component representing this folder structure to filter the bookmarks they wish to use in their online application. If a user doesn't 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 using the Tree

This sample script could be used to assign bookmarks to a folder using a *Tree*:

```
var folderId = TREE_1.getSelectedValue();
```

```
var bookmarkId = LISTBOX_1.getSelectedValue();

BOOKMARKS_1.assignToFolder(folderId, bookmarkId);
```

Get Child Folders of a Hierarchy

This sample script could be used to get the child folders of a hierarchy:

```
var id = TREE_1.getSelectedValue();

var newModel = Bookmark.getFolderTreeModel(false, id);

TREE_1.setModel(newModel);
```

Setting a Tree with Bookmarks

This sample script could be used to set a [Tree](#) with bookmarks:

```
var model = BOOKMARK.getFolderTreeModel(true);

TREE_1.setModel(model);
```

26.9 Setting Bookmarks as Items in a Selection Component

You can directly set the items in a selection component from a returned list of bookmarks.

If you have a large number of bookmarks, it is much faster to add them as list items in a selection component, using the `getBookmarkList()` scripting method. The selection components you can use include the [Dropdown Box](#), [List Box](#), or [Radio Button Group](#) components. A bookmark list item is listed by the bookmark ID, as a unique identifier, and the bookmark title, as the display text.

Use the optional parameters `startIndex` and `endIndex` to allow you to control the number of bookmarks returned, or to return a range of bookmarks. The range is inclusive of the `startIndex` and the `endIndex`, and is 0-indexed.

26.10 Accessing Bookmarks

Access to bookmarks is determined by the setting of the [Access Type](#) property.

You can specify whether a bookmark is visible to all users, or only visible to, and editable by, the user that created it. This is defined using the [Access Type](#) property of the [Bookmarks](#) technical component. Bookmarks can be categorized in the following ways:

Personal Personal bookmarks can only be viewed, edited and deleted by their author.

Global Global bookmarks can be viewed by all users. However, they can only be deleted, overwritten, moved and added to a folder by their author.

Related Information

[Properties of the Bookmarks Technical Component \[page 214\]](#)

26.11 Versioning a Bookmark

The version of a [Bookmarks](#) technical component should be incremented when the bookmark is changed in such a way that renders it very different from the original state.

You can decide which changes to your [Bookmarks](#) technical component warrant an increase to the version of the [Bookmarks](#) technical component. The version number is incremented using the [Version](#) property. When you load a bookmark, it is always the one created in the latest version of the [Bookmarks](#) technical component. To load bookmarks created in an earlier or later version, simply change the version number in the [Version](#) property.

i Note

When you want to return a list of all bookmarks, only bookmarks saved within the current version are returned.

26.12 Deleting a Bookmark

Bookmarks can be deleted.

When you delete a [Bookmarks](#) technical component, the technical component, and all the bookmarks associated with it, are deleted from the application. However, these bookmarks are still stored on the platform. Application users can only delete bookmarks created by them. They cannot delete bookmarks created by other application users.

You can use the scripting API to allow application users to delete their own bookmarks. The following scripting is used:

- `BOOKMARKS.delete(id);`
- `BOOKMARKS.deleteAll();`

26.13 Set and Get Bookmarks at Runtime

Apply scripting methods to allow users to set and get bookmark properties at runtime.

As application designer, you can apply scripting methods to allow users to set and get the following bookmark properties at runtime.

- id
- title
- description
- group
- folder id
- access type

You can use scripting methods to allow application users perform various functions with bookmarks when they run the application. Bookmarks have to be specifically called by the scripting API, unless they are being loaded from a Url.

26.14 Personalization

You can set any bookmark created by a [Bookmarks](#) technical component to be the default view of an application when it initially loads.

Personalization allows you to select a bookmark of your choice to become the default view when you open an application. When you load an application, you can decide to save part, or all of it to create a personalized default application view. The next time you load the application, the personalized version of it is loaded by default. When you select the bookmark you want to use to personalize an application, you are referencing the ID of that bookmark. You can choose to overwrite the personalization you have set. You can change your personalization by referencing a different bookmark, or by changing how the personalization is scripted. Removing personalization from a bookmark does not remove the bookmark itself. It simply removes the reference to this bookmark as the bookmark used to personalize the application. You can use any bookmark created by a [Bookmarks](#) technical component you have access to, to personalize an application.

The personalization feature is enabled by adding the [Personalization](#) technical component to your application. Using the [On Load](#) scripting event, you can decide exactly what parts of the bookmarked application to include in your personalization. Personalization is defined by the unique combination of bookmarked analysis application and application user. Every time personalization is set, the current analysis application state is overwritten by the last personalization saved.

For more information on the [Personalization](#) technical component, you can refer to the chapter called *Personalization Technical Component*.

Note

1. If the version number of the referenced bookmark is incremented, the personalization will be lost.
2. 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

[Personalization Technical Component \[page 225\]](#)

26.14.1 Personalization Technical Component

The [Personalization](#) technical component provides all the functionality required to set any bookmark created by a [Bookmarks](#) technical component to be the default view of an application when it initially loads.

All Lumira Designer personalization functionality is consolidated into a technical component called [Personalization](#). To access personalization functionality, simply add the [Personalization](#) technical component to an application. To add this technical component, select [Personalization](#) from the context menu of the [Technical Components](#) section. The [Technical Components](#) section is in the [Outline](#) view of an application.

The [Personalization](#) technical component contains:

- [On Load](#) event, which is triggered on loading your bookmark.

Related Information

[Personalization \[page 224\]](#)

26.15 Compatibility with Legacy Bookmark Types

What to expect when you open applications in SAP Lumira Designer 2.1 that contain legacy bookmark types and were created before Lumira Designer 2.0.

When you open a 1.x legacy application in Lumira Designer 2.1 and select to migrate the application to 2.1, the application containing legacy bookmarks (standard, fragment and portable fragment bookmarks), will load and behave the same as before. Similarly, when you open the application on the BI platform, and the application contains 1.x bookmarks, it will load and behave the same as before.

If you open an application in BI platform mode, and edit it, the standard bookmarks will be invalidated. Fragment and portable fragment bookmarks will still work as before.

It is recommended that you use the Lumira Designer 2.0 [Bookmarks](#) technical component to create further bookmarks within an application, as previous bookmark technology is now deprecated.

Related Information

[Bookmarks Technical Component \[page 212\]](#)

26.16 Bookmarking Composites

Select composite components as part of the bookmark definition of an application.

An application can contain multiple composites. Each composite can contain its own collection of components. You can select composites to be included as part of the bookmark definition of an application.

To apply bookmark functionality to a composite, add a [Bookmarks](#) technical component to the [Outline](#) view. A [Bookmarks](#) technical component is needed for each composite you want to include in the overall parent bookmark definition. You can then define which components within a composite you would like to bookmark. Set the [Save with Parent Bookmark](#) property value to true, to ensure that the selected composite components are saved as part of the overall parent bookmark definition. The [Save with Parent Bookmark](#) property appears within the [Bookmarks](#) technical component of the composite. The [Save with Parent Bookmark](#) property enables or disables the saving of composites to the bookmark definition. If you set the [Save with Parent Bookmark](#) property to false, the [Bookmarks](#) technical component only saves a reference to a composite. The [Bookmarks](#) technical component does not then save the state of the composite. Only the basic, default properties such as height and width, are then saved to the bookmark definition.

There are two ways to select the parts of the composite to be included in the bookmark. You can use the context menu item [Add to Bookmark Definition](#) from the [Outline](#) view. Or, you can use the [Definition](#) property of the [Bookmarks](#) technical component. For more information, you can refer to the chapters in this guide called [Using the Context Menu to Define Bookmarks](#) and [Using the Definition Property to Define Bookmarks](#).

You can see the selected composites listed for selection with all other application components. Simply view the [Edit Bookmark Definition](#) dialog called by the [Definition](#) property of the [Bookmarks](#) technical component.

Save with Parent Bookmark

The parent bookmark is not necessarily defined by the [Bookmarks](#) technical component added to the main application in the [Outline](#) view. You need to add a [Bookmarks](#) technical component on every level of the application you wish to include in the parent bookmark definition. The parent bookmark can be in the main application, in a composite, or in a nested composite. The parent bookmark is simply the highest one in the set of [Bookmarks](#) technical components.

❖ Example

You could decide not to include a [Bookmarks](#) technical component in the main application. You could instead include one in a composite, and another in a nested composite. The [Bookmarks](#) technical

component in the top composite is then considered the parent bookmark. Once you select the components within each *Bookmarks* technical component, and the *Save with Parent Bookmark* property is set to true, all selected components, within each composite, are included in the parent definition.

i Note

Interface properties cannot be bookmarked. Save these properties instead, to a global variable bookmark.

Versioning

All properties of the *Bookmarks* technical component added to a composite are ignored, except the version number. If the version of the *Bookmarks* technical component added to a composite has changed since saving the overall parent bookmark definition, the associated composite state is not loaded when loading the bookmark. If multiple *Bookmarks* technical components are used in one composite, and if any of the versions have changed, only the *Bookmarks* technical components whose versions have not changed, load the composite state. All *Bookmarks* technical components whose versions have changed, load the composite state.

Related Information

[Adding a Composite to a Parent Bookmark Definition \[page 227\]](#)

26.16.1 Adding a Composite to a Parent Bookmark Definition

Use the *Bookmarks* technical component and the *Save with Parent Bookmark* property to add a composite to a parent bookmark definition.

Context

You have added a composite to an application. You want to include contents of the composite as part of the parent bookmark definition of the application.

Procedure

1. Select the composite from the *Documents* tab, and open it.

The composite opens in the *Outline* view.

2. Using the context menu of the *Technical Components* folder within the *Outline* view, select  *Create*  *Bookmarks* .

A *Bookmarks* technical component is added to the composite *Outline* view.

3. Select the *Bookmarks* technical component from the *Outline* view.

The properties of the *Bookmarks* technical component are available in the *Properties* tab.

4. Using the dropdown, set the *Save with Parent Bookmark* property to true.
Setting the *Save with Parent Bookmark* property to true means that this *Bookmarks* technical component is also saved as part of a parent bookmark definition, when this composite is added to the parent bookmark definition.
5. Select the components you want to include in the bookmark definition by using one of the following:
 - a. On each component, use the context menu item *Add to Bookmark Definition*.
 - b. Use the *Definition* property of the *Bookmarks* technical component, to call the *Edit Bookmark Definition* dialog.

For more information on how to use the *Add to Bookmark Definition* and *Definition* properties, you can refer to the following chapters: *Using the Context Menu to Define Bookmarks* and *Using the Definition Property to Define Bookmarks*.

6. Select the application from the *Documents* tab.
7. Select the composite you want to include as part of the parent bookmark definition. To do this, use either the context menu item of the composite *Add to Bookmark Definition*, or the *Definition* property of the *Bookmarks* technical component.

Only the components that are selected within the bookmark definition for the composite are bookmarked when the property *Save with Parent Bookmark* is set to true. Only the composite, and not its individual components can be added to the bookmark definition for the application. All other components within the application can be added to the bookmark definition separately.

Related Information

[Using the Context Menu to Define Bookmarks \[page 216\]](#)

[Using the Definition Property to Define Bookmarks \[page 217\]](#)

[Using the Context Menu to Remove Item from Bookmark Definition \[page 218\]](#)

[Using the Definition Property to Remove Component from Bookmark Definition \[page 218\]](#)

27 Working with Comments

You can use the *Comments* technical component functionality to allow users to add comments in a variety of contexts to Lumira documents (.lumx files).

The functionality provided by the *Comments* technical component allows users to create comments in Lumira document, and to share those comments across all applications in a Lumira document (.lumx file). To avail of this functionality, you must add the *Comments* technical component to the *Outline* view of the design tool. Comments can be created in local mode, for testing purposes. However, the real use case for this feature is creating and sharing comments using the BI Commentary service on the BI platform. For more information on the administration of comments, you can refer to the “Business Intelligence Platform Administrator Guide” on the SAP Help Portal at <http://help.sap.com>. Comments can be saved and retrieved in local mode and on the BI platform, as a platform service. Comments can be applied to a Lumira document in the following ways:

The entire document: You can apply scripting to create comments on a Lumira document.

A data cell: You can apply scripting to create comments on specific data cells. The comments are displayed when the specific data cell is selected. When you de-select the data cell, the data context is removed and the comment associated with that data cell is hidden.

A dimension: You can apply scripting to add a comment to a specific dimension.

A dimension member: You can apply scripting to add a comment to a specific dimension member.

A context: You can script a document to add specific context to suit requirements, such as a context which represents a filter .

Comment Context

There are different contexts in which the comments apply. The different types of contexts include the following:

- DATA
- MEMBER
- DIMENSION
- CONTEXT
- NONE

When creating a comment with a selection and the context type is not explicitly set, the context type is set to DATA by default.

Scripting methods

When the *Comments* technical component is added to the design tool, the scripting methods associated with the comments functionality are available in the script editor. The following scripting methods can be applied:

- create
- delete
- getComment
- getComments
- setContent

Comments in Composites

It is possible to have multiple composites, each with its own instance of the [Comments](#) technical component. As comments are associated with a Lumira document, users can create composites that can act as specialized comment components. These components could be used for creating and listing comments in a document. For example, it should be possible to create a composite that contains form elements to create comments for an application.

Public or private

You can script comments to be public. If a comment is public, it means that everyone can view it. Comments may also be set to private. If a comment is private, it means that only the creator of the comment may view the comment. To view private comments, you can filter comments based on the CUID. To view private comments you must set the `isPublic` parameter to false when applying the `getComments()` scripting method.

Note

- This concept of public and private comments is specific to Lumira. It does not relate to the view, modify, or delete rights which are associated with comments created or viewed on the BI platform.
- Comments created in an application in local mode within Lumira Designer are not available when running that application on the BI platform. Only comments created in applications while working on the BI platform, are available for viewing. The view rights are determined by the appropriate rights being granted on a BI platform level.

Related Information

[Comment Context Types \[page 231\]](#)

[Displaying Comments \[page 237\]](#)

[Creating a Comment \[page 235\]](#)

27.1 Comment Context Types

Comments can apply to different context types.

When creating and querying comments there are number of different contexts to which the comment can apply:

Context Type	Description
data	The comment applies to a specific cell. This is the default when creating a comment with a context.
member	The comment applies to a member of a dimension.
dimension	The comment applies to a dimension.
context	This allows the user to specify a custom context to which the comment can be associated. For example you could create a context from the currently applied filters on a data source.
none	This indicates that the comment applies to no specific context. This is the default behavior when creating a comment with no context. It is very useful for querying comments that apply across the entire Lumira document.

Related Information

[Working with Comments \[page 229\]](#)

27.2 Model Entity Relationships

The difference between document mode and legacy BI platform mode when working with comments.

When working with the [Comments](#) feature, it is important to bear in mind the following:

Documents Mode

Comments are associated with Lumira documents (.lumx files) in document mode.

- Lumira documents (.lumx files) may contain 0..N applications.
 - Applications contained within Lumira documents may contain 0..1 [Comments](#) technical components.

- Comments created in different applications within the same document are visible across applications.
- Lumira documents (.lumx files) may contain 0..N composites.
 - Composites contained within LUMX documents may contain 0..1 [Comments](#) technical components.
 - Comments created in composites are visible in an application consuming those composite(s), if and only if, the application and composite share the same document.

Legacy BI Platform Mode

Comments are associated with analysis applications in Legacy BI platform mode.

- Applications may contain 0..1 [Comments](#) technical components.
- Comments are not visible across applications as comments can only be associated with one application.

27.3 Creating a Context

There are a few different ways of getting or creating a context to be used in comment creation and querying in a Lumira document.

You can get or create a context in the following ways:

- from a crosstab selection
- from a context as a JSON object
- from a crosstab with background filters
- from a String

Related Information

[Context from Crosstab Selection \[page 232\]](#)

[Context from Crosstab with Background Filters \[page 234\]](#)

[Context as a JSON object \[page 234\]](#)

[Context as a String \[page 235\]](#)

27.3.1 Context from Crosstab Selection

You can create a context from a crosstab selection.

Through scripting you can create a context from a selection on a crosstab. You can create a one-to-one mapping of the [ClickArea](#) of the crosstab and the contextType. This would avoid hard-coding the contextType to a specific area on the crosstab. If you have a context menu on the crosstab, you could use the following scripting sample:

Sample Code

```
var selType = CONTEXT_MENU.getSelectionType() + "";
var sel = CONTEXT_MENU.getSelection();
var comments = COMMENTS.getComments({
    "contextType" : selType,
    "context" : sel
});
```

Crosstabs allow several different selection types and not all can be rendered by the crosstab following comment creation.

Context from Crosstab Selection

Selection Type	Description	Rendered in Crosstab	Click Area	Comment Created
Single Row/Column	Select a single row or column	Yes	MEMBER	Yes
Single Data Cell	Single cell (Data cell or Overall result cell) (RESULT_MEMBER) added to data path in this case	Yes	DATA	Yes
Single Cell	Can select a Data Cell (Overall result cell included) or a Dimension member cell	Yes	MEMBER/CELL	Yes
None	Selection is not possible	N/A	Not specified	No
Multi Row/Column	Select single/multiple rows/columns	YES - if single Row/Column selection made NO - otherwise as crosstab can only render comments for a single selection	MEMBER	Yes

Related Information

[Creating a Context \[page 232\]](#)

27.3.2 Context from Crosstab with Background Filters

You can create a context from a crosstab with background filters.

Through scripting you can create a context from a crosstab with background filters. You can create a one-to-one mapping of the [ClickArea](#) of the crosstab and the contextType. This would avoid hard-coding the contextType to a specific area on the crosstab. When a crosstab is getting comments to display, the background filters of the crosstab data source is taken into account. To create comments with a background filter, it is necessary to use the `getBackgroundFilters()` function of the data source. If you want to specifically click on a data cell to add a comment, you can use the following scripting sample:

Sample Code

Adding background filters to the context

```
var selection = CROSTAB_1.getSelection();
var context = DS_1.getBackgroundFilters(selection);

COMMENTS.create("My New Comment", {
    "context": context,
    "contextType": CommentContextType.DATA
});
```

Related Information

[Creating a Context \[page 232\]](#)

27.3.3 Context as a JSON object

You can create a context as a JSON object.

Through scripting you can create a context as a JSON object. You can use the following scripting sample:

Sample Code

Getting a context from a crosstab selection

```
COMMENTS.create("My CONTEXT Comment", {
    "context": { "education": DS_1.getFilterExt("education").split(";") },
    "contextType": CommentContextType.CONTEXT
});
```

Related Information

[Creating a Context \[page 232\]](#)

27.3.4 Context as a String

You can create a context as a String.

When you do not know in advance which dimensions you want to use for creating a context, you can build one up as a String, through scripting. The following scripting sample converts the filters on all dimensions in the given data source to a String, which can be used as a context for comments.

Sample Code

```
var context = "{";
var first = true; // Just used to add commas
var dimensions = dataSource.getDimensions();
dimensions.forEach(function(dimension, index) {
    var filters = dataSource.getFilterExt(dimension);
    var memberValues = filters.split(";");
    if (filters.length > 0 && memberValues.length > 0) {
        if (!first) {
            context = context + ",";
        }
        first = false;
        var membersAsString = "[";
        memberValues.forEach(function(value, index) {
            if (index > 0) {
                membersAsString = membersAsString + ",";
            }
            membersAsString = membersAsString + "\"" + value + "\"";
        });
        membersAsString = membersAsString + "]";

        context = context + " \"" + dimension.name + "\": " + membersAsString;
    }
});

context = context + "}";
return context;
```

27.4 Creating a Comment

You can create a comment within a Lumira document.

Using scripting, you can create a simple comment or a comment with properties on a Lumira document. You can use the following scripting samples:

Sample Code

Creating a simple comment

```
var commentId = COMMENTS.create("This is a comment");
```

Sample Code

Creating a comment with properties

```
var commentId = COMMENTS.create("This is a comment with properties", {
    "context": {"education": "Graduate Degree"},
    "contextType": CommentContextType.CONTEXT,
    "isPublic": true,
    "dataSource": DS_1
});
```

The following optional properties can be scripted for the comments:

- text
- id
- author
- updateTime
- isPublic
- creationTime

When setting the value of the `isPublic()` property to true, comments are public, or potentially visible to everyone, depending on the view rights set on the BI platform level. When setting the value of the `isPublic()` property to false, comments are private, or visible only to the creator, depending on the view rights set on the BI platform level. If the property is not set, comments are considered to be visible to everyone, depending on the view rights set on the BI platform level.

Related Information

[Working with Comments \[page 229\]](#)

27.4.1 Create Comments with Different Contexts

You can create comments with different contexts.

Using the properties `context` and `contextType` on the [Comment](#) object, you can allow end users to create comments with different contexts, and then filter them, so the comments can be rendered separately by context.

Saving a comment with a user-defined context:

The following sample code suggests a way to save a comment with a user-defined context:

Sample Code

```
var commentId = COMMENTS.create(INPUTFIELD_1.getValue(), {
    "context": {"sales": "2018", "country": "Ireland"},
    "contextType": CommentContextType.CONTEXT,
    "isPublic": true,
    "dataSource": DS_1
});
```

Retrieving comments with the user-defined context:

The following sample code suggests a way to retrieve comments with a user-defined context:

Sample Code

```
var comments = COMMENTS.getComments();
comments.forEach(function(element, index) {
    var context = element.context;
    context.forEach(function(value, key) {
        if(value[0] == "Ireland") {
            FEEDLIST_IRELAND.addItem(comments[index]);
        }
    });
});
```

27.5 Displaying Comments

You can use the [Feed List](#) component to display comments.

Used in conjunction with the [Comments](#) technical component, the [Feed List](#) component offers a nice way to list comments in a Lumira document. Using scripting you can create a connection between the two components so that the comments are treated like feed items of the [Feed List](#). You could use the following sample to display comments in a [Feed List](#):

Sample Code

```
FEEDLIST_1.setItems(COMMENTS.getComments());
```

Note

For more information on scripting the [Comments](#) technical component with the [Feed List](#) component, you cafer to the chapter in this guide called "Working with the Feed List".

Related Information

[Working with Comments \[page 229\]](#)

[Working with the Feed List \[page 284\]](#)

27.6 Querying a Comment

You can query for comments created across all applications in a Lumira document.

Using scripting, you can return comments made in a Lumira application. You can use the following scripting samples to return comments:

Sample Code

Getting a comment by id

```
var id = COMMENTS.create("Some comment");
var comment = getComment(id);
```

Sample Code

Getting all comments

```
var comments = COMMENTS.getComments();
```

Sample Code

Getting a filtered list of comments

```
var comments = COMMENTS.getComments({
  "context": {"[MEASURES_DIMENSION]": ["unit_sales"],"education":["High School Degree"]}, // Returned comments must have this context
  "contextType": CommentContextType.CONTEXT, // Returned comments must have this context type
  "dataSource": DS_1, // Returned comments must have this data source
  "isPublic": true, // Returned comments must be public
  "sortBy": CommentSortBy.CREATED_ON, // Sort by the creation date of the comment
  "order": SortOrder.ASCENDING // Sort in ascending order
});
```

The `getComments` function takes an optional parameter. This optional parameter is an object that contains properties for the query. All of the properties in the query parameter are optional.

27.7 Updating a Comment

You can update the content of the comment in a Lumira document.

Provided you have the necessary permissions, you can update a comment. Only the content of the comment can be updated. You can use the following scripting sample to update comments:

Sample Code

```
COMMENTS.setContent(id, "This is the new content");
```

For more information on permissions, you can refer to the *Administrator Guide: SAP Lumira*.

Related Information

[Working with Comments \[page 229\]](#)

27.8 Deleting a Comment

You can delete a comment from a Lumira document.

Provided you have the necessary permissions, you can delete a comment with a specific id. You can use the following scripting sample to delete comments:

Sample Code

```
COMMENTS.delete(id);
```

For more information on permissions, you can refer to the *Administrator Guide: SAP Lumira*.

Related Information






[Working with Comments \[page 229\]](#)

27.9 Displaying Comments in the Crosstab

If you have created comments for data cells, header cells, or dimensions, you can display them in the crosstab.

After you have created comments for data cells, header cells, or dimensions with the technical component *Comments*, you can display these comments in the crosstab.

To display the comments, set the crosstab property *Comments Visible* to *true*. If there is a comment available for any cell of the result set which the crosstab is currently displaying, the cells will indicate this by displaying a red triangle in the top right corner of the cell.

Neighborhood	Total Population 	Net Income p.c. 	Education Score 
Corcktown-Woodbridge	19,543	146,513	0.82
Downtown	10,854	63,260	0.70
East Side	124,820 	114,479	0.73
Eastern Market Area	32,368	100,880	0.66
Jefferson Corridor	36,679	210,572	0.94
Mid Town	15,300	73,241	0.72
New Center Area	14,678	61,870	0.57
North End	2,449	47,420	0.34
Palmer Park Area	9,758	143,360	0.78
Southwest Detroit	47,566	56,873	0.25
West Side	154,840	110,845	0.79
Overall Result 	468,855	1,129,313	0.66

When you hover over a cell with a comment, a tooltip is displayed with the comment's text.

28 Working with Charts

A [Chart](#) is a graphical representation of the data in an application.

You can add charts to an application to present data graphically. Charts can often emphasize irregularities or trends in 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 graphical 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, by default.

You can use the [Edit Initial View..](#) to specify the content (measures and dimensions) to be used in a [Chart](#). You can also set the chart property [Allow Data Source Modification](#) to true to enable application users to modify the state of the data source within the [Chart](#) itself.

The properties of the [Chart](#) component can be set in three ways:

- general properties are defined in the [Properties](#) view of the Lumira Designer tool.
- additional properties are exposed by selecting the ellipsis button of the [Chart Configuration](#) property within the [Properties](#) view. The additional property options available are based on [Chart](#) type selected.
- a subset of the additional properties are available within the context menu of the [Chart](#) area in the [Configure Chart](#) dialog.

For more information on the chart general properties in the [Properties](#) view, you can refer to the chapter called [Chart](#). For more information on configuring chart types and additional properties in the [Configure Chart](#) dialog, you can refer to the chapters called [Using in the Configure Chart Dialog](#) and [Chart Configuration Context Menu](#).

Note

- The color assigned to measures or dimensions within the scope of a chart, persists for the lifetime of the chart, even if filters are applied to the data, conditions are defined, new measures are added, or if dimension values are changed.
- For more information on considerations when persisting chart colors, you can refer to the SAP Note [2756762](#).
- Analysis applications created in SAP BusinessObjects Design Studio 1.6 that have been created with charts from the commons library of SAP UI5 can still be opened, edited and saved, but only in the BI Platform mode of Lumira Designer. It is not possible to create new charts in the commons mode (by choosing [New](#), [Save As](#) or [Import](#) from the menu of the design tool).

Related Information

[Chart \[page 451\]](#)

[Chart Properties in the Configure Chart Dialog \[page 457\]](#)

[Chart Configuration Context Menu \[page 247\]](#)

[The SAPUI5 m Library as Rendering Mode \[page 45\]](#)

28.1 Adding a Chart to your Application

Add a [Chart](#) component to an application to display data.

Context

You can add a [Chart](#) component to an analysis application to display the data in your data source.

Procedure

1. From the [Chart 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 [Column](#) chart.

28.2 Configuring Initial Settings for Charts

Add measures and dimensions to a [Chart](#).

Context

Add to the initial view the measures and dimensions you want to see in a [Chart](#). 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

- Most [Charts](#) (VizFrame) show a maximum of 10,000 data points. If a [Chart](#) (VizFrame) has more than 10,000 data points, the chart will show the first 10,000 data points along with a message that not all data points are shown. You can retain or change the filter appropriately to view required data points.
- A legacy [Chart](#) can display up to 35,000 data points.
- 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.
- Dual axis, scatter and bubble charts always require measures.

Procedure

1. In the [Outline](#) view, in the context menu of the data source, 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.

Related Information

[Chart \[page 451\]](#)

28.3 Using the Configure Chart Dialog

Use the [Configure Chart](#) dialog to fully configure a chart at design time and to allow the application user to configure a chart while running the application.

Context

You have added a [Chart](#) with an assigned data source to an application.

Procedure

1. Within the chart [Properties](#) tab, select the ellipsis button beside the [Chart Configuration](#) property.

The [Configure Chart](#) dialog appears.

2. Select the preferred chart type from the [Chart](#) tab. Each chart type collection offers a dropdown from which you can select the specific chart type required.

The chart displayed in the [Preview](#) area instantly changes to reflect your chart type selection.

3. To configure the location of the measures and dimensions in the chart, you can move them up and down within the [Measures](#), [Dimensions](#) and [Trellis](#) areas in the [Chart](#) tab. Use the drag and drop motion to move measures and dimensions.

The chart displayed in the [Preview](#) area instantly changes to reflect the selected location of the measures and dimensions.

4. To change the chart type for the current series, use the context menu item [Change Chart Type for Series](#) within the [Measures](#) area.

5. To assign one of the measures to the Primary Axis or Secondary Axis, use the context menu item [Assign to Secondary Axis](#) or [Assign to Primary Axis](#) within the [Measures](#) area.
6. Use the context menu within the chart in the [Preview](#) area of the [Configure Chart](#) dialog to configure a subset of the additional chart properties. The properties available through the context menu are contextually based on the chart type selected.
7. To access all available properties per chart type, select the [Properties](#) tab within the [Configure Chart](#) dialog.
8. To apply a property change to multiple chart elements, select one instance directly in the chart. All instances of the element are then selected by default. You can make a property change, which is then applied to all instances of the chart element.

If you select a column in the chart, all columns for the series are selected. Changing the [Color](#) property applies the color change to all selected columns.

9. To apply a property change to a single instance of a chart element, select only one instance of an element directly in the chart, for example a single column. If you previously had selected all, simply select the one instance you want to change. You can make a property change, which is then applied to the one instance of the chart element selected.
10. Another way to apply a property change to multiple chart elements, is to select a specific chart area from the dropdown in the [Properties](#) tab.

Select [Column](#) from the dropdown. By default, any property change made applies to all columns selected. If no specific columns are selected, the property change is applied to all columns in the chart.

11. If you wish to allow the data source state to be modified by the chart at runtime, set the [Allow Data Source Modification](#) chart property to true. The [Allow Data Source Modification](#) is in the [Property](#) tab of the [Chart](#) component.

When the [Allow Data Source Modification](#) property is set to true, you can modify the data source state, while interacting with the chart. The changes made affect all components bound to the same data. It allows you to add or remove one measure or dimension from the chart or connected [Chart Feeding Panel](#).

12. Return to the [Chart Configuration](#) dialog.

An [x](#) and a [+](#) icon are available beside the [Measures](#) and [Dimensions](#) area within the [Chart Configuration](#) dialog.

13. Select the [+](#) icon beside the [Measures](#) or [Dimensions](#) areas to select the extra measure or dimension you wish to add to the [Chart Feeding Panel](#).
14. Select the [x](#) icon beside the [Measures](#) or [Dimensions](#) areas to remove one measure or dimension from the [Chart Feeding Panel](#).
15. Select the [+](#) icon beside the [Rows](#) and [Columns](#) to add the new measures and dimensions.
16. Select [OK](#).

The newly configured chart is displayed in the [Layout Editor](#).

Related Information

[Chart \[page 451\]](#)

[Chart Configuration Context Menu \[page 247\]](#)

[Chart Types \[page 245\]](#)

[Chart Properties in the Configure Chart Dialog \[page 457\]](#)

[Number Formatting in Charts \[page 252\]](#)

[Customized Number Formatting in Charts \[page 252\]](#)

28.4 Chart Types

A list of all chart types that can be selected for displaying data in your application.

You can use the [Configure Chart](#) dialog within the [Chart Configuration](#) property to select the type of chart to be used in an application. The following chart types are available for selection:

Column and Bar

- Column
- Bar
- Stacked Column
- Stacked Bar
- Marimekko
- Waterfall
- Bullet

Line and Area

- Line
- Stacked Area

Pie

- Pie
- Doughnut

Dots

- Scatter Plot
- Bubble

Tile (Square)

- Heat Map
- Tree Map

Numeric Point

- Numeric Point

Time Series

- Line Chart for Times Series
- Scatter Plot for Time Series
- Bubble Chart for Time Series

Other

- Tag Cloud
- Box Plot
- Radar
- Network
- Funnel
- Parallel Coordinates
- Tree

Related Information

[Chart \[page 451\]](#)

[Using the Configure Chart Dialog \[page 243\]](#)

[Chart Type Picker \[page 458\]](#)

28.5 Chart Configuration Context Menu

You can use the context menu of the [Configure Chart](#) dialog to interact with a chart.

The [Configure Chart](#) dialog offers the following context menu items, depending on the chart type selected, and the area in the [Preview](#) in which the context menu is activated.

Context menu functions for the preview area of the [Configure Chart](#) dialog.

- horizontal
Use [Horizontal](#) to display your chart horizontally.
- vertical
Use [Vertical](#) to display your chart vertically.
- normal stacking
Use [Normal Stacking](#) to stack the data series on top of each other in order.
- 100% stacking
Use [100% Stacking](#) to fill the plot area and draw each point of data with a relative percentage to all the points in the same category.
- show title
Use [Show Title](#) to display a title on the chart.
- show legend
Use [Show Legend](#) to display a legend on the chart.
- show data labels
Use [Show Data Labels](#) to display data labels in the chart.
- show gridlines
Use [Show Gridlines](#) to display gridlines in the chart.
- use measures as a dimension
Use [Use Measures as a Dimension](#), if you want to select where measures appear in the chart, other than in the legend.
- set axis scale
Use [Set Axis Scale](#), if you want to change the axis scale from automatic to a fixed range.
- add trendline
Use [Add Trendline](#), to show data trends in a bar, column, line, scatter, bubble, time line, time scatter or time bubble chart. The trendline is calculated based on the visualized data.
- edit trendline setting
Use [Edit Trendline Setting](#), to extend a trendline beyond the actual data to help predict future values.
- remove trendline
Use [Remove Trendline](#), to remove all trendlines from the chart.
- reference lines
Use [Reference Lines](#), to add a fixed or dynamic reference line to an axis in the chart.
- manage reference lines
Use [Manage Reference Lines](#), to decide which reference lines to display in the chart, to add or remove reference lines, or to edit existing reference lines.

- Use [Choose Colors](#), to select the type of color palette to be used in the chart. You can decide to use [User Defined Palettes](#) or [Standard Palettes](#). Selecting the [Customize Dimension Palette](#) button allows you to define a customized color palette for the dimensions.
- change chart type for series
Use [Change Chart Type for Series](#) to change the type of chart for the current data series. For example, you can change from a line chart to a column chart for a data series.
- assign measure to axis
Use [Assign Measure to Axis](#), if you want to assign the measure to the Primary Axis or Secondary Axis.
- Use [Sort](#), decide whether to sort the chart elements by [Text Ascending](#), [Text Descending](#), or [Sort by Hierarchy](#).
- Use [Display](#), to display chart elements using one of these options:
 - No Display
 - Key
 - Text
 - Key and Text
 - Text and Key
 - Short Text
 - Long Text
 - Text
- Use [Select Hierarchy](#), to apply a hierarchy to the chart elements. When a hierarchy is selected, the following context menu items are also available:
 - Expand All
 - Collapse All
 - Expand to Level

To remove a hierarchy from the chart, simply select [Select Hierarchy](#), and select [No Hierarchy](#) from the drop-down list. This context menu item is available on members and data points only, not on dimensions. When you select a hierarchy from the drop-down list, two other context menu items are available to you - [Drill Down Hierarchy](#) and [Drill Up Hierarchy](#). Select [Remove Filter](#) to return the chart to its original state before the hierarchy was selected.
- Use [Drill Down Hierarchy](#), to drill down the hierarchy nodes. In effect, when you select to drill down the hierarchy, you are expanding a node and setting a filter.
- Use [Drill Up Hierarchy](#), to drill up the hierarchy nodes. In effect, when you select to drill up the hierarchy, you are collapsing a node and replacing a filter with a filter one level up.
- Use [Rank](#), to call up the rank dialog.
- Use [Filter Members](#), to the select filters to apply to a member or data point, and define conditions for it.
- Use [Remove Filter](#), to remove the applied filter from a member or data point.
- Use [Filter by Measure](#), to create a measure filter based on a measure selected from a drop-down.
- Use [Add Dimension](#), to add another dimension to the chart. When you have two dimensions in a legend, the context menu will show the menu options for the inner dimension.
- Use [Swap Dimension With](#), to select a difference dimension to display in the chart.
- Use [Remove Dimension](#), to remove a dimension from the chart.
- Use [Filter](#), to select an individual data point as a filter.
- Use [Filter and Remove Dimension](#), to select an individual data point as a filter, and remove a dimension at the same time.

i Note

The context menu also displays several other analytical actions from other components, such as the [Crosstab](#) or [Spreadsheet](#) components, for example. The context menu items displayed depend on the element selected. For details about other context menu items, you can refer to the chapter called *Using the Context Menu (Technical Component)*.

Related Information

[Using the Context Menu \(Technical Component\) \[page 86\]](#)

28.6 Working with the Chart Color Palette

Define the chart color palette to be used in your chart.

Context

You are in design mode with a chart in the layout panel.

Procedure

1. Select the [Chart Configuration](#) property.
The [Configure Chart](#) dialog opens.
2. Activate the chart context menu and select [Choose Colors](#).
3. To apply a standard color palette to your chart, select one from the [Standard Palettes](#) area of the color widget.
4. To select one of your own color palettes, select from the [User Defined Palettes](#) area of the color widget.
5. To define a customized color palette for the dimensions in your chart, select the [Customize Dimension Palette](#) option in the color widget.
The [Customize Dimension Palette](#) appears.
6. Select a color bubble one by one to customize the color palette you can use in your chart dimensions.
7. Select [OK](#).

28.7 Sharing a Chart Color Palette across Multiple Charts

Create a custom color palette, and share it across multiple charts or applications.

Context

You want to create a custom color palette for a specific department, and share that custom palette as the standard palette, across multiple charts, or even across multiple applications. The steps below describe one of the possible ways you could do this.

Procedure

1. Create a [Chart](#) within a [Composite](#), or simply create a [Chart](#) in a dedicated Lumira document. This [Chart](#) becomes the template chart.
2. Within this template chart, configure the custom dimension palette using the [Customize Dimension Palette](#) property as described in the topic called [Working with the Chart Color Palette](#).
3. To use the custom dimension palette across other charts or applications, the other charts or applications need to be able to connect to this template chart. If you create the template chart within a [Composite](#), the template chart needs an interface function to return the template chart for use.

❖ Example

≡ Sample Code

```
var templateChart = MY_COMPOSITE.getTemplateChart();
COMPONENTS.copyProperties(templateChart, CHART_1,
[CopyPropertiesOptions.KEEP_LAYOUT, CopyPropertiesOptions.KEEP_CHART_TYPE
, CopyPropertiesOptions.KEEP_CURRENT_VALUES]);
COMPONENTS.copyProperties(templateChart, CHART_2,
[CopyPropertiesOptions.KEEP_LAYOUT, CopyPropertiesOptions.KEEP_CHART_TYPE
, CopyPropertiesOptions.KEEP_CURRENT_VALUES]);
COMPONENTS.copyProperties(templateChart, CHART_3,
[CopyPropertiesOptions.KEEP_LAYOUT, CopyPropertiesOptions.KEEP_CHART_TYPE
, CopyPropertiesOptions.KEEP_CURRENT_VALUES]);
```

≡ Sample Code

```
var templateChart = MY_COMPOSITE.getTemplateChart();
var arr =
COMPONENTS.getComponents(ComponentType.com_sap_ip_bi_VizFrame);
arr.forEach(function(c) {
    COMPONENTS.copyProperties(templateChart, c,
[CopyPropertiesOptions.KEEP_LAYOUT, CopyPropertiesOptions.KEEP_CHART_TYPE
, CopyPropertiesOptions.KEEP_CURRENT_VALUES]);
});
```

You can use the above scripting sample as a guide.

Related Information

[Working with the Chart Color Palette \[page 249\]](#)

[Chart Configuration Context Menu \[page 247\]](#)

[Copy Properties \[page 74\]](#)

28.8 Using Ad-hoc Data Comparison

Use the *Enable Ad-hoc Data Comparison* context menu item at runtime only, to compare data points on a bar, column, or line chart.

Context

You have created a column, bar or line *Chart*, assigned a data source, and are viewing the application in a browser.

i Note

The *Enable Ad-hoc Data Comparison* context menu item is available as a runtime feature only. It allows the end user to compare data points while running the application. The data point comparison markers do not persist.

Procedure

1. Select the context menu item *Enable Ad-hoc Data Comparison*.
2. Select one of the data points in the chart that you want to compare to another, from a data point of view.

The data point is highlighted in the chart.

3. Select the second data point in the chart for data comparison.

Both data points are highlighted and the data comparison in percentage value and actual numerical value is displayed. If there is a positive difference between the two data points, this difference is displayed in green. A negative difference between the two data points is displayed in red.

4. Add a third data point for comparison.

A dialog box appears displaying a detailed list of the data comparison between the three values. The dialog also depicts the three data points in a dedicated chart. The data comparison details include the sum, average, maximum, minimum and median results.

5. Select the arrow in the dialog box to display a different view of the data comparison.
6. Continue to select more data points for comparison until you have compared as many data points as required. You can select more data points by using the left mouse-click, or by lasso selection.

28.9 Number Formatting in Charts

The format of numbers in charts can be selected from a list of options.

Number formatting can be applied in many areas of a chart, for example, on a data label or on a value axis. You can also define your own formatting style. For more information, you can refer to the chapter called *Customized Number Formatting in Charts*. To define how the data in the chart is formatted, select one of the following suggested format options from the *Number Format* property dropdown list within the *Configure Chart* dialog:

- Default** applies the same formatting in the chart as applied in the crosstab.
- None** applies no formatting
- #,##0** formats the number with a thousand separator
- #,##0.00** formats the number with a thousand separator, period and two decimal places
- \$#,##0** formats the number as a dollar amount

⚠ Caution

You may see unexpected number formatting in a chart, if your viewing locale is different from the one used when designing the application. This may occur when you select one of the options other than Default, in the *Number Format* property dropdown list, or when you select your own customized number formatting. For more information on regional and language settings, you can refer to the chapters called *Regional Formats for Numbers, Dates and Times* and *Language of Message Texts and Tooltips in %LUMIRA-ABR% Documents and Analysis Applications* in the “Administration Guide: SAP Lumira” on the SAP Help Portal at <http://help.sap.com>.

Related Information

[Customized Number Formatting in Charts \[page 252\]](#)

[Using the Configure Chart Dialog \[page 243\]](#)

28.9.1 Customized Number Formatting in Charts

Customize your own number formatting for charts. Use the *Number Format* property to manually edit the values displayed in the *Number Format* dropdown list.

Context

You want to define your own number formatting style to be applied to charts in an application.

⚠ Caution

You may see unexpected number formatting in a chart, if your viewing locale is different from the one used when designing the application. This may occur when you select one of the options other than Default, in the *Number Format* property dropdown list, or when you select your own customized number formatting. For more information on regional and language settings, you can refer to the chapters called *Regional Formats for Numbers, Dates and Times* and *Language of Message Texts and Tooltips in %LUMIRA-ABR%* *Documents and Analysis Applications* in the “Administration Guide: SAP Lumira” on the SAP Help Portal at <http://help.sap.com>.

Procedure

1. Select a *Chart* component in the *Layout* tab.
2. Select the *Chart Configuration* ellipsis button within the property tab.

The *Configure Chart* dialog appears.

3. Select the *Properties* tab.
4. The *Number Format* property appears in multiple property areas, for example, *Chart Area* or *Data Label*. Within the *Number Format* property, select the dropdown icon.

The current number formatting value is highlighted.

5. Delete this number formatting value.

The field is now clear to allow you to enter your own customized number formatting style.

6. **i Note**

Examples of some of the types of number formatting you can use when defining your own include:

- decimal places
- currency symbol
- thousand separator
- period

Enter your own number formatting style.

Related Information

[Number Formatting in Charts \[page 252\]](#)

29 Working with Conditional Formatting in Crosstabs

There are several ways available to activate conditional formatting in crosstabs.

There are three ways to provide conditional formatting in crosstabs:

- Conditional Formatting Settings as technical component
To enable conditional formatting at design time, the technical component Conditional Formatting Settings has to be added to the application.
- BEx Query Exceptions defined on BEx query level
These conditional formatting rules are defined when the query is created.
- BEx Query Exception formatting defined by the application user at runtime
At runtime, the application user can define additional conditional formatting rules for BEx query exceptions based on measures (key figures in SAP BW). When the application user changes existing rules, only the visualization of existing BEx query exceptions can be changed, not their content.

When working with conditional formatting in crosstabs, consider the following:

- No matter what type of conditional formatting you choose, you have to set the crosstab property *Conditional Formatting Visible* to *true*, otherwise no conditional formatting is displayed in the crosstab.
- Only one conditional formatting type can be applied at a time, with priority as follows:
 1. Technical component *Conditional Formatting Settings*
 2. BEx Query Exceptions

In other words, if you have added the technical component *Conditional Formatting Settings* to your application, no other formatting settings will be applied (even if there are no rules defined for this technical component).

30 Working with Variants

You can use variants in Lumira Designer.

A variant has a technical name, a text description, and a scope. The scope can be either *USER* or *PUBLIC*.

To be able to use variants, you need to set the application property *Variant Mode* to either *Query variants* or *Document variants*. In the *Prompts* dialog box you can select a variant, change its values, or create a new variant. You can edit the technical name, description, and scope of an existing variant.

30.1 Using Query Variants

Query variants store selection data for variables in an SAP BW query.

Prerequisites

To be able to use query variants, you need to set the application property *Variant Mode* to *Query variants*.

Using Variants at Runtime

If there is only one data source in the application that supports variants, only the variant select box will be displayed at runtime.

If there is more than one data source, a data source select box will also be displayed, where the application user must select the data source first.

If the application user selects an existing variant, the corresponding variables will be filled in automatically.

To change the variable values of an existing variant, the application user needs to proceed as follows:

1. Select an existing variant.
2. Change the variable values.
3. Save the changes.

This creates a new user variant with the defined variable values and a generated technical name. The name given by the user is set as description. The new user variant replaces the previous variant that was selected.

To edit the technical name, text, or scope of an existing variant, the application user must press the edit button. This opens the *Edit variants* dialog box, where all variants of the (pre-) selected data source can be edited and deleted.

For variants with user scope, the technical name is generated and cannot be edited. If the scope of the variant is public, the technical name can also be edited and should be unique within the selected data source.

31 Working with Drag and Drop in Applications and Crosstabs

Prerequisites

When working with drag and drop functionality in applications and crosstabs, please note the following points:

- Drag and Drop cannot be used when the crosstab is used in the planning mode, that means if the crosstab contains input ready cells.
- Drag and Drop is not supported for mobile scenarios (for example, applications running on the iPhone or iPad)

Enabling/Disabling Drag and Drop

Drag and Drop can be enabled and disabled on application level and on component level:

- On Application Level
Use the application property [Drag and Drop between Components](#) to specify if drag and drop operations between different components are allowed. This property is set to [false](#) by default, which means that drag and drop operations cannot be carried out between components. Drag and Drop operations within one single component are still possible, if the crosstab (component) level property is set to [true](#) or [Advanced](#) (see next section). Set this property to [true](#) or [Advanced](#), if you want to enable drag and drop operations between components (for example, between the Navigation Panel and the Crosstab).
- On Crosstab (component) Level
Use the crosstab property [Drag and Drop enabled](#) if you want to enable drag and drop operations within the crosstab. The property is set to [false](#) by default, what means that the crosstab does not allow any drag and drop operations. If this property is set to [true](#) or [Advanced](#), the crosstab:
 - enables internal drag and drop operations. Thus the application user can drag and drop dimensions and members within the crosstab and remove dimensions and members by dragging and dropping them outside the area of the crosstab.
 - accepts external drops of dimensions from other components (for example, the navigation panel). This only works if the application property [Drag and Drop between Components](#) is set to [true](#) or [Advanced](#) as well.

For further information about the respective application and crosstab properties, see [Properties of the Application \[page 419\]](#) and [Crosstab \[page 435\]](#).

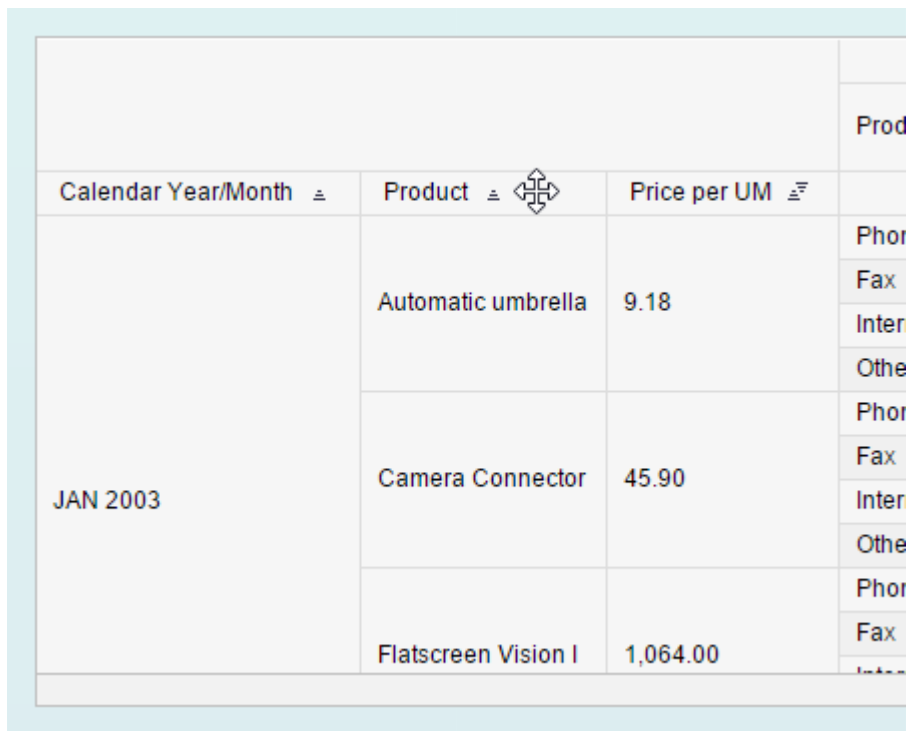
In the following documentation chapters, you will find more specific information about different drag and drop aspects.

31.1 Dragging a Dimension or Dimension Member

General aspects

Dimensions or members that can be dragged can be easily identified: the mouse cursor changes when you hover above them. Drag and drop operations in the crosstab can only be performed for one single dimension or one single dimension member. You cannot select multiple members and drag the members in the selection around the crosstab, even if the *Selection Type* property of the crosstab has been set to *multi*.

The mouse cursor changes to the "move" cursor type (dimension "Product", for example):



			Prod
Calendar Year/Month	Product	Price per UM	
JAN 2003	Automatic umbrella	9.18	Phor
			Fax
			Inter
			Othe
	Camera Connector	45.90	Phor
			Fax
			Inter
			Othe
	Flatscreen Vision I	1,064.00	Phor
			Fax
			Inter

When you start to drag a dimension or member, the drag ghost displays all cells that belong to the respective dimension. This means text fields and attributes are displayed. In this example, the dimension "Product" has the attribute "Price per UM", which is displayed in the drag operation as well:

			Product group
Calendar Year/Month	Product	Price per UM	
JAN 2003	Product	Price per UM	Phone
	Automatic umbrella	9.18	Fax
			Internet
			Others
	Camera Connector	45.90	Phone
			Fax
			Internet
			Others
	Flatscreen Vision I	1,064.00	Phone
			Fax
			Internet

For all drag (and drop) operations, dimension and member data with all related fields like texts or attributes are always treated as a single unit. In this example, this means it is not possible to drag and drop “Price per UM” separately, because it is an attribute for dimension “Product”. This behavior is also valid for the “Product” cell that represents the “Product” dimension’s text. Therefore, you can grab and drag any cell (text, attribute) for a dimension, in order to move the whole cell block that belongs to that dimension.

Drag and drop operations within the crosstab as well as drag drop operations from the navigation panel to the crosstab can be canceled before dropping the element by pressing the **ESC** key on the keyboard.

The dimension header split cell

Typically, when no scaling factor is specified on the rows or on the columns axis in the drilldown, the pivot cell of the dimension header hosts the respective dimension from the rows and from the columns axis. In order to drag a dimension out of this cell, you need to render this cell as a "split" cell with a diagonal separator. The two areas created by this separator determine which dimension will be dragged when dragging is started from a position within the cell. In the following example, the “Currency” dimension on the rows axis is dragged when starting the drag operation from the lower left cell area:

		0BC_COUNT		
	Fiscal year	K4/2004		
		Calendar year, 4 spec. periods 2004		
	Currency Calendar Year/Month	JAN 2004	FEB 2004	MAR 2004
	Euro			
	Euro			
	Euro			
	Euro			
	Euro			
	Euro			
	Euro	1		
	Euro			
	Euro			
	Euro	1		
	Euro			

Starting from the same cell, but using the upper right cell area, the “Calendar Year/Month” dimension on the columns axis is dragged:

		0BC_COUNT		
	Calendar Year/Month	K4/2004		
	Fiscal year	Calendar year, 4 spec. periods 2004		
	Currency Calendar Year/Month	JAN 2004	FEB 2004	MAR 2004
	Euro			
	Euro			
	Euro			
	Euro			
	Euro			
	Euro			
	Euro	1		
	Euro			
	Euro			
	Euro	1		
	Euro			

Structures usually do not have a text rendered in the crosstab. The drag ghost always displays a text, and hence a split cell, as shown below, allows you for example to grab and drag the measure structure out of the pivot cell:

Calendar Year/Month	JAN 2004			
Fiscal year	1/4/2004	Measures Calendar year, 4 spec. periods 2004		
Currency				
	0BC_COUNT	0BC_DATE2	0BC_DATE3	
Euro				
Euro				
Euro				
Euro				
Euro				
Euro	1	1/13/04	1/13/04	
Euro				
Euro				
Euro	1	1/29/04	1/29/04	
Euro				

31.2 Removing a Dimension or Dimension Member

You can remove a dimension or dimension member from the drilldown by dragging the element out of the visible crosstab area. When you do this, the drag ghost will show a trash can symbol to indicate that dropping the element at the current position will remove the element:

			Formula 3			CALC			FORMULA		
			DS30	DS20	Overall Result	DS30	DS20	Overall Result	DS30	DS20	
			Hardware software	Accessories+space		Hardware software	Accessories+space		Hardware software	Accessories+space	
Calendar Year/Month	Product	Price per UM	ST	ST	ST	\$	\$	\$	\$	\$	
JAN 2003	Automatic umbrella	9.18	Phone	1,704,488	1,704,488		1,955,470.00	1,955,470.00		7,821,880.00	
			Fax	2,480,748	2,480,748		2,846,088.50	2,846,088.50		11,384,354.00	
			Internet	1,255,412	1,255,412		1,440,321.00	1,440,321.00		5,761,284.00	
			Others	599,876	599,876		688,267.50	688,267.50		2,753,070.00	
	Camera Connector	45.90	Phone	5,294,964	5,294,964		3,032,347.50	3,032,347.50		12,129,390.00	
			Fax	7,436,942	7,436,942		4,258,818.00	4,258,818.00		17,035,272.00	
			Internet	4,142,948	4,142,948		2,372,611.50	2,372,611.50		9,490,446.00	
			Others	1,874,248	1,874,248		1,073,347.50	1,073,347.50		4,293,390.00	
	Flatscreen Vision I	1,064.00	Phone	2,136,620	2,136,620	2,817,174.50		2,817,174.50	11,268,698.00		
			Fax	3,145,856	3,145,856	4,147,696.50		4,147,696.50	16,590,786.00		
			Internet	1,440,321	1,440,321	1,440,321.00		1,440,321.00	5,761,284.00		
			Others	599,876	599,876	688,267.50		688,267.50	2,753,070.00		

Note

Keep in mind that structures cannot be removed from the drilldown unless they only contain a single member. If you want to remove members of a structure (for example, the measure structure), at least one structure member must remain in the structure. Otherwise you cannot remove a member.

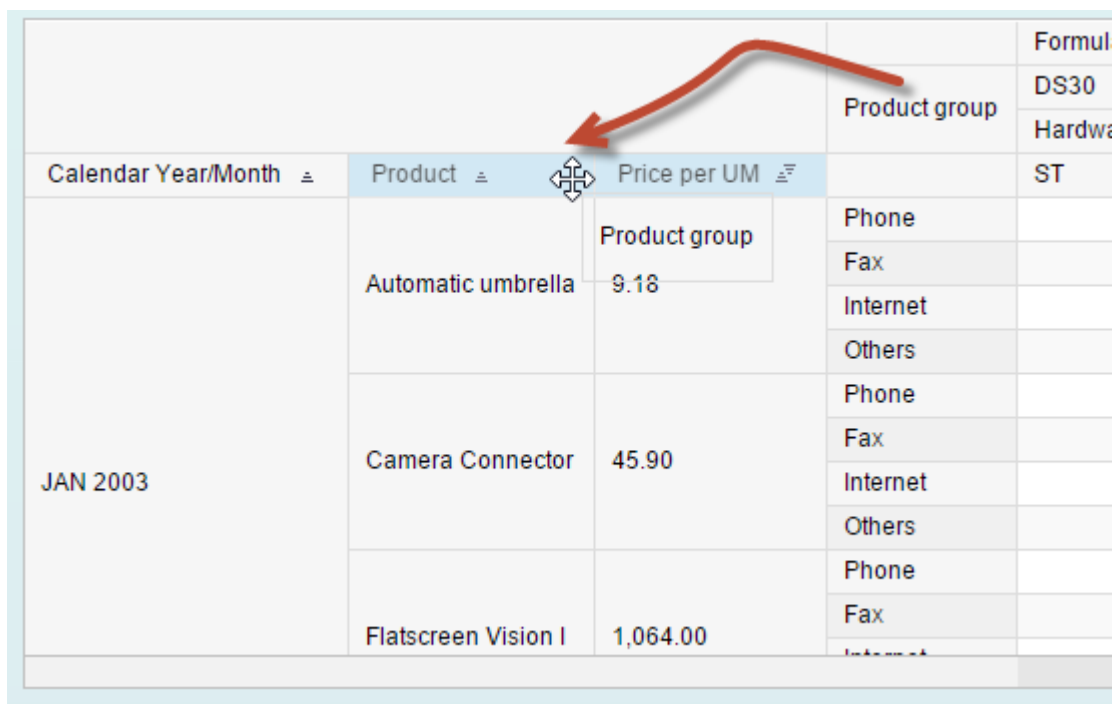
31.3 Dropping Dimensions or Dimension Members

Drop targets for dimensions and dimension members can be other dimensions (or, in the case of members, other members), or the respective drop areas between dimensions and members. Dropping a dimension on another dimension or a member on another member exchanges the two elements, whereas dropping a member/dimension on a respective drop area inserts the member/dimension at the respective drop area position.

A valid drop target is highlighted accordingly, for example, when you drag the “Product Group” dimension onto the “Product” dimension or the “Phone” member onto the “Internet” member, as in the following examples:

Exchange operations

Dimension drop:



Calendar Year/Month	Product	Price per UM	Product group	Formul
JAN 2003	Automatic umbrella	9.18	Product group	DS30
			Phone	Hardwa
			Fax	
			Internet	
	Camera Connector	45.90	Others	
			Phone	
			Fax	
			Internet	
	Flatscreen Vision I	1,064.00	Others	
			Phone	
			Fax	
			Internet	

Note

The drop target highlighting also highlights all cells that belong to the dimension, including text/key and attribute field cells.

A "split cell" that is a pivot cell that hosts both a dimension on the rows axis and columns axis (for more information, see section *The dimension header split cell* in [Dragging a Dimension or Dimension Member \[page 258\]](#)), cannot be a drop target. In order to drop onto a dimension that is contained in a split cell, you first have to swap or move the dimension out of this cell.

Member drop:

			Formula 3	
			Product group	
			DS30	DS20
			Hardware software	Accessori
Calendar Year/Month	Product	Price per UM	ST	ST
JAN 2003	Automatic umbrella	9.18	Phone	
			Fax	
			Internet	
			Others	
	Camera Connector	45.90	Phone	
			Fax	
			Internet	
			Others	
	Flatscreen Vision I	1,064.00	Phone	2,136,620
			Fax	3,145,856
			Internet	1,646,568
			Others	

If you release the left mouse button at the respective positions, the dimensions/members are exchanged with each other.

Insert/move operations

You can perform an insert operations by dragging elements and dropping them on drop areas, which are displayed as horizontal or vertical lines when you hover over them:

The following example shows how to drop a dimension on the rows axis by dragging the “Product Group” dimension between “Calendar Year/Month” and “Product”.

Note

Since “Product” and “Price per UM” belong to the same dimension (“Product”), you cannot position “Product Group” between these two, and hence no drop area appears when you hover over the respective position. The cell to the left of the drop area, is grayed out, except when the drop area at the very left (or very right for RTL) or the drop area at the very top is highlighted. In this case, the grayed out cell is the cell next to (left, right or below) the cell where the dragged dimension will be inserted.

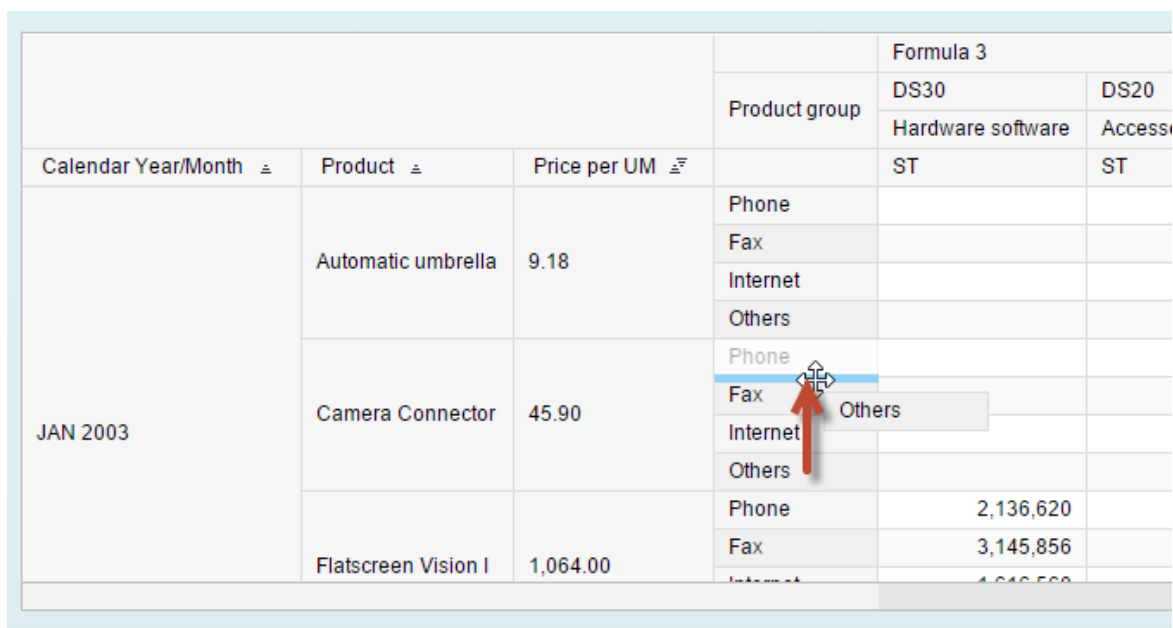
Calendar Year/Month	Product	Price per UM
JAN 2003	Product group	
	Automatic umbrella	9.18
	Camera Connector	45.90
	Flatscreen Vision I	1,064.00

The following example shows how to drop a dimension on the columns axis by positioning the “Distribution Channel” structure between “Product Group” and the measure structure.

If you release the left mouse button, the dragged dimension is moved into the new position.

If you want to insert members, proceed like in the following examples:

- Moving dimension member “Others” ahead of “Fax”. This means insert the member between “Phone” and “Fax”.



Calendar Year/Month	Product	Price per UM	Product group	Formula 3	
				DS30	DS20
				Hardware software	Accessories
				ST	ST
JAN 2003	Automatic umbrella	9.18	Phone		
			Fax		
			Internet		
			Others		
	Camera Connector	45.90	Phone		
			Fax		
			Internet		
			Others		
	Flatscreen Vision I	1,064.00	Phone	2,136,620	
			Fax	3,145,856	
			Internet	1,616,568	
			Others		

- This procedure also works for other elements on the rows axis and columns axis, like inserting “DS30” between “DS20” and the “Overall Result”.

	Product group	Formula 3		Overall Result	CALC
		DS30	DS20		DS30
		Hardware software	Accessories+space		Hardware sof
Price per UM ₪		ST	ST	DS30 ST Hardware software	\$
9.18	Phone		1,704,488	1,704,488	
	Fax		2,480,748	2,480,748	
	Internet		1,255,412	1,255,412	
	Others		599,876	599,876	
45.90	Phone		5,294,964	5,294,964	
	Fax		7,436,492	7,436,492	
	Internet		4,142,948	4,142,948	
	Others		1,874,248	1,874,248	
1,064.00	Phone	2,136,620		2,136,620	2,817,1
	Fax	3,145,856		3,145,856	4,147,6
	Internet	4,646,568		4,646,568	3,134,4

31.4 Drag and Drop Constraints

When using drag and drop, please note the following points:

- Drag and drop only works for single elements under the mouse cursor. This means only one dimension/ dimension member at a time can be dragged and dropped.
- If there are two or more structures on the axes, a structure can only be removed if it contains no more than one member . If there is only one structure, it can always be removed.
- Dimensions can only be dropped on dimensions and the respective drop areas, if the setting *Drag and Drop Enabled* is set to *true*. If this setting is set to *Advanced*, dimensions can also be drag and dropped to data cells.
- Dimension members can only be dropped on dimension members and the respective drop areas.

- Members can only be dragged and dropped within the previous peer dimension's member space. Example:

			Formula 3		
			Product group	DS30	DS20
				Hardware software	Access
Calendar Year/Month	Product	Price per UM		ST	ST
JAN 2003	Automatic umbrella	9.18	Phone		
			Fax		
			Internet		
			Others		
	Camera Connector	45.90	Phone		
			Fax		
			Internet		
			Others		
	Flatscreen Vision I	1,064.00	Phone	2,136,620	
			Fax	3,145,856	
			Internet	1,046,500	

In this example, the structure member "Others" can only be moved within the space that is spanned by the previous peer dimension's member, which is "Camera Connector".

- Hierarchy members can only be dragged and dropped within the same drill level
- Hierarchy members cannot be removed.
- A "split cell" (pivot cell) is not a drop target.
- "Result" members can be dragged, but cannot be dropped on drop areas or other dimension members. For other members you cannot choose result members as drop targets. The only drop action available to "Result" members is to drop them outside the crosstab to remove them from the view.

31.5 Dragging and Dropping Between Crosstab and Navigation Panel

If you want to drag and drop between the navigation panel and the crosstab, note the following points:

- You first have to enable drag and drop between different components by setting the application property *Drag and Drop between Components* to *true*. For more information, see section *Enabling/Disabling Drag&Drop* in [Dragging and Dropping Between Crosstab and Navigation Panel \[page 266\]](#)
- Only dimensions can be dragged from the navigation panel and dropped on crosstab dimensions and their respective drop areas.
- You can cancel the drag operation by pressing the **[ESC]** key on the keyboard.
- The navigation panel can only act as a dimension drag source for the crosstab. Members cannot be dragged from the navigation panel, and the navigation panel is not a drop target for members or dimensions that are dragged from a crosstab.
- If dimensions, which are already contained on any axis of the crosstab drilldown, are dragged from the navigation panel to the crosstab, the crosstab will not accept the drop, neither for dimension cells nor for drop areas. Only dimensions that are not already in the drilldown are accepted. Thus, if you want to reorder

dimensions, you need to do this either within the crosstab or within the navigation panel. You cannot reorder by dragging and dropping a dimension already contained on an axis from the navigation panel to the crosstab.

31.5.1 Dragging and Dropping a Dimension to a Data Cell

If you want detailed information on a specific figure within the crosstab, you can drag and drop a dimension which is currently not visible in the crosstab onto a data cell.

Prerequisites

If you want to use this feature, the crosstab property *Drag and Drop Enabled* must be set to *Advanced*.

Additionally, the dimension must be located on the free axis.

When you drag a dimension from outside the crosstab (for example from the Navigation Panel) onto a data cell in the crosstab, this data cell is marked as droppable upon hover.

			Store Sales		
		Marital Status	M		
First Name <small>⌵</small>	Customer Name <small>⌵</small>	Store City Gender <small>⌵</small>	F <small>⌵</small>	M <small>⌵</small>	Result <small>⌵</small>
A. Catherine	Binkley	5371 Holland Circle			
		Result			
	Result				
Aaron	Lemay	2342 Waltham St.	Store	36.42	36.42
		Result		36.42	36.42
	Story	5179 Valley Ave		50.23	50.23
		Result		50.23	50.23
		Result		86.65	86.65
Abe	Tramel	5006 Highland Drive			
		Result			
	Result				
Abel	Hawkins	1077 Wharf Drive			
		Result			
	Young	1077 Wharf Drive			
		Result			
Result					
Abraham	Swearengin	5006 Highland Drive		26.42	26.42
		Result		26.42	26.42
	Result		26.42	26.42	
Achari	Harp	4365 Indigo Ct		40.72	40.72
		Result		40.72	40.72
	Result		40.72	40.72	

Upon drop, all dimensions on the rows and columns axis except the measure structure are removed from the axes. The dropped dimension is put to the opposing axis of the measure structure. Additionally; all dimension members associated to the data cell are set as filters.

❖ Example

The crosstab shown has the following dimensions on the axes:

- rows axis
 - first name
 - customer name
 - store city
- columns axis
 - measure structure
 - marital status
 - gender

Search

First Name

Customer Name

Store City

Measures

Marital Status

Gender

			Store Sales							
			Marital Status		M			S		
First Name	Customer Name	Store City Gender	F	M	Result	F	M	Result	Result	
A. Catherine	Binkley	5371 Holland Circle				108.94		108.94	108.94	
		Result				108.94		108.94	108.94	
		Result				108.94		108.94	108.94	
Aaron	Lemay	2342 Waltham St.		36.42	36.42				36.42	
		Result		36.42	36.42				36.42	
		5179 Valley Ave		50.23	50.23				50.23	
		Story		50.23	50.23				50.23	
		Result		86.65	86.65				86.65	
Abe	Tramel	5006 Highland Drive					58.75	58.75	58.75	
		Result					58.75	58.75	58.75	
		Result					58.75	58.75	58.75	
Abel	Hawkins	1077 Wharf Drive				60.93		60.93	60.93	
		Result				60.93		60.93	60.93	
		1077 Wharf Drive					37.63	37.63	37.63	
		Young					37.63	37.63	37.63	
		Result				98.56	98.56	98.56		
Abraham	Swearengin	5006 Highland Drive		26.42	26.42				26.42	
		Result		26.42	26.42				26.42	
		Result		26.42	26.42				26.42	
Achari	Harp	4365 Indigo Ct		40.72	40.72				40.72	
		Result		40.72	40.72				40.72	
		Result		40.72	40.72				40.72	

Now the dimension *Store* is dropped onto the data cell *36.42*. This data cell corresponds to the following tuples (highlighted in the above image):

- Rows Tuple
 - First Name = Aaron
 - Customer Name = Lemay
 - Store City = 2342 Waltham St.
- Columns Tuple
 - Measure Structure = Store Sales
 - Marital Status = M
 - Gender = M

After the drop is completed, all these tuples are set as background filters. Additionally, *Store* was set to the rows axis, as the measure structure is on the columns axis.

If any tuple member of the associated data cell is a result member (*Result* or *Overall Result*), no background filter is set for this dimension.

Search

First Name

Aaron

Customer Name

Lemay

Store City

2342 Waltham St.

Measures

Store Sales (store_sales)

Marital Status

M

Gender

M

Store	Store Sales
24	36.42
Overall Result	36.42

31.6 Drag and Drop in Crosstabs with Property "Display Repeated Texts"

If you set the crosstab property *Display Repeated Texts true*, this affects how drag and drop works with members. Dragging and dropping of dimensions will remain unaffected.

In a crosstab with repeated texts, all drag and drop operations are applied logically to the dimension members with the finest drilldown granularity, despite the fact that the visualization of the drag and drop operation is applied to a whole crosstab row or column, and you can start by grabbing any cell in the respective row or column. Hence, it is not possible to drag and drop the lines everywhere. Instead the system checks what is

allowed and what is not allowed, based on the finest drilldown granularity dimension member. The following examples explain this behavior:

Calendar Year/Month	Currency	OBC_VEND1	OBC_COUNT	OBC_DATE2	OBC_DATE3
JAN 2004	Euro	Arrowspace	5	1/30/04	1/5/04
JAN 2004	Euro	Bernardin SA	4	1/28/04	1/14/04
JAN 2004	Euro	C.A.S. Computer Appl			
JAN 2004	Euro	Bush Distribution	5	1/29/04	1/7/04
JAN 2004	Euro	Bush Holdings, Inc.	9	1/24/04	1/1/04
JAN 2004	Euro	C.A.S. Computer Appl	4	1/25/04	1/2/04
JAN 2004	Euro	Result	27	1/30/04	1/1/04
FEB 2004	Euro	Arrowspace	5	2/18/04	2/6/04
FEB 2004	Euro	Bernardin SA	4	2/12/04	2/2/04
FEB 2004	Euro	Bush Distribution	3	2/21/04	2/14/04
FEB 2004	Euro	Bush Holdings, Inc.	2	2/23/04	2/2/04
FEB 2004	Euro	C.A.S. Computer Appl	2	2/26/04	2/6/04

To start drag and drop, the user can grab and start dragging any cell in the row, for example “JAN 2004” or “Euro” or “C.A.S Computer Appl”. The drag visualization shows the whole line, but the drag and drop operation is effectively applied to the member of “OBC_VEND1” because this is the finest granularity dimension on the rows in the given drilldown. Therefore the user is actually logically dragging “C.A.S Computer Appl”.

As a result, the line cannot be dragged and dropped somewhere outside of the “JAN 2004” block of “Calendar Year/Month”. Member reordering of OBC_VEND1 members is only possible within this block. This behavior is very similar to the behavior you would observe if the property *Display Repeated Texts* would be set to *false*.

In fact, using the property *Display Repeated Texts* in the crosstab is even more restrictive when using drag and drop for members. The following example, show the same crosstab as in the example above but with the property *Display Repeated Texts* set to *false*:

Calendar Year/Month	Currency	OBC_VEND1	OBC_COUNT	OBC_DATE2	OBC_DATE3
JAN 2004	Euro	Arrowspace	5	1/30/04	1/5/04
		Bernardin SA	4	1/28/04	1/14/04
		Bush Distribution	5	1/29/04	1/7/04
		Bush Holdings, Inc.	9	1/24/04	1/1/04
		C.A.S. Computer Appl	4	1/25/04	1/2/04
		Result	27	1/30/04	1/1/04
FEB 2004	Euro	Arrowspace	5	2/18/04	2/6/04
		Bernardin SA	4	2/12/04	2/2/04
		Bush Distribution	3	2/21/04	2/14/04
		Bush Holdings, Inc.	2	2/23/04	2/2/04
		C.A.S. Computer Appl	2	2/26/04	2/6/04

In this example, the “JAN 2004” member can be exchanged with the “FEB 2004” member because the drag and drop operation operates on the whole block, which these members span in the dimensions to the right of the crosstab.

Since the drag and drop operation does not operate on a selection in the crosstab, it is not possible to achieve this behaviour as described above when the property *Display Repeated Texts* is set to *true*. This is because there would only be a single row drag and drop action that is applied to a OBC_VEND1 member, and dragging and dropping a single “JAN 2004” row between multiple “FEB 2004” rows would result in an invalid state.

32 Working With Copy and Paste in Crosstabs

You can copy data from the crosstab and paste the copied values into planning-enabled cells of any other crosstab or Microsoft Excel sheet, notepad, etc.

Prerequisites

When working with copy and paste functionality in crosstabs, please note the following points:

- Using the copy functionality in crosstabs works for crosstabs that are in planning or non-planning mode. The paste functionality, on the other hand, is only possible if the crosstab is in planning mode, which means the crosstab has input-enabled cells.
- You can only copy the visible area of the selected crosstab part on your screen. If for example you select a column for copying that can be scrolled, only the visible part of the column on your screen is copied.

Using Data Cells Rectangle Selection as Selection Type

To enable application users to select data by rows or columns (similar to the selection mode [Multiple Rows/Columns](#)), specify [Data Cells Rectangle](#) in the [Selection Type](#) property of the crosstab. With this selection mode, the application user can select multiple data cells freely by clicking a cell, keeping the mouse key clicked, and dragging the cursor to another data cell. The rectangle created by this move, is highlighted. The highlighted cells are selected:

	0BCPL_NUM	0BCPL_NUMFL	0BCPL_PRICE	0BCPL_PRIVA
0BCPL_PLANT	* 10	* 100	EUR	EUR
PLANT001	0.100	0.0150000	0.00	0.00
PLANT001	0.100	0.0150000	0.00	0.00
	0.200	0.0300000	0.00	0.00
PLANT001	0.100	0.0150000	0.00	0.00
PLANT001	0.100	0.0150000	0.00	0.00
	0.200	0.0300000	0.00	0.00
PLANT001	0.100	0.0150000	0.00	0.00

If a rectangular data set is selected and the script function `getSelection()` is called, the first cell in the upper left corner of the selection gets returned.

Copying Data From a Crosstab

To copy data from a crosstab you must have chosen a selection type chosen other than *None*. Press **CTRL** + **C** to start the copy process.

i Note

You can only copy data from a continuous selection, meaning that no blank fields are allowed in the selection when copying.



If the selection is invalid, an error message will inform you that the copy was not accepted:

Pasting Data to a Crosstab



To paste data to a crosstab, select a data cell where the paste area should begin. Press **CTRL** + **V** to start the paste process.

If you have copied data from a crosstab, the scaling factors for data cells are taken into account. This means that copying data from a column with scaling factor *10 to a column with scaling factor *100 will lead to the data being divided by 10 to fit the target scaling factor:

Values upon copy

0BCPL_NUM 	0BCPL_NUMFL 
* 10	* 100
0.100	0.0150000
0.100	0.0150000
0.200	0.0300000
0.100	0.0150000
0.100	0.0150000
0.200	0.0300000
0.100	0.0150000
0.500	0.0750000
0.100	0.0150000
0.100	0.0150000
0.200	0.0300000
0.300	0.0450000
0.400	0.0600000
0.100	0.0150000
0.100	0.0150000
0.100	0.0150000
0.200	0.0300000
0.300	0.0450000
0.700	0.1050000
0.200	0.0300000
0.100	0.0150000
0.300	0.0450000
1.500	0.2250000

Values after paste of column OBCPL_NUM to OBCPL_NUMFL:

OBCPL_NUM 	OBCPL_NUMFL 
* 10	* 100
0.100	0.0100000
0.100	0.0100000
0.200	0.0200000
0.100	0.0100000
0.100	0.0100000
0.200	0.0200000
0.100	0.0100000
0.500	0.0500000
0.100	0.0100000
0.100	0.0100000
0.200	0.0200000
0.300	0.0300000
0.400	0.0400000
0.100	0.0100000
0.100	0.0100000
0.100	0.0100000
0.200	0.0200000
0.300	0.0300000
0.700	0.0700000
0.200	0.0200000
0.100	0.0100000
0.300	0.0300000
1.500	0.1500000

This behavior is also applied if the paste is not within the same crosstab, but to any other crosstab in the application. If you have copied the data from an external source, the data is posted as it is. A calculation regarding scaling factors is not possible.

i Note

As with copying, the pasting is not accepted and an error message is displayed, if the selected area is invalid.

Pasting Data to Microsoft Excel/Notepad

If you paste values to Microsoft Excel, Notepad, or any other external program, the values are copied in a 'What you see is what you get' (WYSIWYG) approach.

0BCPL_PRO	Calendar Year/Month	0BCPL_PLANT	0BCPL_NUM * 10	0BCPL_NUMFL * 100	0BCPL_PRICE EUR	0BCPL_PRIVA EUR	0BCPL_
PROD00003	02.2003	PLANT001	0.100	0.0150000	0.00	0.00	L
	01.2003	PLANT001	0.100	0.0150000	0.00	0.00	
	Result	Result	0.200	0.0300000	0.00	0.00	

This means that scaling factors are not taken into account, and instead the data values are copied as they are displayed in the table:

1	PROD00003	02.2003	PLANT001	0.100	0.0150000	0.00	0.00	100
2	PROD00003	01.2003	PLANT001	0.100	0.0150000	0.00	0.00	100
3	PROD00003	Result	Result	0.200	0.0300000	0.00	0.00	200

	A	B	C	D	E	F	G	H
1	PROD00003	02.2003	PLANT001	0.100	0.0100000	0.00	0.00	100
2	PROD00003	01.2003	PLANT001	0.100	0.0100000	0.00	0.00	100
3	PROD00003	Result	Result	0.200	0.0200000	0.00	0.00	200

33 Working with the Crosstab Sorting

You can sort the crosstab by its data via icons in the column header area, via context menu, or via scripting.

Sorting with Icons in the Column Header

In past releases, when you clicked a sorting icon in the column header of the crosstab, the crosstab was sorted by the corresponding measure. This meant that when there were drilled-down columns of a measure, clicking an icon always sorted the crosstab's data by the measure's aggregate. In other words, clicking a sorting icon in the column header showed the same behavior as the `sortByMeasure` scripting API.

	0BC_ORDW			0BC_TURN		
0BC_REDU	1% Rabatt ▲▼	2% Rabatt ▲▼	Overall Result ▲▼	1% Rabatt ▲▼	2% Rabatt ▲▼	Overall Result ▲▼
0BC_LOC ▲	KG	KG	KG	EUR	EUR	EUR
Goeteborg B aeckebol (PL)	15.700	8.140	23.840	122.44	222.66	345.10
Halle/Leipzig (GR)	4.780	16.890	21.670	40.62	140.69	181.31
Mannheim (GB_ENG)	19.030	4.220	23.250	178.91	26.39	205.30
Spreitenbach (AG) (B)	15.840	7.000	22.840	115.74	86.87	202.61
Utrecht (N)	6.800	20.310	27.110	36.86	109.79	146.65
Wien Nord (GR)	14.370	6.630	21.000	129.94	19.98	149.92
Overall Result	76.520	63.190	139.710	624.51	606.38	1,230.89



	0BC_ORDW			0BC_TURN		
0BC_REDU	1% Rabatt ▲	2% Rabatt ▲	Overall Result ▲	1% Rabatt ▲▼	2% Rabatt ▲▼	Overall Result ▲▼
0BC_LOC ▲▼	KG	KG	KG	EUR	EUR	EUR
Wien Nord (GR)	14.370	6.630	21.000	129.94	19.98	149.92
Halle/Leipzig (GR)	4.780	16.890	21.670	40.62	140.69	181.31
Spreitenbach (AG) (B)	15.840	7.000	22.840	115.74	86.87	202.61
Mannheim (GB_ENG)	19.030	4.220	23.250	178.91	26.39	205.30
Goeteborg B aeckebol (PL)	15.700	8.140	23.840	122.44	222.66	345.10
Utrecht (N)	6.800	20.310	27.110	36.86	109.79	146.65
Overall Result	76.520	63.190	139.710	624.51	606.38	1,230.89

With SAP Lumira 2.3, when you click a sorting icon in the column header of a crosstab it sorts the crosstab's data by the column corresponding to the sorting icon, whether there is a drill-down or not.

	0BC_ORDW			0BC_TURN		
0BC_REDU	1% Rabatt ▴▾	2% Rabatt ▴▾	Overall Result ▴▾	1% Rabatt ▴▾	2% Rabatt ▴▾	Overall Result ▴▾
0BC_LOC ▴▾	KG	KG	KG	EUR	EUR	EUR
Goeteborg B aeckebol (PL)	15.700	8.140	23.840	122.44	222.66	345.10
Halle/Leipzig (GR)	4.780	16.890	21.670	40.62	140.69	181.31
Mannheim (GB_ENG)	19.030	4.220	23.250	178.91	26.39	205.30
Spreitenbach (AG) (B)	15.840	7.000	22.840	115.74	86.87	202.61
Utrecht (N)	6.800	20.310	27.110	36.86	109.79	146.65
Wien Nord (GR)	14.370	6.630	21.000	129.94	19.98	149.92
Overall Result	76.520	63.190	139.710	624.51	606.38	1,230.89



	0BC_ORDW			0BC_TURN		
0BC_REDU	1% Rabatt ▴▾	2% Rabatt ▴▾	Overall Result ▴▾	1% Rabatt ▴▾	2% Rabatt ▴▾	Overall Result ▴▾
0BC_LOC ▴▾	KG	KG	KG	EUR	EUR	EUR
Mannheim (GB_ENG)	19.030	4.220	23.250	178.91	26.39	205.30
Wien Nord (GR)	14.370	6.630	21.000	129.94	19.98	149.92
Spreitenbach (AG) (B)	15.840	7.000	22.840	115.74	86.87	202.61
Goeteborg B aeckebol (PL)	15.700	8.140	23.840	122.44	222.66	345.10
Halle/Leipzig (GR)	4.780	16.890	21.670	40.62	140.69	181.31
Utrecht (N)	6.800	20.310	27.110	36.86	109.79	146.65
Overall Result	76.520	63.190	139.710	624.51	606.38	1,230.89

Sorting via Context Menu


In past releases, when in the context menu you chose **Sort ▸ Ascending/Descending**, the crosstab was sorted by the selected measure. The sorting behaved exactly like the `sortByMeasure` scripting API.

With SAP Lumira 2.3, the context menu supports sorting for measures and drill-downs of a measure. Choosing the context menu causes the crosstab to be sorted by the corresponding column or row.

0BC_REDU	0BC_ORDW	1% Rabatt	Sort	Ascending	Overall Result
0BC_LOC	KG		22 Number Format	Descending	EUR
Goeteborg B aeckebo (PL)	15.70		Add Dynamic Calculation	0	122.44
Halle/Leipzig (GR)	4.78		Calculate Totals As	0	40.62
Mannheim (GB_ENG)	19.03		Add Calculation	0	178.91
Spreitenbach (AG) (B)	15.84		Display	0	26.39
Utrecht (N)	6.80		Filter	0	86.87
Wien Nord (GR)	14.37			0	109.79
Overall Result	76.52			0	19.98
				0	624.51
				0	606.38
				0	1,230.89

0BC_REDU	0BC_ORDW	0BC_TURN				
0BC_REDU	1% Rabatt	2% Rabatt	Overall Result	1% Rabatt	2% Rabatt	Overall Result
0BC_LOC	KG	KG	KG	EUR	EUR	EUR
Utrecht (N)	6.800	20.310	27.110	36.86	109.79	146.65
Goeteborg B aeckebo (PL)	15.700	8.140	23.840	122.44	222.66	345.10
Mannheim (GB_ENG)	19.030	4.220	23.250	178.91	26.39	205.30
Spreitenbach (AG) (B)	15.840	7.000	22.840	115.74	86.87	202.61
Halle/Leipzig (GR)	4.780	16.890	21.670	40.62	140.69	181.31
Wien Nord (GR)	14.370	6.630	21.000	129.94	19.98	149.92
Overall Result	76.520	63.190	139.710	624.51	606.38	1,230.89

Sorting via context menu does work independent of the axis the measure lies on. The cell for which the context menu is opened just needs to be a drill-down of a measure (or a measure itself). In the following picture the crosstab's data is sorted by the row *0BC_ORDW*.



	0BC_REDU	0BC_LOC	Goeteborg B aeckebo (PL)	Halle/Leipzig (GR)	Mannheim (GB_ENG)	Spreitenbach (AG) (B)	Ut...
0BC_ORDW	1% Rabatt	KG	15.700	4.780	19.030	15.840	6...
	2% Rabatt	KG	8.140	16.890	4.220	7.000	20...
	Overall Result	KG	23.250	22.840	23.840	22.840	27...
0BC_TURN	1% Rabatt	EUR	22.66	140.69	178.91	115.74	36...
	2% Rabatt	EUR	45.10	181.31	205.30	202.61	14...

	0BC_REDU	0BC_LOC	Mannheim (GB_ENG)	Wien Nord (GR)	Spreitenbach (AG) (B)	Goeteborg B aeckebo (PL)	Halle...
0BC_ORDW	1% Rabatt	KG	19.030	14.370	15.840	15.700	4.780
	2% Rabatt	KG	4.220	6.630	7.000	8.140	16.890
	Overall Result	KG	23.250	21.000	22.840	23.840	21.670
0BC_TURN	1% Rabatt	EUR	178.91	129.94	115.74	122.44	40.62
	2% Rabatt	EUR	26.39	19.98	86.87	222.66	140.69
	Overall Result	EUR	205.30	149.92	202.61	345.10	181.31

Sorting sortMeasureByColumnOrRow Scripting API

The scripting API `sortMeasureByColumnOrRow(columnOrRow, isSortAscending)` for a data source lets you sort the crosstab by scripting just as you sort via column header icons or via the context menu. Its signature needs a combination of dimension members to determine the column or row which should be sorted and a boolean flag which determines whether the crosstab should be sorted in an ascending or descending order.

❖ Example

In this example the crosstab gets sorted descending by column OBC_ORDW via the following script:

```
DS_1.sortMeasureByColumnOrRow({
  "0MEASURES000000000000000009": "0BC9ORDW",
  "0BC_REDU": "2% Rabatt"
}, false);
```

The script above gives the following sorting result:

	OBC_ORDW			OBC_TURN		
OBC_REDU	1% Rabatt ▲▼	2% Rabatt ▼	Overall Result ▲▼	1% Rabatt ▲▼	2% Rabatt ▲▼	Overall Result ▲▼
OBC_LOC ▲▼	KG	KG	KG	EUR	EUR	EUR
Utrecht (N)	6.800	20.310	27.110	36.86	109.79	146.65
Halle/Leipzig (GR)	4.780	16.890	21.670	40.62	140.69	181.31
Goeteborg B aeckebol (PL)	15.700	8.140	23.840	122.44	222.66	345.10
Spreitenbach (AG) (B)	15.840	7.000	22.840	115.74	86.87	202.61
Wien Nord (GR)	14.370	6.630	21.000	129.94	19.98	149.92
Mannheim (GB_ENG)	19.030	4.220	23.250	178.91	26.39	205.30
Overall Result	76.520	63.190	139.710	624.51	606.38	1,230.89

❖ Example

In this example the crosstab gets sorted ascending by the result row of OBC_TURN via the following script:

```
DS_1.sortMeasureByColumnOrRow({
  "0MEASURES000000000000000009": "0BC9TURN"
}, true);
```

The script above gives the following sorting result:

	OBC_REDU ▲	OBC_LOC	Utrecht (N)	Wien Nord (GR)	Halle/Leipzig (GR)	Spreitenbach (AG) (B)	Mannheim (GB_ENG)	...
OBC_ORDW	1% Rabatt	KG	6.800	14.370	4.780	15.840	19.030	...
	2% Rabatt	KG	20.310	6.630	16.890	7.000	4.220	...
	Overall Result	KG	27.110	21.000	21.670	22.840	23.250	...
OBC_TURN	1% Rabatt	EUR	36.86	129.94	40.62	115.74	178.91	...
	2% Rabatt	EUR	109.79	19.98	140.69	86.87	26.39	...
	Overall Result	EUR	146.65	149.92	181.31	202.61	205.30	...

34 Working with the Undo/Redo Functionality

You can undo or redo steps within an analysis application.

Context

Using scripting you can allow application users to undo or redo a number of steps in an analysis application. Steps refer to an actual state change in an application. Users can revert all changes in an analysis application. They can also go all the way forward to the last state change. For example, if a user changes the data source, such as filtering on `KeyFigure` or `Type`, these changes can be undone or redone within an application. You can also use the scripting API to reset the state of an application, so that the state in the application becomes the new starting point for calculating steps forwards or backwards.

Procedure

1. Create an analysis application.
2. Set ► [Application Properties](#) ► [Behavior](#) ► [Maximum Number of Steps](#) ► to an integer value greater than zero. This property is set to zero or disabled by default. So, if you set the integer value to 10, the user can undo 10 steps within their application. The recommended maximum value is 50. Once the property is set to a value greater than zero, the undo and redo feature is enabled.
3. Script a basic component, such as a [Button](#) using the `State.backOneStep()` ; or `State.forwardOneStep()` ; method. These scripting methods allows the user to undo or redo the last change, one step at a time, at each execution of the scripting method. The number of steps allowed, depends on the configuration of the [Maximum Number of Steps](#) application property.
4. Script another basic component, such as a [Button](#) using the `State.backToStart()` ; method. This scripting method allows the user to undo all changes in the application, back to the original state of the application.
5. Script another basic component, such as a [Button](#) using the `State.forwardToEnd()` ; method. This scripting method allows the user to go forward to the last state change in the current application.
6. Script another basic component using the `clearSteps()` ; method. This scripting method clears all steps in an application and sets the application state, after all steps are cleared, as the starting point. Any undo/redo steps executed from that point are calculated from this newly established starting point.

Results

You have used API methods to allow users to undo and redo changes made to the state of an analysis application.

i Note

Consider the following notes when working with the undo/redo functionality:

- Any state change to the application is a step forward. Therefore, the undo step brings the application user back to a neutral position before the state change was made. Call the `State.backOneStep()` ; method again, to undo the action previous to the state change.
- When background processing is enabled, it adds more state changes while an application loads. Therefore, you might encounter issues when using the undo/redo functionality with background processing.
- The `State.backToStart()` ; method brings you back to the state of the application after *On Startup* scripts and *On Variable Initialization* scripts have finished executing.
- You can only undo the same number of steps as you have created within an application. For example, if you go forward 5 steps in an application, you can only undo 5 steps. The only way to undo more steps is to use the back to start `State.backToStart()` ; method. The `State.backToStart()` ; method brings you back to the starting state of an application, provided the `clearSteps()` ; method is not executed.
- Some steps are not included in the undo/redo steps, for example, opening a dialog or using variable prompts.
- If you change the state of an application and undo that step, you return to the previous state of the application. If, however, you again change the state of the application and try to redo the step to bring you to the state the application was in before you reverted, the redo step does not work. By changing the state of the application after reverting back to it, you have created a new branch of the application that no longer references the other state.
- BW Integrated Planning write-back is not compatible with the undo/redo functionality described in this chapter.
- Only changes made to the application in memory, such as changing a *Chart* type, can be undone or reverted. Anything that persists to disk or platform, such as saving a bookmark, is not undone or reverted with this feature.
- If the *Maximum Number of Steps* application property is set to a value greater than 50, an information message occurs and the property value is set back to the last acceptable number.
- When working with a *Crosstab* component, you can use the context menu instead of scripting to go backwards and forwards within an application.
- The undo/redo functionality is supported on the BI platform and in document mode and local mode.

35 Working with the Feed List

Use the [Feed List](#) basic component to display a list of feed items.

With the [Feed List](#) component, you can enhance how feed items are displayed in an application. For example, you can use the [Feed List](#) with the [Comments](#) technical component. With the [Comments](#) technical component added to the application, the [Feed List](#) component offers a nice way to list comments in a Lumira document. Using scripting you can create a connection between the two components so that the comments are treated like feed items of the [Feed List](#).

Each feed item has an id. The feed item id is always required when combining a selected feed item with another entity. If you try to add a feed item without an id, the feed item will not be added to the [Feed List](#). You can apply selection events to the [Feed List](#), so that when an item is selected you can run a script.

If you do not want the default mapping of [Comments](#) properties to the [Feed List](#) parameters, but prefer more flexibility, you could use the following sample code:

Sample Code

Listing comments in a Feed List:

```
var comments = COMMENTS.getComments();
FEEDLIST_1.removeAllItems();
comments.forEach(function(element, index) {
    FEEDLIST_1.addItem({
        "author" : element.author,
        "icon" : element.icon,
        "authorId" : element.authorId,
        "id" : element.id,
        "info" : element.info,
        "lastModifiedTime" : element.lastModifiedTime,
        "text" : element.text
    });
});
```

Using the [Feed List](#) you can display useful information about each feed item.

Through scripting you can add items to the [Feed List](#) that have the following properties:

- text
- author
- author id
- icon
- info
- updateTime
- id

Multiple lines of text in each feed item is compressed into a structure that is easily displayed, using [...MORE](#) and [LESS](#) ellipsis buttons. You can set and change the author to be displayed against each list item in the [Feed List](#). Icons can be added to each feed list item, as well as a date and timestamp. Extra information can be added to provide some context to each item.

Related Information

[Feed List \[page 478\]](#)

[Comments \[page 73\]](#)

[Displaying Comments \[page 237\]](#)

36 Working with the Grouped List

Use the [Grouped List](#) component to add items to a list, and to group the items into categories.

You can customize exactly how you would like to add, and list items in the [Grouped List](#). Using the [Items](#) property, you can define for each item, the value and the text to be displayed. You can define the group each item belongs to, and the icon associated with each item. You can also define the type of items you want to list. For more information on defining items, you can refer to the chapter called [Grouped List](#).

Sample Use Case for the Grouped List

You can, for example, design an area in an application that is enabled for authoring by the application user at runtime. The application user can then drag and drop items from the [Grouped List](#) into this editable area. Depending on how you design the application, this offers the user the possibility of changing the layout of the application at runtime. To enable this functionality for the application user at runtime, you must do the following:

- Create an editable area in the application, using the [Authoring Area](#) property of the [Authoring](#) technical component.
- Set the [Drag and Drop for Authoring Enabled](#) property of the [Grouped List](#) to true.

For more information on using the [Authoring](#) technical component and the [Grouped List](#) together, you can refer to the chapter called [Using the Grouped List with the Authoring Technical Component](#).

Note

- Within the [Edit Grouped List Items](#) dialog of the [Grouped List](#), the option [Type](#) must have a value, if you wish to use the [Grouped List](#) with the [Authoring](#) technical component.
- The property of the [Grouped List](#) called [Drag and Drop for Authoring Enabled](#) can only be used with the [Authoring](#) technical component.

Related Information

[Grouped List \[page 481\]](#)

[Using the Grouped List with the Authoring Technical Component \[page 288\]](#)

[Adding Items to the Grouped List \[page 290\]](#)

[Component Types Supported in the Grouped List \[page 287\]](#)

36.1 Component Types Supported in the Grouped List

Certain component types can be selected when defining the items listed in a [Grouped List](#) component.

The types of components that can be selected in the [Grouped List](#) basic component are listed in the table below:

Component Types
Button
Checkbox
Checkboxgroup
Crosstab
DataSourceAlias
Datefield
Dimensionfilter
Dropdown
Filterbar
Filterpanel
FormattedTextView
GroupedList
Icon
Image
Inputfield
Link
Listbox
Navigationpanel
ProgressIndicator
PropertyEditor
Radiobuttongroup
SegmentedButton

Component Types

Shape

Spreadsheet

Switch

SwitchBar

Text

TextArea

ToggleButton

Tree

VisualTimer

VizFilterline

VizFrame

VizMap

Related Information

[Grouped List \[page 481\]](#)

[Working with the Grouped List \[page 286\]](#)

36.2 Using the Grouped List with the Authoring Technical Component

Create an area in an application enabled for authoring, and allow the application user to drag and drop items, and to change the layout of an application at runtime.

Context

You want to create an area in an application that is enabled for authoring. This allows the end user to be able to change the layout of the dashboard that you created for them. You want the user to be able to drag items from a [Grouped List](#), and drop these items into this area enabled for authoring. The application should contain a [Composite](#) component.

Note

Only a *Composite* component is supported as an area enabled for authoring.

Procedure

1. Using the *Technical Components* context menu in the *Outline* view, add the *Authoring* technical component to the application.
2. To select the area in the application enabled for authoring by the application user at runtime, use the *Authoring Area* property dropdown of the *Authoring* technical component. Select the *Composite* component you want to enable for authoring.
3. Alternatively, you can use the scripting method `setAuthoringArea()`.
4. Add a *Grouped List* basic component to the application. Make sure that the *Grouped List* is placed outside the assigned area enabled for authoring.
5. Set the *Grouped List* property called *Drag and Drop for Authoring Enabled* to true. (It is false by default.)
This allows the application user to drag items from the *Grouped List* and drop them at runtime, into the area enabled for authoring.
6. Select the ellipsis button beside the *Grouped List* property called *Items* to call the *Edit Grouped List Items* dialog.
7. When defining the value of the *Data Source Alias* type in the *Edit Grouped List Items* dialog, make sure that the data source alias exists in the *Composite* component, which is enabled for authoring.
8. Add as many items as needed to the *Edit Grouped List Items* dialog. For more details on adding items to the *Grouped List*, you can refer to the chapters called *Adding Items to the Grouped List* and *Component Types Supported in the Grouped List*.
9. Run the application.

The application user can drag items from the *Grouped List* onto the area enabled for authoring, to change the layout of the dashboard that you created for them. They can use basic context menu items to interact with the items that they add. For more information on the *Grouped List* context menu items, you can refer to the chapter called *Context Menu Items for Authoring* in the “End User Guide: SAP Lumira Designer” on the Help Portal at <https://help.sap.com>.

Related Information

[Component Types Supported in the Grouped List \[page 287\]](#)

[Adding Items to the Grouped List \[page 290\]](#)

36.3 Adding Items to the Grouped List

You want to add items to the *Grouped List*.

Context

You have added a *Grouped List* basic component to the application.

Procedure

1. Select the *Grouped List* component from within the application.
The *Grouped List* component properties appear.
2. Select the ellipsis button beside the *Grouped List* property called *Items*.
The *Edit Grouped List Items* dialog appears.
3. Select the *Insert* button to insert an item.
The row becomes editable.
4. Add an entry for the *Value* property. *Value* is a mandatory entry, and is the unique identifier for each item in the *Grouped List*.
5. Add an entry for the *Text* property. This is also a mandatory entry. If no text is entered, the value of the item is displayed.
6. Use the *Group* property to define a name for the category or group to which your item entry belongs. The *Group* property is also mandatory.
7. You can also add a icon for the item. The *Icon* property is optional. Select the ellipsis button beside the *Icon* property.
The *Open Image* dialog appears.
8. Select the icon you want to use from the existing folder, or by uploading a new image.
9. Select *Open*.
The *Icon* property is populated with the path to the icon location.
10. Select the *Type* property dropdown to select the type of component you want to add to the *Grouped List*.
For more information about what component types are supported, you can refer to the chapter called *Component Types Supported in the Grouped List*.
If you wish to use the *Grouped List* with the *Authoring* technical component, you must select a value for *Type*.
11. Select *OK*.

The *Grouped List* is populated with all the items entries you created.

Related Information

[Using the Grouped List with the Authoring Technical Component \[page 288\]](#)

[Component Types Supported in the Grouped List \[page 287\]](#)

[Grouped List \[page 481\]](#)

37 Working with Keyboard Shortcuts

Create your own set of keyboard shortcuts using the [Keyboard Shortcuts](#) technical component .

The [Keyboard Shortcuts](#) technical component allows you to create your own set of keyboard shortcuts for use in your applications. You can create your own set of key code combinations to execute scripting methods within your application. When working with the [Keyboard Shortcuts](#) technical component, is it important to take note of the following:

i Note

- Ensure that you do not define key code combinations that mirror standard key combinations associated with different internet browser types. If a browser uses the same keyboard shortcut as one created by you, it may happen that the browser keyboard shortcut takes precedence over the one you have created. To avoid any such issues, carry out browser based testing before adding customized keyboard shortcuts to an application.
- Duplicate key codes for different scriptings methods are not supported.
- The key code combinations must start with [CTRL](#) and can contain one or two other keys. Use only alphanumeric keys and the SHIFT and ALT keys.

37.1 Defining Keyboard Shortcuts

Use the [Keyboard Shortcuts](#) technical component to create key codes that execute scripting functionality within your application.

Context

You have added the [Keyboard Shortcuts](#) technical component to your application.

Procedure

1. Select the [Keyboard Shortcuts](#) technical component from the [Outline](#) view.
2. Select the ellipsis button beside the [Keyboard Shortcuts](#) property.

The [Edit Entries](#) dialog appears.

3. Select the [Description](#) field to create a name for your keyboard shortcut.
4. Select the ellipsis button within the [Key Code](#) field.

The [Key Code](#) dialog appears.

5. Add a key combination of your choice that will be used to call a specific scripting function.
6. Select *OK*.
7. Select the ellipsis button within the *Script* field.

The *Script Editor* dialog appears.

8. Select the script you want to execute when your chosen *Key Code* combination is selected on the user's keyboard.
9. Select *OK*.
10. Select the *Insert* button to add another row within the *Edit Entries* dialog.

An additional row is added, allowing you to add another key code combination.

11. Repeat steps 4 to 11 until you have created all the keyboard shortcuts you wish to have in your application.
12. To remove a row from the *Edit Entries* dialog, select the *Remove* button.

The row is removed.

13. To change the order in which the key codes appear in the *Edit Entries* dialog, select the *Move Up* and *Move Down* buttons.

Related Information

[Working with Keyboard Shortcuts \[page 292\]](#)

[Properties of the Keyboard Shortcuts Technical Component \[page 293\]](#)

37.2 Properties of the Keyboard Shortcuts Technical Component

Definition of *Keyboard Shortcuts* technical component properties.

General Properties of the Keyboard Shortcuts Technical Component

Property	Property Value	Property Description
keyboard shortcuts	none	Select the ellipsis button to call the <i>Edit Entries</i> dialog. This dialog allows you to create a list of key code combinations and associate them with scripting functionality specified by you. When the key combination is selected on the user's keyboard, the specified scripting functionality is executed.

Related Information

[Working with Keyboard Shortcuts \[page 292\]](#)

38 Working with Maps

Use a [Map](#) to display geo-specific data in a layered approach.

Context

To display different layers of geographical information on a map, you can add a [Map](#) component to your analysis application. Each layer displays different information when the user interacts with it. How information is displayed in the [Map](#) depends on how you configure the properties and how you apply scripting. You can connect a different data source to each layer.

Connected data sources must contain geo-specific dimensions or measures, for example, countries, longitude and latitude.

Procedure

1. From the [Chart Components](#) view, drag and drop the [Map](#) component into the layout editor.
2. From the [Outline](#) view, add into the [Data Sources](#) folder, the data sources you wish to work with in your [Map](#).
3. Before the user can interact with the layers of [Map](#), configure the [Edit Initial View...](#) for each data source to ensure that only one geo dimension appears in the [Rows](#). The measures can appear in the [Columns](#) and [Rows](#) areas.
4. Select the [Map](#) component in the layout panel. Select the ellipsis button in the [Map Configuration](#) property to call the [Configure Map](#) dialog. It is from the [Configure Map](#) dialog that you can configure all the [Map](#) properties. For more information on each of these properties, you can refer to the chapters in this guide called [Map](#) and [Properties of the Configure Map Dialog](#).
5. You can allow users to interact in multiple ways with the different [Map](#) layers by using the `onSelect()` ; scripting method. For more information on interacting with the [Map](#), you can refer to the chapter in this guide called [Interacting with Maps](#).

Related Information

[Map](#) [page 460]

[Map Settings](#) [page 296]

[Configuring Initial Map Settings](#) [page 296]

[Interacting with Maps](#) [page 309]

[Configuring Map Layers](#) [page 298]

[Properties of the Configure Map Dialog](#) [page 461]

38.1 Map Settings

Data sources containing geo-specific dimensions or measures are required for the [Map](#) component.

To create a well-designed [Map](#), you should use data sources containing geo-specific dimensions or measures. You also need to have a very good idea of how you would like to display data. You can choose to display data as choropleth (polygons, multi-polygons, lines, and multi-lines), bubble charts, marker and pie charts.

Other important factors that influence a [Map](#)'s appearance are the selected properties in the [Configure Map](#) dialog of the [Map Configuration](#) general property, and other [Map](#) general properties.

Related Information

[Working with Maps \[page 295\]](#)

[Properties of the Configure Map Dialog \[page 461\]](#)

[Map \[page 460\]](#)

38.2 Configuring Initial Map Settings

Configure initial view settings so that the [Map](#) displays data exactly as required.

Context

Use the [Edit Initial View...](#) to configure the initial settings for the [Map](#).

i 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 can appear in the [Columns](#) and [Rows](#) areas.
- For pie chart layers, you require two dimensions in the [Rows](#) area of the [Edit Initial View...](#). The first dimension is the geo-specific dimension and the second dimension is used to determine the pie slices.
- If the SAP BW attributes of the geo dimension contain longitude and latitude, then longitude and latitude are assigned to the map layer based on these BW attributes. Only use the SAP BW attributes to get longitude and latitude values. Do not use the SAP BW attributes elsewhere. Put the longitude and latitude values for the [Map](#) in a measure, if the geo dimension does not contain latitude and longitude BW attributes.

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, from the context menu of the data source, 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. Hide totals. To hide totals, select ► *Totals Display* ► *Hide Totals* ▾ from the context menu of the dimension.
6. The *Map Configuration* property calls up a *Configure Map* dialog, which allows you to configure all properties within the *Map* itself. For a description of all the *Map* properties within the *Configure Map* dialog, you can refer to the chapter called *Properties of the Configure Map Dialog*.
7. The *Show Scaling Factors* property is set to true by default. If you wish to hide the scaling factors on all your *Map* layers, you can set this property value to false.
8. The *Adjust Boundaries* property is set to true to automatically adjust the boundaries of the *Map*.

Related Information

[Working with Maps \[page 295\]](#)

38.3 Binding Data Sources to a Map

You can bind different data sources to different layers of a *Map*.

In the *Map* component, 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 *Map* tab of the *Configure Map* dialog. When you select your data source, ensure that in the *Edit Initial View...*, you have a geographic dimension in the *Rows* area.

The measures can appear in the *Columns* and *Rows* areas.

38.4 Working with GeoJSON

Use GeoJSON to encode 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. This file is then accessed from the *Custom GeoJSON File* property within the *Map* tab of the *Configure Map* dialog. 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 [Configure Map](#) dialog, 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 [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

[Properties of the Configure Map Dialog \[page 461\]](#)

38.5 Configuring Map Layers

Adding multiple layers to a [Map](#) offers many ways to configure how data is displayed.

Once you have configured the data sources, and the initial map settings, you can now begin to add layers to a [Map](#) component. This layered approach to adding data to your [Map](#) allows the application user to drill down into, and interact with the data visualization. Layers can be added by using the [Add Layer](#) button in the [Configure Map](#) dialog. Each layer has a unique ID. With the context menu icon in the [Map](#) tab of the [Configure Map](#) dialog, you can move up, move down, clear, and remove the different layers in your [Map](#). When you then run the application, the [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 choropleth (polygons, multi-polygons, lines, and multi-lines), bubble charts, marker and pie charts. Data sources are assigned to individual [Map](#) layers.

i 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 [Map](#), the scripting methods `getSelectedLayer()` ; and `getSelectedMember()` ; should be used together for optimal results.
- The basemap is created using basemap provider API. Data is rendered on top of the basemap using the Lumira Designer interface. User-specific data is not transferred to Esri or any other basemap provider.
- You can allow the end user to set the `measureColorPalette` and `dimensionColorPalette` by layer, using the scripting method `setLayerColorPalettes()` . The [Map](#) component then automatically chooses which color palette to use when the user changes the chart type.

Related Information

[Working with Maps \[page 295\]](#)

[Binding Data Sources to a Map \[page 297\]](#)

38.5.1 Configuring a Choropleth Layer

You can configure a shapes or choropleth layer in a [Map](#).

Context

A shape layer or choropleth layer depicts a collection of polygons, multi-polygons, lines or multi-lines on a map, for example a country, sales region etc. In the [Configure Map](#) dialog, the choropleth layer is the default layer type. To display choropleths in a [Map](#), a choropleth 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 Lumira Designer. Choropleths 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 [Map](#) component in the [Layout](#) pane. From the [Map Configuration](#) property select the [Configure Map](#) dialog.
2. Ensure that the [Map](#) tab is selected.
3. Select the [Add Layer](#) button if you have already added a layer and wish to add another one.
A new layer of properties is opened above the previous layer.
4. Select the choropleth layer type.
Map properties specific to the choropleth layer type appear.
5. Provide a unique ID for your choropleth layer in order to make it easy to identify when all layers are collapsed. To do this, double-click in the area of the default layer ID, and type your own ID.
6. Select the data source you wish to assign to this choropleth layer by selecting a data source from the [Datasets in Use](#) property.
7. Select the geo information relevant for the choropleth layer. You can select from GeoJSON or Longitude and Latitude.
8. If you are on the BI platform, you can select the ellipsis button within the [Custom GeoJSON File](#) property to select the location of your custom GeoJSON file. Selecting the [Custom GeoJSON File](#) ellipsis button in local mode, opens a [Select Local GeoJSON](#) dialog box, from where you can select your locally stored custom GeoJSON file.
9. Use the [GeoJSON Mapping Property](#) dropdown list to select the property in the GeoJSON file that will be used to display the shape. To assist you with your selection, a text value is available beside the first property within the GeoJSON file. Select the drop down arrow to view it.
10. You can specify how the dimension members will be mapped by selecting either key or text from the [GeoJSON Mapping Type](#) property.

11. If you select Longitude and Latitude, you can enter values for each in the [Longitude](#) and [Latitude](#) properties.
12. Select the + icon beside the [Add Dimensions](#) property to assign dimensions to the choropleth layer.
13. Select the + icon beside the [Add One Measure](#) property to assign a measure to the choropleth layer. Measures determine the color of the choropleth to be represented on the [Map](#).
14. Switch to the [Properties](#) tab. Within this tab, you can set properties based on the following areas: Geo Map, Layers, Data Labels, Legend, and Map Title.
15. Select the [Geo Map](#) property area from the dropdown menu.

Properties related to the map itself appear.
16. To set whether selection mode is exclusive or inclusive, select from the options is the [Selection Mode](#) property. If you want the selection to be cumulative, select Inclusive. Select Exclusive if you do not want the selection mode to be cumulative.
17. To set whether null values are visible on all layers, set the [Null Values Visible](#) property to true.
18. To select which type of map to be used, use the [Map Type](#) dropdown property. You can decide to use the Esri Online Map or your own Customized Online Map.
19. Select Customized Online Map.

Two more properties appear: [Base Map URL](#) and [Base Map Copyright](#).
20. Enter the URL for your preferred basemap. The parameters LOD, X and Y are expanded to provide a URL for each individual basemap. For example, `my .map/ {LOD} / {X} / {Y} .png`.
21. It might be required by the base map URL provider that you display copyright information at the bottom on the map. To provide this text, fill in the details in the free text field of the [Base Map Copyright](#) property.

The text displayed at the bottom right of the map.
22. Select the [Layers](#) property area from the dropdown menu.

Properties related to configuring the map layers appear.
23. Using the dropdown list, select the layer that you want to configure.
24. The [Show this Layer](#) checkbox is selected by default. If you want this layer to be invisible to the application user, you must deselect this checkbox.
25. 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 highest value will use the [End Color](#).

The choropleths in this layer will display on the [Map](#) in the color selected.
26. Set the [Classification Type](#) property. Select from the two options Quantile or Equal Interval. Selecting Quantile to distributes a set of values into groups that contain an equal number of values. Selecting Equal Interval arranges a set of values into groups that contain an equal range of values.
27. Select the [Data Labels](#) property area from the dropdown menu.

Properties related to configuring the data labels of the map appear.
28. Select the [Show Data Labels](#) checkbox to display data labels in the map.
29. Define how the data labels are formatted using the font and color properties.
30. Select the [Legend](#) property area from the dropdown menu.

Properties related to configuring the map legend appear.
31. To display both the legend and legend title in the map, select the checkboxes [Show Legend](#) and [Show Legend Title](#).

32. Define how the title of the legend and items in the legend are formatted using the font and color properties.
33. Select the [Map Title](#) property area from the dropdown menu.

Properties related to configuring the title of the map appear.

34. Select the [Show Map Title](#) checkbox.

Other map title properties appear.

35. Enter text for the title of the map in the free text field of the [Title](#) property.
36. Define how the text of the map title formatted using the font, color and alignment properties.
37. Select [OK](#)

You have configured the choropleth layer of the map.

Related Information

[Configuring Map Layers \[page 298\]](#)

38.5.2 Configuring a Bubble Chart Layer

You can configure a bubble chart layer in a [Map](#).

Context

In the context of the [Map](#) component, bubble chart layers are similar to marker layers, but with a measure assigned to them. The data source assigned to a bubble chart layer must contain longitude and latitude, or reference a GeoJSON file. If longitude and latitude are present, the bubble chart is displayed at these points or coordinates. If a GeoJSON file is referenced, the bubble chart is positioned in the middle of the shape. The bubble size changes, depending on the measures. Bubble chart layer types display a legend.

i 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 can appear in the [Columns](#) and [Rows](#) areas.

Procedure

1. Select the [Map](#) component in the [Layout](#) pane. From the [Map Configuration](#) property select the [Configure Map](#) dialog.
2. Ensure that the [Map](#) tab is selected.
3. Select the [Add Layer](#) button if you have already added a layer and wish to add another one.

A new layer of properties is opened above the previous layer.

4. Select the bubble layer type.

Map properties specific to the bubble layer type appear.

5. Provide a unique ID for your bubble layer in order to make it easy to identify when all layers are collapsed. To do this, double-click in the area of the default layer ID, and type your own ID.
6. Select the data source you wish to assign to this bubble layer by selecting a data source from the [Datasets in Use](#) property.
7. Select the geo information relevant for the bubble layer. You can select from GeoJSON or Longitude and Latitude.
8. If you are on the BI platform, you can select the ellipsis button within the [Custom GeoJSON File](#) property to select the location of your custom GeoJSON file. Selecting the [Custom GeoJSON File](#) ellipsis button in local mode, opens a [Select Local GeoJSON](#) dialog box, from where you can select your locally stored custom GeoJSON file.
9. Use the [GeoJSON Mapping Property](#) dropdown list to select the property in the GeoJSON file that will be used to display the bubble. To assist you with your selection, a text value is available beside the first property within the GeoJSON file. Select the drop down arrow to view it.
10. You can specify how the dimension members will be mapped by selecting either key or text from the [GeoJSON Mapping Type](#) property.
11. If you select Longitude and Latitude, you can enter values for each in the [Longitude](#) and [Latitude](#) properties.
12. Select the + icon beside the [Add Dimensions](#) property to assign geo dimensions to the bubble layer.
13. Select the + icon beside the [Add One Measure](#) property to assign a measure. The [Size](#) of the bubbles is determined by this measure.
14. Select the + icon beside the [Add One Dimension](#) property to assign a dimension. The [Color](#) of the bubbles is determined by this dimension.
15. Select the + icon beside the [Add One Dimension](#) property to assign a dimension. The [Animation](#) of the bubbles is determined by this dimension.
16. Switch to the [Properties](#) tab. Within this tab, you can set properties based on the following areas: Geo Map, Layers, Data Labels, Legend, and Map Title.
17. Select the [Geo Map](#) property area from the dropdown menu.

Properties related to the map itself appear.

18. To set whether selection mode is exclusive or inclusive, select from the options is the [Selection Mode](#) property. If you want the selection to be cumulative, select Inclusive. Select Exclusive if you do not want the selection mode to be cumulative.
19. To set whether null values are visible on all layers, set the [Null Values Visible](#) property to true.
20. To select which type of map to be used, use the [Map Type](#) dropdown property. You can decide to use the Esri Online Map or your own Customized Online Map.
21. Select Customized Online Map.

Two more properties appear: [Base Map URL](#) and [Base Map Copyright](#).

22. Enter the URL for your preferred basemap. The parameters LOD, X and Y are expanded to provide a URL for each individual basemap. For example, `my.map/{LOD}/{X}/{Y}.png`.
23. It might be required by the base map URL provider that you display copyright information at the bottom on the map. To provide this text, fill in the details in the free text field of the [Base Map Copyright](#) property.

The text displayed at the bottom right of the map.

24. Select the [Layers](#) property area from the dropdown menu.

Properties related to configuring the map layers appear.

25. Using the dropdown list, select the layer that you want to configure.
26. The [Show this Layer](#) checkbox is selected by default. If you want this layer to be invisible to the application user, you must deselect this checkbox.
27. If you want to cluster bubbles that are close to each other on the map, you can select the checkbox [Cluster Adjacent Locations](#).
28. Select bubble color palettes from the list of options in the [Color Palette](#) property.
29. Select the [Data Labels](#) property area from the dropdown menu.

Properties related to configuring the data labels of the map appear.

30. Select the [Show Data Labels](#) checkbox to display data labels in the map.
31. Define how the data labels are formatted using the font and color properties.
32. Select the [Legend](#) property area from the dropdown menu.

Properties related to configuring the map legend appear.

33. To display both the legend and legend title in the map, select the checkboxes [Show Legend](#) and [Show Legend Title](#).
34. Define how the title of the legend and items in the legend are formatted using the font and color properties.
35. Select the [Map Title](#) property area from the dropdown menu.

Properties related to configuring the title of the map appear.

36. Select the [Show Map Title](#) checkbox.

Other map title properties appear.

37. Enter text for the title of the map in the free text field of the [Title](#) property.
38. Define how the text of the map title formatted using the font, color and alignment properties.
39. Select [OK](#)

You have configured the bubble layer of the map.

Related Information

[Configuring Map Layers \[page 298\]](#)

[Configuring Initial Map Settings \[page 296\]](#)

38.5.3 Configuring a Marker Layer

You can configure a marker layer in a [Map](#).

Context

You can display data as precise locations on a [Map](#) marker 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 [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 marker layer in your [Map](#):

- You can assign data to a marker layer from the [Map](#) properties [Latitude](#) and [Longitude](#). These properties display dimension members as measures, allowing you to select how to represent longitude and latitude as custom or default markers on a map.
- You can also assign data to a marker layer from the SAP BW attributes in the [Edit Initial View...](#)

Procedure

1. Select the [Map](#) component in the [Layout](#) pane. From the [Map Configuration](#) property select the [Configure Map](#) dialog.
2. Ensure that the [Map](#) tab is selected.
3. Select the [Add Layer](#) button if you have already added a layer and wish to add another one.
A new layer of properties is opened above the previous layer.
4. Select the marker layer type.
Map properties specific to the marker layer type appear.
5. Provide a unique ID for your marker layer in order to make it easy to identify when all layers are collapsed. To do this, double-click in the area of the default layer ID, and type your own ID.
6. Select the data source you wish to assign to this choropleth layer by selecting a data source from the [Datasets in Use](#) property.
7. Select the geo information relevant for the choropleth layer. You can select from GeoJSON or Longitude and Latitude.
8. If you are on the BI platform, you can select the ellipsis button within the [Custom GeoJSON File](#) property to select the location of your custom GeoJSON file. Selecting the [Custom GeoJSON File](#) ellipsis button in local mode, opens a [Select Local GeoJSON](#) dialog box, from where you can select your locally stored custom GeoJSON file.
9. Use the [GeoJSON Mapping Property](#) dropdown list to select the property in the GeoJSON file that will be used to display the shape. To assist you with your selection, a text value is available beside the first property within the GeoJSON file. Select the drop down arrow to view it.
10. You can specify how the dimension members will be mapped by selecting either key or text from the [GeoJSON Mapping Type](#) property.

11. If you select Longitude and Latitude, you can enter values for each in the [Longitude](#) and [Latitude](#) properties.
12. Select the [+](#) icon beside the [Add Dimensions](#) property to assign dimensions to the marker layer.
13. Switch to the [Properties](#) tab. Within this tab, you can set properties based on the following areas: Geo Map, Layers, Data Labels, Legend, and Map Title.
14. Select the [Geo Map](#) property area from the dropdown menu.

Properties related to the map itself appear.

15. To set whether selection mode is exclusive or inclusive, select from the options is the [Selection Mode](#) property. If you want the selection to be cumulative, select Inclusive. Select Exclusive if you do not want the selection mode to be cumulative.
16. To set whether null values are visible on all layers, set the [Null Values Visible](#) proeprty to true.
17. To select which type of map to be used, use the [Map Type](#) dropdown property. You can decide to use the Esri Online Map or your own Customized Online Map.
18. Select Customized Online Map.

Two more properties appear: [Base Map URL](#) and [Base Map Copyright](#).

19. Enter the URL for your preferred basemap. The parameters LOD, X and Y are expanded to provide a URL for each individual basemap. For example, [my.map/{LOD}/{X}/{Y}.png](#).
20. It might be required by the base map URL provider that you display copyright information at the bottom on the map. To provide this text, fill in the details in the free text field of the [Base Map Copyright](#) property.

The text displayed at the bottom right of the map.

21. Select the [Layers](#) property area from the dropdown menu.

Properties related to configuring the map layers appear.

22. Using the dropdown list, select the layer that you want to configure.
23. The [Show this Layer](#) checkbox is selected by default. If you want this layer to be invisible to the application user, you must deselect this checkbox.
24. Use the [Marker Color](#) dropdown to customize the color of the marker that indicates the point on the [Map](#).
25. You can upload to the platform any pixel based image file to replace the standard [Map](#) marker, however it is recommended to use PNG files. You can have a different marker for each points layer. Select the icon beside the [Custom Marker](#) to select custom markers that are stored on the platform. Selecting the [Custom Marker](#) icon in local mode, opens a [Select Local Image](#) dialog box, from where you can select your locally stored custom marker.

The marker and corresponding legend changes to display the customized marker you have selected.

26. To select an anchor point for the custom marker, select from the options provided in the [Vertical Anchor](#) dropdown list.
27. If you want to cluster points on the map that are close to each other, you can select the checkbox [Cluster Adjacent Locations](#).
28. Select the [Data Labels](#) property area from the dropdown menu.

Properties related to configuring the data labels of the map appear.

29. Select the [Show Data Labels](#) checkbox to display data labels in the map.
30. Define how the data labels are formatted using the font and color properties.
31. Select the [Legend](#) property area from the dropdown menu.

Properties related to configuring the map legend appear.

32. To display both the legend and legend title in the map, select the checkboxes [Show Legend](#) and [Show Legend Title](#).
33. Define how the title of the legend and items in the legend are formatted using the font and color properties.
34. Select the [Map Title](#) property area from the dropdown menu.

Properties related to configuring the title of the map appear.
35. Select the [Show Map Title](#) checkbox.

Other map title properties appear.
36. Enter text for the title of the map in the free text field of the [Title](#) property.
37. Define how the text of the map title formatted using the font, color and alignment properties.
38. Select [OK](#)

You have configured the marker layer of the map.

Related Information

[Configuring Map Layers \[page 298\]](#)

38.5.4 Configuring a Pie Chart Layer

You can configure a pie chart layer in a [Map](#).

Context

In the context of the [Map](#) component, pie chart layers are similar to points layers, but with a measure assigned to them. The data source assigned to a pie chart layer must contain longitude and latitude or reference a GeoJSON file. If longitude and latitude are present, the pie chart is displayed at these points or coordinates. If a GeoJSON file is referenced, the pie chart appears in the middle of the shape. The pie slices change, depending on the measures. The pie chart layer types display a legend.

i 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 can appear in the [Columns](#) and [Rows](#) areas.
- For pie chart layers, you require two dimensions in the [Rows](#) area of the [Edit Initial View...](#). The first dimension listed is the geo-specific dimension and the second dimension is used to determine how the pie is sliced.

Procedure

1. Select the [Map](#) component in the [Layout](#) pane. From the [Map Configuration](#) property select the [Configure Map](#) dialog.
2. Ensure that the [Map](#) tab is selected.
3. Select the [Add Layer](#) button if you have already added a layer and wish to add another one.
A new layer of properties is opened above the previous layer.
4. Select the pie layer type.
Map properties specific to the pie layer type appear.
5. Provide a unique ID for your pie layer in order to make it easy to identify when all layers are collapsed. To do this, double-click in the area of the default layer ID, and type your own ID.
6. Select the data source you wish to assign to this pie layer by selecting a data source from the [Datasets in Use](#) property.
7. Select the geo information relevant for the pie layer. You can select from GeoJSON or Longitude and Latitude.
8. If you are on the BI platform, you can select the ellipsis button within the [Custom GeoJSON File](#) property to select the location of your custom GeoJSON file. Selecting the [Custom GeoJSON File](#) ellipsis button in local mode, opens a [Select Local GeoJSON](#) dialog box, from where you can select your locally stored custom GeoJSON file.
9. Use the [GeoJSON Mapping Property](#) dropdown list to select the property in the GeoJSON file that will be used to display the pie. To assist you with your selection, a text value is available beside the first property within the GeoJSON file. Select the drop down arrow to view it.
10. You can specify how the dimension members will be mapped by selecting either key or text from the [GeoJSON Mapping Type](#) property.
11. If you select Longitude and Latitude, you can enter values for each in the [Longitude](#) and [Latitude](#) properties.
12. Select the [+](#) icon beside the [Add Dimensions](#) property to assign geo dimensions to the pie layer.
13. Select the [+](#) icon beside the [Add One Measure](#) property to assign a measure. The [Size](#) of the pie slices is determined by this measure.
14. Select the [+](#) icon beside the [Add One Dimension](#) property to assign a dimension. The [Color](#) of the pie slices is determined by this dimension.
15. Switch to the [Properties](#) tab. Within this tab, you can set properties based on the following areas: Geo Map, Layers, Data Labels, Legend, and Map Title.
16. Select the [Geo Map](#) property area from the dropdown menu.
Properties related to the map itself appear.
17. To set whether selection mode is exclusive or inclusive, select from the options is the [Selection Mode](#) property. If you want the selection to be cumulative, select Inclusive. Select Exclusive if you do not want the selection mode to be cumulative.
18. To set whether null values are visible on all layers, set the [Null Values Visible](#) property to true.
19. To select which type of map to be used, use the [Map Type](#) dropdown property. You can decide to use the Esri Online Map or your own Customized Online Map.
20. Select Customized Online Map.

Two more properties appear: [Base Map URL](#) and [Base Map Copyright](#).

21. Enter the URL for your preferred basemap. The parameters LOD, X and Y are expanded to provide a URL for each individual basemap. For example, `my.map/{LOD}/{X}/{Y}.png`.
22. It might be required by the base map URL provider that you display copyright information at the bottom on the map. To provide this text, fill in the details in the free text field of the *Base Map Copyright* property.

The text displayed at the bottom right of the map.

23. Select the *Layers* property area from the dropdown menu.

Properties related to configuring the map layers appear.

24. Using the dropdown list, select the layer that you want to configure.
25. The *Show this Layer* checkbox is selected by default. If you want this layer to be invisible to the application user, you must deselect this checkbox.
26. If you want to display a doughnut chart, select the checkbox called *Show as Doughnut Chart*.
27. Select pie color palettes from the list of options in the *Color Palette* property.
28. Select the *Data Labels* property area from the dropdown menu.

Properties related to configuring the data labels of the map appear.

29. Select the *Show Data Labels* checkbox to display data labels in the map.
30. Define how the data labels are formatted using the font and color properties.
31. Select the *Legend* property area from the dropdown menu.

Properties related to configuring the map legend appear.

32. To display both the legend and legend title in the map, select the checkboxes *Show Legend* and *Show Legend Title*.
33. Define how the title of the legend and items in the legend are formatted using the font and color properties.
34. Select the *Map Title* property area from the dropdown menu.

Properties related to configuring the title of the map appear.

35. Select the *Show Map Title* checkbox.

Other map title properties appear.

36. Enter text for the title of the map in the free text field of the *Title* property.
37. Define how the text of the map title formatted using the font, color and alignment properties.
38. Select *OK*

You have configured the pie layer of the map.

Related Information

[Configuring Map Layers \[page 298\]](#)

[Configuring Initial Map Settings \[page 296\]](#)

38.6 Interacting with Maps

There are a number of ways that you can allow users to interact with the [Map](#) component.

Through scripting and working with the [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 [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 [On Select](#) 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 [Map](#) to allow the user to select a different basemap while running the application.

Center the map

You can script the [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.

Center the map by all layers

You can script the [Map](#) to change the center of the map around all the layers. This repositions the map so that all layers are visible.

Multiple selection

You can select multiple sets of data points on 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.



Related Information

[Working with Maps \[page 295\]](#)

39 Working with Real-Time Dashboards

Connect to a streaming data source to implement real-time functionality.



You can 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). You can then create and run your own complex event processing (CEP) applications to derive continuous intelligence from streaming event data in real time. There are many applications for real-time functionality, including in the area of operational systems. In operational systems, data may only be relevant as it occurs. It may not be necessary to persist the data, for example utility companies monitoring smart buildings and the financial capital markets.

The real-time functionality is implemented through a connection to a streaming data source. Examples of streaming data sources are SAP HANA Smart Data Streaming (HANA Streaming or SDS), or SAP ESP (Event Stream Processor). Within the *Outline* view in the designer, you can select from the context menu  [Add Custom Data Source](#) . The streaming data can then be connected to any OOTB *Chart* or *Map*, to visualize the data.

You can also use the *Timer* technical component to poll a data source at regular intervals to update a chart. You can apply this technique in cases where you do not need real-time streaming using WebSockets to push to the browser.

Note

- At least one dimension must be one of the following types when connecting your *Streaming Data Source* to a *Time Series* chart type:
 - bigdatetime
 - seconddate
 - msdate
 - timeOnly dimensions of this type are allowed to be used in the *Time Dimension* feed.
- Before working with real-time feature, refer to the following SAP Note:

SAP Note Number	Description
2460525 	WebSocket support required when using Streaming Data Source in SAP Lumira Designer.
2526738 	Some chart types cannot render when using Streaming Data Source

Related Information

[How to Create Real-Time Dashboards with Streaming Data Sources \[page 312\]](#)

[Streaming Data Source Additional Properties \[page 316\]](#)

[Working with the Outline View \[page 69\]](#)

39.1 How to Create Real-Time Dashboards with Streaming Data Sources

You can connect charts to create visualizations of streaming data.

Context

A streaming data source such as SAP HANA SDS (Smart Data Streaming) or SAP ESP (Event Stream Processor) can connect to Lumira Designer charts to create visualizations of streams of real-time data. You need to have an instance within SAP ESP 5.1 (SP09 or above) or SAP HANA SDS and have a project running.

i Note

- The authentication for SAP HANA SDS is done by the HANA server. Therefore, Lumira Designer does not support SAP BI platform authentication with SAP HANA SDS.
- 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 Lumira Designer go to ► [Outline View](#) ► [Data Sources](#) ►.
2. In the context menu, select the option [Add Custom Data Source](#).
3. Select [Streaming Data Source](#).
4. In the [Properties](#) view, select [Additional Properties](#) for the [Streaming Data Source](#).
Three [Additional Properties](#) tabs appear. The first tab is called [General Configuration](#). This tab 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 or SAP HANA SDS project is running.
6. Enter your user SAP ESP or SAP HANA SDS password and username credentials. Use your ESP Studio or HANA 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.

i 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 or SAP HANA SDS. 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 or SAP HANA SDS 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. **i Note**

Measures must be in the columns area when connecting your [Streaming Data Source](#) to a [Map](#).

Select your measures, dimensions, rows and columns to configure your query for the [Streaming Data Source](#) initial view.

15. Select a [Chart](#) or [Map](#) component and add to your canvas.
16. Assign the [Streaming Data Source](#) to your selected component.
17. If you wish to view the data at design time, select the [Show 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 any time to get a modified view of your data.

Related Information

[Streaming Data Source Additional Properties \[page 316\]](#)

[Working with Real-Time Dashboards \[page 311\]](#)

39.2 How to Create Real-Time Dashboards with a Single Pull-Based Data Source

Set an application to update by the defined interval.

Context

You wish to have your dashboards update at a set interval rather than every time there is a new event. You can create almost real-time visualizations when connected to SAP HANA or SAP BW (pull based) data sources by using the [Timer](#) technical component in conjunction with an OOTB [Chart](#) or [Map](#) within Lumira Designer. The

[Timer](#) technical 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.

The [Timer](#) component is activated through the optional [Technical Component](#) called [Timer](#). It must be added in the [Outline](#) view of the designer application before this functionality is available.

To reload a single data source, follow the steps below:

Procedure

1. Add a [Chart](#) or [Map](#) component onto your canvas and assign a data source. The data source can be either SAP BW or SAP HANA.
2. Add the [Timer](#) component, which is located in the [Outline](#) view under ► [Technical Components](#) ► [Create Child](#) ► [Timer](#) .
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 by the defined number of seconds you set for the [Interval in Milliseconds](#) component property.

Related Information

[Assigning a Data Source to a Component \[page 144\]](#)

[Working with the Outline View \[page 69\]](#)

[Working with Real-Time Dashboards \[page 311\]](#)

39.3 How to Create Real-Time Dashboards with Multiple Pull-Based Data Sources

Set multiple data sources with multiple chart types in a dashboard to update at regular intervals.

Context

It may not be necessary to load all the data at regular intervals as the data sources might not all contain new data. To increase performance you can implement a separate data source that executes quickly and

determines if there is new data across the multiple data sources. 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 or two [Map](#) components onto 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](#) view. You can set the [Visible](#) property of this component to **false**.
4. Add the [Timer](#) technical component, from within the [Outline](#) view.
5. Within the [Timer](#) properties, change the [Interval in Milliseconds](#) property to your chosen interval value.
6. Reloading a slower performing data source may impede user interaction. To alleviate this, move the reloading to the background processing. To move the reloading to the background processing, open the [Script Editor](#) of the [On Timer](#) property within the [Timer](#) component 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

```
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 \[page 313\]](#)

[Working with Real-Time Dashboards \[page 311\]](#)

[Working with the Outline View \[page 69\]](#)

39.4 Streaming Data Source Additional Properties

The *Streaming Data Source* additional properties are described in the table below.

Property Name	Property Value	Property Description
configuration	<ul style="list-style-type: none">hostport	Host and REST port on which your project is running.
security	<ul style="list-style-type: none">HTTPS	The default value is <i>True</i>
credentials	<ul style="list-style-type: none">usernamepassword<i>Connect</i> button	<ul style="list-style-type: none">ESP Studio or HANA Studio username and password.Connects to SAP ESP or SAP HANA SDS instance.
data source	<ul style="list-style-type: none">workspaceprojecttopicretention policyupdate interval (ms)	<ul style="list-style-type: none">ESP or HANA SDS projects are stored within the workspace.Projects within a workspace.Retention Policy: In SAP ESP or SAP HANA SDS, a topic is either a stream or a window. If the topic you select is a stream, a <i>Text</i> field will appear. You will be required to specify the <i>Retention Policy</i> for a stream. This option will allow you to specify the maximum number of rows that the <i>Chart</i> or <i>Info Chart</i> 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 <i>Retention Policy</i> text field will not appear as SAP ESP or SAP

Property Name	Property Value	Property Description
		<p>HANA SDS handles the retention policy of windows.</p> <ul style="list-style-type: none"> Update Interval (ms): The interval for displaying new events in milliseconds. By default the display updates with new data every second.
data types	a list of data types	<p>Currently SAP ESP and SAP HANA SDS have no concept of a measure or a dimension. As a result of this, some default rules have been applied to decide if a column is a measure or dimension. These rules are:</p> <ul style="list-style-type: none"> If a Column is Primary Key in SAP ESP or SAP HANA SDS, it is a dimension. If a column is a number in SAP ESP or SAP HANA SDS, it is a measure. Any remaining columns are dimensions. <div> i Note You will also be able to configure some columns. </div>
measures	primary values	<p>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.</p>
dimensions	<ul style="list-style-type: none"> rows columns 	<p>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. Select the Rows or Columns option to decide where to position the dimensions.</p>
measures	<ul style="list-style-type: none"> rows columns 	<p>Select the Rows or Columns option to decide where to position the measures.</p>
live preview	checkbox	<p>To preview the live stream you must select the Show Live Preview checkbox. Until you select Show Live Preview, any charts connected to the data source will show no data available. Show Live Preview will not persist and if connection is lost or the chart is closed</p>

Property Name	Property Value	Property Description
		and reopened it must be selected again.
		<div> i Note Show Live Preview is unchecked by default. </div>

Related Information

[Working with Real-Time Dashboards \[page 311\]](#)

[How to Create Real-Time Dashboards with Streaming Data Sources \[page 312\]](#)

40 Working with Scorecards

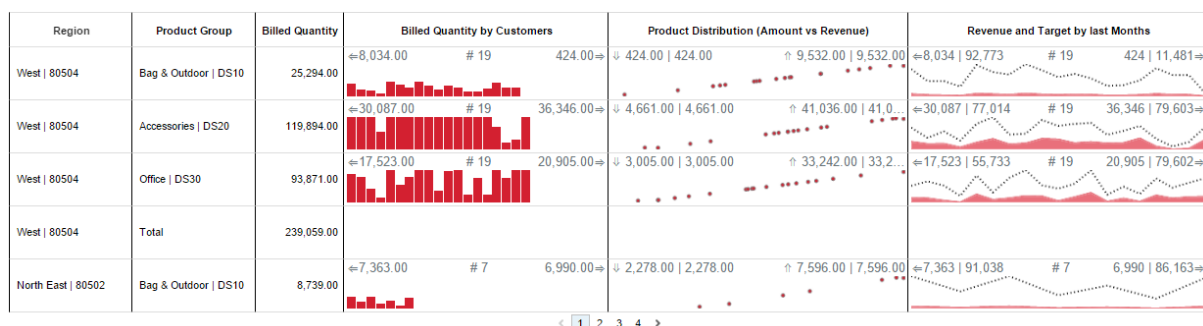
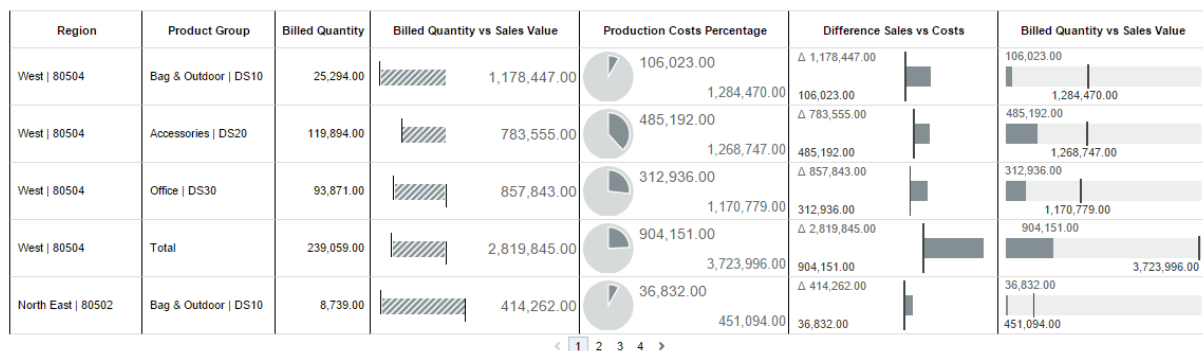
The Scorecard component is used for creating dashboards and reports with controlled visualization content and restricted navigation possibilities. Scorecards can help you to make complex result sets browsable by using user friendly visualizations (like charts) and highly customizable layouting properties.

It is not recommended to use the Scorecard component in applications with dynamic data sources and data sources with free multidimensional navigation (slice and dice). The predefined data selections in the Scorecard can turn invalid in such scenarios.

Using the Scorecard

You can add a Scorecard component to an analysis application to allow users to view the data as a highly customizable table as well as micro charts in columns. Depending on your configuration of the Scorecard column properties, each column can display information in the form of a text, image or charts. Each column property can be connected to a selection of the assigned data source.

The following pictures show examples of applications using scorecards. This first one shows a scorecard with several comparison charts and the second trend charts :



1. Drag and drop the [Scorecard](#) component from the [Components](#) view into the layout editor.
2. Add a data source to your application (in the [Outline](#) view right-click [Data Sources](#) and choose [Add Data Source...](#)) and assign it to the Scorecard.

3. (Optional) You can configure the Data Source by using the [Edit Initial View...](#) function (right-click the data source in the [Data Source](#) folder and choose [Edit Initial View...](#)). For best fit, place the dimensions in rows and measures in columns.
4. Select the [Scorecard](#) component in the [Layout](#) folder of the [Outline](#) view and configure the Scorecard properties in the [Properties](#) and [Additional Properties](#) views.
5. (Optional) You can allow users to interact in multiple ways with the Scorecard component by using the `onSelect()` ; scripting method. For more information on interacting with the Scorecard, see [Interacting with Scorecards \[page 327\]](#).

40.1 Scorecard Configuration

To create a well-designed Scorecard, you must use a data source containing dimensions in rows. These dimensions are made available for the row scope definition. For the scorecard you have to place the measures in columns. The scorecard does not display any content when measures are placed in rows. You can nevertheless add more dimensions into columns (in addition to the measures).

As soon the data source drilldown is defined, you can start configuring the scorecard.

To configure a scorecard, follow the steps in sequential order.

1. Define the row scope
2. Generate the initial scorecard

Caution

Once you have started the configuration of the scorecard, it is not recommended to make any changes in the data source definition (for example, removing or adding dimensions). For this reason, it is recommended to use a separate data source which is assigned to the scorecard component or to assure that there is no runtime functionality for this data source which can influence the drill down state. Filtering on the dimensions is allowed and will not invalidate the scorecard component.

Related Information

[Defining the Row Scope \[page 320\]](#)

[Generating the Initial Scorecard \[page 321\]](#)

40.1.1 Defining the Row Scope

In the [Additional Properties](#) sheet of the scorecard, choose a dimension of your data source out from the dropdown box for the row scope definition. You can select the first, second or any other dimension from the rows. The chosen row in the scorecard has the scope of the selection. Every row will become a single member selection of the chosen dimension and all dimensions which are placed higher in the rows. The other

dimensions which are not contained in the row scope definition are free for any use in the columns (for example, for charts).

❖ Example

The data source assigned to the scorecard has four dimensions in the rows (in the sequential order): “Customer”, “Region”, “Product Group” and “Time”.

You have selected as row scope the dimension “Customer”. The system assigns every row to a concrete customer as the member of the dimension Customer (for example, “SAP SE”). You can then create charts in other columns for “Region”, “Product Group” and “Time Dimensions”.

You have selected as row scope the dimension “Region”. The system assigns every row to a concrete customer and region as combination of the members of the dimensions “Customer” and “Region” (for example, “SAP SE | EUROPE” or “SAP SE | ASIA”). You can then create charts in other columns for product group and time dimensions. As the dimension “Region” is already in the row scope, it cannot be used for any other charts. You can assign a text to the members of the region dimension as text content.

40.1.2 Generating the Initial Scorecard

The scorecard component is a highly customizable component. It does not reflect the data source result set directly like the crosstab component does. It allows you to specify every column of the scorecard separately and decide which result set selection should be used for any visualization in the concrete column.

The configuration of columns can be done manually by using the [Add New Column](#) function or you can generate an initial scorecard for a quicker startup by clicking on the respective button.

When you use the manual creation, you can define column by column of the scorecard.

When you use the automatic generation of an initial scorecard, the following columns are generated by the system based on the result set:

- for every dimension included in the rows scope – one column with dimension members
- for the first measure – one dimension displaying the measure sum of the row scope definition
- for the first dimension which is outside of the row scope – a column with trend chart displaying the measure members for that dimension as line or column chart.
- if the data source has more than one dimension – a column with comparison chart displaying the delta between the first and second measure

i Note

Keep in mind that the automatic generation has only a “show case” character to speed up the initial creation. It does not analyze in full extend any specialties of the data source – for example, it is assuming that all dimensions have result members activated. You have to check all selections and properties of the generated columns and assure that the selected data is fulfilling your requirements.

40.2 Basic Scorecard Concepts

The Scorecard consists of the following areas:

- **Group Headers**
Group Headers allow you to group columns in custom groups, which can be used for special descriptions.
- **Columns**
Columns allow you to visualize the result set, based on chosen selections and visualizations.

A column is defined by:

- the column header. The header allows you to place a description of the column.
- the column content. The column content allows you to visualize the data per row, based on the row scope definition and the chosen visualization.

While configuring the columns, you can

- add a new column after the last existing column
- insert a new column before the selected one
- delete existing columns
- copy existing column (copy is inserted to the right of (in case of RTL to the left of) the existing column)
- move a column up and down (the group assignment is adjusted automatically)

When configuring group headers, you can:

- group two or more group headers together
- ungroup a header, which was grouped before

40.3 Configuring Initial Scorecard Settings

Once you have generated an initial scorecard, you can start to configure the scorecard settings in detail. To configure the scorecard, you have to set the properties in the [Properties](#) and the [Additional Properties](#) sheet.

The properties in the properties sheet affect global settings like:

- heights for group headers, headers and rows
- interaction behavior
- navigation behavior

The properties in the additional properties sheet are used for the column definition and customizing.

40.3.1 Configuring Scorecard Columns

To configure a scorecard column, select a column in the [Columns and Group Headers](#) area in the Additional Properties sheet and click on the [Column Properties](#) tab.

Configuring the scorecard width

A column in the scorecard component has a [Width](#) property in pixels and can be configured with [Automatic Column Size](#). All columns marked as [Auto Size](#) are resized accordingly (the column width is increased or decreased to fit the available free space and the column width ratio is retained). A column can also be hidden (property [Column Hidden](#)). To make a column visible, use the corresponding scripting method.

Using the property Column Content Appends to Previous Column

By using this property, you can change the visualization of a column and display the content as part of the previous column, appended vertically below the content of the previous column but in the same column. This allows you to define a vertical layout in one column (the master column is always the previous column). Using this function, you can create a text with a chart below it. The appended content is defined in the current column, but will take the width property of the previous column. This property does not affect the first column as there is no previous column for it.

Using the property Column Content Height Ratio

By using this property, you can change the height ratio for all columns that are selected by the property [Column Content Appends to Previous Column](#). The default value is [1](#). For example, if your row height is defined for [120](#) pixels, you have one appended column and the property is set to [2](#) for the master column, and [1](#) for the appended column. The master column will take $2/(2+1) = 2/3 = 80$ pixels. The appended column will take $1/(2+1) = 1/3 = 40$ pixels.

Example: In the following example, the property [Column Content Appends to Previous Column](#) is not set active (standard columns definitions):

Product Group	Billed Quantity	Billed Quantity for Region		Billed Quantity vs Sales Value	
		⇐7.070.593,00	# 4	3.236.162,00⇒	
Bag & Outdoor DS10	23.556.976,00				23.556.976,00
					267.644.177,00
					291.201.153,00
		⇐38.948.284,00	# 4	17.431.338,00⇒	
Accessories DS20	126.873.813,00				126.873.813,00
					207.216.219,00
					334.090.032,00

In this second example, the property [Column Content Appends to Previous Column](#) has been set for the column [Billed Quantity](#): The content defined in column [Billed Quantity](#) appears appended vertically in the previous column [Product Group](#). As the property [Column Content Height Ratio](#) is set to [1](#) in both columns, the height distribution is 1:1.

Product Group	Billed Quantity for Region		Billed Quantity vs Sales Value	
	⇐7.070.593,00	# 4	3.236.162,00⇒	
Bag & Outdoor DS10	23.556.976,00			23.556.976,00
				267.644.177,00
				291.201.153,00
	⇐38.948.284,00	# 4	17.431.338,00⇒	
Accessories DS20	126.873.813,00			126.873.813,00
				207.216.219,00
				334.090.032,00

Using the property Cell Selection Allowed

This property allows you to disable cell selection. This can be helpful to reduce the communication between the application and Lumira Designer runtime if you do not use any events on cell selection. This property is not selected by default.

Using the property Cell Selects Entire Row

Use this property if you want to select the entire row when the cell is clicked. This property does not have any effect if your selection mode is set to *None*. If you have activated the multiple row selections mode (*Multi* or *Multi (Toggle)*, all other rows will be deselected as using this property results in the selection of a single row. This property is not selected by default.

Using the property Merge Rows with Same Text Value

Use this property if you want to remove duplicate texts in the same column. This property cannot be used with the property „*Column Content Appends to Previous Column*“. This property is not selected by default.

Example: The following example shows a scorecard with the characteristics *Region* and *Product Group*. In the second graphic, the property *Merge Rows with Same Text Value* has been selected for the column *Product Group*.

Product Group	Region	Billed Quantity
Bag & Outdoor DS10	Central 80501	7.070.593,00
Bag & Outdoor DS10	North East 80502	1.591.776,00
Bag & Outdoor DS10	South East 80503	11.658.445,00
Bag & Outdoor DS10	West 80504	3.236.162,00
Bag & Outdoor DS10	Summe	23.556.976,00
Accessories DS20	Central 80501	38.948.284,00
Accessories DS20	North East 80502	7.847.460,00
Accessories DS20	South East 80503	62.646.731,00
Accessories DS20	West 80504	17.431.338,00
Accessories DS20	Summe	126.873.813,00
Office DS30	Central 80501	26.049.267,00

< 1 2 >

Product Group	Region	Billed Quantity
Bag & Outdoor DS10	Central 80501	7.070.593,00
	North East 80502	1.591.776,00
	South East 80503	11.658.445,00
	West 80504	3.236.162,00
	Summe	23.556.976,00
Accessories DS20	Central 80501	38.948.284,00
	North East 80502	7.847.460,00
	South East 80503	62.646.731,00
	West 80504	17.431.338,00
	Summe	126.873.813,00
Office DS30	Central 80501	26.049.267,00

< 1 2 >

40.3.2 Configuring Scorecard Headers and Group Headers

As a header is bound to the column, you need to select the column first, in order to specify the header. Select a column in the [Columns and Group Headers](#) area in the [Additional Properties](#) sheet, click on the [Header Layout](#) tab, to see all available properties and set them according to your needs. The header layout properties consist of the [Column Header Layout](#) and the [Column Header Font](#) properties.

You can hide the header area if you want to show merged columns with a single group header description.

The group header is initially invisible and can be activated by selecting the [Show Group Header](#) function. Then you can select a group header that you want to define. Group headers are special because they can be merged between columns. This allows you to use a description across multiple columns. There are no other special settings for the group header. The [Group Header Content](#) is a text, which can be directly maintained or bound to the data selection.

40.3.3 Configuring the Scorecard Cell Content

Each cell (whether it is a header, group header or content cell) has a bunch of properties for layout and text design. Text properties are only available for cells which support the text inside.

In the layout area, you can find different properties for the borders, padding, and background settings. In the text area, you can define the font name, size, color and all other settings.

There are two special settings for the cell layout and cell font:

- **Cell Visible**
This property hides the cell, other properties do not apply
- **Text Wrapping**
This property allows the text to be wrapped. A wrapped text is not positioned in the vertical middle, but to the top of the cell. You can influence the position by changing the top padding of the cell.

40.3.4 Defining the Scorecard Column Content

The main content of the scorecard are the columns. A column consists of four parts:

- column content
- column cell layout
- column header layout
- column properties

The column content is dependent of the selected column template. The scorecard has three column template types:

- content display for values, texts and images – for single selection content
- comparison chart as Bar, Bullet, Delta Bar, Fraction Pie – for comparison of two single selections
- trend chart as Line, Column, Bubble – for list of multiple cells selection

The properties that are available for every column template type can be maintained manually (in this case, the properties are set with same value for all rows) or they can be bound to a selection and are evaluated for each row. The binding allows you to reflect the data conditions in the row visualizations.

After each property description you can find functions which helps you with the configuration:

- common properties (like cell background) can be applied to all columns
- header properties can be applied also to group headers
- group header properties can be applied to all group headers
- all properties can be reset to a default value (the configuration will be removed)

40.3.5 Defining the Textual Content

The template for textual content can be used to display members from dimensions, attributes and the data from selected measures. It allows you to set basic properties for the content, and activate icons or images. The main property is the [Text](#) property. You can select any content to display as text in the cell.

40.3.6 Configuring Trend Charts

Trend charts can help you to visualize data series from dimensions which are not included in the row scope definition.

The column chart can be used when you have a small number of data points and only one axis.

The line chart can be used when you have a bigger number of data points and only one axis. In addition, you can add target data points which are visualized by a dotted line.

The bubble chart can be used when you want to compare the distribution of data and you have more than one axis (two or three). The bubble chart displays the actual data in the Y axis, the target data in the X axis and can visualize the bubble size on an additional Z axis.

40.3.7 Configuring Comparison Charts

Comparison charts can help you to visualize data and compare it to other data.

The bar chart is used for single value display.

The bullet chart is used to display two values (actual and target) with a third value in the background (forecast). In addition you can define up to five threshold values.

The delta bar chart is used to compare the delta value between two values.

The fraction pie chart is used to express a fraction as a percentage of the pie.

40.4 Interacting with Scorecards

The scorecard component supports interaction in the form of selections.

Cell selection is activated by default. In the *onSelect* event you can access the current selection of the dimensions specified in the row scope and the ID of the selected column.

You can use the property *Selection Type* to activate the row selection. The property is set to *None* by default. You can activate *Single Selection*, *Multiple Selection* and *Multiple (Toggle) Selection*.

In the event you get the current selection of the dimensions specified in the row scope. If the user selects a cell, he/she can access the selected column ID. If the user selects a row, the column ID is empty.

40.5 Binding the Properties of Scorecards to Data Sources

Property binding allows filling in the value of the property by using the data source. Depending on the property type, there are different ways to fill the proper content for the scorecard property:

- direct binding: The bound value is used directly as property value
- indirect (conditional) binding: The bound value must be translated into the proper property value

Additionally, depending on the property type it is also possible to bind an aspect item of the data set to the property. The data set "aspect" can include the following items:

- value of a data cell
- content of master data information
- content of a dimension or measure
- other metadata content which is available on the data set

Property Binding Basics

Almost all scorecard properties in the [Additional Properties](#) sheet can be bound to a data source. The binding reads out the value of the chosen aspect and places the value into the property value. In case of direct binding the value is used directly in the property. All values will be validated according to the property type. If the bound value is returning a value which cannot pass the validation, the default value will be applied. It is also possible to specify conditional binding. There are up to nine conditions possible.

Conditional Binding

Conditional binding enables the creation of indirect binding of data source aspects into the property value. You can specify up to nine different conditions. The system will read out the bound value from the data source and evaluate the conditions taking the [Conditional Format Value](#) of the conditional binding key which equals to the bound value.

Example: You want to bind a background color to the conditional format. If the format is returning “3” you want to make the color “red”. For this you have to select the binding type [Conditional Format Value](#) and select its properties. Then you have to activate conditional binding and insert a new entry (in addition to the default entry). The key of the entry must be **3** and the value **red**.

i Note

Conditional binding can be used for all binding types, there is nevertheless a restriction to nine values.

40.5.1 Properties with Single Values

The majority of the properties which can be bound to a data source are of the the single value type. This means, in case you bind these properties to an aspect of the data source, these properties must get a single value as content. There are multiple binding types to select for single value properties:

- [Single data cell value](#)
- [Conditional Format Value](#)
-> details
- [Attribute Member Content](#)
- [Dimension Member Content](#)
- [Multiple Cell Count](#)
- [Dimension Metadata Content](#)

- [Measure Metadata Content](#)
- [Attribute Metadata Content](#)

An example for a single value property is the [Cell Background](#).

Depending on the binding type you have selected for a single value property, you have to set further properties relating to the chosen binding type.

Related Information

[Binding Type Settings \[page 329\]](#)

40.5.2 Properties with Multiple Values

Properties for values in charts enable the selection of multiple cells. The result of this binding is represented by multiple cells with values. For this property type, you can only select [Multiple Cell Values](#) as the binding type. An example for a property with multiple values is “Actual Values” in trend charts.

Related Information

[Binding Type Settings \[page 329\]](#)

40.6 Binding Type Settings

Binding Type Settings for Single Value Properties

Depending on the binding type you have selected for a single value property, you have to set further properties relating to the chosen binding type:

- binding type: Single Data Cell Value

Property	Description
Selected Data Cell	Specifies the cell that should provide its value (content)

Property	Description
Content Type	<p>Specifies the type of content to be used in the cell:</p> <ul style="list-style-type: none"> Value as Number The mere value, unformatted as float value is displayed. The scaling factor defined in the initial view is applied. Float values use the "." character as decimal separator; thousands separators are not applied and all decimals with value 0 are ignored. The value as number is language independent. Example: "241.80" becomes "241.8"; 241.000 becomes "241". Value as Formatted String The value is displayed as in the crosstab component, respecting the scaling factor and the decimal places. Those values are language dependent and are not recommended for use in conditional binding.

Selection is Relative in Row	This property is selected by default. In this case the chosen cell applies the row scope definition. When you uncheck this property, the selection is always bound to the selected cell – in every row.
------------------------------	---

- binding type: Conditional Format Value
This binding type can be used to access the content of the conditional formatting (BEx Exceptions) for the selected data cell.

Property	Description
Selected Data Cell	Specifies the cell that should provide its value (content)
Conditional Format	Specifies the conditional format which should be selected from the cell.

- binding type: Attribute Member Content
This binding type can be used to access the attributes of a dimension.

Property	Description
Dimension	Specifies the dimension which should provide its attributes.
Attribute	Specifies the attribute which should provide its content.
Attribute Content	Specifies the content of the attribute – text or key.

- binding type: Dimension Member Content
This binding type can be used to access the members of a dimension.

Property	Description
Dimension	Specifies the dimension which should provide its members.
Member Display	<p>Specifies the content of the member object:</p> <ul style="list-style-type: none"> Value Presentation the member presentation selected in the query or the initial view Value Key the member key Member Type the type of the member; can be used for hierarchy members Member Node State the state of the member, can be used for hierarchy members Member Level the level of the member; can be used for hierarchy members
Member Presentation Part	<p>Specifies the Member Presentation Part. Only if you have selected <i>Value Presentation</i> for the <i>Member Display</i>, the following choices are available:</p> <ul style="list-style-type: none"> Default The default is the complete member presentation. First Part This is the first part of the value presentation. Last Part This is the last part of the value presentation.

- binding type: Multiple Cell Count
This binding type can be used to access the count of the selected cells.

Property	Description
Selected Data Cells	Specifies the selected data cells.
Selection Mode	Specifies the behavior of the selection – with or without results.

- binding type: Measure Metadata Content
This binding type can be used to access the dimension metadata.

Property	Description
Measure	Specifies the dimension which should provide its meta-data.
Measure Display	Specifies the content of the metadata – text or key. In addition, the following measure attributes can be selected: <ul style="list-style-type: none"> ○ Unit of Measure ○ Scaling Factor ○ Format String

- binding type: Attribute Metadata Content
This binding type can be used to access the attribute metadata of a dimension.

Property	Description
Dimension	Specifies the dimension which should provide its attributes.
Attribute	Specifies the attribute which should provide its content.
Attribute Display	Specifies the content of the attribute – text or key.

Binding Type Settings for Multiple Values Properties

Depending on the binding type you have selected for a multiple values property, you have to set further properties relating to the chosen binding type:

This binding type can be used to access multiple cells. This property is used for selections in trend charts.

Property	Description
Selected Data Cells	Specifies the selected data cells.
Selection Mode	Specifies the behavior of the selection – with or without results.

41 Working with the Scroll Container

The *Scroll Container* component enables or disables horizontal or vertical scrolling on supported container components at design time, or runtime of the application.

Add a *Scroll Container* component to another container component, to allow application users to scroll horizontally or vertically within the container component. Using scrolling allows the users to view all items within the container component. Simply drop the *Scroll Container* into one of the supported container components. Users can then scroll horizontally, vertically or both, depending on the property settings at design time or runtime. When the *Horizontal Scrolling Enabled* or *Vertical Scrolling Enabled* design properties are set to true, the scroll bars are displayed, when they are required.

When the user holds and moves an item within the *Scroll Container*, the scroll bars automatically scroll in the same direction. This automatic scrolling enables the user to see to where they are moving an item within the *Scroll Container*. Touch scrolling is also possible on all touch enabled devices using SAP BusinessObjects Mobile, Open Document, the BI platform, or in local mode

The *Scroll Container* can be used with the following container components:

- dialog
- panel
- popup
- story
- tabstrip
- pagebook (vertical scrolling only)

i Note

- Define the *Width* and *Height* properties of the *Scroll Container* to ensure that scrollbars are shown. If the size settings are set to auto and the top, left, right, and bottom margins are not well defined, the *Scroll Container* container might expand, and hide the scrollbars.
- With the *Pagebook* containers on SAP BusinessObjects Mobile or touch enabled devices, the swipe function is used to navigate through pages or tabs. We recommend that only vertical scrolling should be used with the *Pagebook* component.
- When adding items to the *Scroll Container*, make sure that you do not place items outside the top and left margins of the *Scroll Container*. Items move back inside the *Scroll Container* automatically when items are placed in these locations.
- The *Adaptive Layout* and the *Grid Layout* containers, do not require scrolling.
- You cannot interact with the *Scroll Container* scroll bars if the child component decorators overlap with the scroll bars.
- At design time, the scroll bar interaction does not work when the *Scroll Container* is inside a *Dialog*, *Pagebook* or a *Popup*.

- For more information about *Scroll Container* properties, you can refer to the chapter in this guide called *Scroll Container* [page 503].
- You should read the following SAP Note before using the *Scroll Container* component:

SAP Note Number	Description
2566896 	Using the Scroll container inside a pagebook

Related Information

[Scroll Container \[page 503\]](#)

[Enabling and Disabling Scrolling \[page 334\]](#)

[Adding Items to the Scroll Container \[page 335\]](#)

[Moving Items Within the Scroll Container \[page 335\]](#)

41.1 Enabling and Disabling Scrolling

Use properties to enable and disable scrolling.

Context

You have added a *Scroll Container* to your application.

Procedure

1. Select the *Scroll Container* from the *Outline* view, or within the *Layout* pane in the design tool.
The *Scroll Container* properties appear.
2. To allow horizontal scrolling in your application, set the *Horizontal Scrolling Enabled* property to true.
The horizontal scroll bar then appears in the *Scroll Container*, when it is required.
3. To allow vertical scrolling in your application, set the *Vertical Scrolling Enabled* property to true.
The vertical scroll bar then appears in the *Scroll Container*, when it is required.
4. To disable horizontal scrolling, set the *Horizontal Scrolling Enabled* to false.
5. To disable vertical scrolling, set the *Vertical Scrolling Enabled* to false.

Related Information

[Adding Items to the Scroll Container \[page 335\]](#)

41.2 Adding Items to the Scroll Container

Use the [Scroll Container](#) to create an area in a supported container component that can scroll horizontally, vertically or both.

Context

The [Horizontal Scrolling Enabled](#) and [Vertical Scrolling Enabled](#) properties of the [Scroll Container](#) are set to true.

Procedure

1. Select the component you want to add to the [Scroll Container](#).
2. Drop the selected component within the outline of the [Scroll Container](#) in the layout area. Or, drop the selected component within the [Layout](#) folder of the [Outline](#) view of the design tool.

The component is now within the scrollable area of the [Scroll Container](#).

Related Information

[Moving Items Within the Scroll Container \[page 335\]](#)

[Enabling and Disabling Scrolling \[page 334\]](#)

41.3 Moving Items Within the Scroll Container

Use the automatic scroll bars to see where you are going when changing the location of items in the [Scroll Container](#).

Context

You have dropped a component into the [Scroll Container](#), and the [Horizontal Scrolling Enabled](#) and [Vertical Scrolling Enabled](#) properties of the [Scroll Container](#) are set to true.

i Note

- The component you want to move must already be dropped within the [Scroll Container](#). If you try to drag and hover a component over the [Scroll Container](#) without dropping it first, the automatic scrolling does not occur.
- The component you want to move must be within 20 pixels of the [Scroll Container](#).

Procedure

1. Select the component you want to move within the [Scroll Container](#).
2. Hold and move the component vertically downwards or upwards at the edge of the [Scroll Container](#). The component you are moving must stay inside the [Scroll Container](#) for the auto scrolling to occur.

The vertical scroll bar automatically follows in the direction in which you are moving the component.
3. Hold and move the component horizontally left or right at the edge of the [Scroll Container](#). The component you are moving must stay inside the [Scroll Container](#) for the auto scrolling to occur.

The horizontal scroll bar automatically follows in the direction in which you are moving the component.

Related Information

[Adding Items to the Scroll Container \[page 335\]](#)

[Enabling and Disabling Scrolling \[page 334\]](#)

42 Working with the Story Component

Use the [Story](#) to create a presentation-style application that uses visualizations, text, graphics, illustrations, shapes, and other customizations to describe data.

You use the [Story](#) component to group and order the content of your application into tab-like pages. A [Story](#) may have a single visualization or story pages full of visualizations. When you add a [Story](#) to your application it contains a [Filter Bar](#) on the [Story](#) level. The [Story](#) component allows you to apply a filter across multiple data bound components within the same [Story](#). Explore and analyze the data using filters, controls, calculations, conditional text, and other tools.

Related Information

[Story \[page 503\]](#)

42.1 Adding Content to the Story

Add story pages to the [Story](#) component.

Context

You have a [Story](#) component in an application and you wish to add story pages and components to it.

Procedure

1. Select the [Story](#) component from the [Outline](#) view.
2. Use the context menu to select ► [Create](#) ► [Story Page](#) ▾.
A new page is added to the [Story](#) in the [Layout](#) area with the default name [STORYPAGE_x - New Page](#).
3. Repeat step 2 until you have added the required number of story pages.
4. Select the [STORYPAGE_1 - New Page](#) within the [STORY_1](#) in the [Outline](#) view.
5. Using the context menu, select [Create](#) and choose a component type you wish to add to the story page.
6. Repeat step 5 until you have added the required number of pages.

The selected components are all added to the story pages in the [Story](#) component according to your specifications.

Related Information

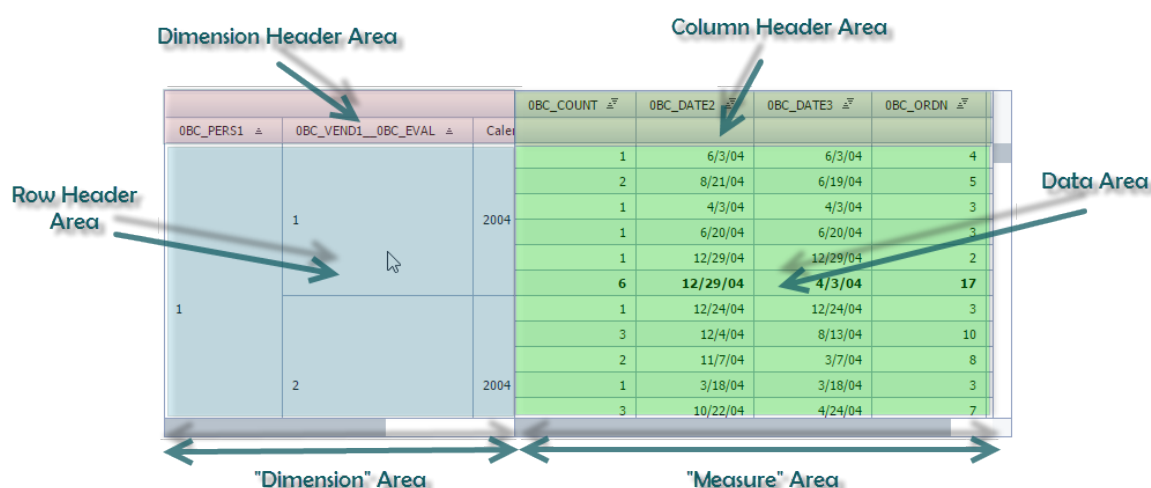
[Working with the Story Component \[page 337\]](#)

43 Using Header Scrolling in Crosstabs

With SAP Design Studio 1.6, the header scrolling function of the crosstab component has been changed. Incompatible changes and improvements with regard to earlier versions (DS 1.4 and DS 1.5) have been made in this feature, so starting from release 1.6, this documentation is the only source of reference.

Terminology and Basic Concepts

The following picture displays the main concepts and their terms when using the header scrolling function:



- "Dimension Area" represents the combined dimension/row header area of the crosstab
- "Measure Area" represents the combined column header/data area of the crosstab
- "Content Width" represents the overall width of the combined "Dimension" Area and "Measure" Area that results from setting the crosstab's *Width* property. If the width property is set to *auto*, the "Dimension" Area and "Measure" Area will be dependent on the crosstab's container width and/or the width of the browser window.

Crosstab Properties

The header scrolling function of the crosstab is controlled by the following properties:

- Properties in section *User Interactivity*
 - *Horizontal Scrolling for Header Enabled*: true/false (default: false)
 - *Horizontal Header Resizing Enabled*: true/false (default: false)
- Properties in section *Display*

- *Maximum Width of Header Area*: width in pixel / auto (default: auto)

When using these properties, note the following:

- If scrolling of the dimension area is needed, the property *Horizontal Scrolling for Header Enabled* needs to be set to *true*. Only if this property is set to "true", the remaining properties (*Horizontal Header Resizing Enabled* and *Maximum Width of Header Area*) are taken into account to enable finer control of the way the dimension area behaves.
- If *Horizontal Header Resizing Enabled* is set to *true*, the vertical border between dimension area and measure area can be moved horizontally by the user by dragging and dropping that border horizontally, thus altering the width ratio between dimension area and measure area. This happens so that the crosstab overall width remains unaltered if possible. However, if the user makes the dimension area width quite small and there are not enough columns in the measure area to fill the remaining horizontal space to fill all of the content width, the crosstab will horizontally -shrink. If the dimension area is enlarged horizontally again, the crosstab will horizontally expand up to the content width again. If *Maximum Width of Header Area* is set to a pixel value, the user cannot enlarge the dimension area beyond that value (see also the following "Maximum Width of Header Area" bullet point).
- "Maximum Width of Header Area":
 - When the property is set to "auto", the width of the dimension area can take any value within the horizontal boundaries of a fully rendered dimension area.
 - When the property has been set to a pixel value, the width of the dimension area can only be lower than or equal to that value. The automatic dimension area width determination and the user's possibility to alter the width of the dimension area if *Horizontal Header Resizing Enabled* is set to "true" take a non-auto pixel width setting into account. If that pixel value is set, the crosstab tries to render the dimension header area width in this size. However, if the property's pixel value is larger than the fully rendered width of the dimension area, the dimension area only takes Maximum Width up as much horizontal space as it actually needs. The Maximum Width up as much horizontal space as it actually needs. The of Header Area dimension area width is not artificially enlarged (cells made wider) to stretch out to the given pixel width.

i Note

- There is no need for any type of horizontal scrolling in the crosstab (which means if dimension area and measure area fully fit into the content width, will not be applied).
- The property *of Header AreaMaximum Width of Header Area* cannot be set to 0 and the dimension area cannot be excluded from the crosstab by trying to do so.
- The actual rendered width of the dimension area in case it is limited by a pixel setting may differ by a few pixels from the exact property value, depending on the SAPUI5 theme used.

43.1 "Horizontal Scrolling Enabled" Set to "False"

If you set the crosstab property *Horizontal Scrolling Enabled* to *false*, the crosstab applies the default rendering, which means the dimension area is always rendered to its full horizontal extent, and whatever horizontal space remains of the content width is used for the measure area. If the width of the dimension area exceeds the content width, there is no scrolling of the dimension area available and columns extending beyond the crosstab boundaries are not displayed or reachable.

Also, if the dimension area takes up most of the content width so that only a very small portion of the measure area can be rendered, the dimension area does not shrink (and is hence scrollable) to make more room available for the measure area.

43.2 "Horizontal Scrolling Enabled" Set to "True"

If you set the crosstab property *Horizontal Scrolling Enabled* to *true*, consider the following two scenarios:

1. Width of the dimension area and the measure area combined is smaller than the content width

In this case there is no need for any scrolling at all. Both dimension area and measure area render to their full horizontal extent without any scrollbars. Also, the application user will not be able to manually move the border between dimension area and measure area even if the property *Horizontal Header Resizing Enabled* is set to "true". The *Maximum Width of Header Area* property is not applied either.

2. Width of the dimension area and the measure area combined exceeds the content width

Depending on whether the width of the dimension area is limited or not, take these two scenarios into account:

- *Maximum Width of Header Area* set to "auto", general strategy
- *Maximum Width of Header Area* set to a pixel value

Related Information

[General Strategy if "Maximum Width of Header Area" Set to "Auto" \[page 341\]](#)

[Behavior after Manual Changing of Header With \[page 345\]](#)

[Crosstabs With a Dimension Area or a Measure Area Only \[page 350\]](#)

["Maximum Width of Header Area" Set To A Pixel Value \[page 351\]](#)

43.2.1 General Strategy if "Maximum Width of Header Area" Set to "Auto"

The general strategy is to distribute the widths of the dimension area and the measure area roughly in a 1:1 ratio across the available content width (it is not always exactly 1:1!). However, this is not always possible, for

example, if the dimension area has many columns and the measure area has only few columns or just a single column. In that case, the dimension area width is determined so that the full width of the measure area can be rendered within the content width. This might result in the scrollable dimension area being much wider than the measure area.

Examples:

- Crosstab with a roughly 1:1 width distribution between dimension area and measure area

OBC_PERS1	OBC_VEND1__OBC_EVAL	OBC_COUNT	OBC_DATE2	OBC
1	1	1	6/3/04	
		2	8/21/04	
		1	4/3/04	
		1	6/20/04	
		1	12/29/04	
		6	12/29/04	
	2	1	12/24/04	
		3	12/4/04	
		2	11/7/04	
		1	3/18/04	
		3	10/22/04	
		1	6/26/04	

- Crosstab with scrollable dimension area and fully rendered measure area that only has two columns.
Dimension area width > 50% of content width

OBC_PERS1	OBC_VEND1__OBC_EVAL	Calendar Year	OBC_	OBC_COUNT	OBC_DATE2
1	1	2004	bank c	1	6/3/04
			by bill	2	8/21/04
			check	1	4/3/04
			credit	1	6/20/04
			EC car	1	12/29/04
			Resu	6	12/29/04
	2	2004	bank c	1	12/24/04
			by bill	3	12/4/04
			cash	2	11/7/04
			check	1	3/18/04
			coupo	3	10/22/04
			credit	1	6/26/04

- Shrinking the above example with the two columns measure area. Now, the full measure area cannot be rendered any more without the dimension area width falling below 50% of the Content Width. Hence, a 1:1 width distribution is rendered again

OBC_PERS1	OBC_VE	OBC_COUNT	OBC_D
1	1	1	
		2	
		1	
		1	
		1	
		6	1
	2	1	
		3	
		2	
		1	
		3	
		1	

If the property *Horizontal Header Resizing Enabled* is set to “true”, the user may move the border between dimension area and measure area. Depending on how many columns are available in the measure area, shrinking the dimension area may alter the content width so that the crosstab width gets smaller.

Examples:

- Grabbing the border between dimension area and measure area

OBC_PERS1	OBC_VEND1__OBC	OBC_ORDN	OBC_ORDW
			KG
1	1	4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480
		17	23.390
	2	3	7.710
		10	20.130
		8	13.510
		3	8.390
		7	12.380

- Manually enlarged dimension area

			0BC_ORDN
0BC_PERS1	0BC_VEND1__0BC_EVAL	Calendar Y	
1	1	2004	
	2	2004	

- Manually shrunk dimension area

		0BC_ORDN	0BC_ORDW	0BC_TURN
0BC_PERS1		KG		EUR
1	1	4	1.460	54.04
		5	10.050	135.93
		3	2.220	73.27
		3	5.180	50.36
		2	4.480	18.40
		17	23.390	332.00
	2	3	7.710	15.82
		10	20.130	146.11
		8	13.510	88.65
		3	8.390	95.50
		7	12.380	173.37

- The default layout when the measure area is small enough

		OBC_ORDN	OBC_ORDW
OBC_PERS1	OBC_VEND1__OBC_EVAL		KG
1	1	4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480
		17	23.390
	2	3	7.710
		10	20.130
		8	13.510
		3	8.390
		7	12.380

- The dimension area of the above example has now been manually shrunk. Observe that the overall crosstab width is now smaller since there are not enough columns in the measure area to fill the width up to the available content width.

		OBC_ORDN	OBC_ORDW
OBC_PERS1			KG
1		4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480
		17	23.390
		3	7.710
		10	20.130
		8	13.510
		3	8.390
		7	12.380

43.2.2 Behavior after Manual Changing of Header With

Once the user has altered the dimension area width manually by using the border handle, all following width resize operations that may be applied to the crosstab as a whole will try to preserve that dimension area width, except in case the crosstab has been enlarged so that the dimension area can be rendered without a scrollbar. If the crosstab is then shrunk again, the automatic algorithm applies and any previously made dimension area width setting will be lost.

When the crosstab width is shrunk so that the content width becomes smaller than the dimension area width that was set by the user, the user setting will also be discarded and the auto-width distribution strategy applies.

When the dimension area width is manually altered, subsequent resize operations on the crosstab as a whole might lead to the crosstab not taking up the full available space.

Please see the following example sequence:

- 1. Two identical crosstabs, but in the lower crosstab the dimension area was manually expanded by the user.

		OBC_ORDN	OBC_O
OBC_PERS1	OBC_VEND		KG
1	1	4	
		5	
		3	
		3	
		2	

		OBC_ORD
OBC_PERS1	OBC_VEND1__OBC_EVAL	
1	1	

- 2. Increasing the crosstab width. Note how the upper crosstab uses the auto 1:1 width distribution while the lower crosstab preserves the dimension area width set by the user.

		OBC_ORDN	OBC_ORDW
OBC_PERS1	OBC_VEND1__OBC_EVAL		KG
1	1	4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480

		OBC_ORDN	OBC_ORDW
OBC_PERS1	OBC_VEND1__OBC_EVAL		KG
1	1	4	1
		5	10
		3	2
		3	5
		2	4

3. Further increasing the crosstab width. Note how in the upper crosstab the automatic algorithm fills the available content width, while the lower crosstab preserves the user's dimension area width setting, thus resulting in the crosstab not using up all the available space.

			OBC_ORDN	OBC_ORDW
OBC_PERS1	OBC_VEND1__OBC_EVAL	Calendar Y		KG
1	1	2004	4	1.460
			5	10.050
			3	2.220
			3	5.180
			2	4.480

		OBC_ORDN	OBC_ORDW
OBC_PERS1	OBC_VEND1__OBC_EVAL		KG
1	1	4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480

4. Increasing the crosstab width so everything can be rendered without any scrollbars. In this case, the user set header width is also discarded since header scrolling is no longer necessary.

				OBC_ORDN ㉔	OBC_ORDW ㉔
OBC_PERS1 ㉔	OBC_VEND1__OBC_EVAL ㉔	Calendar Year ㉔	OBC_PAYM ㉔		KG
1	1	2004	bank collection	4	1.460
			by bill	5	10.050
			check	3	2.220
			credit card	3	5.180
			EC card	2	4.480
			Result	17	22.300

				OBC_ORDN 27	OBC_ORDW 27
OBC_PERS1 26	OBC_VEND1__OBC_EVAL 26	Calendar Year 26	OBC_PAYM 26		KG
1	1	2004	bank collection	4	1.460
			by bill	5	10.050
			check	3	2.220
			credit card	3	5.180
			EC card	2	4.480
			Result	17	22.300

5. Again horizontally shrinking the crosstab shows that the user-set dimension area width is not being preserved, thus resulting in both crosstabs behaving exactly the same.

OBC_PERS1	OBC_VEND1__OBC_EVAL	OBC_ORDN	OBC_ORDW
			KG
1	1	4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480

OBC_PERS1	OBC_VEND1__OBC_EVAL	OBC_ORDN	OBC_ORDW
			KG
1	1	4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480

6. Moving the lower crosstab's handle to shrink the dimension area. Note the size change in the crosstab due to insufficient columns in the measure area.

		OBC_ORDN	OBC_ORDW
OBC_PERS1	OBC_VEND1__OBC_EVAL		KG
1	1	4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480

		OBC_ORDN	OBC_ORDW
OBC_PERS1	OBC_VEND1		KG
1	1	4	1.460
		5	10.050
		3	2.220
		3	5.180
		2	4.480

7. Shrinking the crosstab so that the content width gets smaller than the user-set dimension area width. In this case, the user-set dimension area width is discarded and the auto-layout applies, resulting in upper and lower crosstab behaving identically again.

	OBC_ORDN
OBC_PERS1	
1	

	OBC_ORDN
OBC_PERS1	
1	

Also, be aware that through setting the dimension area width manually and then shrinking the crosstab, situations like the following may arise where the measure area gets very small while the auto-layout constantly provides a usable layout: Unusable measure area due to manual dimension area width setting in the lower crosstab. Upper crosstab with auto-layout provides usable width distribution.

	0BC_ORDN	
0BC_PERS1		
1		4
		5
		3
		3
		2

0BC_PERS1	0BC_VEND	
1	1	

43.2.3 Crosstabs With a Dimension Area or a Measure Area Only

In some cases, there may only be a dimension area or a measure area without the respective other area present, for example, if there is only the measure structure on the columns axis of the drilldown as shown in the following picture:

0BC_DATE3	0BC_DATE2	0BC_COUNT	0BC_TURN	0BC_ORDN	0BC_ORDW
			EUR		KG
1/1/04	12/31/04	2,000	109,393.65	5,935	10,684.600

When the crosstab is shrunk horizontally, scrolling is also displayed for this case, but even if the property *Horizontal Header Resizing Enabled* is set to *true* and/or there is a pixel value set for the property *Maximum Width of Header Area*, those properties are not applied. Hence, the user is not able to move any area border interactively (because there is none) and no maximum header width pixel value is taken into account. The result may appear as follows:

C_COUNT	0BC_TURN	0BC_ORDN	0BC_ORDW
	EUR		KG
2,000	109,393.65	5,935	10,684.600

The above crosstab has shrunk so that it becomes horizontally scrollable. No maximum header width is applied and no interactive resizing of an area border is possible since there is only one area.

43.2.4 "Maximum Width of Header Area" Set To A Pixel Value

All given facts and aspects described in the sections if the property *Maximum Width of Header Area* is set to *auto* are also valid if the same property is set to a pixel value. The only difference is that the width of the dimension area never exceeds the given pixel value. Also, the user cannot drag the border between dimension area and measure area beyond what would be allowed by the maximum width setting. The crosstab tries to make the dimension area width match the width setting of *Maximum Width of Header Area* if possible. This means, if the dimension area provides enough data or wide enough cells to actually take up the space required to fill the width given through the *Maximum Width of Header Area* property.

Besides, the behavior of *Maximum Width of Header Area* is similar to if the Dimension Area width was set interactively by the user as described above.

Examples:

- Maximum Width of Header Area* set to 600px. However, the dimension area is not wide enough (only around 480px) to extend this far. Hence, the dimension area width grows no larger than its actual size.

OBC_PERS1	OBC_VEND1__OBC_EVAL	Calendar Year	OBC_PAYM	OBC_DATE3	OBC_DATE2	OBC_COUNT
	1	2004	bank collection	6/3/04	6/3/04	1
			by bill	6/19/04	8/21/04	2
			check	4/3/04	4/3/04	1

- Same settings as described in the example above, but the property *Maximum Width of Header Area* is set to 300px.

OBC_PERS1	OBC_VEND1__OBC_EVAL	Calendar Year	OBC_DATE3	OBC_DATE2	OBC_COUNT	OBC_TURN	OBC_ORDN
	1	2004	6/3/04	6/3/04	1	EUR	4
			6/19/04	8/21/04	2		5
			4/3/04	4/3/04	1		3

- Same settings as described in the example above, but shrunk horizontally. Note that, as if it was set by user interaction, the dimension area width is preserved according to the *Maximum Width of Header Area* setting.

			OBC_D
OBC_PERS1	OBC_VEND1__OBC_EVAL	Cale	
	1	2004	

- Shrinking the crosstab further beyond the *Maximum Width of Header Area* value. In that case, auto-layout (1:1) applies.

		OBC_DATE3
OBC_PERS1	OBC	
		6/3/04
		6/19/04
		4/3/04
	1	6/3/04

- Enlarging the crosstab again. Note that the *Maximum Width of Header Area* value is applied again when the overall crosstab width gets big enough.

			OBC_DAT
OBC_PERS1	OBC_VEND1__OBC_EVAL	Cale	
	1	2004	

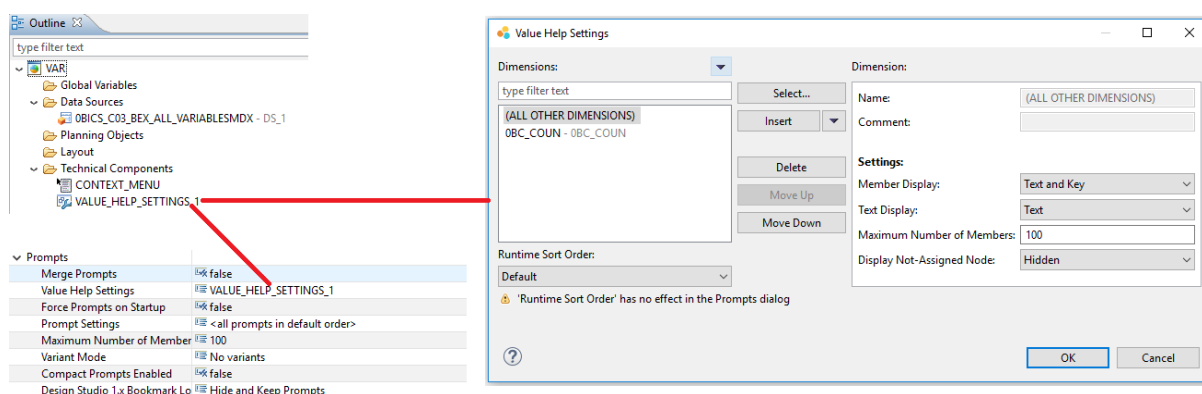
- Enlarging the crosstab further. If the crosstab is big enough to accommodate everything so that scrolling is no longer necessary, *Maximum Width of Header Area* no longer applies.

OBC_PERS1	OBC_VEND1__OBC_EVAL	Calendar Year	OBC_PAYM	OBC_DATE3	OBC_DATE2	OBC_COUNT	OBC_TURN	OBC_ORDN	OBC_ORDW
			bank collection	6/3/04	6/3/04	1	54.04	4	1.460
			by bill	6/19/04	8/21/04	2	135.93	5	10.050
			check	4/3/04	4/3/04	1	73.27	3	2.220
			credit card	6/20/04	6/20/04	1	50.36	3	5.180
1	1	2004							

44 Using the Value Help Settings for the Prompts Dialog

With the application property *Value Help Settings* you can use the *Value Help Settings* technical component for the *Prompts* dialog.

You can configure the value help settings for all dimensions provided by the data source that has been assigned to the application.



Note

Keep the following in mind:

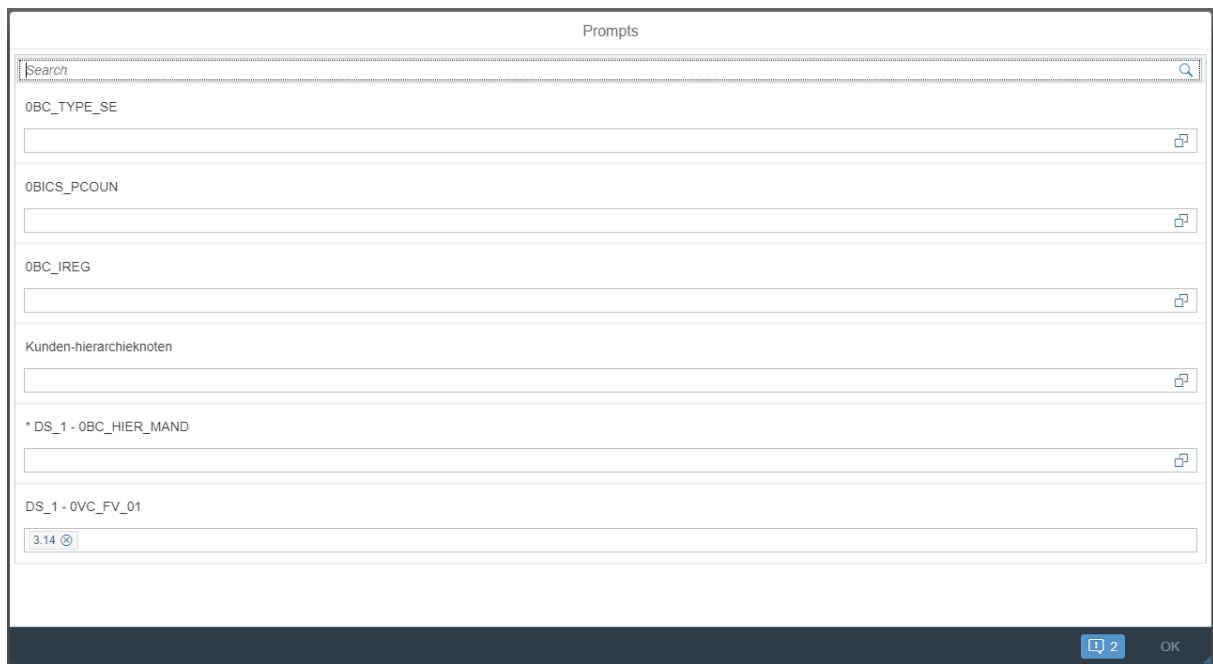
- If a variable is bound to a specific dimension, all the items in the variable's value help are affected by the configuration you make in the Value Help Settings.
- The configuration of the settings is only valid for none- hierarchy dimensions.
- When no specific configuration is listed for a dimension in the Value Help Settings dialog, the configuration of *(ALL OTHER DIMENSIONS)* is applied.

The *Runtime Sort Order* function in the *Value Help Settings* dialog has no effect in the *Prompts* dialog.

45 Using the Compact Layout Mode for Prompts Dialog

With the compact layout presentation mode, you can make more efficient use of the space in the prompts dialog.

By default, all variable names and input fields are listed vertically in the Prompt Dialog, which takes up lots of space:



The screenshot shows a dialog box titled "Prompts". At the top is a search bar with the placeholder text "Search" and a magnifying glass icon. Below the search bar, there is a list of prompts, each consisting of a variable name and an input field. The prompts are: "OBC_TYPE_SE", "OBICS_PCOUN", "OBC_IREG", "Kunden-hierarchieknoten", "* DS_1 - OBC_HIER_MAND", and "DS_1 - OVC_FV_01". Each input field has a small square icon with a plus sign in the bottom right corner. The last input field, "DS_1 - OVC_FV_01", contains the value "3.14" and a small circular icon with a cross. At the bottom right of the dialog, there is a blue button with a speech bubble icon and the number "2", and an "OK" button.

To use the compact mode, set the application property *Compact Prompts Enabled* to *true*. The variable name and its corresponding input field are on the same line in this mode.

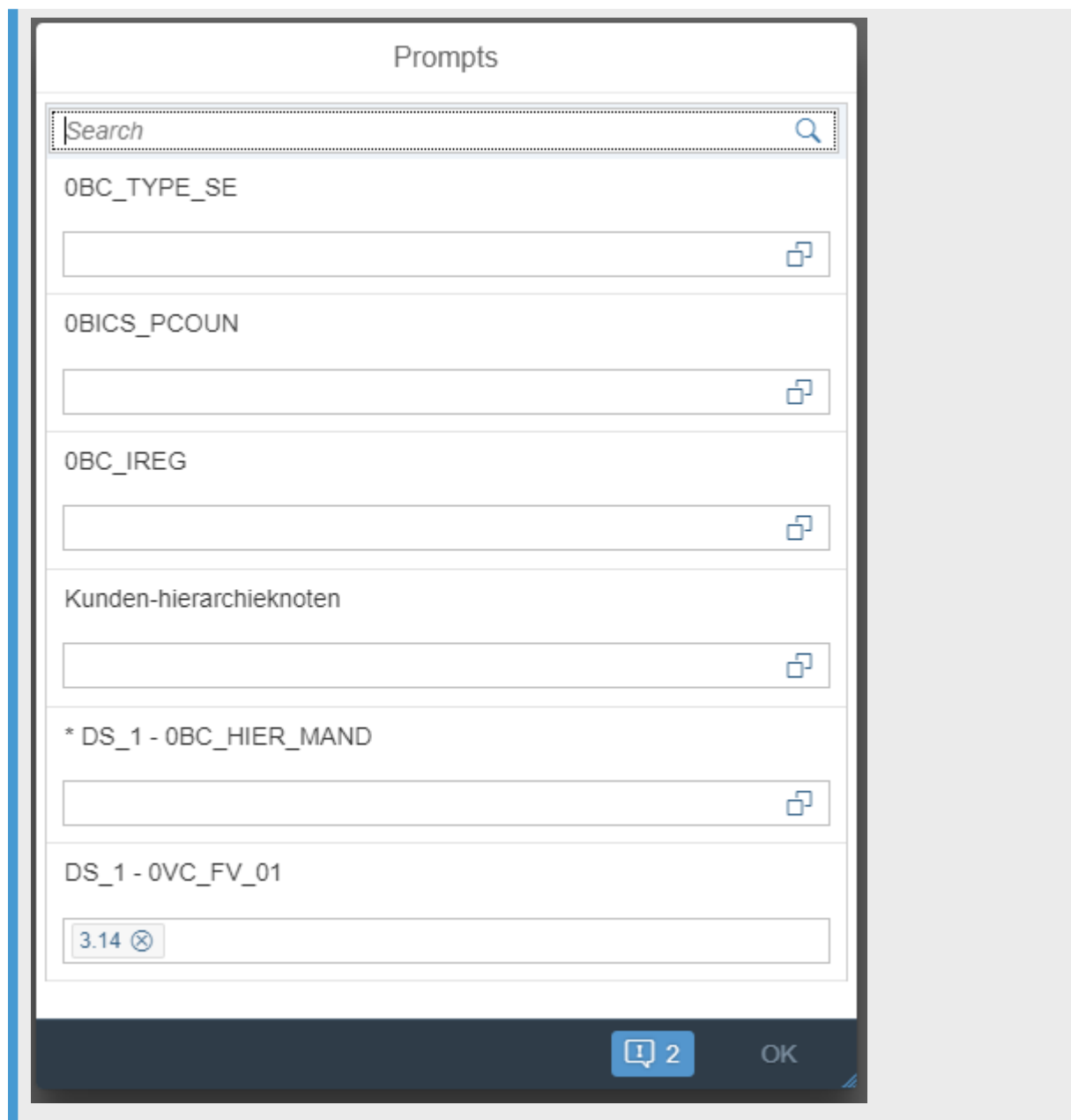
Prompts

OBC_TYPE_SE	<input type="text"/>	
OBCS_PCOUN	<input type="text"/>	
OBC_IREG	<input type="text"/>	
Kunden-hierarchieknoten	<input type="text"/>	
* DS_1 - OBC_HIER_MAND	<input type="text"/>	
DS_1 - OVC_FV_01	<input type="text" value="3.14"/>	

2
 OK

i Note

If the screen width is below a certain value the layout is displayed vertically as in the default display.



46 Working with 1.6 Applications

When working with 1.6 applications in SAP Lumira Designer 2.x, there are some things you need to take into account.

When you use 1.6 analysis applications (applications created with SAP BusinessObjects Design Studio 1.6), you have generally two options:

- You can keep existing applications in the 1.6 format (.biapp file) and edit, save, and run them in both legacy modes of SAP Lumira Designer 2.x: SAP BusinessObjects BI Platform mode, and Local mode.

i Note

SAP BusinessObjects Design Studio 1.6 applications that contain SAP HANA JDBC connections must be adapted to SAP HANA HTTP connections. The adaptation will cause invalid bookmarks. For further information, please see [Exchanging SAP HANA JDBC Connections in 1.6 Analysis Applications \[page 359\]](#) and SAP Note [2538142](#).

The 1.6 applications can be based either on the SAPUI5 commons or the SAPUI5 m rendering mode. If you keep existing applications in the 1.6 format, you cannot use any feature of SAP Lumira Designer and have no interoperability with SAP Lumira Discovery.

The following considerations need to be expected after execution of a 1.6 application within SAP Business Objects BI Platform (legacy) mode:

- 1.6 Applications (Commons Mode):
 - These will render as before, in BI Platform (Legacy) mode, following upgrade to 2.x Lumira add-on for BI Platform.
 - When opening 1.6 commons mode applications within the 2.x Lumira Designer client, these applications are editable as before, but the following should be noted : For legacy classic Chart component and Info Chart component, scripting objects have been deprecated – existing scripting will work without issue, but the script objects will be displayed in the editor with a strikethrough, and context assistance will not be available.

- 1.6 Applications (M Mode)

When opening existing 1.6 M Mode applications within the 2.x Lumira Designer client or on the BI Platform, the following should be noted :

- Auto-migration takes place for 1.6 Info Chart and Geo Map components, as the Info Chart component has been replaced by the new 2.x chart (VizFrame) component in M Mode applications in Lumira 2.x release. These applications will now display with the 2.x Chart (VizFrame) and 2.x VizMap components instead.
- The remainder of the application will remain unchanged.
- In relation to the auto-migration of 1.6 Info Chart components to 2.x Chart (VizFrame) components, the same points as listed in SAP note [2538142](#) are also relevant here.

- You can transform existing 1.6 applications to the 2.x format (.lumx file).

These 1.6 applications must be based on the SAPUI5m rendering mode. If they are based on the SAPUI5 commons rendering mode, you need to migrate them first to the SAPUI5 m rendering mode before transforming them to the 2.x .lumx file format.

Once the applications are transformed to the 2.x format, you can use new features of SAP BusinessObjects Lumira Designer 2.x and the interoperability functions with SAP Lumira Discovery.

Related Information

[Transforming 1.6 Applications to 2.x Applications \[page 358\]](#)

[Exchanging SAP HANA JDBC Connections in 1.6 Analysis Applications \[page 359\]](#)

[Using Automated Migration to Migrate 1.x Applications From SAP HANA JDBC to SAP HANA HTTP \[page 360\]](#)

46.1 Transforming 1.6 Applications to 2.x Applications

To use and edit 1.6 applications in SAP Lumira Designer 2.x, you need to transform the applications from `.biapp` to `.lumx` files.

Context

The transformation needs to be done in one of the legacy modes of SAP Lumira Designer *SAP BusinessObjects BI Platform (Legacy)* mode or the 2.x, either in the *Local Mode (Legacy)* of the design tool.

i Note

We recommend that you create a copy of the original 1.6 application, because migration to the required SAPUI 5 m rendering mode will overwrite the existing application.

Procedure

1. (only valid for 1.6 applications created in the SAPUI5 commons mode) Migrate your 1.6 application based on the SAPUI5 commons mode to the SAPUI5 m mode by choosing *Migrate to SAPUI5 m Mode* in the *Tools* menu.

After you have done this migration, you might need to manually adjust your application's layout:

- Adjust the customer stylesheet if needed
- Adjust the component position sizes if needed

2. 2.x, either inExport the application that you migrated to the SAPUI5 m rendering mode by choosing *Application > Export...* in the menu of the design tool and follow the *Export Application* wizard.

If your application reference resource files are located on the BI platform, and you do not change resource file locations during the transformation process, you would not need to export referenced files. Uncheck the respective *Export Referenced Files* checkbox in the *Content* dialog box of the *Export Application* wizard.

3. Choose *Tools > Preferences* in the design tool menu and switch to *SAP Lumira Documents* as general start-up mode.

You are prompted to restart Lumira Designer. Accept the restart.

4. In the [Documents](#) view, create a new document by right-clicking the appropriate folder depending on where your application was located, and choosing [Create Document...](#)

Application Location	Folder to Choose
The BI platform	The Remote Documents folder
Your PC	The Local Documents folder

5. Import the previously exported .ZIP file into the newly created Lumira document: Choose your application, right-click it and choose [Import Application](#).

Your application is created as a 2.x application within the Lumira document (.lumx file).

Results

You have transformed a 1.6 application to a 2.x application and can now use all SAP Lumira Designer 2.0 features and interoperability with SAP Lumira Discovery. After transforming, you need to pay attention to the following details:

- Applications still using legacy SAP HANA JDBC connections need adaptations.
- Bookmarks based on a 1.6 application will become invalid. You need to create the new bookmarks.

For more information, please see:

- SAP Note [2538142](#) 
- [Exchanging SAP HANA JDBC Connections in 1.6 Analysis Applications \[page 359\]](#)


46.2 Exchanging SAP HANA JDBC Connections in 1.6 Analysis Applications

To run SAP BusinessObjects Design Studio 1.x applications in SAP Lumira Designer 2.0, you'll need to exchange any SAP HANA JDBC connections with SAP HANA HTTP connections and adjust scripts accordingly

Context

Before this is done the data source cannot be initialized. After changing the connections you'll need to adjust scripts in those applications accordingly.

! Restriction

When you have exchanged the SAP HANA JDBC connection with an SAP HANA HTTP connection, the existing bookmarks that contain a SAP HANA JDBC connection do not work anymore. You have to create new bookmarks and replace the old ones. For further information, please see SAP Note [2538142](#) .

Procedure

1. Configure an HTTP connection on the BI platform and SAP HANA (if necessary).

For more information, see “Creating SAP HANA HTTP Connections” in the *Administrator Guide: SAP Lumira* on SAP Help Portal at <http://help.sap.com>.

2. In your application, select the data source that contains the JDBC connection and replace it with an HTTP connection:
 - a. In the *Properties* sheet of the data source, select the property *Connection* and press *Exchange Data Source*.
 - b. In the dialog box that appears, click *Browse* and select your HTTP connection.
 - c. Enter your system credentials if prompted.
 - d. Click OK.

Your SAP HANA HTTP connection is displayed in the *Connection* property of your data source.

3. Adjust places in the application's scripts where the names of SAP HANA hierarchies occur. Use the content assistance in the script editor for the correct values and names.

46.3 Using Automated Migration to Migrate 1.x Applications From SAP HANA JDBC to SAP HANA HTTP

To run SAP BusinessObjects Design Studio 1.x applications in SAP Lumira Designer 2.0, you'll need to exchange any SAP HANA JDBC connections for SAP HANA HTTP connections and adjust scripts accordingly. You can make this exchange manually by using the Exchange Data Source workflow or you can use the automated migration possibilities offered by Lumira Designer.

It is not enough to just point affected data sources to a different SAP HANA connection. As part of the migration, existing initial views of affected data sources need to be modified as well. You can do this in the property sheet of a data source, using the *Exchange Data Source* workflow, or it can happen during an import of an exported 1.x application via the backend connection mapping wizard.

SAP Lumira offers a migration tool that automates these migration tasks.

Two different scenarios are supported:

- Opening a SAP Lumira 1.x application scans it for JDBC-based SAP HANA connections and offers to migrate the application on the spot.
- You can launch a mass migration of SAP Lumira 1.x applications stored on the BI Platform via the *Tools* menu in Lumira Designer.

i Note

All migrations take place in the legacy modes (local mode and BI Platform mode). Migrations cannot be triggered in Lumira Documents mode; here you have to migrate 1.x apps during the import into a Lumira document by mapping the JDBC-based SAP HANA connection to an HTTP-based SAP HANA connection in the import wizard.

The main mode for migrating applications is the legacy BI Platform mode. Below you'll find workflows for the above-mentioned scenarios:

- 1. If you open an application and the system detects the need for migration, a dialog box appears asking you if the migration should be started, and what the implications of the migration will be.
- 2. You can cancel the migration and the application will continue to load.
- 3. If you choose to migrate, a backend connection-mapping dialog box is displayed, showing all affected JDBC-based SAP HANA connections in the left column. In the right column, you must select an existing HTTP-based SAP HANA connection.

i Note

The **OK** button of the dialog box will not be enabled until all connections are mapped (in contrast to the backend connection mapping dialogs elsewhere in Lumira Designer, where the mapping is optional).

4. After pressing **OK**, the application is migrated and opened afterwards.
- 1. If you choose **Tools > Migrate Unsupported SAP HANA Connections**, a dialog box appears explaining what the migration does.
 - 2. If you choose to continue, a dialog box is displayed where the you can pick a root folder on the BI Platform.
This folder is now scanned recursively for applications that require a migration.
 - 3. If no application is found, a corresponding info dialog box is displayed.
Otherwise, a dialog box with the search results is displayed, and you can decide to either start the migration workflow or cancel the migration.
The remaining part is the same as for the single-application migration described above.

47 Advanced Design Tasks

47.1 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 `CTRL` + `Space` on your keyboard.

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 \[page 363\]](#)

[Error Analysis in Scripting \[page 365\]](#)

[Component API Reference \[page 506\]](#)

47.1.1 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 and backend validation

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. Some of these checks are fast. However, others involve calls to a backend server, for example to validate the name of a dimension or a hierarchy, and this can slow down working with larger scripts significantly, especially if you are working over a WAN connection. For this reason, you can explicitly disable backend validation (leaving all other checks in place) either in the [Preferences](#) page of the design tool or directly in the script editor when you work on a script:

- Choose **Tools** > **Preferences** > **Application Design** > **Script Editing** > **Disable backend validation**.
- In the script editor click  and check the option **Disable Backend Validation**.

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 (inplace help)

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.

You can disable the inplace help either in the [Preferences](#) page of the design tool or directly in the script editor when you work on a script:

- Choose **Tools > Preferences > Application Design > Script Editing > Disable inplace help**.
- In the script editor click  on the right and check the option [Disable Inplace Help](#).

i Note

If you experience performance issues on operating systems with limited memory and large applications with complex scripts, SAP recommends to disable the inplace help.

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**, all available functions for the data source alias will be displayed automatically.

Relevance-ordered proposals and contextsensitive 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.

47.1.2 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).

i 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  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.

i 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 Lumira Designer. 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

For further information see the following methods of the class `Application` in the *Component API Reference* on the SAP Help Portal at <https://help.sap.com/viewer/index>:

- `log`
- `alert`
- `createErrorMessage`
- `createInfoMessage`
- `createWarningMessage`

47.1.3 Navigating Between Scripts

You can navigate from one script to another directly in the script editor.

In the script editor press and hold the `CTRL` key in an empty area of the script editor and hover over the script text.

The script text that you hover turns into a blue clickable hyperlink, if it is a call to one of the following elements:

- a global script function
- an interface function
- an explicitly called event handler (like `BUTTON_1.onClick()`)
- the method `APPLICATION.doBackgroundProcessing()`

If you click the link, the following actions take place:

- The system checks if the current script has been modified. If the script has been modified, a dialog box is displayed where you can either apply the modifications, discard the modifications or cancel the navigation.
- First the current script editor is closed, and the new script editor dialog is opened.

In the new script editor dialog a navigation toolbar is displayed, where you can navigate back and forth to any of the previously visited scripts.

i Note

For technical reasons, navigation is not possible, if the initial script editor dialog is displayed on top of another dialog (like editing a dialog button script, editing a keyboard shortcut script).

No hyperlinks will be displayed in this case.

47.1.4 Using Offline Data Sources in Scripting

You can specify the current document directly in special calls to get all offline data sources of the document at once.

In previous SAP Lumira releases, assigning an offline data source to a data source alias at runtime was only possible via the `CONNECTION.getConnections()` API, which, if the document contains at least one offline data source, also returns a special connection representing the document itself. This connection can then be passed to APIs like `CONNECTION.setSystemId()` or `DataSourceAlias.assignDataSource()`. Since the call to `CONNECTION.getConnections()` can cause performance issues, it's better to be able to specify the current document directly in these API calls.

Example

This example shows how to use offline data sources in scripting to populate a dropdown box with all offline data sources in the current document.

First you need to pass in the document ID in a call to `CONNECTION_1.setSystemId()` (assuming you have a backend connection component called `CONNECTION_1` in your application). Unfortunately, this document ID is a GUID, and for local documents this GUID changes as soon as you rename the LUMX file in the file system. This means your script code would break.

To avoid this, you can use the special alias `DOCUMENT` that represents the current document.

Write in your script: `CONNECTION_1.setSystemId(" (DOCUMENT) ")`, followed by a call to `CONNECTION_1.searchDataSources ("*")`. This will return all offline data sources.

If invoked for a `DataSourceConnection` parameter, the content assist proposal popup in the script editor (click `CTRL` + `SPACE`) has a new entry *The current document*. This inserts the new alias into the script text.

47.2 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. 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 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.

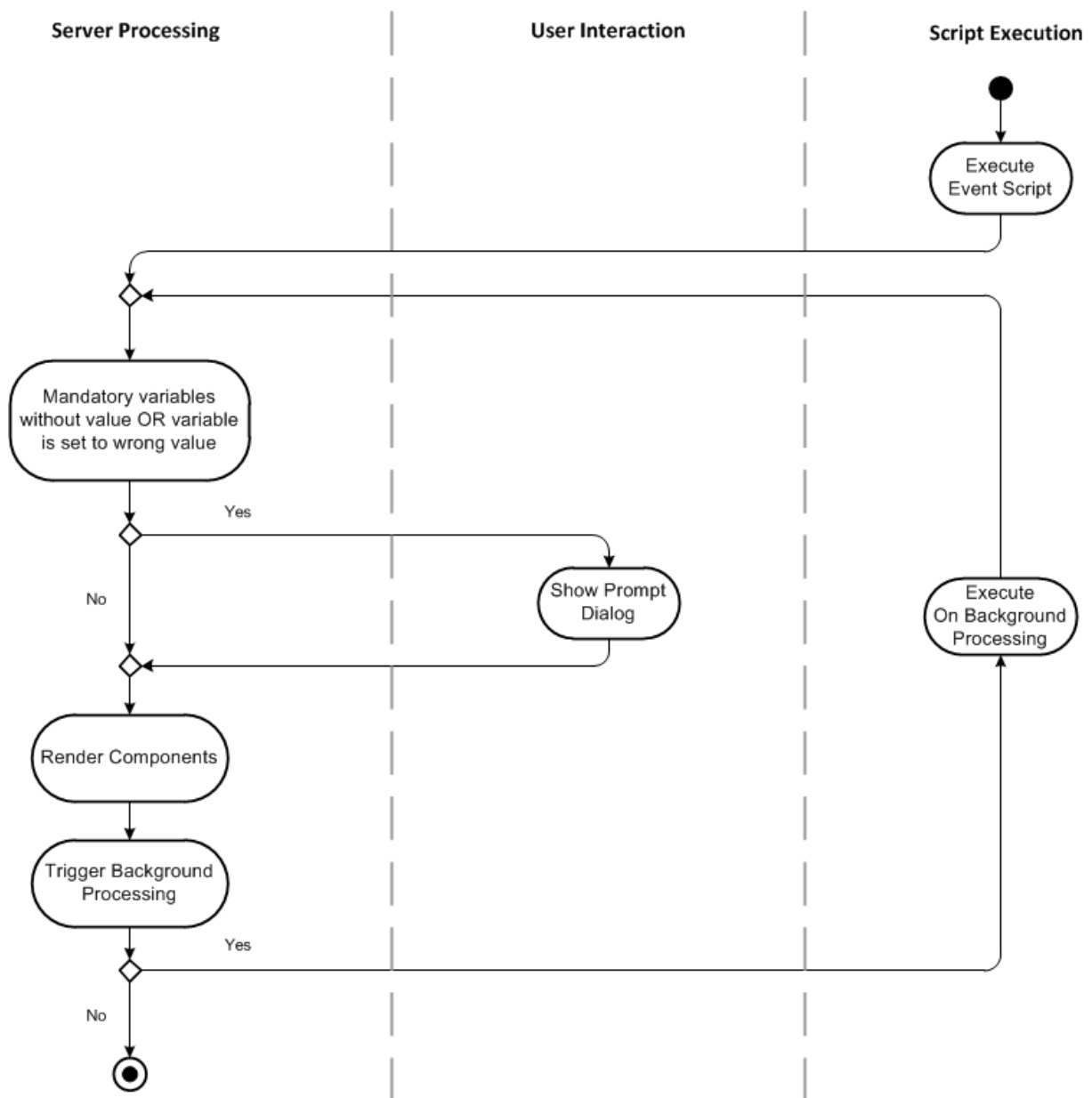
To expand or collapse the nodes and the entire hierarchy, press [Expand All](#) (+) or [Collapse All](#) (-).

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.

47.3 Script Processing Flow in Applications

The following graphic describes the processing flow for scripts that you create and insert into your analysis application.

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.

i 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

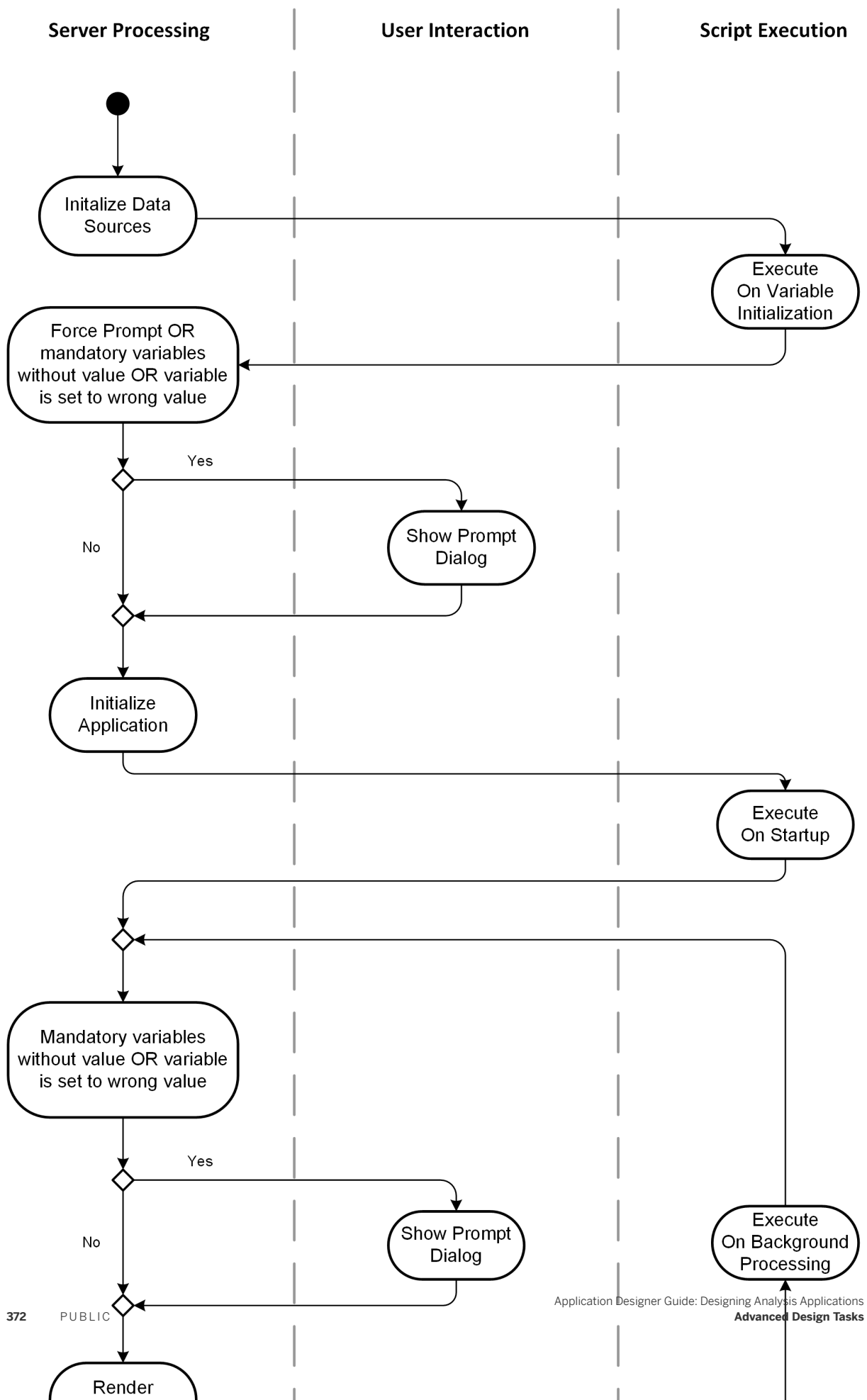
sources - one after the other. However, to avoid an endless loop, you must integrate a condition (in your script) that terminates the loop. For example, you can specify the number of calls by using a script like this: In this case, a global variable counter has been defined and initialized with 0. After 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 \[page 419\]](#)

```
counter = counter + 1;  
if (counter < 3 ) {  
    APPLICATION.doBackgroundProcessing();  
}
```

47.4 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.

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.

47.5 Business Cases

47.5.1 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:
`DS_2.setFilter("country" ,`

```
MAIN_CROSSTAB.getSelectedMember( "country" );  
TABSTRIP_1.setSelectedTabIndex(1);
```

47.5.2 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`.

i 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`:

```
TOOLBAR_BACKGROUND_2.setVisible(true);  
TOOLBAR_LAYOUT.setVisible(true);  
BUTTON_FILTEROFF.setVisible(true);  
BUTTON_FILTERON.setVisible(false);
```

For `BUTTONFILTER_OFF`:

```
TOOLBAR_BACKGROUND_2.setVisible(false);
```

```
TOOLBAR_LAYOUT.setVisible(false);  
BUTTON_FILTERON.setVisible(true);  
BUTTON_FILTEROFF.setVisible(false);
```

47.5.3 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...](#) [Page 4](#)). 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();
}
```

47.5.4 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):

BW query with a structure in the rows and the columns

Structure in the Rows		Structure in the Columns		
		Actual	Plan	Forecast
BW Key Figure 1 (Measure 1 = Sales)	Hierarchy Node 1 (= World)	Cell 1		
	Hierarchy Node 2 (= Europe)	Cell 2		
	Hierarchy Node 3 (= DE)	Cell 3		
BW Key Figure 2 (Measure 2 = Costs)	Hierarchy Node 1			
	Hierarchy Node 2			
	Hierarchy Node 3			

i Note

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):

```
DS_3.getDataAsString("DCNFKHLVQTUD5AIFYGF4HQF4J",
    {"DCNFKHLVQVGT4FC4BHDPM8JRN":"DCNFKHLVQSUZORHJ7MGXE26MR"});
```

In this business case a specific structure element is selected within a structure in the columns with "DCNFKHLVQVGT4FC4BHDPM8JRN=DCNFKHLVQSUZORHJ7MGXE26MR": "DCNFKHLVQSU ZORHJ7MGXE26MR" (for example, Actual). As no further specifications are made, the first cell (Cell 1) is read out.

Annotations:

- DCNFKHLVQTUD5AIFYGF4HQF4J is the UID (unique ID) of the key figure *Sales* in the BW key figure structure in the rows.
- DCNFKHLVQVGT4FC4BHDPM8JRN 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)

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):

```
DS_3.getDataAsString("DCNFKHLVQTUD5AIFYGF4HQF4J",  
{ "DCNFKHLVQVGT4FC4BHDPM8JRN": "DCNFKHLVQSUZORHJ7MGXE26MR" } );
```

In this business case a specific structure element is selected within a structure in the columns with "DCNFKHLVQVGT4FC4BHDPM8JRN:DCNFKHLVQSUZORHJ7MGXE26MR" (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:

- DCNFKHLVQTUD5AIFYGF4HQF4J is the UID (unique ID) of the key figure *Sales* in the BW key figure structure in the rows.
- DCNFKHLVQVGT4FC4BHDPM8JRN 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)
- "OPROFIT_CTR": "HIERARCHY_NODE/OHIER_NODE/EUROPE" filters the hierarchy node "EUROPE" (which represents a text node in 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:

```
DS_3.getDataAsString("DCNFKHLVQTUD5AIFYGF4HQF4J",  
{ "DCNFKHLVQVGT4FC4BHDPM8JRN": "DCNFKHLVQSUZORHJ7MGXE26MR", "OPROFIT_CTR": "DE" } );
```

In this business case, a specific structure element is selected within a structure in the columns with "DCNFKHLVQVGT4FC4BHDPM8JRN" : "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:

- DCNFKHLVQTUD5AIFYGF4HQP4J is the UID (unique ID) of the key figure [Sales](#) in the BW key figure structure in the rows.
- DCNFKHLVQVGT4FC4BHDPM8JRN 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)
- OPROFIT_CTR__ZKBUSAREA=LS is the technical name of the characteristic value to be filtered (for example, DE, which is part of the hierarchy)

47.5.5 Using the Browser Close Warning Message

You can display a warning message when the application user quits the application.

If the application user closes the browser window or a tab, you can display a warning message by using the `APPLICATION.showCloseWarning` API.

Selecting `true` will display the warning if the user closes the Web browser. Selecting `false` turns the warning off. In combination with `true` you can also pass a custom message. Some browsers (like Microsoft Internet Explorer) show the custom text. Many newer browsers like Google Chrome only show a standard message.

Usage for Planning Applications

For planning applications you should add the following snippet to the [On Resultset Changed](#) event of the planning data source(s):

```
if (Planning.hasClientChanges() || Planning.hasUnsavedChanges()) {  
    APPLICATION.showCloseWarning(true, "Unsaved Planning Changes");  
} else {  
    APPLICATION.showCloseWarning(false);  
}
```

This will show the warning message as long as there are unsaved changes to the data source and turn it off when the application user saves the changes

47.6 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.

i 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:

```
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:

```
if (enabled) {  
    button.setCSSClass("enabledCSSClass");  
} else {  
    button.setCSSClass("disabledCSSClass");  
}
```

47.7 Enabling Text Translation in Analysis Applications

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.

Context

When using BI platform mode, you can provide your analysis applications in different languages, and configure the analysis applications accordingly.

i Note

This is not supported for Lumira documents in Lumira Documents mode.

i 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:

Text Type	Example
Static text	Button label that stays constant during application lifetime
Dynamic text	Message text that a script joins together from multiple parts

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](#).
3. Choose [Text Pool](#).

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.

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 \[page 381\]](#)

[Working With Translatable Texts in Analysis Applications \[page 382\]](#)

47.7.1 Using Scripts for Translatable Dynamic Texts

Application designers can write a script that joins together translatable texts from multiple parts.

Context

You can use this function to create message texts, for example.

i Note

This is not supported for Lumira documents in Lumira Documents mode.

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:

```
var  
    translated = TEXT_POOL.Key1;  
    var placeholdersReplaced = Convert.formatString(translated,  
    [ "Replacement1" , "Replacement2" ] );  
    TEXT_1.setText (placeholdersReplaced);
```

47.7.2 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. The texts are stored separately for each analysis application.

i Note

This is not supported for Lumira documents in Lumira Documents mode.

i 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:

Text_Key	Text	Language
BUTTON_1.TEXT	Action	EN
BUTTON_1.TEXT	Aktion	DE

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:

Text_Key	Text	Language
BUTTON_2.TEXT	Action	EN

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.

When transporting analysis applications from one system to another, the texts are automatically transported together with the analysis application.

All the texts in an analysis applications text pool are picked up by the Translation Management Tool. For more information on working with the Translation Management Tool, you can refer to the guide called *Translation Management Tool User Guide* on the SAP Help Portal at <http://help.sap.com>.

i Note


- So that correct localization is picked up, the text pools which are present in analysis applications and are translated in the Translation Management Tool, require locale and country information. In the Translation Management Tool, you can select how text is translated based on language (country). For example, you can select German (Austria), or German (Germany).

❖ Example

With German (Austria) selected in the Translation Management Tool, you should open the document using an OpenDocument URL, with parameters set as "&language=de &country=at".
With German (Germany) selected in the Translation Management Tool, you should open the document using an OpenDocument URL, with parameters set as "&language=de& country=de".

If the country information is missing, then the localization will show only the source content.

- Before working with the Translation Manager Tool, you should refer to the following SAP Note:

SAP Note Number	Description
2666111 	Translated strings from Translation Manager are lost when you save as a document (LUMX) from Designer Client to Add-on.

47.8 Deploying SDK Extensions

In addition to the standard palette of components in SAP Lumira Designer, you can install 3rd party components developed with the Component SDK. Adding these 3rd party components, known as Component SDK extensions, to your SAP Lumira Designer installation enables you to create and execute local analysis applications containing these SDK extensions.

Furthermore, you can install new chart types developed with the Visualization SDK. These Visualization SDK extensions are added to the list of additional charts in the chart type picker. You can create and execute local analysis applications containing these new chart types.

Before application users can launch Lumira documents or analysis applications containing SDK extensions from the BI platform, your administrator needs to deploy these locally installed SDK extensions to the BI platform. For more information, see “Deploying SDK Extensions” in the corresponding *Administrator Guide: SAP Lumira* on SAP Help Portal at <http://help.sap.com>.

For more information about creating Component SDK extensions using the Component SDK, see the *Developer Guide: Component SDK* on the SAP Help Portal at <http://help.sap.com>.

For more information about creating Visualization SDK extensions using the Visualization SDK, see the *Visualization Extension Plugin for SAP Web IDE Guide* on SAP Help Portal at <http://help.sap.com>.

Related Information

[Chart Type Picker \[page 458\]](#)

47.8.1 Installing Component SDK Extensions to SAP Lumira Designer

Context

You can add extensions developed with the Component SDK to your SAP Lumira Designer installation as new components.

Procedure

1. In SAP Lumira Designer, choose **Tools** > *Install Extension to Lumira Designer...*.
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 Component SDK 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 Lumira Designer to restart.

Results

The SDK extension components appear in the [Components](#) view of SAP Lumira Designer as new components.

The components are stored under `<user home directory>\LumiraDesigner-config\plugins` .

47.8.2 Installing Visualization SDK Extensions to SAP Lumira Designer

You can add Visualization SDK extensions, developed with the Visualization SDK, to your SAP Lumira Designer installation as new chart types available in the chart type picker.

Procedure

1. In SAP Lumira Designer, choose **Tools** > [Install Visualization SDK Extension...](#) .
2. Depending on where the Visualization SDK 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 Lumira Designer, choose [Yes](#).

Results

When you create or open a Lumira document or an analysis application with a chart type picker, the Visualization SDK extensions you have added are listed as additional chart types.

Next Steps

In the [Properties](#) view of the [Chart Type Picker](#) component, under [Display](#), choose [Additional Types...](#) The Visualization SDK extensions are listed as new chart types in the [Additional Charts](#) dialog box.

Related Information

[Chart Type Picker \[page 458\]](#)

47.8.3 Removing Extensions from SAP Lumira Designer

Context

You can remove SDK extensions that you have added to your SAP Lumira Designer installation as follows:

Procedure

1. In SAP Lumira Designer, 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 Lumira Designer to restart.

Results

The SDK extension components are removed from the *Components* view of SAP Lumira Designer. Visualization SDK extensions are removed from the list in the *Additional Charts* dialog box.

47.9 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

- “Using the Report-Report Interface” on SAP Help Portal at <http://help.sap.com>
- “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 Lumira Designer 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 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 Lumira Designer on BI Platform*.

47.10 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.

i 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 one local mode platform to another local mode platform.

- You can use the export for regular backups of your complex analysis applications.

i Note

Exporting analysis applications is available in BI platform mode and local mode.

Procedure

1. Choose ► *Application* ► *Export* ► in the design tool.
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.

i 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.

i 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 \[page 389\]](#)

47.11 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 Lumira Designer (former 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.

i Note

Importing analysis applications is possible in all modes (Lumira Documents mode, BI platform legacy mode 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.

i 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 one local mode platform to another local mode platform.

Procedure

1. If you work in Lumira Documents mode right-click the document of your choice and choose [Import Application...](#). If you work in BI platform legacy mode or local mode, choose ► [Documents](#) ► ► [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.

i Note

In local mode, the default option is to import the files (otherwise the application will be incomplete after the import), whereas in BI platform 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 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.

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.

47.12 Exporting Applications As Templates

Context

You can create analysis applications and provide them as templates for your fellow application designers.

i Note

Exporting analysis applications as templates is available in BI platform mode and local mode.

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.

i 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.

If you have not defined a template category, you need to jump to the [Preferences](#) page by clicking [Configure Categories...](#) Here you can add your template categories that indicate the target device types recommended for a specific template, for example.

9. Under [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 [Icon](#), you can define the icon that symbolizes your template in the [New Application](#) dialog box.

You can select the required icon in the dropdown box or choose [Browse...](#) to select a new icon from your file repository.

i Note

Images larger than 48x48 pixels will be scaled down when displayed in the [New Application](#) dialog box.

11. Under [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.

12. 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.

13. (optional) To access the created `content.biapp` file quickly, 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 Lumira Designer, and can also select the template you have exported.

Related Information

[Maintaining Settings in Lumira Designer \[page 34\]](#)

47.13 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.

i Note

When using the API, note that the Backend Connection Component relates to the API object type `CONNECTION`. For more information, see chapter “Connection” in the Component API reference.

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.

i 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.

The use of the backend connection component is not supported for Universe 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 ([Title](#) 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:

Data Source Browser dialog box for SAP HANA connections	Data Source Browser dialog box for SAP BW connections
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.	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

You can also set a configuration string to configure the dialog directly in the script, following a JSON syntax, for example `CONNECTION.showDataSourceBrowser({'title': 'JSON customized datasource browser', 'width': '90%', 'height': '90%', 'defaultTab': 'rolesTab', 'hiddenTabs': ['foldersTab', 'workspacesTab']})`;

Make sure the JSON syntax follows the example (including ' ') and that there are no line breaks in the JSON string.

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

```
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

```
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:

```
var ds = CONNECTION.getSelectedDataSource();
DS_2.assignDataSource(ds.connection, ds.type, ds.name, true);
```

i 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.

47.14 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 Lumira Designer 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)" in the *Component API Reference* on SAP Help Portal and [Binding the Properties of Standard Components to Data Sources \[page 145\]](#)

Prerequisites

When using parallel query execution, note the following points:

- Parallel query execution can only be used for a Lumira Designer 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.
 - with merged variables

Variables used in the application need to be unmerged in order to use parallel query execution. If the application is set to use merged variables, the system ignores the processing group definition and an error message is displayed. If the application needs both parallel query execution and variable merging, then new Lumira Designer script methods can be used, which can emulate variable merging behavior.

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:

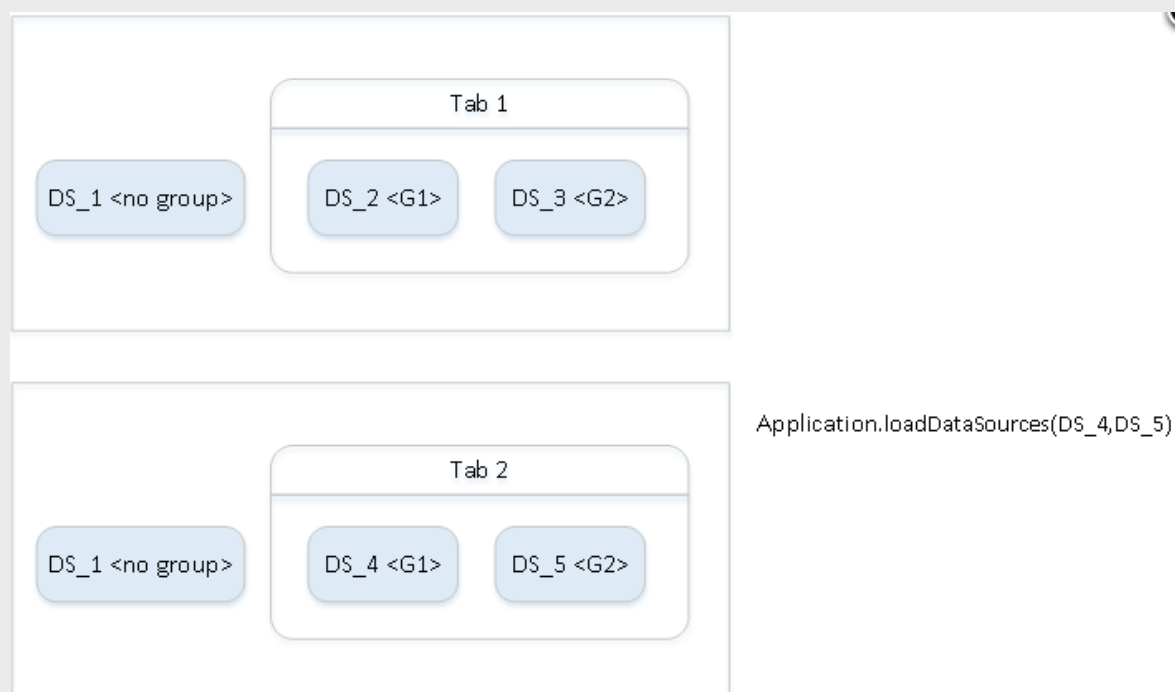
- In general, you should not execute queries with a very short runtime in parallel as the overhead might be bigger than the performance improvement achieved by parallelization.
- When designing applications with SAP BW data sources, be aware that using additional processing groups causes additional load 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.
- Each processing group represents a session in a data source system (SAP BW, SAP HANA). Sessions are a scarce resource in terms of scalability. Therefore we highly recommend to use a processing group for multiple data sources. This keeps the number of processing groups per application as low as possible. The number of processing groups per application should not exceed the single digit range.

For further information, see the respective blog at <http://scn.sap.com/community/businessobjects-design-studio/blog/2015/09/28/parallel-processing-and-scripting>.

❖ 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).

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.



47.15 Unmerging Prompts (Variables)

Prerequisites

When unmerging prompts, note the following points:

- Unmerging prompts works only for a Lumira Designer deployment on 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” in the SAP BW documentation on SAP Help Portal at <https://help.sap.com>.

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 Lumira Designer for (un)merging variables

In the Lumira Designer, 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,

```
DS_2.setVariableValueExt("OBC_HIER_MAND",  
DS_1.getVariableValueExt("OBC_HIER_MAND"));
```

```
DS_2.setVariableValueExt("OBC_HIER_MAND",  
DS_1.getVariableValueExt("OBC_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,

```
DS_2.setVariableValueExt("OBC_HIER_MAND",  
DS_1.getVariableValueExt("OBC_HIER_MAND"));  
DS_2.loadDataSource();
```

47.16 Working with Background Processing

As an application designer you can control the sequence and timing of data loading in your applications using the Background Processing concept.

SAP Lumira Designer provides a set of data-bound components. The data for these components does not necessarily have to be loaded when the component becomes visible (for example, at application startup, when tabs of a tab strip are switched, and so on). You can control the sequence and timing of data loading using the Background Processing concept. This can be useful to make the application more responsive and improve the application's perceived performance.

Getting Started

To understand the application and the benefits of background processing, let's look at the following scenario: Your application has two crosstabs, two charts, and four data sources. Each data source is assigned to one of the components. The following table shows the components and their data sources:

Component	Data Source
CROSSTAB_DS_1	DS_1
CHART_DS_2	DS_2
CROSSTAB_DS_3	DS_3
CHART_DS_4	DS_4

The startup time of this application is determined by the total time it takes to load the four data sources, which may be too long for the application user. There are several solutions for this scenario using the Background Processing concept:

- Using Simple Background Processing
- Using Background Processing with Multiple Processing Steps
- Using Background Processing Beyond Application Startup

These examples show you how to work with Background Processing. Your application appears more responsive and improves its perceived performance. This effect is achieved by a combined use of the [On Background Processing](#) script and the `doBackgroundProcessing()` method.

The examples show that Background Processing is not limited to a two-step action performed at one point in time in the [On Startup](#) script and at a single later point in time in the [On Background Processing](#) script of your application.

You can create a multi-step action, performed at one point in time in the [On Startup](#) script and at several later points in time in the [On Background Processing](#) script of your application. This is achieved by calling the `doBackgroundProcessing()` method again in the [On Background Processing](#) script.

You can certainly also call the `doBackgroundProcessing()` method not only in the [On Startup](#) script but in any other script. The example in “Using Background Processing Beyond Application Startup” shows this using the [On Select](#) script of a tab strip.

47.16.1 Using Simple Background Processing

Using simple background processing can improve the perceived performance of your application.

Procedure

1. Decide which data source should have its data displayed first.

Depending on the intent of your application, this could be the fastest of your data sources or the data source with the most relevant data for the application user. Let's assume this is data source DS_1, and that there are three other data sources: DS_2, DS_3, and DS_4.

2. Set the *Load in Script* property for data sources DS_2, DS_3, and DS_4 to *true*. This will skip loading the data sources on application startup and load the data sources later by an explicit script command.
3. Add the following commands to the script of your application's *On Startup* event:

```
// Show loading indicators (optional, but good practice)
CHART_DS_2.showLoadingState();
CROSSTAB_DS_3.showLoadingState();
CHART_DS_4.showLoadingState();
// Trigger background processing
APPLICATION.doBackgroundProcessing();
```

The important part of this script is the call of the `doBackgroundProcessing` method, which executes the application's *On Background Processing* script at a later point in time. The first three lines display an indicator that the respective components are being loaded. This is optional, but good practice, as this indicates to the application user that data is loaded and will be displayed soon.

At application startup, the first component, CROSSTAB_DS_1, already shows its data from data source DS_1. The remaining components do not show their data yet, but an indicator that they are being loaded.

After the application has been rendered, it executes its script of the *On Background Processing* event. This script is used to load the remaining data sources.

4. Add the following commands to your application's *On Background Processing* script:

```
// Load data sources
DS_2.loadDataSource();
DS_3.loadDataSource();
DS_4.loadDataSource();
// Hide loading indicators (optional, but good practice)
CHART_DS_2.hideLoadingState();
CROSSTAB_DS_3.hideLoadingState();
CHART_DS_4.hideLoadingState();
```

In this script the calls of the method `loadDataSource()` trigger the actual loading of each of the data sources. After they have been loaded, the components display the data of their data sources. The last three lines remove the loading indicator from the components. Again, this is optional, but good practice when you have shown loading indicators in the application's *On Startup* script.

Note that you are not limited to apply background processing strictly in the fashion demonstrated in the example above, that is, by picking a single data source that is loaded on startup and loading the remaining data sources later. Applying the concept in a broader sense, you can split your data sources in two sets:

- a first set which should be loaded on startup (often the fastest data sources or the data sources with the most relevant data to the application user, depending on your application's intent)
- a second set which should be loaded later.

47.16.2 Using Background Processing with Multiple Processing Steps

To improve your application's perceived performance of data loading even more, you can load the data sources not only in two distinct sets but in multiple sets, or, in other words, in multiple processing steps. This makes the data appear in the components one after the other.

Context

In this example the four data sources described in “Working with Background Processing” are loaded in four distinct steps. The basic approach is that the “On Background Processing” script is called not just once, but once for every processing step. In order to distinguish which step is currently being processed, we introduce a global variable indicating the current processing step.

Procedure

1. Add the global variable **currentBackgroundStep** of type String. The actual name of the variable is up to you as long as you use it consistently. This variable will contain values like **Load Crosstab DS_1** or **Load Chart DS_2**. This makes it easy for you to identify the current processing step. Another solution would be to use a global variable of type Integer, where the value of the Integer indicates the current processing step number. However, for better readability it is recommended to use the more descriptive approach with a String value.
2. Add the following commands to your application's *On Startup* script:

```
// Show loading indicators (optional, but good practice)
CHART_DS_2.showLoadingState();
CROSTAB_DS_3.showLoadingState();
CHART_DS_4.showLoadingState();
// Trigger next background processing step
currentBackgroundStep = "Load Chart DS_2";
APPLICATION.doBackgroundProcessing();
```

In this script, the first processing step is indicated by the value **Load Chart DS_2**. It is stored in the global script variable **currentBackgroundStep**. Then the `doBackgroundProcessing` method is called, which executes the application's *On Background Processing* script at a later point in time.

3. Add the following commands to your application's *On Background Processing* script:

```
if (currentBackgroundStep == "Load Chart DS_2") { // Block 1
    // Load data
    DS_2.loadDataSource();
```

```

// Hide loading indicator (optional, but good practice)
CHART_DS_2.hideLoadingState();
// Trigger next background processing step
currentBackgroundStep = "Load Crosstab DS_3";
APPLICATION.doBackgroundProcessing();
} else if (currentBackgroundStep == "Load Crosstab DS_3") { // Block 2
// Load data
DS_3.loadDataSource();
// Hide loading indicator (optional, but good practice)
CROSSTAB_DS_3.hideLoadingState();
// Trigger next background processing step
currentBackgroundStep = "Load Chart DS_4";
APPLICATION.doBackgroundProcessing();
} else if (currentBackgroundStep == "Load Chart DS_4") { // Block 3
// Load data
DS_4.loadDataSource();
// Hide loading indicator (optional, but good practice)
CHART_DS_4.hideLoadingState();
// Final step, clear processing step variable, do not call
doBackgroundProcessing()
currentBackgroundStep = "";
}

```

When this script is executed for the first time, the instructions in Block 1 are executed: First, data source DS_2 is loaded. This makes its data appear in CHART_DS_2. Then, the loading indicator in this chart is removed (this is optional, but good practice after having displayed the loading indicator). Then the value of the current background processing step is changed to Load Crosstab DS_3. Finally, the next background processing step is enabled by calling the `doBackgroundProcessing` method. After the display of the CHART_DS_2 is updated, the application's [On Background Processing](#) script is executed again with the updated value of `currentBackgroundStep`.

When this script is executed for the second time, the instructions in Block 2 are executed: First, data source DS_3 is loaded. This makes its data appear in CROSSTAB_DS_3. Then, the loading indicator in this crosstab is removed (again, this is optional, but good practice after having displayed the loading indicator). Then the value of the current background processing step is changed again to Load Chart DS_4. Finally, the next background processing step is enabled by calling the `doBackgroundProcessing` method. After the display of CROSSTAB_DS_3 is updated, the application's [On Background Processing](#) script is executed again with the updated value of `currentBackgroundStep`.

When this script is executed for the third time, the instructions in Block 3 are executed: First, the data source DS_4 is loaded. This makes its data appear in CHART_DS_4. Then the loading indicator in this chart is removed (again, this is optional, but good practice after having displayed the loading indicator). Further, optionally, the value of the current background processing step is changed to an empty value. This time the `doBackgroundProcessing` method is not called. This ends the execution of this script, the loading of the data sources, and the Background Processing.

When you execute this application, first the data of CROSSTAB_DS_1 appears, then, with each background processing step, the data of the next component is displayed. This gives the application the impression of being even more responsive and improves again the application's perceived performance.

47.16.3 Using Background Processing Beyond Application Startup

Loading data sources with Background Processing is not limited to be used application startup only. It can also be used to load data sources both at application startup and when switching tabs of a tab strip.

Context

The following scenario is used for this example: Your application has a tab strip component with two tabs. In each tab there are two crosstabs and one chart. The application has six data sources. Each data source is assigned to one of the crosstab and chart components. The following table shows the tabs, their components, and their data sources:

Tab	Component	Data Source
TAB_1	CROSSTAB_DS_1	DS_1
TAB_1	CHART_DS_2	DS_2
TAB_1	CROSSTAB_DS_3	DS_3
TAB_2	CROSSTAB_DS_4	DS_4
TAB_2	CHART_DS_5	DS_5
TAB_2	CROSSTAB_DS_6	DS_6

The application loads data source DS_1 immediately at startup, and DS_2 and DS_3 using Background Processing in processing steps triggered by the application startup. It immediately loads data source DS_4 when the tab strip is switched to TAB_2, and loads data sources DS_5 and DS_6 using Background Processing in processing steps triggered by switching to TAB_2.

Procedure

1. Set the *Load in Script* property of DS_1 to *false* and set the *Load in Script* properties of the remaining data sources to *true*.

Data source DS_1, which is assigned to CROSSTAB_DS_1, is loaded immediately at application startup. The other data sources are loaded later.

2. Add a global script variable **currentBackgroundStep** of type String to your application.
3. Add the following commands to your application's *On Startup* script:

```
// Show loading indicators (optional, but good practice)
CHART_DS_2.showLoadingState();
CROSSTAB_DS_3.showLoadingState();
// Trigger next background processing step
```

```
currentBackgroundStep = "Load Chart DS_2";
APPLICATION.doBackgroundProcessing();
```

With this application, data source DS_1 is loaded immediately at application startup. Its data is displayed in CROSSTAB_DS_1. Then, the application's *On Startup* script is executed. This script first shows the loading indicators for the two remaining components on TAB_1, CHART_DS_2 and CHROSSTAB_DS_3. Again, this is optional, but good practice. Then, the value of the current background processing step is changed to Load Chart DS_2. Finally, the next background processing step is enabled by calling the doBackgroundProcessing method. After the display of the CHART_DS_2 is updated, the application's *On Background Processing* script is executed with the updated value of currentBackgroundStep.

4. Add the following commands to your application's *On Background Processing* script:

```
if (currentBackgroundStep == "Load DS_2") { // Block 1
    // Load data
    DS_2.loadDataSource();
    // Hide loading indicator (optional, but good practice)
    CHART_DS_2.hideLoadingState();
    // Trigger next background processing step
    currentBackgroundStep = "Load DS_3";
    APPLICATION.doBackgroundProcessing();
} else if (currentBackgroundStep == "Load DS_3") { // Block 2
    // Load data
    DS_3.loadDataSource();
    // Hide loading indicator (optional, but good practice)
    CROSSTAB_DS_3.hideLoadingState();
    // Final step, clear processing step variable, do not call
    doBackgroundProcessing()
    currentBackgroundStep = "";
}
```

This script uses Background Processing to load data sources DS_2 and DS_3.

5. Add the following commands to the tab strip's *On Select* script of your application:

```
if ((DS_4.isInitialized() == false) && (TABSTRIP_1.getSelectedTab() ==
"TAB_2")) {
    // Load data
    DS_4.loadDataSource();
    // Show loading indicators (optional, but good practice)
    CHART_DS_5.showLoadingState();
    CROSSTAB_DS_6.showLoadingState();
    // Trigger next background processing step
    currentBackgroundStep = "Load DS_5";
    APPLICATION.doBackgroundProcessing();
}
```

If TAB_2 was selected by the application user for the first time (we check this if data source DS_4 has been already initialized), then this script is executed. First, it loads data source DS_4 immediately. Its data is displayed in CROSSTAB_DS_4. Then, the script displays the loading indicators for the two remaining components on TAB_2, CHART_DS_5 and CROSSTAB_DS_6. Again, this is optional, but good practice. Then the value of the current background processing step is changed to Load Chart DS_5. Finally, the next background processing step is enabled by calling the doBackgroundProcessing method. After the display of the CROSSTAB_DS_4 is updated, the application's *On Background Processing* script is executed with the updated value of currentBackgroundStep.

6. Replace your application's *On Background Processing* script with the following commands:

```
if (currentBackgroundStep == "Load DS_2") { // Block 1
    // Load data
    DS_2.loadDataSource();
    // Hide loading indicator (optional, but good practice)
```

```

    CHART_DS_2.hideLoadingState();
    // Trigger next background processing step
    currentBackgroundStep = "Load DS_3";
    APPLICATION.doBackgroundProcessing();
} else if (currentBackgroundStep == "Load DS_3") { // Block 2
    // Load data
    DS_3.loadDataSource();
    // Hide loading indicator (optional, but good practice)
    CROSSTAB_DS_3.hideLoadingState();
    // Final step, clear processing step variable, do not call
doBackgroundProcessing()
    currentBackgroundStep = "";
} else if (currentBackgroundStep == "Load DS_5") { // Block 3
    // Load data
    DS_5.loadDataSource();
    // Hide loading indicator (optional, but good practice)
    CHART_DS_5.hideLoadingState();
    // Trigger next background processing step
    currentBackgroundStep = "Load DS_6";
    APPLICATION.doBackgroundProcessing();
} else if (currentBackgroundStep == "Load DS_6") { // Block 4
    // Load data
    DS_6.loadDataSource();
    // Hide loading indicator (optional, but good practice)
    CROSSTAB_DS_6.hideLoadingState();
    // Final step, clear processing step variable, do not call
doBackgroundProcessing()
    currentBackgroundStep = "";
}

```

The added Blocks 3 and 4 implement the loading of data sources DS_5 and DS_6 respectively, using Background Processing. They have the same structure as Blocks 1 and 2.

This shows that the application of Background Processing is independent of how Background Processing was triggered (at application startup, by switching tabs, and so on). Background Processing is not limited to use at application startup only!

When you start the application, first, the tab strip is displayed, initially showing TAB_1. In the tab, CROSSTAB_DS_1 is displayed with its data loaded from data source DS_1, while CHART_DS_2 and CHART_DS_3 are displayed without data. Then, using Background Processing, data sources DS_2 and DS_3 are loaded one after the other. The chart CHART_DS_2 and the crosstab CROSSTAB_DS_3 each update their display as soon as their data source has been loaded during Background Processing.

When you switch the tab strip to TAB_2, then CROSSTAB_DS_4, CHART_DS_5, and CHART_DS_6 are displayed. Then the [On Select](#) script of the tab strip is executed. It first loads data source DS_4, which is displayed by CROSSTAB_DS_4. Then, using Background Processing, data sources DS_5 and DS_6 are loaded one after the other. The chart CHART_DS_5 and the crosstab CROSSTAB_DS_6 each update their display as soon as their data source has been loaded during Background Processing.

47.17 Using Roundtrip Optimization

You can use roundtrip optimization to improve the performance of analysis applications, especially if the application is running over an internet connection with high latency.

In previous Lumira releases, some actions the user performed (such as clicks) led to roundtrips from Web browser to server - even if this was not really needed.. These roundtrips were to synchronize a state change to the server that would be needed later for a different user interaction. However, too many server roundtrips

reduce application performance. To optimize roundtrip performance, the number of server roundtrips is reduced by queuing (holding back) changes caused by clicking in the browser. These changes are only sent to the server in a roundtrip when it's necessary to keep the application's state on the server consistent.

Example: Input Field with a Button

In this example, there is an application with an input field `INPUTFIELD_1` and a button `BUTTON_1` with the following `ON_CLICK` event handler:

```
var text = INPUTFIELD_1.getValue();
APPLICATION.alert("Text was " + text);
```

In previous releases, that current value of `INPUTFIELD` was sent on each change (to be more precise, when `INPUTFIELD_1` lost focus). With roundtrip optimization, the value is sent when the application user clicks on `BUTTON_1`. There is no visible change for the application user.

Example: Input Field without a Button

In this application no button has been added. Instead the `ON_CHANGE` event of the inputfield `INPUTFIELD_1` is used with the following script:

```
var text = me.getValue();
APPLICATION.alert("Text was " + text);
```

In this case the value is sent and the script is executed in the same roundtrip. Again, there is no visible change for the user.

Technical Details

Each user interaction with a component is classified in the following groups:

- immediate
Since the user expects a result for his interaction, an immediate roundtrip is started.
- queued
The user interaction is queued and sent with the next input of type immediate.
- queued only one
If the user interacts the same way for multiple times (like changing a text several times) only the last interaction is queued.

There is also the case in which the application user clicks repeatedly very fast, and the roundtrip for one click is not completed yet before the following click. With roundtrip optimization, all queued commands are sent in a single roundtrip, before all queued commands have been replayed one by one.

What You Should Know As...

The following sections explain what different user roles need to know about roundtrip optimization.

- **Administrator**
As an administrator, you can turn off roundtrip optimization completely or set it to a certain level if you face problems with it. For more information, see chapter “Configuring Roundtrip Optimization for Analysis Applications on the BI Platform” in the *Administrator Guide: SAP Lumira* on the SAP Help Portal at <http://help.sap.com>
- **Application Designer**
As an application designer, you can turn off roundtrip optimization completely or set it to a certain level if you face problems with it in one specific application. In the startup script of the application, you use the same levels as the administrators, but the setting affects only the current application:

```
APPLICATION.setInternalValue("ROUNDTRIP_OPTIMIZATION", "1");
```

For more information about the levels to control roundtrip optimization, see [Configuring Roundtrip Optimization for Analysis Applications in Lumira Designer \[page 409\]](#).

i Note

There is a typical case when roundtrip optimization can cause a problem. Assume you're using the application from the example above, with an input field and a button component. But in this case you have activated undo/redo functionality with the methods `State.backOneStep()` and `State.forwardOneStep()`. When you use roundtrip optimization, the text changes are not sent back and thus are not recorded on the undo stack. As a workaround, just ensure that the `ON_CHANGE` event of `INPUT_FIELD1` is not empty. You could enter a script line like this:

```
// This script ensures that ON_CLICK causes a round trip from Browser to server
```

- **SDK Component Developer**
As an SDK Component developer, you can control whether a round trip is triggered or not. For more information, see “Roundtrip Optimization” in *Developer Guide: Component SDK* on the SAP Help Portal at <http://help.sap.com>.

Related Information

[Configuring Roundtrip Optimization for Analysis Applications in Lumira Designer \[page 409\]](#)

47.17.1 Configuring Roundtrip Optimization for Analysis Applications in Lumira Designer

You can configure roundtrip optimization for all analysis applications in Lumira Designer.

Context

To improve the performance of analysis applications, user actions that lead to a roundtrip from the Web browser to the server and are not needed instantly are queued on the Web browser until there is a user interaction in the application that triggers a command that is really useful for the application user.

For more information about roundtrip optimization, see “Using Roundtrip Optimization” in the *Application Designer Guide: Designing Analysis Applications* on SAP Help Portal at <http://help.sap.com>.

You can influence roundtrip optimization for all analysis applications on the BI platform server. You turn it off or set it to a certain degree by specifying the required value with a Java VM argument on the BI platform server. Application designers may want to configure the roundtrips between the Web browser and the embedded Web server of Lumira Designer and set roundtrip optimization to the same value. This allows them to simulate the behavior of the applications when executing them locally in Lumira Designer.

You can influence roundtrip optimization for all analysis applications in Lumira Designer by setting a Java VM argument in the `SapLumiraDesigner.ini` file called `ROUNTRIP_OPTIMIZATION` to one of the following values:

Value	Description
0	Roundtrip optimization is turned off. All analysis applications are processed as in Lumira < 2.2 releases.
1	The system only optimizes situations where the user clicks quickly before the busy indicator appears.
2	The system queues user inputs until the user performs an input that requires an immediate roundtrip as the user expects a result.
3	This is the default value. Full optimization, including the avoidance of replaying redundant commands on server side.

Procedure

1. Go to the directory on your machine where you installed Lumira Designer.
2. Open the `SapLumiraDesigner.ini` file.

3. Enter the following JAVA VM argument with the required value at the end: –
`DROUNDTRIP_OPTIMIZATION=<value>`

❖ Example

```
-DROUNDTRIP_OPTIMIZATION=0
```

4. Choose **File** > **Save**.
5. Restart Lumira Designer.

47.18 Optimizing Application Startup

There are two ways to speed up application startup.

You can use the following features to make application startup faster:

- Use optimized components that load JavaScript and CSS Files only when needed.
- Use the application property *Fast Startup Enabled*.

Related Information

[Using Optimized Components \[page 410\]](#)

[Using the Fast Startup Application Property \[page 411\]](#)

47.18.1 Using Optimized Components

Some components have been optimized for application startup.

For a faster application startup, these components load all necessary JavaScript code and CSS files only when the component first shows up on the screen. If, for example, you have a Chart in your application with the property *Visible* set to *false*, the JavaScript code is not loaded until a script changes the property value of *Visible* to *true*. This also applies if, for example, it is on the second page of a Pagebook or Tabstrip container component.

The following components are optimized for application startup:

- Chart (Vizfragem)
- Feeding Panel
- Chart Type Picker
- Map
- Map Builder
- Map Property Editor
- PDF Export

- Document
- Icon
- Formatted Text View
- Text Area
- Segmented Button
- Scorecard

SDK components are normally also automatically optimized, unless you disable this function. For more information see attribute `loadIncludesOnlyIfVisible` in element “Component” of chapter “Elements of the Contribution XML Schema Definition File” in the “Developer Guide: Component SDK” on the SAP Help Portal at <http://help.sap.com>.

47.18.2 Using the Fast Startup Application Property

You can use the application property *Fast Startup Enabled* to optimize application startup.

When you set the property to true, the following actions take place in the background:

- On the server side, data sources are initialized and data is retrieved, for example from BW or SAP HANA.
- On the Web browser side, JavaScript code and CSS files are loaded and executed.

When using this property, you will see the progress indicator. In the background, the application starts rendering without data. Once the data is available from the back end, the charts and tables are updated.

You don't need any extra script code to use this property. Some events are delayed automatically. With *Fast Startup Enabled* set to *true*, the startup order is as follows:

1. The page is loaded and the progress indicator is shown.
2. JavaScript code and CSS files are loaded on the Web browser side.
3. In parallel, all data sources are initialized and loaded.
4. Waiting for all data to be available.
5. The On Variable Initialization event is executed.
6. The prompt dialog box is displayed if needed.
7. The On Before Prompt Submit event is executed.
8. The On Startup event is executed.
9. Data is displayed in all data-bound components.

48 Troubleshooting

48.1 Auto-Recovering Applications

When you create an application in the Lumira Designer, 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.

48.2 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 Lumira Designer. 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.

48.3 Setting Network Connections If Logon Problems Occur

Context

After installing or deinstalling SDK extensions and restarting Lumira Designer, you might not be able to log on to your BI platform. The *Authentication* dropdown box in the logon dialog box for Lumira Designer 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, Lumira Designer 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 Lumira Designer locally by pressing *Work Offline* in the logon dialog box.
2. Navigate to **Tools** > *Preferences* > *Application Design* > *Network Connections*.
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 Lumira Designer.

Results

This procedure only needs to be performed once.

48.4 Activating Runtime Traces

Context

You can record traces to analyze problems in Lumira Designer.

Procedure

1. In Lumira Designer, 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>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen\logs\RSTT` folder.

The design tool also connects to the SAP 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 BW trace tool environment (transaction code RSTT in the connected BW system) is activated.

For information on how to replay and maintain the traces, see "Trace Tool Environment" in the *SAP NetWeaver Library*, at <https://help.sap.com>.

The recorded traces help us to resolve problems that you report to SAP as customer messages.

48.5 Activating SAP JCo Traces

Context

You can record SAP JCo traces to analyze problems in Lumira Designer.

Procedure

1. In Lumira Designer, choose ► *Tools* ► *Preferences* ► *Application Design* ► *Support Settings* ►.
2. Select the *Activate SAP JCo Trace* checkbox.
3. Click *OK*.
4. Restart Lumira Designer 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 \[page 416\]](#)

48.6 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 Lumira Designer, go to ► *Tools* ► *Preferences* ► *Application Design* ► *Support Settings* ► and choose *Activate Runtime Profiling*.
2. Execute the required analysis application in your preferred platform mode.
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 [Runtime 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 BW or SAP HANA) and on the BI platform.
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 SAP 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.

48.7 Collecting Support Information

Context

If you encounter problems in Lumira Designer, you can collect the relevant information to send to SAP in a .zip file.

Procedure

1. In Lumira Designer, choose  [Help](#)  [Collect Support Information...](#) .
2. Read and accept the terms of the legal agreement and click [Next](#).
3. Select the target folder for the .zip file and click [Next](#).
4. Check the screenshot for confidential information and decide if you want to add the screenshot to the .zip file by clicking [Yes](#) or [No](#).

5. (Optional) If you want to add analysis applications, click [Next](#) and select the required analysis applications.

i Note

In Lumira Documents mode, you can only add one analysis application. In BI Platform mode and local mode, you can add several analysis applications to the zip file.

6. (Optional) Click [Next](#) and decide which log files you want to add.
7. Click [Finish](#).
The support information is saved in the file `DS_Support.zip`.
8. To view the content of the `.zip` file, click [View File Contents](#).

Results

You can attach the `.zip` file to a customer message and send it to SAP.

48.8 Managing Logs in Lumira Designer

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 **Error Log** view of Lumira Designer. 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.

i Note

With regard to log files, it is recommended you also refer to the chapter in this guide called “General Security Recommendations”.

48.9 Best Practices

48.9.1 Improving Performance of Analysis Applications

As the topic **Improving Performance of Analysis Applications** is subject of regular and frequent changes, we recommend to read carefully the information in the SAP Note (Performance Hints for Design Studio Applications) [1931691](#) .

49 User Interface Reference

49.1 Properties of the Application

Applications have general properties and properties for behavior, display, prompts, planning, and events.

General Properties

Property	Property Value	Property Description
name	NameOfYourApplication	Displays the application name that you entered when creating the application.
type	NameOfObject	Specifies the object type that the properties belong to. For an application, the type is Analysis Application .
description	DescriptionOfYourApplication	Displays the application description that you entered when creating the application. If you did not enter a description, you can do so now.
content version	number	Displays the version number of the application.
folder	YourFolderOnTheBIPlatform	Specifies the folder on the BI platform where the application is stored. This is only relevant for deployment on the BI platform and if you work with Remote Documents . It is not relevant for the Lumira Documents mode).
created by	UserNameForOperatingSystem	Displays the user name in the operating system.
creation time	none	Displays the time stamp when the application was created.

Property	Property Value	Property Description
last modified by	none	Displays the user who made the last change.
<div> i Note <p>This property is not available if your Lumira Designer is deployed on the BI platform.</p> </div>		
last modification time	none	Displays the time stamp of the last change.

Behavior Properties

Property	Property Value	Property Description
fast startup enabled	false (default), true	<p>With this property you can optimize and speed up the application startup.</p> <p>If you set the property to true, the following actions take place in the background:</p> <ul style="list-style-type: none"> On the server side, data sources are initialized and data is retrieved, like from BW or SAP HANA. On the Web browser side, Java-Script code and CSS files are loaded and executed. <p>For more information, see Using the Fast Startup Application Property [page 411]</p>
maximum number of steps	integer	Specifies the number of state changes backwards or forwards permitted within an application. Setting the property to a value more than zero enables the undo and redo functionality. The default value is zero or disabled. The maximum number of steps is 50.

Property	Property Value	Property Description
drag and drop between components	true, false	Specifies if drag and drop operations between different components are allowed or not. This property is set to <i>false</i> by default, meaning that drag and drop operations cannot be carried out between components. Drag and drop operations within one single component are still possible. Set this property to <i>true</i> if you want to enable drag and drop operations between components (for example, between the Navigation Panel and the Crosstab). For Drag&Drop operations between the crosstab and other components, the crosstab property <i>Drag and Drop Enabled</i> has also to be set to <i>true</i> . For further information about drag and drop, see Crosstab [page 435] and Working with Drag and Drop in Applications and Crosstabs [page 257] .
disable browser context menu	true, false	Specifies if the browser context menu is globally disabled for the application.

Display Properties

Property	Property Value	Property Description
SAPUI5 m mode	true, false	<p>Specifies the rendering mode for the application and its components. This property is read-only and just for information purposes. You set the rendering mode in the <i>New Application</i> dialog box when creating the application.</p> <p>Lumira Designer 1.6 now also supports the so-called Main part of the SAPUI5 library (sometimes just referred as "m", these are controls in name space sap.m of SAPUI5). The Main (m) part was developed with the mobile use case in focus; these components are therefore specialized for mobile devices. For further information, see</p>

Property	Property Value	Property Description
		<p>The SAPUI5 m Library as Rendering Mode [page 45].</p> <p>However, the usage of the SAPUI5 m library is not restricted to mobile scenarios; it also supports desktop applications. In order to adjust the visualization accordingly there are two form factors for the m mode: the compact form factor for the desktop and the cozy form factor with more spacing and padding for mobile use cases. For further information, see the application property Compact Form Factor.</p>
browser title	text	<p>Specifies the browser tab name of the running application.</p> <p>If no value or only spaces are entered, the previous default caption is displayed in the browser. Otherwise the changed value (without spaces at the beginning and end) is used.</p>
theme	SAP Belize, SAP Blue Crystal, SAP Simplite, SAP High Contrast Black	<p>Specifies the theme of the application.</p> <p>The default theme for analysis applications is SAP Belize. The SAP High Contrast Black theme supports people with visual impairments.</p> <p>For interoperability scenarios with Lumira Discovery you can use the SAP Simplite theme.</p>
compact form factor	true, false	<p>This property is only relevant for applications and components that are rendered in the SAPUI5 m mode.</p> <p>Specifies the form factor of the SAP UI5 m rendering mode. If set to <i>true</i>, the components are rendered in desktop style of the SAPUI5 m mode. For further information, see The SAPUI5 m Library as Rendering Mode [page 45]</p>
custom CSS	none	<p>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</p>

Property	Property Value	Property Description
		<p>cascading style sheets (CSS) and Web design.</p> <div> ! Restriction <p>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.</p> </div>
position of message button	bottom right, bottom left, top right, top left	Specifies the position of the message button. The message button displays errors, warnings and information relating to the application at runtime.
loading indicator delay	default: 1000	Specifies the delay in milliseconds before the loading indicator is displayed.
displayed message types	none, errors, warnings and errors, all	Specifies the content in the message button.

Prompts Properties

Property	Value	Description
merge prompts	true, false	Specifies if the same prompts (variables) that are used in different 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 397].

Property	Value	Description
value help settings	technical name of Value Help Settings technical component	Specifies the technical component Value Help Settings which you can use to define the value help settings for dimensions.
force prompts on startup	true, false	Specifies whether the prompts dialog box is displayed at runtime as soon as the application is displayed.
prompt settings		Specifies which prompts should be displayed in which order.
maximum number of members	your number ; default: 100	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.
variant mode	no variants, query variants document variants	Specifies the variant mode.
compact prompts enabled	true, false	Enables the compact mode for prompts. This mode lets you make more efficient use of the space in the prompts dialog. The variable names and the corresponding input fields are arranged horizontally one behind the other in this mode.
design studio 1.x bookmark loading	hide and keep prompts, hide and clear prompts, show and keep prompts, show and clear prompts	Specifies the behavior when loading a Design Studio 1.x bookmark with prompts.

Planning Properties

Property	Value	Description
planning connection	none or listed planning system	Specifies the back end connection used for planning scenarios.

Property	Value	Description
planning model	none or listed planning models	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 Lumira Designer's planning functionality on the server to behave differently. For more information, see "SAP Business Planning and Consolidation, version for SAP NetWeaver" on SAP Help Portal at https://help.sap.com .

Events

Property	Value	Description
on variable initialization	none	<p>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:</p> <ul style="list-style-type: none">• before the event <i>On Startup</i>• before the first variable submit <p>Opens the script editor. With this property/event, you can enable user interaction with the application by writing scripts. Click <code>CTRL+Space</code> 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:</p> <ul style="list-style-type: none">• for the object APPLICATION<ul style="list-style-type: none">◦ <code>setVariableValue</code>◦ <code>setVariableValueExt</code>◦ <code>alert</code>◦ <code>createErrorMessage</code>◦ <code>createWarningMessage</code>• 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

Property	Value	Description
		<div> set. All variables set using event <i>On Variable Initialization</i> are set as default values in the variable screen. </div>
on startup	none	<p>Opens the script editor. With this property/event, you can enable user interaction with the application by writing scripts. Click <code>CTRL+Space</code> to see the list of available methods for the application, data source alias and the components. Choose one of these methods.</p> <p>For further information, see the method descriptions in the API reference.</p>

Property	Value	Description
on background processing	none	<p>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.</p> <div> <p>i Note</p> <ul style="list-style-type: none"> To trigger this event, you must use the script method <code>doBackgroundProcessing</code>. This method executes the contents of the event. The execution is run after the script (which calls <code>doBackgroundProcessing</code>) has finished and the result has been sent to the front end. </div> <p>Click <code>CTRL</code> + <code>Space</code> 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).</p> <p>For further information, see the method descriptions in the API reference.</p> <p>There are two main scenarios in which the <i>On Background Processing</i> might be especially useful:</p> <ul style="list-style-type: none"> guided navigation <p>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. <i>On Background Processing</i> can be used here to load all three possible data sources in the background, while the user is still look-</p>

Property	Value	Description
		<p>ing at the first data source to decide which path he/she will take.</p> <ul style="list-style-type: none"> • tile effect <p>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:</p> <pre> 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(); } </pre>
on before prompts submit		<p>You can use this event in combination with the Prompts Settings property. For more information, see chapter Unmerging Prompts (Variables) [page 397]</p>

49.2 Properties of the Data Source Alias

≡ Syntax

Data source aliases have the following properties:

Property	Property Value	Property Description
name	name of data source alias	Displays the name of the data source alias, for example, DS_1.
type	type of data source alias	Displays the type of the data source alias.
load in script	false, true	Specifies whether the data source alias is initialized in the script, instead of automatically loading the data source when the application is started.
data source: name	name of underlying data source	Displays the name of the underlying data source, for example, the name of query or query view.
data source: connection	name of connection	Displays the connection of the data source.
data source: type	type of data source	Displays the type of the data source, for example, query view.
processing group		<p>This property can only be used if the Lumira Designer runs in BI platform mode or local. Also this property cannot be used for planning enabled queries and for merged variables.</p> <p>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).</p> <p>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:</p> <ul style="list-style-type: none"> • For specifying a group at the beginning of design process, type in capital letters a group name, for example GROUP1. • The first typed in group name will be listed in the property if you want to set the property for another data source. • 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

Property	Property Value	Property Description
		<p>following configurations are behaving different:</p> <ul style="list-style-type: none"> Configuration 1: DS_1 (<no group>), DS_2 (GROUP) Configuration 2: DS_1 (GROUP1), DS_2 (GROUP2) <p>Configurations like (2) where all data sources are assigned to a non-default group are not recommend since it unnecessary allocates an additional new thread.</p> <p>For more information on limiting the number of sessions and threads, see "Configuring the Number of Sessions for Parallel Query Execution" in the <i>Administrator Guide: SAP Lumira Designer based on BI Platform</i></p> <p>For further background information on working the parallel query execution, see XXXX</p>
text	your Text	<p>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, <i>DS_1</i>. This property can be useful when unmerging variables. For further information, see Unmerging Prompts (Variables) [page 397].</p>
on result set changed	none	<p>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. For further information, see the method descriptions in the Component API reference.</p> <p>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:</p>

Property	Property Value	Property Description
		<ul style="list-style-type: none"> if a data source is initially assigned. if the query model was changed (for example, by using a filter command) during the roundtrip. if planning operations are performed (for example, when entering planning data or executing planning functions). <p>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.</p> <div> <p>i Note</p> <p>This event is intended for updating components after data changes. During this event, you should not change the data source itself, by setting filters, for example. Otherwise unpredictable behavior may happen or it may result in endless loops.</p> <p>As a simpler alternative to using the <i>on result set changed</i> property, consider using property-binding. For more information, see Binding the Properties of Standard Components to Data Sources [page 145].</p> </div>

49.3 General Properties for All Components

The following general properties are available for all components:

Property	Property Value	Property Description
name	String	Defines the unique name of a component. If you do not enter a name, the system takes the default name (for example CHART_1).

Property	Property Value	Property Description
type	String	Displays the type of the component, for example, crosstab, chart, checkbox.
visible	true / false	Specifies whether a component is visible or not.
enabled	true / false	Specifies whether a component is enabled. Disabled components do not allow user interaction.

i 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.

49.4 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:

Property	Property Value	Property Description
CSS class	Your CSS class	Defines an additional CSS class for custom CSS. The CSS classes must have the format myclass (and not .myclass). You do not have to use this property to be able to use Custom CSS .

49.5 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:

Property	Property Value	Property Description
Top Margin	Numeric value in pixels or auto	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 <code>auto</code> .
Left Margin	Numeric value in pixels or auto	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 <code>auto</code> .
Bottom Margin	Numeric value in pixels or auto	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 <code>auto</code> .
Right Margin	Numeric value in pixels or auto	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 <code>auto</code> .
Height	Numeric value in pixels or auto	Specifies the height of a component. Enter the numeric value in pixels, or set the value to <code>auto</code> .
Width	Numeric value in pixels or auto	Specifies the width of a component. Enter the numeric value in pixels, or set the value to <code>auto</code> .

❖ 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.

i 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`.

49.6 Table Components

49.6.1 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 without any data. As soon as you assign a data source to the crosstab, the data of the data source is displayed in the crosstab.

The [Crosstab](#) component has the following specific properties:

Data Binding Properties

Property	Property Value	Property Description
data source	name of data source alias (for example, "DS_1")	<p>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.</p> <div>i Note<p>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.</p></div>

Optimization for Low Data Volume Properties

Property	Property Value	Property Description
pixel-based scrolling	true, false	<p>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.</p> <div>i Note<p>This property is mandatory for setting the following properties of row and column limits:</p><ul style="list-style-type: none">For iPad applications, SAP recommends that you limit the total number of cells (the product of rows*columns) to 500, to improve performance and user experience.For desktop browser applications with pixel-based scrolling, the total number of cells can be set higher, depending on the performance of the client PC (recommended maximum number of cells 5000).</div>
row limit	numeric value for rows; default: 20 (max)	<p>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.</p>
column limit	numeric value for columns; default: 20 (max)	<p>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.</p>

User Interactivity Properties

Property	Property Value	Property Description
selection type	none (default), single row/column, multi row/column, single data cell, single cell	<p>Specifies if and which kind of data selection is offered for the application user at runtime:</p> <ul style="list-style-type: none"> If set to value <i>None</i>, selection is completely disabled. The user cannot select anything in the crosstab. There is no hovering effect even if the property <i>Enable Hover Effect</i> is set to <i>true</i>. If set to value <i>Single Row/Column</i>, the user can select rows and columns in the crosstab by hovering over and clicking the members of the required dimensions. <div> <p>i Note</p> <p>If you open an application with the formerly used property <i>Selection Enabled</i> set to <i>true</i> 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 <i>Enable Selection</i> property automatically transferred to <i>Enable Selection</i> with value <i>Single Row/Column</i>.</p> </div> <ul style="list-style-type: none"> If set to value <i>Multi Row/Column</i>, multiple selection is enabled. This means you can select multiple rows and columns by pressing Ctrl + Left mouse button on the required rows and columns. To remove a single selected row or column from multiple selected rows or columns, press Ctrl + Left mouse button on the selected row or column again. You can also achieve multi selection by choosing a member of an outer dimension with children. <div> <p>i Note</p> <ul style="list-style-type: none"> Multiple selection is not possible on mobile devices. It is not possible to specify multiple selection by clicking the left mouse key on a start row/column and then clicking Shift + Left mouse button on the end row/column. Each time the user performs an individual selection, an <i>On Select</i> event is triggered. </div>

Property	Property Value	Property Description
		<ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> If set to value <i>Single Data Cell</i>, 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>. If set to value <i>Single Cell</i>, the selection of a single dimension or measure is enabled. For data cells you can either choose the value <i>Single Cell</i> or <i>Single Data Cell</i> as the result is the same. The application designer can get the full selection with the API method <code>getSelection</code> or get the relevant members using <code>getSelectedMember</code>.
selectable area	all (default), rows, columns	<p>Specifies if the application user can select the rows or columns in the 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).</p> <p>i Note</p> <p>This property is only useful, if you have set the property <i>Enable Selection</i> to <i>Multi</i> or <i>Single</i>.</p>
enable hover effect	true (default), false	<p>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.</p> <p>i Note</p> <p>On mobile devices, there are no hover effects as there is no mouse pointer.</p>

Property	Property Value	Property Description
single on select event	false (default), true	<p>Specifies the behavior of the <i>On Select</i> event when multiple rows or columns in the crosstab are selected with the <code>CTRL</code> key pressed (also see property <i>Selection Type</i>, value <i>Multi</i>). When selecting multiple rows or columns in the crosstab with the <code>CTRL</code> key pressed, the <i>On Select</i> event will be fired on each single selection click, thus executing any script that is set for the On Select event. This may not always be desired. Instead, you might want to first finish the multi-selection and then have the On Select event executed only once. This parameter controls exactly this behavior.</p> <p>When set to <i>false</i> (default), each click with the <code>CTRL</code> key pressed will fire an On Select event. When set to <i>true</i>, each click with the CTRL key pressed will NOT fire an On Select event, but there will be only one single On Select event fired when the user releases the <code>CTRL</code> key. This means that the user can finish multi-selection before an event is fired.</p>
hierarchy navigation enabled	false, true	<p>Specifies whether the application user can expand or collapse hierarchy nodes, if the crosstab contains one or more hierarchies or hierarchical structures. If this property is set to <i>false</i>, 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.</p>
sorting enabled	true, false	<p>Specifies whether the end user can sort columns. If the property is set to <i>false</i>, the icons for sorting are not shown in the column headers.</p>
column resizing enabled	true, false	<p>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 <i>false</i>, the hover spot for the double-click resize will not be available.</p>

Property	Property Value	Property Description
horizontal header resizing enabled	true, false	<p>Specifies whether the application user can move the vertical border between the dimension area and measure area by dragging and dropping this border horizontally and thus altering the width ratio between dimension area and measure area. The default value is <i>false</i>.</p> <div> <p>Note</p> <p>If you have specified a maximum header width in pixels, the user cannot extend the width of the dimension area beyond this value.</p> </div> <p>For further information, see Using Header Scrolling in Crosstabs [page 339]</p>
horizontal scrolling for header enabled	true, false	<p>Specifies whether the header area will have a horizontal scrollbar. When set to <i>false</i> (default), the crosstab will not have a scrollable header area at runtime. Only if this property is set to <i>true</i>, the other properties relating to the header scrolling function in the crosstab (<i>Horizontal Header Resizing Enabled</i> and <i>Maximum Width of Header Area</i>) are taken into account to enable finer control of the way the dimension area behaves. For further information, see Using Header Scrolling in Crosstabs [page 339].</p>
context menu enabled	true, false	<p>Specifies whether a context menu can be displayed on the crosstab. The context menu can only be displayed on the crosstab, if the property is set to <i>true</i> and the technical component CONTEXT_MENU is available in the application (in the <i>Outline</i> view in the <i>Technical Components</i> folder). If the property is set to <i>false</i>, 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 Using the Context Menu (Technical Component) [page 86]</p>

Property	Property Value	Property Description
drag and drop enabled	true, false, advanced	<p>Specifies if the crosstab is drag and drop enabled. This property is set to <i>false</i> by default. If set to <i>true</i>, the crosstab</p> <ul style="list-style-type: none"> enables internal drag and drop operations. Thus the application user can drag and drop dimensions and members within the crosstab and remove dimensions and members by dragging and dropping them outside the area of the crosstab. accepts external drops of dimensions from other components (for example the navigation panel). This only works if the application property <i>Drag and Drop between Components</i> is set to <i>true</i> as well. If you set to <i>Advanced</i>, you get all the options that are offered when you set the property to <i>true</i>. Additionally, this enables to drag and drop dimensions outside the crosstab to data cells. <p>For further information about Drag and Drop, see Working with Drag and Drop in Applications and Crosstabs [page 257].</p>

Display Properties

Property	Property Value	Property Description
CSS class	Your CSS class	Defines an additional CSS class for custom CSS. The CSS classes must have the format myclass (and not .myclass). You do not have to use this property to be able to use <i>Custom CSS</i>
units and scaling factors	display both in header, display units in data cells, do not display	Specifies how units and scaling factors are shown in the crosstab. Measures can have scaling factors and units or currencies. If you have chosen <i>Display both In Header</i> , 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).
always fill	true, false	Specifies whether the component space should be filled as defined by the layout properties. If you set the property value to <i>false</i> , 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 <i>true</i> , the crosstab is adjusted and displayed in the exact size defined in the layout properties.

Property	Property Value	Property Description
conditional formatting visible	true, false	<p>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.</p> <div> <p>i Note</p> <ul style="list-style-type: none"> 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 <i>SAP BusinessObjects Analysis, edition for Microsoft Office User's Guide</i> on SAP Help Portal at https://help.sap.com. </div>
conditional formatting settings	conditional formatting rules	<p>Specifies the conditional formatting rules that should be applied. You can only use this property if you have added from the <i>Outline view</i> the technical component <i>Conditional Formatting Settings</i> to your application. By clicking the <i>Value</i> row for this property, the <i>Conditional Formatting Rule Manager</i> is opened, where you can change existing or add new rules for conditions.</p> <div> <p>i Note</p> <p>You can only use this property, if you have set the property <i>Conditional Formatting Visible</i> to <i>true</i>.</p> </div>
comments visible	true, false	Specifies if comments are displayed in the crosstab.

Property	Property Value	Property Description
column widths	numeric values ≥ 0	<p>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.</p> <p>A valid column width setting consists of two parts:</p> <ul style="list-style-type: none"> • 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 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. <div> <p>i Note</p> <p>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.</p> </div> <p>If the application user is allowed to resize columns, the resulting width will take precedence over any width set using the API method <code>setColumnWidth</code> or by the crosstab property Column Widths. Subsequent API calls to change the column width of a user-resized column will have no effect. If you do not want the user to be able to resize the crosstab columns at runtime, you can disable this feature by setting the crosstab property Column Resizing Enabled to false.</p>

Property	Property Value	Property Description
maximum width of header area	auto, integer values > 0	Specifies the header area width of the crosstab. If the property is set to auto (default setting), the width of the dimension area can take any value within the horizontal boundaries of a fully rendered dimension area. If set to a pixel value, the width of the dimension area can only be lower than or equal to that value. The automatic dimension area width determination and the user's possibility to alter the width of the dimension area if the property Horizontal Header Resizing Enabled is set to true , take a non-auto pixel width setting into account. If that pixel value is set, the crosstab will try to render the dimension header area width in this size. However, if the property's pixel value is larger than the fully rendered width of the dimension area, the dimension area will only take up as much horizontal space as it actually needs. The dimension area width is not artificially enlarged (cells made wider) to stretch out to the given pixel width. For further information, see Using Header Scrolling in Crosstabs [page 339]
<div>i Note</div> <div>This property cannot be used to hide the header area.</div>		
display repeated texts	true, false	Specifies if multiple header cells with the same texts are merged into single cells with row spans or column spans.
alternating row style	off, data area only, data area/last dimension	Specifies the display style of the crosstab rows.
title display	none, title text, data source	Specifies whether the crosstab has a title or not. If you set the value Title Text you can enter in the following property Title Text a free text as title. If you set the value Data Source , the title will be composed of the dimensions on the rows and columns axes
title text	your text	Specifies the title text for the crosstab. You can enter your text in the Value row of the property.

Planning Properties

Property	Property Value	Property Description
number of new rows	numeric value	Specifies the number of new rows for manual planning applications.
position of new rows	bottom, top	Specifies the position of new rows for planning applications.

Property	Property Value	Property Description
maximum number of value help members	numeric value (default: 100)	Specifies the maximum number of members displayed in a value help for non-hierarchical dimensions.
value help settings	technical name of Value Help Settings technical component	Specifies the technical component Value Help Settings which you can use to define the value help settings for dimensions.

Events

Property	Property Value	Property Description
on select	none	<p>Opens the script editor. With this property/event, you can enable user interaction with the crosstab by writing scripts. Click <code>CTRL +Space</code> to see the list of available methods for the application, the data source alias and the crosstab. Choose one of them.</p> <p>For further information, read the relevant method description in the Component API reference.</p> <div> <p>i Note</p> <p>The On Select event of the crosstab is triggered when a value is selected and also when a value is deselected.</p> </div>

49.6.2 Scorecard

Use the [Scorecard](#) to make complex result sets browsable using user friendly visualizations.

The [Scorecard](#) component is used for creating dashboards and reports with controlled visualization content and restricted navigation possibilities. Scorecards can help you to make complex result sets browsable by using user friendly visualizations (like charts) and highly customizable layout properties.

Properties: Display

Property	Property Value	Property Description
CSS class	Your CSS class	Defines an additional CSS class for custom CSS. The CSS classes must have the format myclass (and not .myclass). You do not have to use this property to use Custom CSS
selection type	none (default), single, multi, multi (toggle)	Defines the selection type. You can use the property Selection Type to activate the row selection. The property is set to None by default. You can activate Single , Multiple and Multiple (Toggle) selection types.
row height	integer	Defines the height of your rows.
header height	integer	Defines the height of your headers.
group header height	integer	Defines the height of your group headers.

Properties: Events

Property	Property Value	Property Description
on select	none	<p>Opens the script editor. With this property/event, you can enable user interaction with the scorecard by writing scripts. Click <code>CTRL+Space</code> to see the list of available methods for the application, the data source alias and the scorecard. Choose one of them.</p> <p>Cell selection is activated by default. In the onSelect event you can access the current selection of the dimensions specified in the row scope and the ID of the selected column. For further information, read the relevant method description in the Component API reference.</p>

49.6.3 Spreadsheet

The [Spreadsheet](#) component displays multi-dimensional data in a grid with analytic functions. In addition, the Spreadsheet component offers an easy-to-use copy function for copying cells and sheet areas. The spreadsheet is mainly intended for use with planning applications with input-ready queries.

Restrictions

When using the spreadsheet component, note the following restrictions:

- The universal display hierarchy of SAP BW data sources.
- Only one hierarchy per axis is supported.
- The Key and Text of dimensions are only displayed in one cell/column and not in separate cells/columns.
- Attributes are not supported
- In SAP UI5 m rendering mode, the Spreadsheet component is only supported in the Compact Form Factor.
- The Spreadsheet component can only be used with the Blue Crystal theme
- The Spreadsheet component does not allow any interaction in mobile applications.

The Spreadsheet component has the following specific properties:

Properties User Interactivity

Property	Property Value	Property Description
dimension toggles visible	false (default), true	Specifies if the dimension toggles are visible. With the dimension toggles, the user can hide one or more dimension rows or columns. This can be useful if the query has many dimensions and the user wants to restrict which dimensions are displayed.
hierarchy navigation enabled	false, true (default)	Specifies if the application user can expand or collapse hierarchy nodes, if the spreadsheet contains one or more hierarchies or hierarchical structures. If this property is set to <i>false</i> , 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.
context menu enabled	true (true), false	Specifies if a context menu can be displayed on the crosstab. The context menu can only be displayed on the spreadsheet, if the property is set to <i>true</i> and the technical component CONTEXT_MENU is available in the application (in the <i>Outline</i> view in the <i>Technical Components</i> folder). If the property is set to <i>false</i> , the context menu cannot be displayed, regardless of whether the technical component CONTEXT_MENU is part of the application. The entries displayed in the context menu depend on the data source element that the user clicks on. For more information, see Using the Context Menu (Technical Component) [page 86]

Properties: Display

Property	Property Value	Property Description
CSS class	Your CSS class	Defines an additional CSS class for custom CSS. The CSS classes must have the format myclass (and not .myclass). You do not have to use this property to use Custom CSS
conditional formatting visible	true, false	<p>Specifies if conditional formats (exceptions), which have been defined in the BEX Query Designer or in SAP BusinessObjects Analysis, edition for Microsoft Office (Analysis), are shown in the spreadsheet.</p> <div><p>i Note</p><ul style="list-style-type: none">• 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 Lumira Designer, 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 <i>SAP BusinessObjects Analysis, edition for Microsoft Office User's Guide</i> on SAP Help Portal at http://help.sap.com/boaa.</div>
display repeated texts	true, false	Specifies if multiple header cells with the same texts are merged into single cells with row spans or column spans.

Events

Property	Property Value	Property Description
on select		<p>Opens the script editor. With this property/event, you can enable user interaction with the spreadsheet by writing scripts. Click <code>CTRL</code> + <code>Space</code> to see the list of available methods for the application, the data source alias and the spreadsheet. Choose one of them.</p> <p>For further information, read the relevant method description in the API reference.</p> <div><p>i Note</p><p>The <i>On Select</i> event is triggered when the user clicks on a cell of the spreadsheet</p></div>

Properties: Planning

Property	Property Value	Property Description
maximum number of value help members	numeric value (default: 100)	Specifies the maximum number of members displayed in a value help for non-hierarchical dimensions.
paste new lines enabled	false (default), true	Specifies if the application user can insert new empty rows or paste copied rows or areas of the spreadsheet below the current last line of the spreadsheet.

49.7 Chart Components

49.7.1 Chart

Add charts to an application to present data in a graphical way. Charts can emphasize irregularities or trends in 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, by default.

i Note

When working with the script editor and viewing lists of global script functions, you see that the [Chart](#) component has a different technical name. Instead of the display name **Chart**, its technical name is **VizFrame**.

The properties of the [Chart](#) component can be set in three ways:

- general properties are defined in the [Properties](#) view of the application
- additional properties are available by selecting the ellipsis button of the [Chart Configuration](#) property. The additional property options available are based on [Chart](#) type selected.
- a subset of the same additional properties is available within the context menu of the [Chart](#).

The chart general properties in the [Properties](#) view are described in the table called “Properties”. For more information on the how to set additional chart properties available, you can refer to the chapter called *Using the Configure Chart Dialog*.

Properties

Property	Property Value	Property Description
data source	Name of data source alias, such as DS_1 .	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.

i Note

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.

Property	Property Value	Property Description
allow data source modification	true, false	Sets whether the data source state can be modified by the Chart . If set to true, the user can modify the data source state, while interacting with the Chart . The changes made affect all components bound to the same data. It allows the application user to add or remove one measure or dimension from the Chart or connected Chart Feeding Panel . This property setting modifies the row and column items in the data source state. If set to the default value false, the data source state cannot be modified by the Chart and the component strictly follows the initial data source state.
data selection	Selection string expressed in JSON notation generated from the data selection.	<p>Dialog box allows you to select multiple rows or columns from the data result set to create a separate chart. Select the columns or rows that you want to appear in a separate chart.</p> <div> <p>i Note</p> <p>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.</p> </div>

Property	Property Value	Property Description
chart configuration	column chart	<p>Select the ellipsis button to call the <i>Configure Chart</i> dialog box. The <i>Configure Chart</i> dialog box allows you to change chart type, properties, and chart feeding in the one location. The dialog box contains the following:</p> <ul style="list-style-type: none"> • Layout area to display the current chart. You can see changes to the chart type, properties, and chart feeding reflected in this layout area. • Chart tab offers you a selection of chart types from which to choose to display data. • Properties tab offers you a selection of context based additional properties based on the selected element. • Chart feeding panel allows you to manipulate measures and dimensions. <div> <p>i Note</p> <p>For more details, you can refer to the chapters called <i>Chart Properties in the Configure Chart Dialog</i> and <i>Using the Configure Chart Dialog</i>.</p> </div> <div> <p>i Note</p> <p>If you have created and installed new chart extensions using the SAP Lumira SDK, these extensions also appear in the <i>Additional Charts</i> drop-down list. For more information about creating SAP Lumira SDK extensions, see the “SAP Lumira Visualization Extensions Developer Guide” on the SAP Help Portal at https://help.sap.com. Extensions developed with the SAP Lumira SDK can only be deployed to the BI platform.</p> </div>
conditional formatting visible	true, false	Allows you to select whether conditional formatting is visible in a chart.

Property	Property Value	Property Description
conditional formatting settings	none	Select to call a dialog box called <i>Conditional Formatting Rule Manager</i> to allow you to apply rules to measures or dimension members within a selected chart. These rules change the appearance of the chart when specific conditions are met.
show totals	<ul style="list-style-type: none"> • true • false 	When you select true , the totals for each dimension that you have added in the <i>Edit Initial View...</i> dialog box are displayed in the chart. Select false to hide these totals. The default setting is <i>false</i> .

Property	Property Value	Property Description
show scaling factors	<ul style="list-style-type: none"> • true • false 	Select true to show scaling factors in the chart. The scaling factors displayed are the ones defined for the measures in the Edit Initial View... dialog box. Select false to hide scaling factors. The default setting is false .

i Note

1. If the measures are in the columns, the scaling factors display in brackets in the axes, to the right of the measures. If your measures are in the rows, the scaling factors display in brackets in the legend, to the right of the measures.

❖ Example

Measure A (*1000)

2. When you swap axes, the scaling factors 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 appears beside the measures without the currency value.

4. There are two ways to define scaling factors:
 - using the context menu on the measure in the [Edit Initial View...](#) and selecting the required scale
 - scripting a component with an on-click event

Property	Property Value	Property Description
show expanded nodes	<ul style="list-style-type: none"> • true • false 	<p>Displays or hides the expanded hierarchy nodes in a chart. The default setting is <i>true</i>. When you set the property to false, the dimension with the expanded hierarchy is not displayed in the chart. This removes some unnecessary nodes from the chart, and helps make the chart easier to read.</p>
dimension label	<ul style="list-style-type: none"> • initial view definitions • text • key 	<ul style="list-style-type: none"> • Select <i>Initial View Definitions</i> to use the definitions that are set in the <i>Edit Initial View...</i> dialog box. • Select the <i>Text</i> option to set all the members to be displayed as text. <div> <p>i Note</p> <p>This setting overrides the settings in the <i>Edit Initial View...</i> dialog box.</p> </div> <ul style="list-style-type: none"> • Select <i>Key</i> to use the <i>Key</i> member display. <div> <p>i Note</p> <p>This setting overrides the settings in the <i>Edit Initial View...</i> dialog box.</p> </div>
on select	none	<p>The <i>On Select</i> event of the chart is triggered when a value is selected or deselected. This property enables you to assign a custom handler to the <i>OnSelect</i> event. To enable user interaction with the chart, use this property to assign a custom handler to the <i>OnSelect</i> event.</p> <ol style="list-style-type: none"> 1. Choose the <i>Browse</i> button to open the <i>Script Editor</i> dialog box. 2. Press <code>CTRL+SPACEBAR</code> to see the list of available methods for the application, the data source alias, and the chart. 3. Choose one of them.

Related Information

[Chart Properties in the Configure Chart Dialog \[page 457\]](#)
[Using the Configure Chart Dialog \[page 243\]](#)

49.7.1.1 Chart Properties in the Configure Chart Dialog

Use the chart properties in the [Properties](#) tab to configure the settings of the [Chart](#) component.

You can select from a range of properties within the [Configure Chart](#) dialog within the [Chart Configuration](#) property. The types of properties offered to you, depend on the type of chart and the type of property you have selected. The following property categories are available, depending on the chart type selected:

- column and bar
- data label
- horizontal axis
- horizontal axis title
- vertical axis
- vertical axis title
- chart area
- chart title
- plot area
- marker
- legend

Related Information

[Chart](#) [page 451]

[Using the Configure Chart Dialog](#) [page 243]

49.7.2 Chart Feeding Panel

Use the [Chart Feeding Panel](#) chart component to display all the measures and dimensions bound to a chart.

The [Chart Feeding Panel](#) offers the application user a representation of all the measures and dimensions bound to that chart in the [Edit Initial View...](#) It also allows the user to move measures and dimensions up and down within their respective areas in the [Chart Feeding Panel](#) component. The chart reflects any changes they make to the position of measures and dimensions in the [Chart Feeding Panel](#). You cannot drag measures into the dimensions area or drag dimensions into the measures area. Changing the position of measures and dimensions in the [Chart Feeding Panel](#) and then changing the chart type using the [Chart Type Picker](#), means that the measures and dimensions remain in their new position. The ability to modify how data is fed to the chart, gives the application user greater freedom when running their application. The [Chart Feeding Panel](#) is bound to a chart using the [Chart Reference](#) property in the designer.

The [Chart Feeding Panel](#) property is described in the table.

i Note

The [Chart Feeding Panel](#) component has a different technical name and display name. You can see this difference when working with the script editor and viewing lists of global script functions. The display name is **Chart Feeding Panel** and the technical name is **FeedingPanel**.

Property	Property Value	Property Description
chart reference	dropdown list	Specifies which chart component is referenced by the Chart Feeding Panel .

Related Information

[Using the Configure Chart Dialog \[page 243\]](#)

49.7.3 Chart Property Editor

Use the [Chart Property Editor](#) to allow the application user to change the properties of a data bound chart when running the application.

The [Chart Property Editor](#) is a chart component that you can bind to a data bound chart, through the [Chart Reference](#) property in the designer. It allows the application user to edit the look and feel of the data bound chart by manipulating the properties of the chart.

The [Chart Property Editor](#) property is described in the property table.

Note

The [Chart Property Editor](#) component has a different technical name and display name. You see this when working with the script editor and viewing lists of global script functions. The display name is **Chart Property Editor** and the technical name is **PropertyEditor**.

Property	Property Value	Property Description
chart reference	dropdown list	Specifies which chart component is referenced by the Chart Property Editor .

Related Information

[Using the Configure Chart Dialog \[page 243\]](#)

49.7.4 Chart Type Picker

Use the [Chart Type Picker](#) chart component to select a chart type when running an application.

The [Chart Type Picker](#) chart component allows the application user to select an alternative chart type to display data, when running the application. You can decide, through scripting, how the charts behave when the user runs the application. A dropdown list beside each group of charts, displays a list of each chart type within each

group. You can manipulate the *Chart Type Picker* properties to add additional visualization types to the existing palette of chart types. If you add additional visualization types to the *Chart Type Picker*, they are grouped together in a new group.

i Note

When working with the script editor and viewing lists of global script functions, you can see that the *Chart Type Picker* component has a different technical name. Instead of the display name **Chart Type Picker**, its technical name is **SwitchBar**.

The *Chart Type Picker* properties are described as follows:

Property	Property Value	Property Description
chart reference	dropdown list	Specifies which chart component in the application is changed when a different chart type is selected in the <i>Chart Type Picker</i> . The chart component type change to reflect this chart type selection once the selection made is different to the chart component's current type.
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: <ul style="list-style-type: none">• type• name• icon

i 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.
- If you have created and installed new chart extensions using the SAP Lumira SDK, these extensions will also appear in the *Chart Type Picker* component under the *Additional Charts* chart type category. For more information about creating

Property	Property Value	Property Description
		SAP Lumira SDK extensions, see the “SAP Lumira Visualization Extensions Developer Guide” on the SAP Help Portal at https://help.sap.com .

Related Information

[Using the Configure Chart Dialog \[page 243\]](#)

49.7.5 Map

Use the [Map](#) component to display different layers of geographical information.

The [Map](#) chart component allows users to drilldown through the different layers to reveal data in various ways. You can apply four different types of layers to your [Map](#) - choropleth (polygons, multi-polygons, lines, and multi-lines), bubble charts, marker and pie charts.

You can assign a different data source to each layer.

The configuration of all layers and properties of the [Map](#) component is done using the [Configure Map](#) dialog. To access the [Configure Map](#) dialog, select the ellipsis button beside the [Map Configuration](#) property of the [Map](#) component.

i Note

The [Map](#) component has a different technical name and display name. You can see the different names when working with the script editor and viewing lists of global script functions. The display name is **Map** and the technical name is **VizMap**.

General Properties

The general properties of the [Map](#) are as follows:

Property	Property Value	Property Description
map configuration	none	Calls the Configure Map dialog, which allows you to configure all properties and settings associated with the Map . For all properties within the Configure Map dialog you can refer to the chapter called <i>Properties of the Configure Map Dialog</i>
show scaling factors	true, false	Select true to show scaling factors in the map. The scaling factors displayed are the ones defined for the measures in the Edit Initial View... dialog box. Select false to hide scaling factors. The default setting is <i>false</i> .
adjust boundary	true, false	Select true to automatically adjust the boundaries of the Map to suit the current size.

Related Information

[Working with Maps \[page 295\]](#)

[Properties of the Configure Map Dialog \[page 461\]](#)

49.7.5.1 Properties of the Configure Map Dialog

The [Configure Map](#) dialog contains all settings and properties for the [Map](#) component.

The [Configure Map](#) dialog contains the following two categories:

- Map
- Properties

The table called *Map* contains all the map builder properties in the [Map](#) section of the [Configure Map](#) dialog.

The table called *Properties* contain all other properties in the [Properties](#) section of the [Configure Map](#) dialog.

Map:

Property	Property Value	Property Description
add layer	button	Select the + icon to add a new layer to the Map .

Property	Property Value	Property Description
context menu	cog wheel with the following context menu items: <ul style="list-style-type: none"> • move up • move down • clear • delete layer 	Select Move Up to change the position of the current map layer so it is in a higher position in the layer hierarchy. Select Move Down to change the position of the current map layer so it is lower in the layer hierarchy. Select Clear to remove any value settings. Select Delete Layer to remove the current layer from the layer hierarchy.
data point type	choropleth, bubble, marker, pie	Select the data point type from the options available.
datasets in use	drop down list of available data sources beside an ellipsis button	Enables you to select a data source to assign to the current layer in the Map .
geo information	geoJSON, latitude and longitude	Allows you to switch between the GeoJSON and Longitude and Latitude geo types.
custom geoJSON file	none	Specifies the file path to the GeoJSON file. In BI platform mode, select the ellipsis button within the Custom GeoJSON File property. The Custom GeoJSON File property allows you to select the location of your custom GeoJSON file. In local mode, select the Custom GeoJSON File ellipsis button, to open a Select Local GeoJSON dialog box. From the Select Local GeoJSON dialog box, select your locally stored custom GeoJSON file.
geoJSON mapping property	drop down list	Provides a list of all available properties in the GeoJSON file. To assist you with your selection, a text value is available beside the first property within the GeoJSON file. Select the dropdown arrow to view it.
geoJSON mapping type	key, text	Specifies whether key or text of the geo dimension member is used when mapping to the GeoJSON.
latitude	dropdown	Enables you to assign a latitude measure name to points and charts layers.
longitude	dropdown	Enables you to assign a longitude measure name to points and charts layers.
geo dimension	none	Allows you to add the geo dimensions to the map layer.

Property	Property Value	Property Description
size	none	<p>Select the + icon beside the Add One Measure property to assign a measure. The Size of the bubbles and pie slices is determined by this measure.</p> <div> i Note <p>This property is available for bubble and pie layer types.</p> </div>
color	none	<p>Select the + icon beside the Add One Measure property to assign a measure. The Color of the choropleth is determined by this measure.</p> <div> i Note <p>This particular color property to add one measure is available choropleth layer type only.</p> </div> <p>Select the + icon beside the Add One Dimension property to assign a dimension. The Color of the bubbles and pie slices is determined by this dimension.</p> <div> i Note <p>This property is available for bubble and pie layer types.</p> </div>
animation	none	<p>Select the + icon beside the Add One Dimension property to assign a dimension. The Animation of the bubbles is determined by this dimension.</p> <div> i Note <p>This property is available for bubble layer type only.</p> </div>

Properties:

Property Area	Property Type and Value	Property Description
geo map	selection mode: <ul style="list-style-type: none">• exclusive• inclusive	Sets whether selection mode is exclusive or inclusive. Exclusive selection mode means that the selection is not cumulative. Inclusive selection mode means that the selection is cumulative.
	null values visible: <ul style="list-style-type: none">• true• false	Sets whether null values are visible on all layers.
	map type: <ul style="list-style-type: none">• ESRI online map• customized online map	Allows you to select the type of map to be used.
	base map url <ul style="list-style-type: none">• url	<p>The URL template, which is used to provide the basemap. The default basemap is the Esri map. You can also enter the URL for your preferred 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>.</p> <div>i Note<p>LOD indicates 'level of detail'. It is also sometimes referred to as 'Z'.</p></div> <p>A basemap is optional.</p>
	base map copyright: <ul style="list-style-type: none">• text	Text 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.
layers	layer drop down	Select from the drop down list of layers to select the layer to be configured.
	show this layer: <ul style="list-style-type: none">• checkbox	Sets the visibility of the layer. When this checkbox is selected, the layer is visible on the Map . This checkbox is selected by default.

Property Area	Property Type and Value	Property Description
	marker color	<p>Use the Marker Color dropdown to customize the color of the marker that indicates the point on the Map.</p> <div> i Note This property is available for the marker layer type only. </div>
	custom marker	<p>Select the icon beside the Custom Marker to select custom markers that are stored on the platform. Selecting the Custom Marker icon in local mode, opens a Select Local Image dialog box, from where you can select your locally stored custom marker.</p> <div> i Note This property is available for the marker layer type only. </div>
	vertical anchor: <ul style="list-style-type: none"> • top left • top center • top right • center left • center • center right • bottom left • bottom center • bottom right 	<p>Select from the options provided in the Vertical Anchor dropdown list to select an anchor point for the custom marker.</p> <div> i Note This property is available for the marker layer type only. </div>
	cluster adjacent locations	<p>Select the checkbox Cluster Adjacent Locations if you want to cluster bubbles and markers that are close to each other on the map.</p> <div> i Note This property is available for bubble and marker layer types only. </div>

Property Area	Property Type and Value	Property Description
	show as doughnut chart	<p>If you want to display the pie chart as a doughnut chart, you can select the checkbox called Show as Doughnut Chart.</p> <div> i Note This property is available for the pie layer type only. </div>
	start color: <ul style="list-style-type: none"> hex color picker 	You can use a color picker to specify the start color of a shape layer color gradient.
	end color: <ul style="list-style-type: none"> hex color picker 	You can use a color picker to specify the end color of a shape layer color gradient.
	color palette	<p>Select bubble or pie slice color palettes from the list of options in the Color Palette property.</p> <div> i Note This property is available for bubble and pie layer types only. </div>
	classification type: <ul style="list-style-type: none"> equal interval quantile 	<p>Defines the type of classification to be applied to a shape layer. Select the Quantile option to distribute a set of values into groups that contain an equal number of values. Select the Equal Interval option to arrange a set of values into groups that contain an equal range of values.</p>
data labels	show data labels: <ul style="list-style-type: none"> checkbox 	Select the Show Data Labels checkbox to display data labels in the map layer.
	<ul style="list-style-type: none"> font style font size bold italics color 	You can define how the data labels are formatted.
legend	show legend: <ul style="list-style-type: none"> checkbox 	Select the Show Legend checkbox to display a legend in the map layer.

Property Area	Property Type and Value	Property Description
	show legend title: <ul style="list-style-type: none"> checkbox 	Select the Show Legend Title checkbox to display a title in the legend in the map layer.
	<ul style="list-style-type: none"> font style font size bold italics color 	You can define how the text in the legend title and legend items is formatted.
map title	show map title: <ul style="list-style-type: none"> checkbox 	Select the Show Map Title checkbox to display a title in the map layer.
	title: <ul style="list-style-type: none"> free text field 	Define a title for the map by populating the Title free text field.
	<ul style="list-style-type: none"> font style font size bold italics color left align center right align 	You can define how the text in the map title and legend items is formatted.

49.8 Filter Components

49.8.1 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 application 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:

Property	Property Value	Property Description
data source	name of data source alias (for example "DS_1")	Displays all data source aliases. If you have created several data sources for

Property	Property Value	Property Description
		<p>the application, you can change the data source for the filter panel by choosing the relevant data source alias.</p> <div> <p>i Note</p> <ul style="list-style-type: none"> 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. 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. </div>
target data sources (opional)	names of target data source aliases (for example, "DS_1; DS_2")	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 sources aliases added to the application and separated by a semicolon.
dimension	dimension	Specifies the dimension to be filtered. You can select a dimension using the Dimension Selection dialog box or search for a dimension.
dimension name	true, false	Specifies whether the dimension name of the selected dimension is displayed in the dimension filter.
member display	key, text, key and text, text and key	Specifies the dimension member display.
visualization mode	default, dropdown, list , text, range, date, complex	Specifies the visualization mode of the component.
maximum number of members	your number , default: 100	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

Property	Property Value	Property Description
		displayed. Instead, the application user has to limit the number by searching for the text or key of one or more values.
value help settings	technical name of Value Help Settings technical component	Specifies the technical component Value Help Settings which you can use to define the value help settings for dimensions.
on apply	none	<p>Opens the script editor. With this property/event, you can enable user interaction with the dimension filter by writing scripts. The On Apply event is triggered when the application user presses the Apply button of the dimension filter at runtime. 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 dimension filter. Choose one of the methods.</p> <p>For further information, read the relevant method description in the Component API reference.</p>

49.8.2 Filter Bar

The [Filter Bar](#) is the horizontal filter element for applications rendered in the SAPUI5 m rendering mode.

Initially, the filter area of the filter bar is not displayed, even if you have chosen a dimension in the [Dimensions](#) properties. Instead the application user has to click on [Show Filter Bar](#) at runtime. To display the filters in the filter bar, the application user has to click on [Filters](#). The [Filter](#) dialog box is then displayed, which enables the user to set filters. The application user has to check the measures and/or dimensions to be displayed in the filter area and click on [Go](#). In the [Filter](#) dialog box the user can also enter values directly or use the value help for the chosen measures and/or dimensions. The filter is applied after the user has clicked the [Go](#) button. Alternatively, the application user can enter values directly or use the value help for the chosen measures and/or dimensions when the chosen measures and/or dimensions are displayed in the filter area.

Once filters have been added, they can be changed directly in the filter bar. As is the case in the [Filter](#) dialog box, changes are only submitted when the user clicks on the [Go](#) button.

If the user cancels the Filter dialog box after having made some changes, the changes are discarded and not displayed in the filter area.

Note

If one or more filters are set (either in the filter bar or in the dialog box), a counter showing the number of set filters is included behind the [Filters](#) link.

If there is not enough space to show all selected filters in the filter bar, an [Show More](#) button is displayed. Clicking the [Show More](#) button will expand the filter bar to show all selected filters.

The [Filter Bar](#) component has the following specific properties:

Property	Property Value	Property Description
filter area visible	true, false (default)	Specifies whether the filter area is initially displayed. This can be changed at runtime by using the Show Filter Bar / Hide Filter Bar link.
show only filtered dimensions	true, false (default)	Specifies whether the dimensions are automatically displayed for which filter values are set.
dimensions	dimension	Specifies the dimension to be filtered. You can select a dimension using the Select Dimensions dialog box after having assigned a data source to the filter bar in its data binding properties.
value help settings	technical name of Value Help Settings technical component	Specifies the technical component Value Help Settings which you can use to define the value help settings for dimensions.

49.8.3 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](#).

The [Filter Line](#) component has the following specific property:

Property	Property Value	Property Description
measure visible	true, false	Select true to include measures in the list of data source filters in the Filter Line menu.

Related Information

[Filter Panel \[page 471\]](#)

49.8.4 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 application 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 the maximum number of members set in the properties sheet, 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:

Property	Property Value	Property Description
data source	name of data source alias (for example "DS_1")	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.
<div> <div> Note</div> <ul style="list-style-type: none"> 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. Keep in mind that the data source is filtered automatically and you do not have to add it to the <i>Target Data Sources</i> property. If you change the data source, the system automatically refreshes the list of the <i>Target Data Sources</i>. </div>		
target data sources (optional)	names of target data source aliases (for example "DS_1; DS_2")	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 have to be existing data

Property	Property Value	Property Description
		sources aliases that have been added to the application. After you have selected the target data sources in the Select Target Data Sources dialog box, the target data sources are displayed, separated by semicolons.
mode	filtering, filtering and navigation	<p>Enables you to set the function scope of the filter panel:</p> <ul style="list-style-type: none"> • Filtering enables the application user to filter data. • Filtering and Navigation enables the application user both to filter data and to add or remove dimensions from rows or columns of the result set by using the corresponding buttons.
dimensions	dimensions	Opens the Select Dimensions dialog box. Here you can select and sort dimensions of the data source you have assigned to the filter panel.
member display	key, text, key and text, text and key	Specifies the display mode of the dimension members.
title	yourTitle	You can set a title for the filter panel. The title is displayed in the middle of the filter panel header.
auto apply	true, false	Specifies if all changes made by the application user at runtime are applied automatically. If set to false , the application user has to click on GO after making changes in the filter panel. If set to true, all changes in navigation will be applied immediately.
maximum number of members	your number , default: 100	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.
value help settings	technical name of Value Help Settings technical component	Specifies the technical component Value Help Settings which you can use to define the value help settings for dimensions.
on apply	none	Opens the script editor. With this property/event, you can enable user interaction with the filter panel by writing scripts. Click <code>CTRL+Space</code> to

Property	Property Value	Property Description
		<p>see the list of available methods for the application, the data source alias and the filter panel. Choose one of the methods. The On Apply event is rendered when the application user presses the Apply button at runtime. The filter value is set first.</p> <p>For further information, read the relevant method description in the Component API reference.</p>

49.8.5 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:

Property	Property Value	Property Description
dimensions	dimensions	Opens the Select Dimensions dialog box. Here you can select and sort the dimensions of the data source that you have assigned to the navigation panel.
mode	navigation, list only	Specifies the mode used to display the different parts of the component. If you choose Navigation , the component is displayed with the rows and columns area (navigation area). If you choose List Only , the dimensions are displayed in a list, without the rows and columns area. In this case, no drag and drop is possible; the application user can use the context menu instead.
list measures separately	true, false; default: false	Specifies how the measures are displayed in the list. If set to true , the measures (key figures) are listed separately above the dimensions.
auto apply	true, false	Specifies if all changes made by the application user at runtime are applied automatically. If set to false , the application user has to click on GO after making changes in the navigation

Property	Property Value	Property Description
		panel. If set to true, all changes in navigation will be applied immediately.

49.9 Basic Components

49.9.1 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:

Property	Property Value	Property Description
text	default text: button	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.
icon	name of icon; save location	Specifies the icon to be displayed on the button. Click on the Browse.. button right to the property and choose in the Open Image (for the SAP Lumira Documents mode) or Select Local Image (for the local mode) dialog box, the icon you want to add.
on click	script	<p>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 CTRL + Space to see the list of available methods for the application, the data source alias and the button. Choose one of them.</p> <p>For further information, read the relevant method description in the Component API reference.</p>

49.9.2 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:

Property Type	Property Value	Property Description
text	default text: checkbox	Specifies the text to be displayed on the right of the checkbox. You can overwrite the default text and enter your own text.
checked	false/true	Specifies whether the checkbox is initially selected.
on click	script	<p>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+Space</code> to see the list of available methods for the application, the data source alias and the checkbox. Choose one of them..</p> <p>For further information, read the method description in the Component API reference.</p>

49.9.3 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:

Property Type	Property Value	Property Description
items	list of items	<p>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:</p> <ul style="list-style-type: none">value

Property Type	Property Value	Property Description
		<p>The value property is mandatory for all items.</p> <ul style="list-style-type: none"> text (optional) <p>The text property is optional. If you enter a text for an item, this text is displayed right of the checkbox. If no text is entered, the value for the item is displayed.</p> <p>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.</p>
on select	script	<p>Opens the script editor. With this property, you can enable user interaction with the checkbox group by writing scripts. The On Select event is triggered when the application user activates one checkbox. In the script editor, press <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.</p> <p>For further information, read the method description in the Component API reference.</p>

49.9.4 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:

Property Type	Property Value	Property Description
date	date	Specifies the initial date in the format <code>yyyymmdd</code> .
on select	script	<p>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 date field. Choose one of them..</p>

Property Type	Property Value	Property Description
		For further information, read the relevant method description in the Component API reference.

49.9.5 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:

Property Type	Property Value	Property Description
items	list of items	<p>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:</p> <ul style="list-style-type: none"> value The value property is mandatory for all items. text (optional) 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. <p>If you want a specific item to be displayed by default, select the relevant item and click Set as Default. This is shown as the default entry in the dropdown box.</p>
on select	script	<p>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 <code>CTRL</code> <code>+Space</code> to see the list of available methods for the application, the data source alias and the dropdown box. Choose one of them.</p>

Property Type	Property Value	Property Description
		For further information, read the relevant method description in the Component API reference.

❖ Example

You have created an application with a crosstab. You have assigned the data data source with 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:

```
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.

49.9.6 Feed List

The [Feed List](#) is a basic component that allows you to display a list of feed items.

The [Feed List](#) component has the following specific properties:

Property	Property Value	Property Description
icon visible	boolean	Displays an icon for each feed item.
icon url template	none	Generates an icon URL for each of the feed items. It makes use of placeholder texts that can be replaced with a property of a feed item.
<div>❖ Example</div> <p>Given the following url template: <code>https://people.sap.com/images/{author}.png</code>, a feed item with the author X will have the following url set for its icon: <code>https://people.sap.com/images/X.png</code>.</p>		
<div>i Note</div> <p>Any of the feed item properties can be used in the url template.</p>		
date format	short, medium, long	Selects the format of the date for each feed item.

Related Information

[Working with the Feed List \[page 284\]](#)

49.9.7 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:

Formatted Text View Editor

Input Type	Input Options
functions	<ul style="list-style-type: none">• cut• copy• paste• paste as plain text• undo• redo
font	<ul style="list-style-type: none">• bold• italic• remove format
styles	<ul style="list-style-type: none">• block styles<ul style="list-style-type: none">◦ address• inline styles<ul style="list-style-type: none">◦ cited work◦ inline quotation
paragraph format	<ul style="list-style-type: none">• normal• h1• h2• h3• h4
list	<ul style="list-style-type: none">• insert/remove numbered list• insert/remove bulleted list
layout	<ul style="list-style-type: none">• decrease indent• increase indent

Supported HTML Tags

The following table lists all the supported HTML Tags for this component.

HTML Tag	Description
abbr	abbreviation
acronym	acronym
address	format as address
blockquote	blockquote
br	line break
cite	cite
dd	definition item
dfn	definition
div	div
dl	definition list
dt	definition topic
em	emphasis
h1	header level 1
h2	header level 2
h3	header level 3
h4	header level 4
kbd	text to be entered via keyboard
li	list item for "ol" and "ul"
code	inline code
ol	ordered list (numbered list)
p	section symbol
pre	pre-formatted text
q	quotation
samp	sample(block)
span	span
strong	strong
ul	unordered list
var	text that is variable

Note

- Header tags must be on the top level.
- If styled accordingly in your chosen theme, you can either use `` for italic font and `` 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](#)

49.9.8 Grouped List

Use the [Grouped List](#) basic component to add items to a list, and to group the items into categories.

The [Grouped List](#) allows you to add data sources, composites or components to a list, and to categorize these items into groups. Similar to the [List Box](#), you can insert items and adjust the order in which they appear in the list. A new value can be added at design time, using the properties. A new value can be added at runtime, through scripting. If the list grows beyond the size set in the design tool, the scroll bar appears. The [Grouped List](#) component has the following specific properties:

Property Type	Property Value	Property Description
drag and drop for authoring enabled	boolean	When set to true, you can drag and drop items from the Grouped List onto the area of the application enabled for authoring. The Authoring technical component allows you to provide application users with the ability to change the layout of applications that you created for them. Default value is false.
<div><div>i Note</div><ul style="list-style-type: none">• The Drag and Drop for Authoring Enabled property is used with the Authoring technical component only.• The Authoring technical component is supported in SAP Lumira Documents mode only.• For more information on using the Grouped List with the Authoring technical component, you can refer to the chapter called <i>Using the Grouped List with the Authoring Technical Component</i>.</div>		
items	list of items	Opens the Edit Grouped List Items dialog box. Here, you can insert items, change the order of items, or remove them. Each item has the following properties:

Property Type	Property Value	Property Description
		<ul style="list-style-type: none"> value The <i>Value</i> property is mandatory for all items. The value is the unique identifier of each grouped list item. text The <i>Text</i> property is mandatory. If you enter a text for an item, this text is displayed. If no text is entered, the value of the item is displayed. group The <i>Group</i> property is mandatory. It categorizes the item into a group. icon The <i>Icon</i> property is optional. Icon image for the item. Insert an icon of your choice using the ellipsis button to call the <i>Open Image</i> dialog. type The <i>Type</i> property is optional. Select object type of the item from a dropdown list of component types. <div> <p>i Note</p> <ul style="list-style-type: none"> If you wish to use the <i>Grouped List</i> with the <i>Authoring</i> technical component, you must select a value for Type. </div> <p>If you want a specific item to be displayed by default, select the relevant item and click <i>Set as Default</i>. This item is shown as the default entry in the <i>Grouped List</i>.</p>

Related Information

[Working with the Grouped List \[page 286\]](#)

[Component Types Supported in the Grouped List \[page 287\]](#)

[Using the Grouped List with the Authoring Technical Component \[page 288\]](#)

49.9.9 Icon

The *Icon* component displays icons. The icons can either be chosen from a set of built-in icons delivered by SAP or from a custom font in TTF or OTF format.

The *Icon* component has the following specific properties:

Property	Property Value	Description
CSS class	Your CSS class	Defines an additional CSS class for custom CSS. The CSS classes must have the format myclass (and not .myclass). You do not have to use this property to use <i>Custom CSS</i> .
custom font	font	Specifies the custom font. With the file picker you can specify the font file in the .ttf or .otf format. Ensure that the font file's <i>Font embeddability</i> property has been set to <i>Installable</i> , otherwise the icons cannot be rendered in Internet Explorer. This property can be found in the <i>Details</i> tab of the files properties using Windows™ Explorer. The corresponding font icons are displayed on the <i>Additional Properties</i> sheet on the <i>Custom Font</i> tab in design studio and are available for selection.
color	color	Specifies the color of the icon itself.
background color	color	Specifies color of the icon background.
background shape	Rectangle, Ellipsis, None	Specifies the shape of the icon background. The shapes are <i>RECTANGLE</i> , <i>ELLIPSIS</i> or <i>NONE</i> (transparent).
size factor	numeric value	Specifies the size of the icon compared to the bounding rectangle (valid range from 0.0 to 1.0).
icon URI	URI	Specifies the icon URI. The icon URI must have the following structure sap-icon://name or sap-icon://collection/name

Property	Property Value	Description
tooltip	text	Specifies the tooltip that is displayed when the user hovers over the component at runtime. Type in the text that should be displayed as a tooltip.
on click	script	<p>Opens the script editor. With this property, you can enable user interaction with the icon component by writing scripts. The On Click event is triggered when the application user clicks on the icon. 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, the icon and other components of the application. Choose one of them.</p> <p>For further information, read the method descriptions in the Component API reference.</p>

49.9.10 Image

Using the image component, you can enhance applications by adding images.

The *Image* component has the following specific properties:

Property Type	Property Value	Property Description
image	image location/name	<p>Specifies the image that is displayed. Specifies the image to be displayed. Click on the <i>Browse...</i> button right to the property and choose in the <i>Open Image</i> (for the SAP Lumira Documents mode) or <i>Select Local Image</i> (for the local mode) dialog box, the image you want to add.</p> <p>There are several possible locations where you can store the images and icons that you want to use in your application:</p> <ul style="list-style-type: none"> in the application directory <code><userhome>\Analysis-workspace\com.sap.ip.bi.zen</code>

Property Type	Property Value	Property Description
		\repository \MyApplication \MyImage.jpg Type in the image name, for example, MyImage.jpg . <ul style="list-style-type: none"> in a subfolder of the application directory, for example, <userhome>\Analysis-workspace \com.sap.ip.bi.zen \repository \MyApplication \MySubFolder \MyImage.jpg Type in the image name, for example, MyImage.jpg. Internet or intranet For more information, see the <i>Administrator Guide: SAP Lumira</i> at http://help.sap.com .
hover image	image location/name	Specifies the image that is displayed when the user hovers with the mouse over the component. Type in the image name, for example, MyImage.jpg .
click image	image location/name	Specifies the image that is displayed when the user clicks the image. Type in the image name, for example, MyImage.jpg .
opacity	Opacity percentage value (0 - 100: 0 = transparent, 100 = solid).	Specifies the image opacity.
on click	none	Opens the script editor. With this property/event, you can enable user interaction with the image by writing scripts. Click CTRL+Space to see the list of available methods for the application, the data source alias and the image. Choose one of them.. For further information, read the relevant method description in the Component API reference.

49.9.11 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:

Property Type	Property Value	Property Description
value	string	Specifies the initial value that the application user can overwrite during runtime.
on change	none	<p>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.</p> <p>For further information, read the relevant method description in the Component API reference.</p>

49.9.12 Link

The [Link](#) component can be used to trigger a navigation in an application.

i Note

This component is only available in the SAPUI5 m rendering mode.

The [Link](#) component has the following specific properties:

Property Type	Property Value	Property Description
Text	some text	Specifies the link text.
Tooltip	true, false	Specifies whether the component has a tooltip at runtime and which tooltip.
URL	absolute URL	<p>Specifies the URL of the link. The URL should be entered in the following form: http://www.yourlink.linkdomaine.</p> <p>The URL is opened in a new browser window.</p> <div><h3>i Note</h3><p>If you have set both, the URL property and the On Click event, both properties are executed.</p></div>
Style	normal, subtle, emphasized	Specifies the style of the link text.
On Click	none	Opens the script editor. With this property/event, you can enable user

Property Type	Property Value	Property Description
		interaction with the link by writing scripts. Click <code>CTRL</code> + <code>+</code> + <code>Space</code> to see the list of available methods for the application, the data source alias and the link. Choose one of them. For further information, read the relevant method description in the Component API reference.

49.9.13 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:

Property Type	Property Value	Property Description
items	list of items	<p>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:</p> <ul style="list-style-type: none"> 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. <p>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.</p>
selection mode	single selection, single selection radio buttons: right; single selection radio buttons: left; multiple selection	Specifies the behavior as well as the visualizations for the list box.
on select	none	<p>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 <code>CTRL</code> + <code>Space</code> to see the list of available methods for the</p>

Property Type	Property Value	Property Description
		application, the data source alias and the list box. Choose one of them.
		For further information, read the relevant method description in the Component API reference.

49.9.14 Progress Indicator

The *Progress Indicator* component provides a graphical representation of the progress of a process. To indicate the progress, the Progress Indicator is filled with a color.

i Note

This component is only available in the SAPUI5 m rendering mode.

The *Progress Indicator* component has the following specific properties:

Property Type	Property Value	Property Description
Tooltip	true, false	Specifies whether the component has a tooltip at runtime and which tooltip.
Text	some text	Specifies the text displayed on the bar..
Percentage Value	value	Specifies the progress expressed as a number between 0 and 100
State	none (default), error, warning, success	Specifies the state of the bar. Depending on the state, the color of the percentage value bar changes: <ul style="list-style-type: none"> • none - blue color • error - red color • warning - orange color • success - green color

49.9.15 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:

Property Type	Property Value	Property Description
items	list of items	<p>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:</p> <ul style="list-style-type: none"> 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. <p>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.</p>
columns	numeric value for columns, default: 1	<p>Specifies the number of columns for the radio button group. The specified columns are filled by the system with the specified items.</p> <p>Example: You have specified five items and two columns. The items are displayed as follows:</p> <ul style="list-style-type: none"> line 1: item 1 and item 2 line 2: item 3 and item 4 line 3: item 5 <div> <p>→ Remember</p> <p>If you specify several columns, you need to enlarge the width of the component. Otherwise the columns will not be fully or correctly displayed.</p> </div>
on select	script	<p>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 <code>CTRL</code> <code>+Space</code> to see the list of available methods for the application, the data source alias and the radio button group. Choose one of them.</p>

Property Type	Property Value	Property Description
		For further information, read the relevant method description in the Component API reference.

49.9.16 Segmented Button

The *Segmented Button* basic component is a horizontal component comprising a configurable number of buttons. These buttons can display text, images or both. You add buttons, text, and images using the *Edit Entries* dialog box. The *Segmented Button* responds automatically and resizes the buttons to fit proportionally within your component. The number of entries listed within the *Segmented Button* is displayed in the *Items* property. You can script it to interact with another component, for example, a Pagebook.

Note

- The Segmented Button is only available in *SAP UI5 m Mode*.
- The scripting method `setHeight()` and the component property *Height* have no effect on the *Segmented Button*. The *Segmented Button* responds automatically to accommodate the buttons added by you.

Property	Property Value	Property Description
items	list of items	<p>Open the <i>Edit Entries</i> dialog box. Here, you can insert items, change the order of items or remove them. Each item has the following properties:</p> <ul style="list-style-type: none"> • value The value property is mandatory for all items. • text (optional) The text property is optional. Enter a text for an item, and the text is displayed. If no text is displayed, the value for the item is displayed. • Icon The icon property is optional. Enter an image or icon for an item, and the icon is displayed.

49.9.17 Shape

The *Shape* component allows you to display scalable vector graphics (SVG) in your application. You can select an SVG shape by setting the *Shape URL* property. Enter a URL or data URL, or select an SVG file. You can also select one of the predefined shapes from the *Additional Properties* view. A data URL starts with "data:" and contains the file content encoded in the URL string.

The [Shape](#) component has the following specific properties:

Property	Property Value	Property Description
Line Color	color	Specifies the color of the shape line.
Fill Color	color	Specifies the fill color of the shape.
Line Width	numeric value in pixels	Specifies the line width of the shape. Enter the numeric value in pixels.
Shape URL	URL	Specifies the URL or data URL of the shape. The former must reference a file stored on the BI platform. To automatically create the valid URL, use the value help for the Shape URL to pick an SVG file.
Tooltip	text	Specifies the tooltip that is displayed when the user hovers over the component at runtime. Type in the text that should be displayed as a tooltip.
On Click	script	Opens the script editor. With this property/event, you can enable user interaction with the image by writing scripts. Click <code>CTRL</code> + <code>Space</code> to see the list of available methods for the application, the data source alias and the image. Choose one of them. For further information, read the relevant method description in the Component API reference.

49.9.18 Switch

The Switch component is a simple on/off switch designed for mobile devices.

i Note

This component is only available in the SAPUI5 m rendering mode.

The [Switch](#) component has the following specific properties:

Property	Property Value	Property Description
Tooltip	true, false	Specifies whether the component has a tooltip at runtime and which tooltip.
On	true, false	Specifies the state of the component.
Mode	OnOff, Blank, Accept/Reject	Specifies the mode of the component. The default shows <i>On</i> and <i>Off</i> translated to the current language. <i>Blank</i> does not have any text.

Property	Property Value	Property Description
		AcceptReject shows accept and reject icons.
On Change	none	Opens the script editor. With this property/event, you can enable user interaction with the switch by writing scripts. Click <code>CTRL</code> + <code>+</code> + <code>Space</code> to see the list of available methods for the application, the data source alias and the switch. Choose one of them. For further information, read the relevant method description in the Component API reference.

49.9.19 Text

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:

Property	Property Value	Property Description
Text	Some Text	Here you can enter the text that you want to add to your application.
CSS Style	YourCssStyle	<p>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, <code>color: pink;</code>. You can specify one of the following CSS attributes:</p> <ul style="list-style-type: none"> • font size • font color • weight (bold) • style (italic) • text decoration (underline) <p>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.</p>

Property	Property Value	Property Description
		! 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.
tooltip	yourTooltipText	Specifies the tooltip text for the component.

❖ Example

An example of CSS code for formatted text.

```
font-size: 15px; weight: bold; font-style: italic; text-decoration: underline; color: red;
```

49.9.20 Text Area

The [Text Area](#) component is an input field that allows multiple line input.

i Note

This component is only available in the SAPUI5 m rendering mode.

The [Text Area](#) component has the following specific properties:

Property	Property Value	Property Description
Editable	true, false	Specifies whether the text area is editable at runtime. Non-editable text areas do not allow you to change the multi-line text.
Value	some text	Opens the text editor where you can enter text in lines.
Tooltip	true, false	Specifies whether the component has a tooltip at runtime and which tooltip.
On Change	none	Opens the script editor. With this property/event, you can enable user

Property	Property Value	Property Description
		interaction with the text area by writing scripts. Click <code>CTRL</code> + <code>+</code> + <code>Space</code> to see the list of available methods for the application, the data source alias and the text area. Choose one of them.. For further information, read the relevant method description in the Component API reference.

49.9.21 Toggle Button

You use the [Toggle Button](#) component to enable quick interaction possibilities within your application for the application user. When the application user clicks on the toggle button at runtime, it remains pressed until the user clicks on it again.

The [Toggle Button](#) component has the following specific properties:

Property	Property Value	Property Description
Tooltip	text	Specifies the tooltip that is displayed when the user hovers over the component at runtime. Type in the text that should be displayed as a tooltip.
Text	default text: toggle button	Specifies the text to be displayed on the toggle button. You can overwrite the default text and enter your own text. The text is centered on the toggle button.
Pressed	false, true	Specifies whether the button is pressed. The default is false .
Icon	name of icon; save location	Specifies the icon to be displayed on the toggle button. Click on the Browse.. button right to the property and choose in the Open Image (for the SAP Lumira Documents mode) or Select Local Image (for the local mode) dialog box, the icon you want to add.
On Click	your script	Opens the script editor. With this property/event, you can enable user interaction with the text area by writing scripts. Click <code>CTRL</code> + <code>+</code> + <code>Space</code> to see the list of available methods for the application, the data source alias and the Toggle Button. Choose one of them. For further information, read the relevant method description in the Component API reference.

49.9.22 Tree

The [Tree](#) basic component can be bound to a hierarchical dimension in a data source. If the dimension has no hierarchy, the [Tree](#) component will display a simple list. If the dimension has a hierarchy, the [Tree](#) component will reflect the actual result set as it appears in the crosstab. If a change is made within the [Tree](#) component, the result set in the crosstab and other components bound to the same data source will change accordingly. Similarly, if you change the result set in the crosstab, the [Tree](#) component will also change. The columns of the [Tree](#) component reflect the display settings of the dimension. For example, if you change the dimensions display settings from Text and Key to Key in the context menu on the crosstab, the tree will display only one column. Your selection will persist if you select a dimension member in the [Tree](#) component and then change the order of your dimension members, for example, by sorting. If you select a child element in the hierarchy and then collapse the parent node, your selection is not persisted.

Note

- The [Tree](#) component does not support line breaks when displaying members. It only supports single line texts.
- It does not support hierarchies that open upwards.

Property	Property Value	Property Description
display column headers	false,true	Specifies whether column headers are displayed in the Tree component. The default value is false.
tree items	bound, not bound	Specifies whether the component is bound to a data source.
(source) type	hierarchical dimension members binding	Specifies the binding type.
(source) dimension	dropdown list	Specifies the dimension on the data source bound to the component.
(source) maximum number of entries	integer	Specifies how many entries are displayed. The default value is 100.
selection mode	single, multi	Specifies how many members can be selected.

Related Information

[Adding a Data Source \[page 113\]](#)

[Binding the Properties of Standard Components to Data Sources \[page 145\]](#)

[Assigning Bookmarks to a Folder \[page 220\]](#)

49.9.23 Visual Timer

The *Visual Timer* is a basic component.

In contrast to the technical component *Timer*, the *Visual Timer* component is visually displayed. You can use it to execute scripts periodically or to start scripts when the user interacts with the application. It is also recommended to use this component with the `releaseAndSuspend` method of the application object to release all server-side resources held by this application.

The *Visual Timer* component has the following specific properties:

Property	Property Value	Property Description
Visible	true, false	Specifies whether a component is visible or not. In contrast to all other components the <i>Visual Timer</i> component keeps running in the background, if the property is set to <i>false</i> and if the <i>Visual Timer</i> is running.
Duration	positive numbers without separator	Specifies the time interval in seconds when the <i>On Timeout</i> event is called. Type in positive numbers without separator.
Restart Timer	Never, On Interaction, On Timeout	<p>Specifies the restart option for the <i>Visual Timer</i> component. The following restart options are offered:</p> <ul style="list-style-type: none">• Never If you choose this option, the visual timer runs only once and is not restarted again at all.• On Interaction If you choose this option, the visual timer is restarted when the application user interacts with the application.• On Timeout If you choose this option, the visual timer is restarted right after the end of the duration time.
CSS Class	your CSS class or a predefined CSS class	<p>Defines an additional CSS class for custom CSS. The CSS classes must have the format myclass (and not .myclass). You do not have to use this property to be able to use <i>Custom CSS</i>.</p> <p>With Lumira Designer you get the following predefined CSS classes for the <i>Visual Timer</i> component:</p> <ul style="list-style-type: none">• zenTimerStarted

Property	Property Value	Property Description
		<ul style="list-style-type: none"> zenTimerStopped zenTimer zenTimerElapsed <p>The zenTimerElapsed css class will only be applied to the visual timer component if the Restart Timer property has been set to On Interaction.</p> <p>To be able to use these CSS classes you have to add them using the Custom CSS property of the application.</p>
On Timeout	your script	<p>Specifies an action to be triggered when the timer interval has elapsed. This opens the script editor. With this property/event, you can enable user interaction with the visual timer by writing scripts. Click <code>CTRL</code> + <code>+</code> + <code>Space</code> to see the list of available methods for the application, the data source alias and the visual timer. Choose one of them. For further information, read the relevant method description in the Component API reference.</p>

49.10 Container Components

49.10.1 Adaptive Layout

Use the [Adaptive Layout](#) to create an adaptive application that can be viewed on multiple devices.

The [Adaptive Layout](#) allows you to preview different layouts at design time, so you can develop one application that will work on large or small devices. It has the following specific properties:

Property Type	Property Value	Property Description
vertical spacing	none, small, medium (default), large	The vertical spacing between blocks.
horizontal spacing	none, small, medium (default), large	The horizontal spacing between blocks.
minimum width of medium viewport	default value 640	The width at which the layout of the application transitions from a small to a medium viewport.

Property Type	Property Value	Property Description
minimum width of large viewport	default value 1024	The width at which the layout of the application transitions from a medium to a large viewport.
minimum width of extra large viewport	default value 1920	The width at which the layout of the application transitions from a medium to an extra large viewport.

49.10.2 Dialog

You use the [Dialog](#) component to create a dialog box with or without buttons. The dialog can only be used in the root layout and you can only set the height and width of the component. You can place further elements and items or components inside the dialog but not in the header area or footer.

i Note

This component is only available in the SAPUI5 m rendering mode.

The [Dialog](#) component has the following specific properties:

Property Type	Property Value	Property Description
draggable	true, false (default)	Specifies whether the dialog component is draggable during runtime. If false is selected, the dialog component cannot be moved and will always be centered on the screen.
resizable	true, false (default)	Specifies whether the dialog component can be resized during runtime. If false is selected, the setting of the properties Width and/or Height cannot be changed by the application user.
fullscreen	true, false (default)	Specifies whether the dialog component is displayed in fullscreen mode. If set to true , the Draggable and Resizable properties are ignored and the dialog component extends to the full screen size.
title	text	Specifies the title to be displayed in the header area (title bar).
buttons	none	Specifies the functionality and behavior of the buttons within the footer. When you click on the ... button in the property sheet, the Add/Edit Dialog Footer Buttons dialog box is displayed. Here you can insert and remove buttons and edit the button

Property Type	Property Value	Property Description
		script. For each button, you can specify the button text and whether it is enabled and/or visible.

49.10.3 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:

Property Type	Property Value	Property Description
number of rows	numeric value, default: 1	Specifies the number of rows in the grid.
number of columns	numeric value, default: 1	Specifies the number of columns in the grid.

49.10.4 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.

i 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:

Property	Property Value	Property Description
selected page index	index	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.
page pre-loading	off, previous and next page, all pages	Specifies the strategy for loading pages.

Property	Property Value	Property Description
show page indicator	true, false	<p>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.</p> <div> i Note <p>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 five. If you are on page two, you can only change to page one or three.</p> </div>
show navigation controls	true, false	<p>Shows buttons to the right and left of the current page to navigate between pages. The buttons become visible when the mouse hovers over the respective areas. Set this to false and disable the property Show Page Indicator if your pagebook navigates between pages using scripts and interactive navigation by users should not be possible.</p>
on select	none	<p>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.</p> <p>For further information, read the relevant method description in the Component API reference.</p>

49.10.5 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:

Property	Property Value	Property Description
CSS style	CSS style	<p>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, <code>color: pink;</code>. You can specify one of the following CSS attributes:</p> <ul style="list-style-type: none"> • font size • font color • weight (bold) • style (italic) • text decoration (underline) <p>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.</p> <div> <p>i Note</p> <p>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.</p> </div>
on click	none	<p>Opens the script editor. With this property/event, you can enable user interaction with the panel by writing scripts. Click <code>CTRL +Space</code> to see the list of available methods for the application, the data source alias and the panel. Choose one of them.</p> <p>For further information, read the relevant method description in the Component API reference.</p>

49.10.6 Popup

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).

i Note

Please be aware of the following items for the popup component in SAPUI5 m mode:

- It is possible to open the popup (in SAPUI5 m mode) centered. To open the popup centered, use the function `open(true)` (see also API section for more details).
- If multiple popups are opened in one script call, the foreground order is determined by the order in the application, not by the order in the script.
- If multiple popups are opened one after another in separate script actions (one roundtrip per script action), then the execution order determines the foreground order. The last executed popup will be the foremost.

The *Popup* component (common mode) has the following specific properties:

Property	Property Value	Property Description
modal	true, false	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.
autoclose	true, false	Specifies whether the component is automatically closed, when the user interacts outside the popup but inside the application.
animation	no animation, flip animation, pop animation, horizontal slide animation, vertical slide animation	Specifies the animation effect when the popup is displayed or closed.

Additional, changed or removed properties in SAPUI5 m Mode

Due to the different rendering in SAPUI5 m mode, the popup has a different properties sheet in the m mode:

- Property *Background Design* has been added.

Property	Property Value	Property Description
background design	true, false	Specifies the background design of the popup.

- Property [Animation](#) has been removed.

49.10.7 Scroll Container

Use the [Scroll Container](#) component to enable or disable horizontal or vertical scrolling within supported container components at runtime or design time.

The [Scroll Container](#) component has the following specific properties:

Property	Property Value	Property Description
horizontal scrolling enabled	boolean	Enable or disable horizontal scrolling.
vertical scrolling enabled	boolean	Enable or disable vertical scrolling.

49.10.8 Story

Use the [Story](#) component to create a presentation-style application that uses visualizations, text, graphics, illustrations, shapes, and other customizations to describe data.

Use the [Story](#) component to group and order the content of an application into tab-like pages. Explore and analyze the data using filters, controls, calculations, conditional text, and other tools. A story may have a single visualization or pages full of visualizations. When you add a [Story](#) to an application, it contains a [Filter Bar](#) on the [Story](#) level. The [Story](#) component allows you to apply a filter across multiple data bound components within the same [Story](#).

The [Story](#) component has the following specific properties:

Property Type	Property Value	Property Description
read-only story	boolean	Specifies if the story can be edited or if it is read only. The default value is false.
selected page index	integer	Specifies the first visible tab of the story. This index is zero based.
show filter bar	boolean	Specifies if a filter bar is displayed.
caching mode	none, all pages, dynamic	<p>Specifies the caching mode for the Story component.</p> <ul style="list-style-type: none"> • If the value None is selected only the current Story Page loads at a given point in time.

Property Type	Property Value	Property Description
		<ul style="list-style-type: none"> If the value All Pages is selected, all pages are cached and story pages load quickly when switching from page to page. If the value Dynamic is selected, story pages are cached based on demand. Story pages are cached after they have been selected.

49.10.8.1 Story Page

Use the [Story Page](#) as the location for the components in the [Story](#).

The [Story Page](#) has the following specific properties:

Property Type	Property Value	Property Description
text	none	The text that is displayed on the Story Page tab within the Story .
aspect ratio	standard (4:3); extended (16:9); BI launchpad; custom. (Default = extended)	The ratio of the width to the height of the Story Page .
background color	hex, color picker	You can use a color picker to specify the background color of the Story Page .
background image	none	You can specify the background image of the Story Page .
opacity	integer (default = 100)	You can set the transparency of the background image or the background color of the Story Page .
page width	integer	You can set the width of the Story Page . <div> i Note This property can be applied only when the Custom option is selected within the Aspect Ratio property. </div>
page height	integer	You can set the height of the Story Page . <div> i Note This property can be applied only when the Custom option is </div>

Property Type	Property Value	Property Description
		selected within the Aspect Ratio property.

49.10.9 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:

Property Type	Property Value	Property Description
selected tab index	index	Specifies the initially displayed tab of the tabstrip. The index 0 represents the first tab of the tabstrip, 1 represents the second tab, 2 represents the third tab etc.
on select	script	<p>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.</p> <p>For further information, read the relevant method description in the Component API reference.</p>

50 Component API Reference

The former API Reference is no longer part of the *Application Designer Guide*.

You can find the new Component API Reference on the SAP Help Portal page of SAP Lumira .

The new online format offers you the following possibilities and advantages:

- navigate the hyperlinks of the classes' documentation
- search classes
- group classes by name or type
- generate a print version of a class's documentation

51 Working with the Legacy Local Mode of Lumira Designer

This section provides information on differences you may encounter when working with Lumira Designer in the legacy local mode. Tasks that are similar to Lumira Designer when connected to the BI platform are not listed separately in this section. The relevant information is contained in the sections above.

You can use the [Local Mode \(Legacy\)](#) to create and save analysis applications on your local machine. In the legacy local mode, the technical prerequisites are minimal. Once the BI platform infrastructure has been set up in your company, application designers who have been working in local mode can easily switch the startup mode to [SAP BusinessObjects BI Platform \(Legacy\)](#) in the [Preferences](#) dialog box and connect to the BI platform. Application designers can then upload their locally saved analysis applications to the BI platform.

! Restriction

Local mode in Lumira Designer was not designed with offline use of Lumira Designer in mind. You should not use it for downloading analysis applications from the BI platform, changing and saving them locally, and uploading the applications again to the BI platform.

i Note

We recommend to use the [SAP Lumira Documents](#) mode. It allows both, working with Lumira documents in BI platform folders and also offline work in a folder on your local machine.

51.1 Launching SAP Lumira Designer

Context

To launch SAP Lumira Designer, choose ► [Start](#) ► [All Programs](#) ► [SAP Business Intelligence](#) ► [SAP Lumira](#) ► [Lumira Designer](#) . The design tool is launched with the welcome page.

Related Information

[Starting with the Welcome Page \[page 28\]](#)

51.2 Creating New Analysis Applications

Context

When creating new analysis applications, you can select different templates. You can choose between blank templates or predefined templates that correspond to various design and business needs. When you choose a 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. Enter a unique name for your application in the field under [Application Name](#).
 - b. Choose the template you want to use for your application. A short description of the template is displayed at the right side of the dialog box.
 - c. Choose [Create](#). The template is opened in the editor where you can change and edit it.

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 ready-to-run templates according to your needs. For more information, see the links in the *Related Topics* section.

Related Information

[Adding Components to an Application \[page 46\]](#)

[Adding a Data Source \[page 113\]](#)

[Assigning a Data Source to a Component \[page 144\]](#)

[Using the Script Editor \[page 362\]](#)

[Saving an Application \[page 149\]](#)

[Executing an Application \[page 518\]](#)

[Properties of the Application \[page 419\]](#)

51.3 Maintaining Settings in Lumira Designer in Local Mode

You can define settings for Lumira Designer 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, Lumira Designer starts in [SAP Lumira Documents](#) mode after installation. If you want to use SAP Lumira Designer in a different mode, select the required platform mode:

- [SAP Lumira Documents](#)
This is the default mode that supports the creation of Lumira documents, which may contain any number of analysis applications, composites, MIME objects and/or offline datasets created with Lumira Discovery. This mode can handle both local documents and documents stored on the BI platform.
- [SAP BusinessObjects BI Platform \(Legacy\)](#)
Supports the creation of analysis applications in the legacy format of Design Studio 1.6 that are stored on the BI platform. You can also open, edit, and save analysis applications created with Design Studio 1.6 on the BI platform in this mode.
- [Local Mode \(Legacy\)](#)
Supports the creation of analysis applications in the legacy format of Design Studio 1.6 that are stored on your local machine. You can also open, edit, and save analysis applications created with Design Studio 1.6 on your local machine in this mode.

After restarting Lumira Designer, you need to log on to the selected platform.

i Note

In the [Logon to <selected platform>](#) dialog box, you can still decide to log on locally by clicking [Work Offline](#).

- If the startup mode is set to [SAP Lumira Documents](#) SAP Lumira Designer starts in the local mode of the SAP Lumira Designer mode.
- If the startup mode is set to [SAP BusinessObjects BI Platform \(Legacy\)](#) SAP Lumira Designer starts in [Local Mode \(Legacy\)](#).

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 Lumira Designer.

Embedded Web Server

You can define the network port for the embedded Web server of Lumira Designer. 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. Lumira Designer 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 Lumira Designer that is running.

- SAP HANA data sources used in the analysis application: the application designer of the PC that Lumira Designer 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 Lumira Designer 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

Lumira Designer 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>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen`. The content of this file is encrypted using Eclipse secure store technology.

i 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 Lumira Designer and opening the affected application again, the application designer is informed that an auto-saved version of the application exists.

- If the application 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 application 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 application designers manually enter non-existent values. To activate the warnings, select the [Display warnings for manually entered invalid values](#) checkbox.

Script Editing

If you run into performance issues when editing complex scripts, you can disable the backend validation of your scripts by selecting [Disable backend validation](#). When backend validation is enabled, the script editor's automatic syntax checks involve calls to a backend server, for example to validate the name of a dimension or a hierarchy. This can slow down working with larger scripts significantly, especially if you are working over a WAN connection. For this reason, you can explicitly disable backend validation (leaving all other checks in place).

In addition, you can disable the inplace help of the script editor by selecting [Disable inplace help](#).

i Note

If you experience performance issues on operating systems with limited memory and large applications with complex scripts, SAP recommends to disable the inplace help.

These script editing settings are also available in the script editor. For more information, see “Content Assistance in the Script Editor” in the *Application Designer Guide: Designing Analysis Applications* under [► Help > Help Contents ►](#) in Lumira Designer.

Prompt Handling

In the [Prompts](#) dialog box, application designers and application users set values for prompts. For SAP 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 Lumira Designer (if there are mandatory variables without default values, or if variables have invalid default values, for example). SAP Lumira Designer stores all valid prompt values of each analysis application in the user's cache file in `<user home directory>\LumiraDesigner-workspace\.metadata\.plugins\com.sap.ip.bi.zen\cache`. This provides application designers with a smooth workflow in Lumira Designer. 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.

i 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 Lumira Designer 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 Lumira Designer 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 Lumira Designer.

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 Lumira Designer.

Network Connections

Under ▮ ▮ [Tools](#) ▸ [Preferences](#) ▸ [Application Design](#) ▸ [Network Connections](#) ▮, you can specify the proxy settings to be used when opening connections.

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. For more information, see the links in the *Related Information* section below.

Under *SAP HANA Data Sources (HTTP)*, you can specify that you use an HTTP proxy for communication with the SAP HANA system. If you have configured that your SAP HANA system can only be accessed using a proxy, select the *Use HTTP Proxy* checkbox.

- Under *Proxy Host*, enter the name of the system hosting the proxy service used by the HTTP destination.
- Under *Proxy Port*, enter the port to connect on the system hosting the proxy service.

You can also use this setting for support purposes: Using Fiddler as a proxy, you can record Fiddler traces, which help SAP diagnosing your issues with SAP HANA HTTP data sources. To set Fiddler as a proxy, select the *Use HTTP Proxy* checkbox.

- Under *Proxy Host*, enter **localhost**.
- Under *Proxy Port*, enter **8888**.

For more information, see SAP Note [2166049](#) .

In the status line of Lumira Designer, the indicator *HTTP Proxy: Off/On* shows you if this setting is switched on or not. By double-clicking on the indicator you can quickly access the *Preferences* page to change your current setting.

Scripting

Under **Tools** > **Preferences** > **Scripting**, you can specify settings for syntax coloring and maintain templates for scripting.

Related Information

[Activating Runtime Traces \[page 414\]](#)

[Activating SAP JCo Traces \[page 414\]](#)

[Configuring the Report-Report Interface for Analysis Applications in Local Mode \[page 521\]](#)

51.4 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`

- Internet or intranet

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.

51.5 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 Lumira Designer can access SAP HANA systems or SAP 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 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.

i 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](#).

Related Information



[Selecting a Data Source \[page 117\]](#)

51.5.1 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 Lumira Designer can access SAP HANA systems or SAP 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 BW connection or a SAP HANA connection.
 - For SAP BW connections, click  [Launch SAP Logon](#) in the upper right corner of the [Connections maintained in SAP Logon](#) area.
 1. In the [SAP Logon](#) dialog box, click [New](#).
 2. 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.
 1. Click [Add...](#)
 2. Select the driver [HDBODBC32](#).

i Note

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.

51.6 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](#) . 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.

51.7 Executing an Application

Context

You can always execute an application locally in your Web browser while you are working on it, and you don't need to save it first. This enables you to easily check your design steps in the application.

Procedure

Click ► [Application](#) ► [Execute Locally](#) ► in the menu or toolbar. The application is displayed in your Web browser.

51.8 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 ► [Tools](#) ► [Preferences](#) ►.
- Execute the application on your desktop Web browser.

Context

To execute the 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 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 *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.

51.9 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.

51.10 Coordinating the Translation of Translatable Texts in Legacy 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 Lumira Designer, the original texts entered by the application designer are always displayed.

Example

The `localization.properties` file can contain the following texts:

```
# XMSG  
BUTTON_1.TEXT=Filter
```

```
# XMSG
CHECKBOX_1.TEXT=Show Totals
# XMSG
TEXT_1.TEXT=Please enter your filter criteria
# XMSG
TEXT_POOL.Key1=This is {0}.
```

51.11 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

- “Using the Report-Report Interface” on SAP Help Portal at <https://help.sap.com>
- “Query as a Recipient” on SAP Help Portal at <https://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 Lumira Designer and the target query is displayed as an analysis application, you need to specify a generic analysis template.

Procedure

1. In Lumira Designer, 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 SAP Lumira 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.

52 Terminology Essentials

Before you read any further, it might be useful to understand some basic OLAP and analysis design tool terminology.



(analysis) application	An entity created in the design tool of SAP Lumira Designer. An application usually consists of user interface components like charts, crosstabs and buttons and data source aliases.
connection	A representation of BW or SAP HANA systems (in the design tool) that is used to add data sources.
(user interface) component	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.
data source	A BW query or query view, a SAP HANA analytic or calculation view, used in the design tool.
data source alias	An instance of a data source at runtime of an application and an entity in the design tool.
data binding	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.
dimension	A collection of related data members, which represents one aspect of a business; for example, products or sales.
event	A system notification about a specific user interaction, for example, a click on a button.
Lumira document	File which may contain any number of analysis applications, composites, MIME objects and/or offline datasets created with Lumira Discovery.
measure	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.
script	Series of statements which are created by the user of the design tool using the script editor. 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.
script editor	A tool within the design tool to specify the actions that should take place when an event is triggered by an application user.
statement	A programmatic instruction within a script. The execution of a statement is typically triggered by user interaction with the component.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.