PUBLIC

SAP BusinessObjects Business Intelligence platform
Document Version: 4.1 Support Package 8 – 2016-07-04

SAP BusinessObjects BI Developer's Guide for Web Intelligence and the BI Semantic Layer

Content

1 Document Version History. . . . 4

2 About This Guide. . . . 6

3 Audience. . . . 7

4 Conventions in This Guide. . . . 8

5 Setting Up Web Intelligence as an Embedded Applet. . . . 9

5.1 Prerequisites. . . . 9

5.2 Where to Find the Embedded Applet. . . . 10

5.3 Package Content. . . . 10

The Embedded Applet Utility. . . . 10

The Embedded Applet Properties File. . . . 10

The webiApplet Folder Content. . . . 11

The JSP and Image Sample Files. . . . 11

5.4 Managing the Locale. . . . 12

5.5 Managing the Session. . . . 13

5.6 Deploying the Embedded Applet Sample. . . . 13

To Create the Deployment Root Directory. . . . 13

To Deploy the Embedded Applet. . . . 13

To Test the Deployment. . . . 14

6 Customizing Web Intelligence User Interfaces. . . . 15

6.1 Customizing Web Intelligence interface elements by user group and folders. . . . 15

Customization Interface. . . . 15

User Interface Elements tab. . . . 16

Features tab. . . . 25

Customization rules. . . . 26

To customize the Web Intelligence interface appearance. . . . 26

6.2 Web Intelligence content alignment. . . . 28

To set content alignment for the Web Intelligence Applet interface. . . . 28

7 Customizing Web Intelligence with UI Extension Points. . . . 29

7.1 About the JavaScript APIs. . . . 29

7.2 UI Extension Points Task Sequence. . . . 30

7.3 About the Extension Bundle. . . . 30

7.4 Where to Find the Bundle Host. . . . 31

7.5 Prerequisites. . . . 31

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

7.6 To Import the Bundle Host. 31
7.7 To Create an Extension Bundle. 32
7.8 To Declare the Extension Bundle Contribution. 32
7.9 To Implement the IExtension Interface. 33
getExtensionProperties. 33
getContribution. 34
7.10 To Append Contribution Files. 35
7.11 To Develop with the JavaScript APIs. 35
7.12 To Make the Extension Visible on the Interface. 38
7.13 To Test the Extension Bundle. 38
7.14 To Build the Extension Bundle. 39
7.15 To Deploy the Extension Bundle in Production. 40
7.16 About the Web Intelligence UI Extension Point Sample. 40
To Use the Extension Sample. 41

8 Exposing Web Intelligence Features with REST Web Services. 42

9 Consuming BI Semantic Layer Universes with REST Web Services. 43

10 Developing Applications to Design and Administrate Universes. 44

11 Creating a Data Access Driver. 45

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1 Document Version History

The following table provides an overview of the most important document changes.

Table 1:

Version	Date	Change
SAP BusinessObjects BI Developer's Guide for Web Intelligence and the BI Semantic Layer 4.1 Support Package 8	June 2016	You can hide data source items from the *Create a Document* dialog box (or *New Document* dialog box in Web Intelligence Rich Client), the *Query Panel*, and the *New Data Provider* dialog box in Design mode of SAP BusinessObjects Web Intelligence through customization in the CMC. See User Interface Elements tab [page 16]. SAP recommends you use the asynchronous mode when developing extensions, see To Develop with the JavaScript APIs [page 35]. This document now provides a note about WebGL support for extension points, see Customizing Web Intelligence with UI Extension Points [page 29].
SAP BusinessObjects BI Developer's Guide for Web Intelligence and the BI Semantic Layer 4.1 Support Package 6	June 2015	The Embedded Applet has been modified to follow the SAP branding standard. A new image folder must be deployed along with the applet. You can hide the SAP Marketplace button from the Status Bar of SAP BusinessObjects Web Intelligence through customization in the CMC. This document now provides sections on the JavaScript APIs used to develop an extension, and a link to the API reference. The chapter "Developing Applications to Design and Administrate Universes" now mentions the Universe Design Tool COM SDK. This document now provides a chapter about creating drivers with the Driver Development Kit.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Version	Date	Change
SAP BusinessObjects BI Developer's Guide for Web Intelligence and the BI Semantic Layer 4.1 Support Package 5	November 2014	First release of the document. The document contains the Extension Points documentation.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

2 About This Guide

The SAP BusinessObjects BI platform 4.1 is the key foundation for your analytics applications. It comes with a comprehensive set of tools from which you can pick up the one that suits the technologies you are ready to use and your business objectives. To this end, the *SAP BusinessObjects BI Developer's Guide for Web Intelligence and the BI Semantic Layer* is your new entry point to learn how to develop applications, using SDKs, samples, and extension framework, to enforce and take advantage of the Web Intelligence and BI Semantic Layer capabilities.

This guide provides the following information:

- A description of the Embedded Applet for Web Intelligence and how to deploy its sample in your portal
- How to customize the DHTML or Java Web Intelligence user interface via the Central Management Console
- How to extend Web Intelligence features using UI extension points
- How to use REST APIs to work with Web Intelligence documents and reports in non-SAP client tools
- How to use REST APIs to access universes and run queries in non-SAP client tools
- How to create, edit, secure and deploy universes with the BI Semantic Layer Java SDK
- How to create a JavaBean driver or an Open driver with the Driver Development Kit

This guide relates to the SAP BusinessObjects Business Intelligence platform 4.1 Support Package 8 release.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

3 Audience

As it serves as en entry point to the Web Intelligence and BI Semantic Layer customization area, the *BI Developer's Guide* is intended for various readers.

This guide is for you if:

- You are an SAP consultant who wants to help SAP partners and customers in their BI platform customization project
- You are an SAP partner who would like to provide customizations and extensions of Web Intelligence to their customer
- You are an SAP BusinessObjects administrator who wants to use Web Intelligence in their corporate portal
- You are a JavaScript developer responsible for developing extensions to Web Intelligence user interfaces
- You are a Java developer responsible for developing applications that perform creation, editing, and publication tasks on UNX and UNV universes
- You are a developer responsible for writing programs that access and consume the BI platform web services
- You are a developer responsible for developing data access drivers to help the BI platform to communicate with your company's data sources

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

4 Conventions in This Guide

In this guide, the placeholder `<bip-install-dir>` is the install root path of the SAP BusinessObjects Bi platform. On Microsoft Windows, the default `<bip-install-dir>` stands for the `C:\Program Files (x86)\SAP Business Objects\SAP BusinessObjects Enterprise XI 4.0` directory.

The placeholder `<tomcat-dir>` stands for the `C:\Program Files (x86)\SAP BusinessObjects\tomcat` directory.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

5 Setting Up Web Intelligence as an Embedded Applet

You can set up the Web Intelligence Java applet to run as an applet embedded in your own portal, rather than running it from the BI launch pad. The Embedded Applet provides the same functionality as the Web Intelligence applet that is launched through the BI launch pad.

End-users will be able to open Web Intelligence documents stored in your 4.1 BI platform repository directly from your portal. Using this applet, documents can be viewed, refreshed, printed, and saved locally as snapshots. No refresh will be available for documents saved locally.

A JSP sample demonstrating how to include the Embedded Applet in your own web application is also provided.

Restriction

- With the Embedded Applet, you cannot use the following services:
  - Scheduling reports
  - Checking the Document History
  - Using the 'Send To' function
- You cannot change the drill options managed via the BI launch pad Web Intelligence preferences.
- You cannot set a language different to your portal language. The Embedded Applet accepts the Language parameter, but your deployment will have to pass it to the applet. This is not exposed to end-users.

5.1 Prerequisites

Software Requirements

- The SAP BusinessObjects BI platform servers 4.1
- Apache Tomcat web application server
- Java SE Runtime Environment 7

User Rights

In the Embedded Applet, the user rights are managed in the same way as in the BI launch pad version of the applet: the CMC administrator assigns user rights and authorizations, and these settings are applied when the user launches the applet and logins to the session. If you use your own login method, then all users will be able to perform all operations.

To set the user rights for the Embedded Applet, you must have:

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

- Access to the Central Management Console (CMC)
- The authorization to edit the settings of the Web Intelligence Adaptive Processing Server
- The rights to manage the portal

5.2 Where to Find the Embedded Applet

The Web Intelligence Embedded Applet is a ZIP file installed by default with the SAP BusinessObjects BI platform servers. It is located in `<bip-install-dir>\Samples\webi\WebIntelligenceEmbeddedApplet.zip`.

5.3 Package Content

The `WebIntelligenceEmbeddedApplet.zip` archive file contains the following series of folders:

- `js`
- `jsp`
- `lib`
- `sample`
- `webiApplet`

Table 2:

Folder	Description
`js`	The Embedded Applet utility
`jsp`	The configuration files of the Embedded Applet
`lib`	The list of mandatory JAR files to make the Embedded Applet work
`sample`	The list of JSP and image sample files to create and manage a user session to the CMS
`webiApplet`	The Embedded Applet resources

5.3.1 The Embedded Applet Utility

The `applet_util.js` file helps the embedded applet to detect the web browser in which the applet is launched, the operating system of the machine, and the Java version.

5.3.2 The Embedded Applet Properties File

The `embeddedapplet.properties` file provides the necessary values for the configuration of the applet, which are used in the `appletpopup.jsp` file.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

The following table lists the property description and values that configure the applet sample.

Table 3:

Property	Description	Value
portalroot	The end of the portal URL	/BOE/portal/1303180624
portal_port	The port of the portal URL	8080
help_url	The end of the help URL	/AnalyticalReporting
gateway_url	The end of the gateway URL	/rebean3ws/services/Gateway
applet_url	URL of the applet on the web application server	/webiApplet

5.3.3 The webiApplet Folder Content

The `webiApplet` folder mainly contains the Embedded Applet resources:

- The Embedded Applet JAR file (`webiapplet.jar`)
- JAR files to manage applet localization in different languages
- Splash screens used when the applet is loading

5.3.4 The JSP and Image Sample Files

The Embedded Applet provides a series of JSP and image sample files that allow you to login to the applet in your portal. The image files are stored in the img subfolder.

> **Note**
>
> These files are only samples. You do not need to use them to create your own application.

Table 4:

JSP File	Description
`index.jsp`	The sample index file that represents the portal in which you want to use the Embedded Applet
`loginForm.jsp`	The form that you use to get login information. The form calls the `login.jsp` file.
`login.jsp`	The JSP file used to login the end-user to the portal and create an IEnterpriseSession. It directs to the `appletpopup.jsp` file.
`appletpopup.jsp`	The JSP file used to load the Embedded Applet in the portal with the session created in the `login.jsp` file.
`closeSession.jsp`	The file used to close the session. The login form page is displayed again after the session is closed.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Table 5:

Image File	Description
`center_normal_logon.gif`	Background image for the Logon button. It is used in the `LoginForm.jsp` file.
`uhBodyTextureTop.png`	Portal background image used in the `index.jsp` file
`uhLogo2.png`	SAP logo used in the `index.jsp` file

Embedded Applet CallBack

The `appletpopup.jsp` file calls the `AppletCallBack_updateDocumentTitle` function when the end-user is performing any of the following actions on a Web Intelligence document that could lead to a change to the current document name:

- Create
- Open
- Save as
- Close

This JavaScript function must be present in the parent of the iFrame which has the `appletpopup.jsp` file on it. It is implemented in the `index.jsp` file.

IEnterpriseSession Attributes

One of the roles of the `login.jsp` file is to provide the attributes of the IEnterpriseSession object with correct values. This enables the creation of the user session on the CMS.

Table 6:

Attribute	Description
`WebIEmbeddedApplet_EnterpriseSession`	The IEntrepriseSession object
`WebIEmbeddedApplet_CMSName`	The name of the CMS which the applet will connect to
`WebIEmbeddedApplet_PortalPort`	The sample gateway port used to create the gateway URL
`WebIEmbeddedApplet_ProductLanguage`	The locale to use for the applet localization

5.4 Managing the Locale

The locale is not the one managed through the end-user properties in the BI launch pad. You must set the locale through the `WebIEmbeddedApplet_ProductLanguage` attribute.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

5.5 Managing the Session

The Embedded Applet does not manage the lifecycle of the IEnterpriseSession object. So your portal implementation should take care of the creation and deletion of the session.

5.6 Deploying the Embedded Applet Sample

This section describes the deployment of the applet sample provided in the `WebIntelligenceEmbeddedApplet.zip` file.

5.6.1 To Create the Deployment Root Directory

Before you deploy the Embedded Applet, you should configure the web application server where you want to run the Embedded Applet, for example Apache Tomcat.

1. Stop Apache Tomcat.
2. Create the `EmbeddedAppletTest` folder as the root directory for the applet sample deployment on the web application server, under `<tomcat-dir>/webapps`.

   If your Apache Tomcat is brand new, the directories under `<tomcat-dir>/webapps` should look like the following:
   - `docs`
   - `EmbeddedAppletTest`
   - `examples`
   - `host-manager`
   - `manager`
   - `ROOT`

5.6.2 To Deploy the Embedded Applet

You must make sure that `JAVA_HOME` environment variable is set correctly, for example:

```
JAVA_HOME= C:\Program Files (x86)\Java\jre7
```

1. Copy to `<tomcat-dir>\webapps\EmbeddedAppletTest`:
   - The `webiApplet` folder
   - The `js` folder
   - The `sample` folder content, including the `img` subfolder
   - The `appletpopup.jsp` file from the `jsp` folder of the ZIP file

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

2. Create the `<tomcat-dir>\webapps\EmbeddedAppletTest\WEB-INF\classes` directory.
3. Copy all the properties files from the `jsp` folder of the ZIP file to this folder (`embeddedapplet.properties`, `webi_applet_jars.properties`, and `webi_applet_lang_jars.properties`).
4. Copy the `lib` folder to `<tomcat-dir>\webapps\EmbeddedAppletTest\WEB-INF`.
5. Start Apache Tomcat.

The final folder hierarchy should contain the following folders and files:

- `img`
- `js`
- `webiApplet`
- `WEB-INF`
- `appletpopup.jsp`
- `closesession.jsp`
- `index.jsp`
- `login.jsp`
- `loginForm.jsp`

The `WEB-INF` folder should contain the following subfolders:

- `classes`
- `lib`

5.6.3 To Test the Deployment

1. Open an Internet Explorer browser window on the same machine where you have deployed the Embedded Applet.
2. Go to `http://localhost:8080/EmbeddedAppletTest/`.
   You should see the login form displayed on a web page called Web Intelligence Embedded Applet.
3. Login to the portal as an end-user can do.
   This starts SAP BusinessObjects Web Intelligence as an applet embedded in the portal.
4. Perform any action on a Web Intelligence document, save your modifications and close the document.
5. Click *Logoff* in the upper-right corner of the web page to close SAP BusinessObjects Web Intelligence and the user session.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

6 Customizing Web Intelligence User Interfaces

You can simplify the appearance of Web Intelligence Rich Client and the DHTML and Java interfaces of SAP BusinessObjects Web Intelligence by hiding some functionalities through the CMC.

6.1 Customizing Web Intelligence interface elements by user group and folders

In the CMC, you can customize the appearance of Web Intelligence interface elements for a user, depending on the user group they belong to and the folders containing Web Intelligence documents. For example, an entire toolbar or specific items in a toolbar, and the access to specific document modes. You can also customize Web Intelligence by enabling extensions.

All interface elements appear by default. If you do not want specific elements to appear, you deselect them in the CMC. All extension points are disabled by default. If you want to make them available to users, you enable them in the CMC.

> **Note**
>
> - The customization and the enabled extension points are applicable to all Web Intelligence application clients: HTML, Java Applet, and Rich Client.
> - It might happen that the customization and enabled extension points do not work on Web Intelligence Rich Client because of proxy or DNS configuration. To solve this problem, log in to the CMC with the IP address of the server instead of the server name when you customize Web Intelligence. This IP address will be used as a reference during customization.

6.1.1 Customization Interface

The Customization section contains the following section and tabs:

- Customized folders section
  On this section, you can select folders containing Web Intelligence documents for which you want to customize user interface and enable extensions.
- User Interface Elements tab
  On this tab, you can select individual interface elements to hide, such as a toolbar or tab, or their subelements, for example a button command. You can also select data sources to hide when creating a new document (*Create a document* dialog box in the HTML and Applet interfaces and *New Document* dialog box in Web Intelligence Rich Client), adding a new query (*Query Panel*) and adding a new data provider to the document (*New data provider* dialog box) in Design mode.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

> **Note**
>
> Users can still edit existing documents based on data sources to which they don't have access.

- Features tab
  On this tab, you can choose to hide all user interface elements related to a function; for example, Refresh.
- Extensions tab
  On this tab, you can enable Web Intelligence user interface extensions that you have created and deployed in your installation.

6.1.2 User Interface Elements tab

The user interface elements that you can customize are identified in the subsequent sections and diagrams.

Splash Screen

You can hide the SAP BusinessObjects Web Intelligence splash screen that appears by default when a user opens Web Intelligence.

Application Contextual Menu

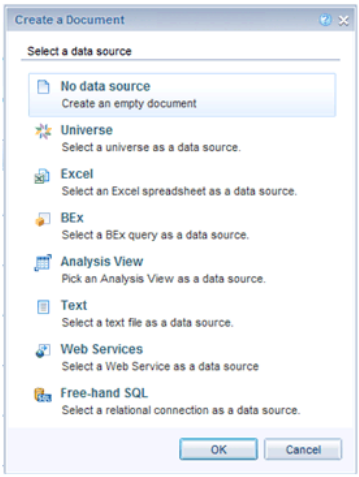
This menu appears when a user right-clicks in the Web Intelligence screen. You can hide the whole menu or any of its menu items.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

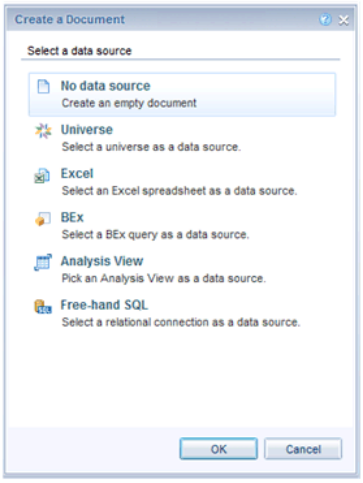
Create a Document Dialog Box

This dialog box appears when a user wants to create a document. You can hide any of the data sources that display in the dialog box. If all the data sources are selected or the *Create a Document* parent checkbox is selected in the CMC, then the dialog box does not display. A blank document is created instead.

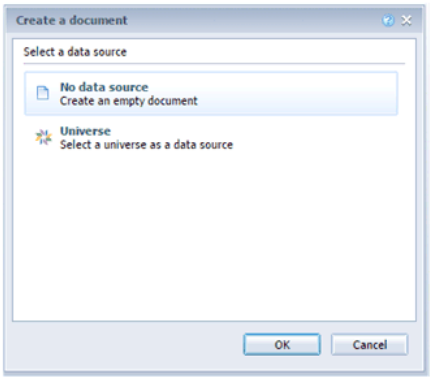
Web Intelligence Rich Client



Web Intelligence Applet interface



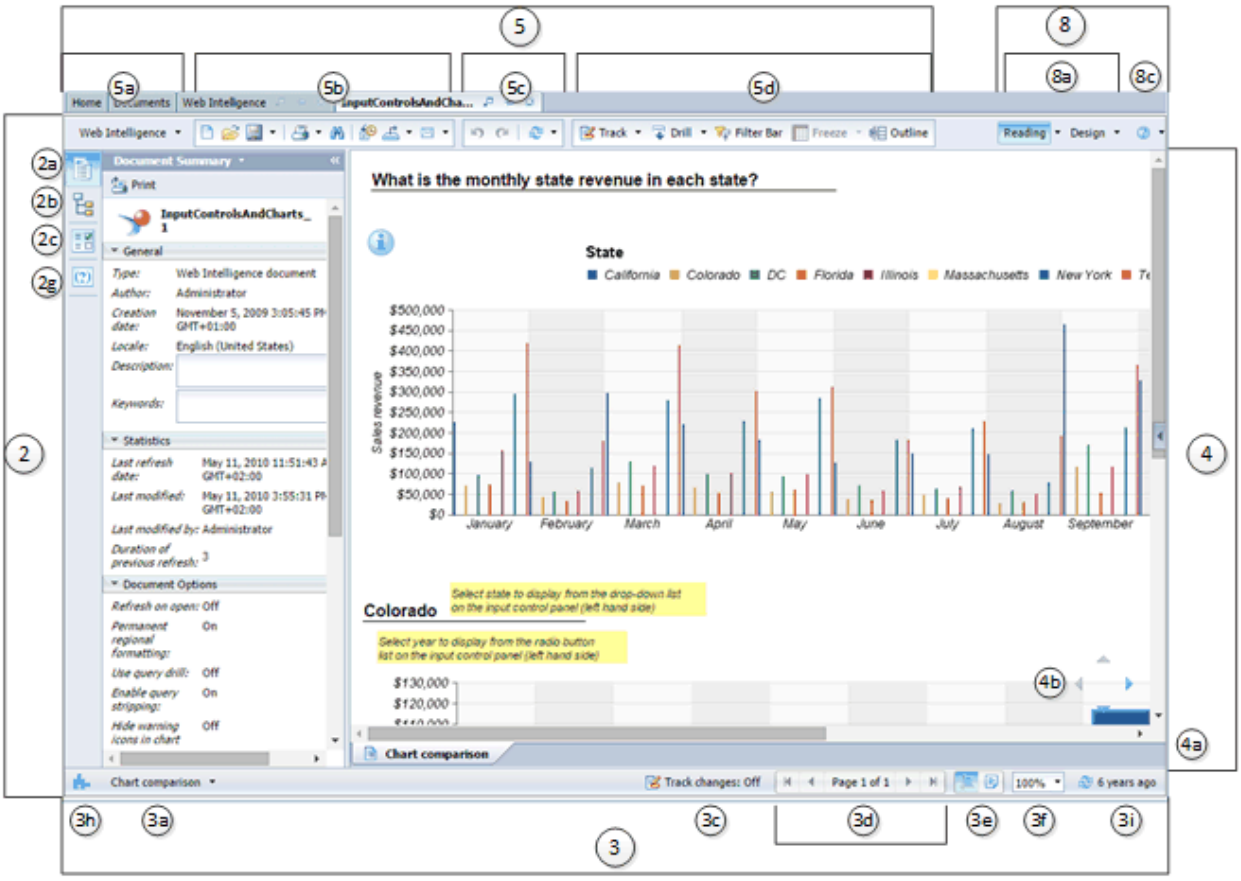
Web Intelligence HTML interface



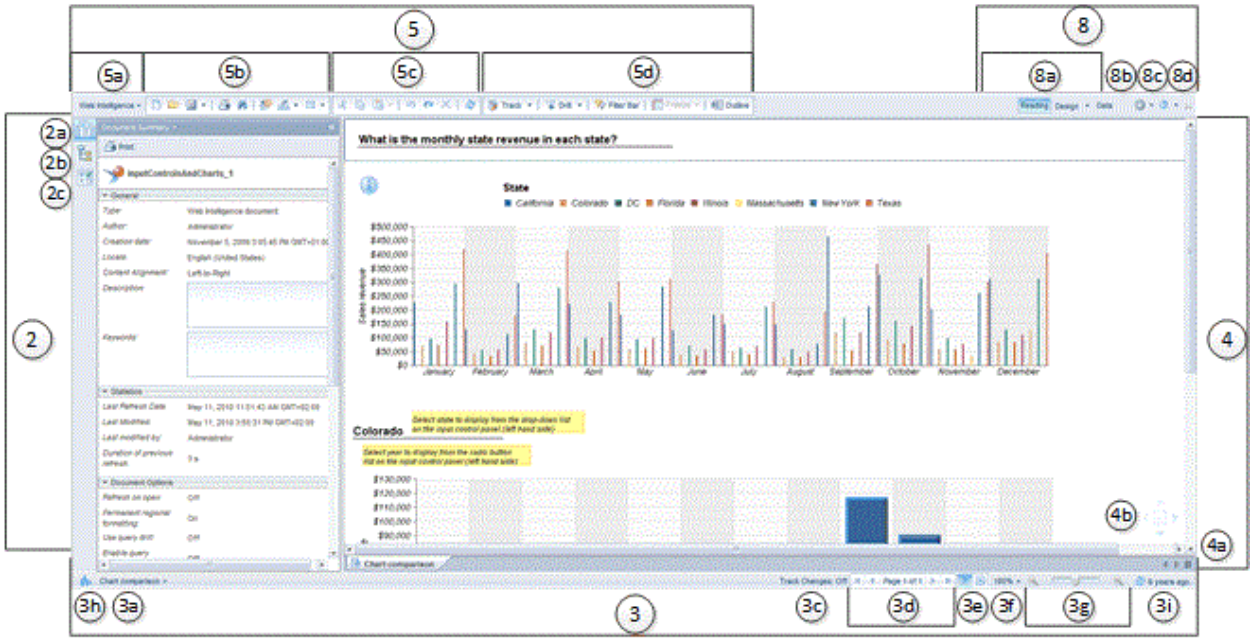
Find the next interface elements in the English examples of the Reading Mode below.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Web Intelligence HTML interface

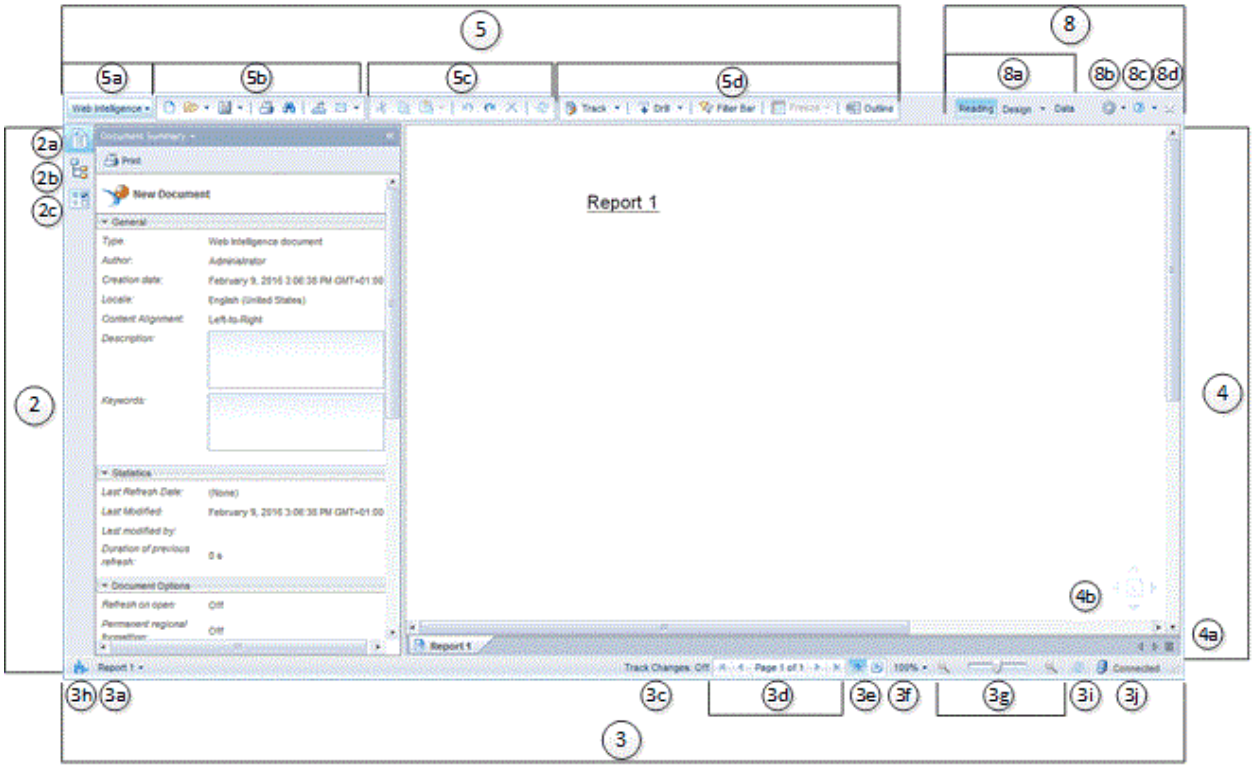


Web Intelligence Applet interface



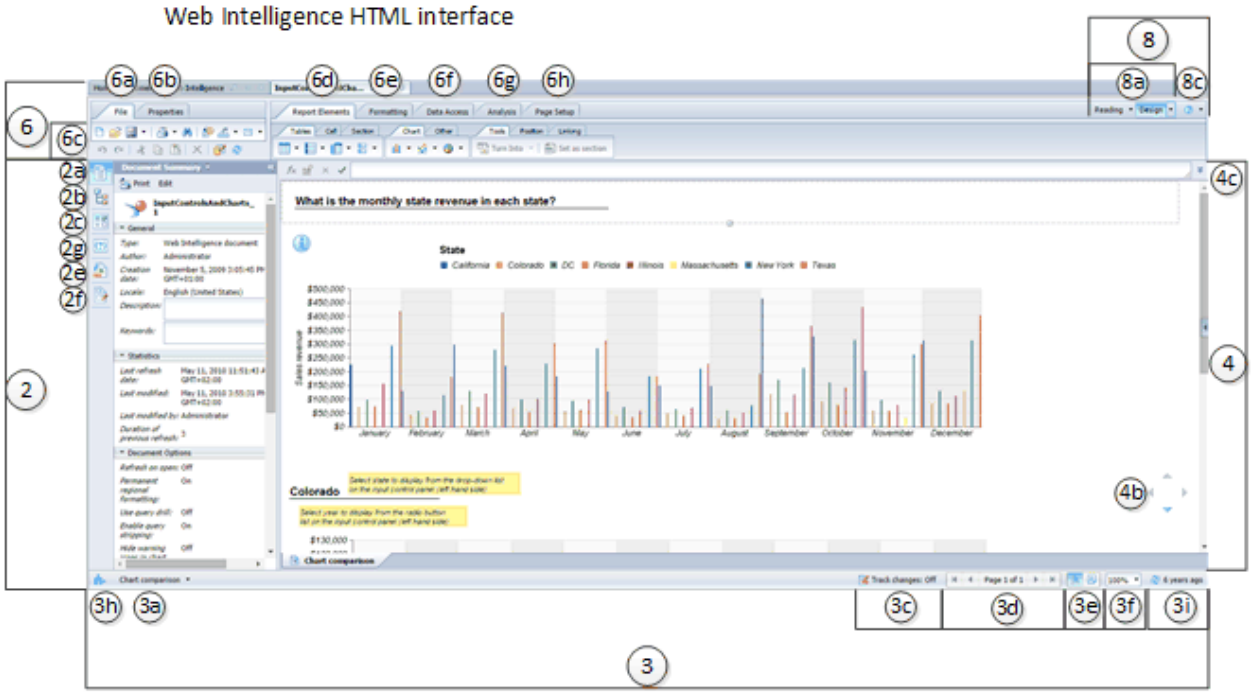
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Web Intelligence Rich Client



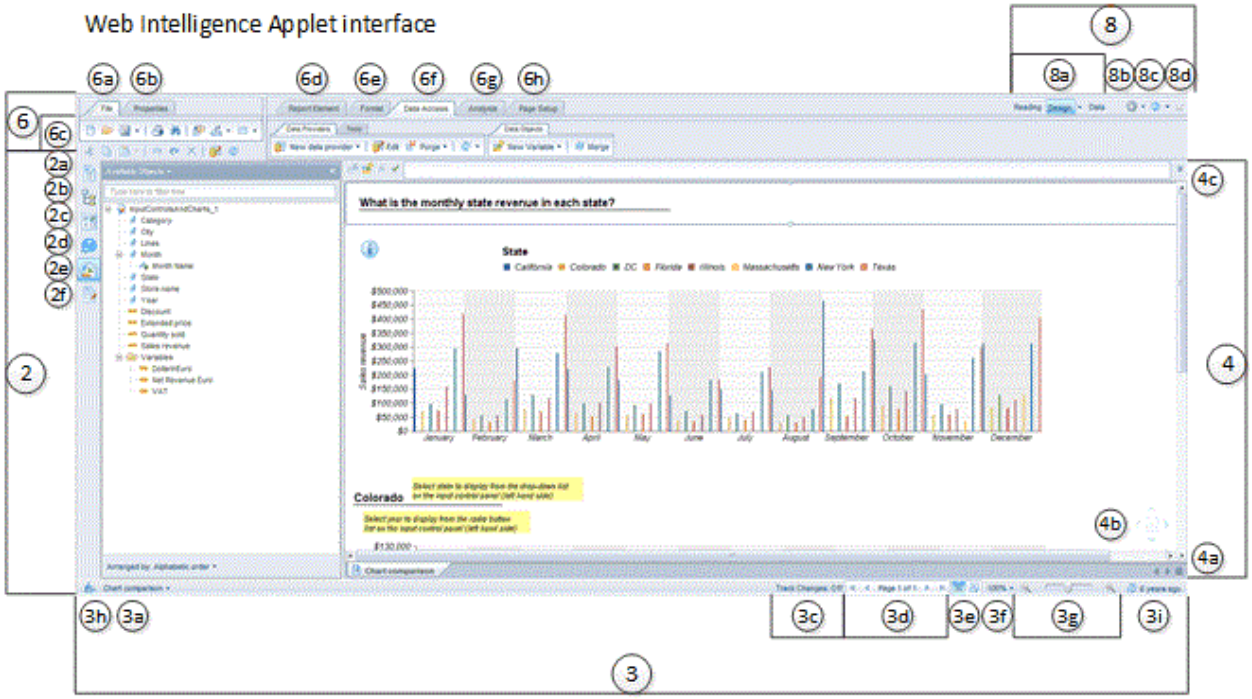
Find the interface elements in the English examples of the Design Mode below.

Web Intelligence HTML interface

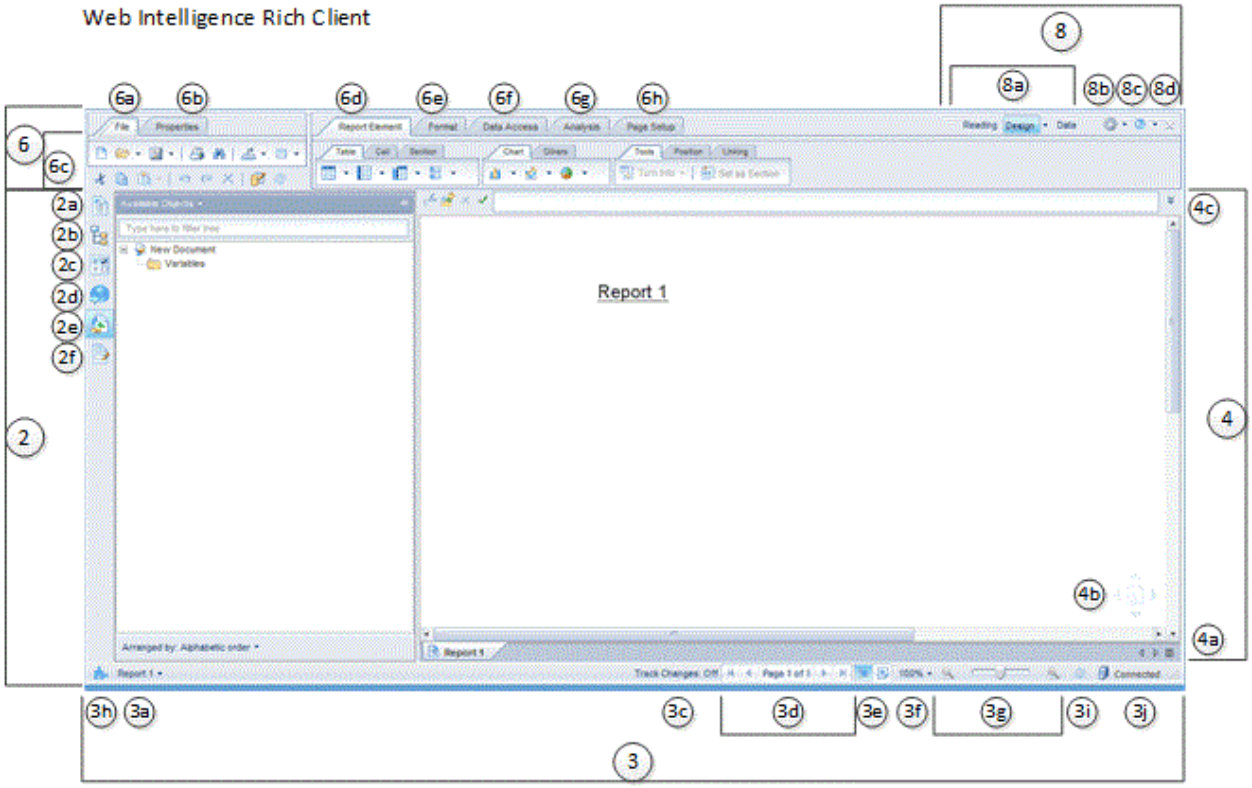


PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Web Intelligence Applet interface



Web Intelligence Rich Client



© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Side Panel

The Side Panel (2) next to the report panel allows users access to various information tabs.

Subelement List:

- Document Summary (2a)
- Navigation Map or Report Map (2b)
- Input Controls (2c)
- Web Service Publisher (2d)
- Available Objects (2e)
- Document Structure and Filters (2f)
- User Prompt Input (2g)
- Data (only in Data mode)

Status Bar

The Status Bar (3) is the bar where the user sees information on document action statuses and can perform zoom, page navigation, and formula bar activation tasks.

Subelement List:

- SAP Marketplace (3h)
- Report dropdown list (3a)
- Printing status icon (3b)
- Track data changes (3c)
- Page Navigation (3d)
- Pagination Mode (3e)
- Zoom List (3f)
- Zoom Slider (3g)
- Workspace Status

  The Workspace Status indicator (⚠) appears between the Zoom Slider and the Last Refresh Date if a problem occurs in the workspace.
- Last Refresh Date (3i)
- Connection Status (3j)

Report Zone

Subelement List:

- Report Tabs (4a)
- Bi-directional Page Scrolling (4b)
- Formula Bar (4c)

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Reading Mode Toolbar

You can hide the following toolbars (5) displayed in Reading mode.

Subelement List:

- Web Intelligence dropdown list (5a)
- File Group (5b)
- Standard Actions Group (5c)
- Analysis Group (5d)

Design Mode Toolbar

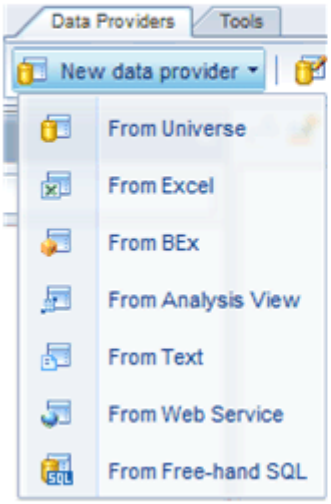
You can hide the toolbars and tabs (6) displayed in Design mode.

Subelement List:

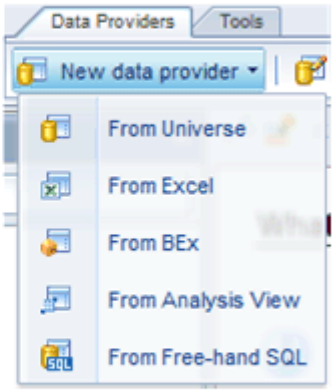
- File tab (6a)
- Properties tab (6b)
- Standard Actions Group (6c)
- Report Elements tab (6d)
- Format tab (6e)
- Data Access tab (6f)
- Analysis tab (6g)
- Page Setup tab (6h)

Under Data Access tab, a *New data provider* parent subtab allows you to choose the data source used when the user creates a data provider. You can hide the whole menu or any of its menu items.

Web Intelligence Rich Client



Web Intelligence Applet interface



© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Initial Toolbar

The initial toolbar (7) appears when a user opens the Web Intelligence application and when no document is open.

Subelement List:

- Web Intelligence dropdown list (7a)
- File Group (7b)

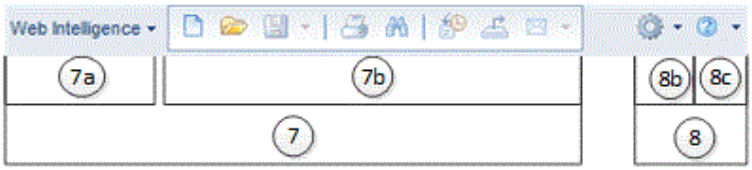
Application Control Toolbar

This toolbar (8) also appears in the upper toolbar of Web Intelligence when no document is open.

Subelement List:

- Application mode buttons (8a)
- Tools (8b)
- Help (8c)
- Close (8d)

Web Intelligence Applet interface
Web Intelligence Rich Client



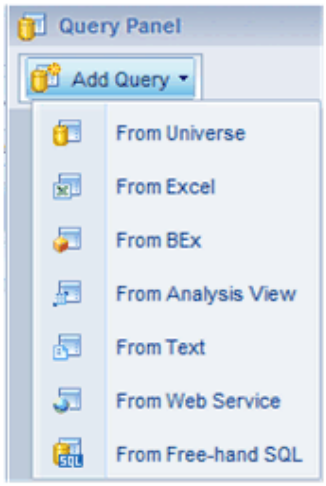
Web Intelligence HTML interface

Query Panel Toolbar

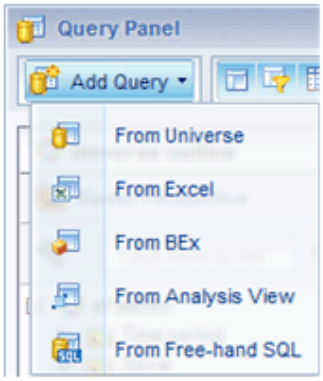
Options under *Query Panel Toolbar* > *Add Query* are used to choose the possible data sources when adding a new query. You can hide the whole menu or any of its menu items.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Web Intelligence Rich Client



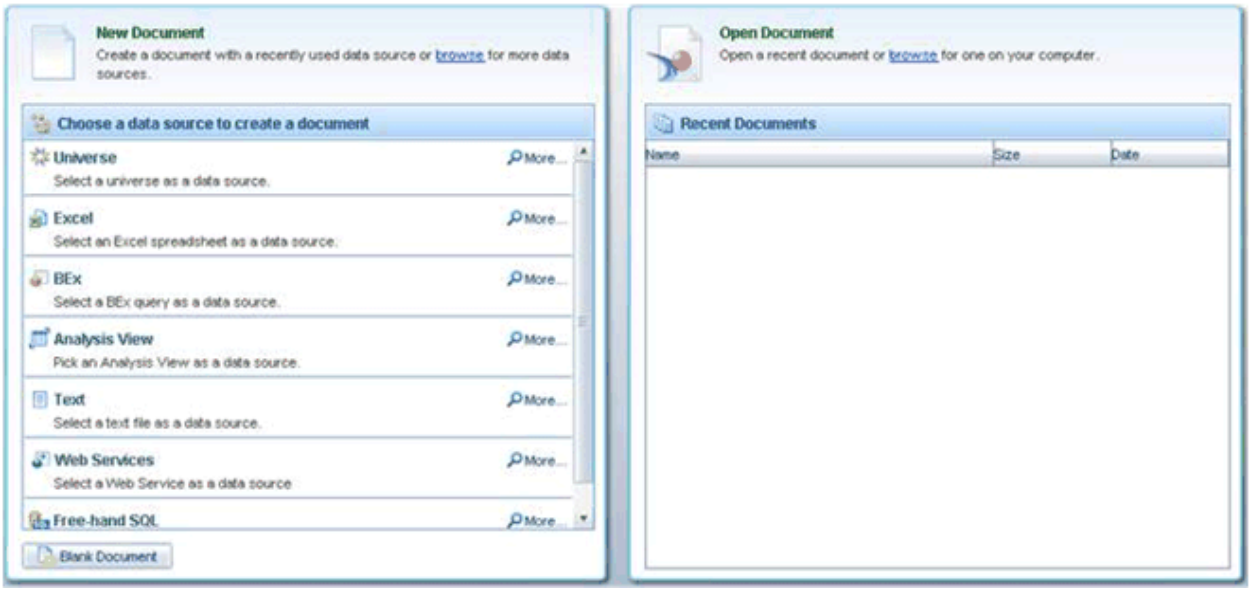
Web Intelligence Applet Interface



Web Intelligence HTML Interface

Welcome Screen

The *New Document - Open Document* dialog box appears when Web Intelligence Rich Client opens and allows the user to select a data source to create a document. You can hide any of the data sources that display in the dialog box. If all the data sources are selected in the CMC, then the dialog box is empty. If the *Welcome screen* parent checkbox is selected, then the dialog box does not display.



Shortcuts

Keyboard shortcuts, for example, CTRL + N or CTRL + S.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

> **Note**
>
> In Data Mode, the following items can be hidden:
>
> Table 7:

Web Intelligence Interface	Elements and Subelements

Applet	• Side Panel (2): Data icon
	• Status Bar (3): Last Refresh Date (3i)
	• Application Control Toolbar (8):
	  ○ Application mode buttons (8a)
	  ○ Tools (8b)
	  ○ Help (8c)
	  ○ Close (8d)

Rich Client	• Side Panel (2): Data icon
	• Status Bar (3):
	  ○ Last Refresh Date (3i)
	  ○ Connection Status (3j)
	• Application Control Toolbar (8):
	  ○ Application mode buttons (8a)
	  ○ Tools (8b)
	  ○ Help (8c)
	  ○ Close (8d)

6.1.3 Features tab

You can choose on this Feature tab to disable some functions at once, without having to disable them one after the other.

Table 8:

Feature Item	Description	Affects the following interface items
Refresh	Users can refresh documents to update the data from the data source.	The Refresh button in the Standard Action Group toolbar used in the Reading and Design modes.
Drill	Users can drill up and down on data in a document.	The Drill button available in the following locations: • The Analysis Group toolbar in the Reading mode. • The Interact subtab under the Analysis tab in Design mode.
Reading Mode	Users can view a document in Reading mode.	The Reading button in the following locations: • Application Contextual Menu • Application Control Toolbar

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Feature Item	Description	Affects the following interface items
Design Mode	Users can view a document in Design mode.	The Design button in the following locations: • Application Contextual Menu • Application Control Toolbar
Data Mode	Users can view a document in Data mode.	The Data button in the following locations: • Application Contextual Menu • Application Control Toolbar

6.1.4 Customization rules

The following rules are used to define customizations to apply to a user:

- If the user belongs to different groups, only the customization defined to the group whose ID is lower applies. The customization defined for the other groups containing the user does not apply.
- For nested folder structure, the immediate parent folder of the document that has been added in the list of customized folders define customizations for the document for user interface elements, features, and extensions.
- The customization defined for Default Folders apply for the documents stored in Personal Documents and Inboxes, and for documents for which the parent folder is not customized.
- The customization defined for user interface elements have priority over customization defined for features as features is only a shortcut to enable all user interface elements.

6.1.5 To customize the Web Intelligence interface appearance

You can customize the appearance of the Web Intelligence user interface by hiding menu items, subitems, and features for a selected User Group and document folder.

1. Log into the CMC as an Administrator.
2. From the *Organize* list, select *Users and Groups*.
3. In the *Group Hierarchy* list, select a user group.
4. In the *Actions* list, select *Customization*.
5. In the *Customized folders* section, do one of the following:

What You Want to Do	Steps
Defining the default customization	1. Select *Default Folders* in the *Customized folders* area.
Adding the document folders for which you want to apply customization for the selected user group	1. Click *Add Folder*. 2. Select the folders. The folders displays in the *Customized folders* area.
Avoiding redefining the same customization for other folders	1. In the *Customized folders* area, select the folder from which you want to copy the customization.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

What You Want to Do	Steps
	2. In the dropdown list, click *Duplicate Customization*. 3. Select the folder to which you want define the customization. 4. Click *Paste Customization*. 5. Go to step 7.
Removing the customization for a specific folder	1. In the *Customized folders* area, select the folder. 2. In the dropdown list, click *Remove Folder*. 3. Go to step 7. **Note** You cannot remove *Default Folders*.

6. Do one of the following:

What You Want to Do	Steps
Having items or extensions hidden in Web Intelligence	Deselect them in the *User Interface Elements*, *Features*, or *Extensions* tab.
Having hidden items or extensions appear in Web Intelligence	Select them in the *User Interface Elements*, *Features*, or *Extensions* tab.

If you select or deselect a parent item, then all its children are also selected or deselected.

If you deselect all children of a parent item, the parent item is not deselected. However it is hidden in Web Intelligence.

7. Click *Save & Close*.

When you save the customization, all users of the selected group will see these changes the next time they log on to BI launch pad and open Web Intelligence.

Note

We recommend that you log on to BI launch pad as a user from the group you have just customized, start Web Intelligence, and verify that the interface corresponds to your customization settings.

Related Information

User Interface Elements tab [page 16]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

6.2 Web Intelligence content alignment

Choose the way document content will be aligned (left-to-right or right-to-left) when users create Web Intelligence documents.

For the Web Intelligence Applet interface, you can set the content alignment in the CMC. Choose from these options:

- *Right-to-Left only when both the Preferred viewing and Product locales are set to Right-to-Left languages* (the default option)
- *Right-to-Left or Left-to-Right depending on the user's Preferred viewing locale*
- *Always Right-to-Left*
- *Always Left-to-Right*

> **Note**
>
> The content alignment setting applies to all users.

For the Web Intelligence Rich Client interface, the content alignment is determined by the locales set in the BI launch pad preferences:

- The system uses right-to-left alignment only when both the Preferred Viewing Locale and Product Locale are set to right-to-left languages.
- In all other cases, the content alignment is left-to-right.

> **Note**
>
> For information about how to set locales, see the *Business Intelligence Launch Pad User Guide*.

> **Note**
>
> Content alignment applies only at document creation time, and does not affect existing documents.

6.2.1 To set content alignment for the Web Intelligence Applet interface

Set content alignment for the Web Intelligence Applet interface.

1. Log into the CMC as an Administrator.
2. From the *Manage* list, select *Applications*.
3. Select *Web Intelligence*.
4. Click *Manage* > *Properties*.
5. Scroll down to the *Content Alignment for New Documents* section and select the appropriate option.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

7 Customizing Web Intelligence with UI Extension Points

You can customize the DHTML and Java interfaces of SAP BusinessObjects Web Intelligence and the Web Intelligence Rich Client interface using extensions on Microsoft Windows.

An extension contributes to the user interface by adding one or more UIElements, for example a button in the left-side pane, an icon button, a drop-down list, or a text field. Extensions provide end-users with advanced functionality to interact with Web Intelligence documents and reports via this UIElement.

> **Note**
>
> Web Intelligence Rich Client and the Java interface of SAP BusinessObjects Web Intelligence do not support extensions that use WebGL.

7.1 About the JavaScript APIs

You make the extension interact with the application by using the Web Intelligence Application and Service JavaScript APIs.

The Web Intelligence Application JavaScript functions allow you to set up the Web Intelligence DHTML or Java interface for interaction with documents and reports. You can for example listen to and dispatch events, update the DHTML client context, display wait cursor and dialog boxes. The Web Intelligence Service JavaScript functions allow you to develop the functionality provided by the extension.

To do so, you create a target page such as an HTML or a JSP page that includes the appropriate JavaScript API files.

> **Note**
>
> You can also use these functions with Web Intelligence Rich Client in Connected mode only, because the extension is installed on the BI platform server.

Table 9:

Documentation Deliverable	Description
SAP BusinessObjects Web Intelligence UI Extension Points JavaScript API reference	The official JavaScript functions reference. The 4.1 SP6 version of this document is the latest one for 4.1 BI platform releases.

Related Information

To Append Contribution Files [page 35]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

To Develop with the JavaScript APIs [page 35]

7.2 UI Extension Points Task Sequence

Here is the series of tasks to perform for customizing SAP BusinessObjects Web Intelligence with an extension:

1. Build your development environment.
2. Create the extension point.
3. Declare the extension to the extension point.
4. Implement the extension using the `IExtension` interface.
5. Create a function for your extension with the help of the Javascript APIs.
6. Test the extension on your development environment.
7. Build the extension JAR file.
8. Deploy the JAR file on the BI platform server and Apache Tomcat server of your production environment.
9. In the customization panel of the CMC, select the extension that you want to make available to specific users, groups of users, or folders.

7.3 About the Extension Bundle

The extension bundle is a fragment linked to the `webpath.AnalyticalReporting` bundle host. In the BI platform OSGI framework, the bundle host and its fragment bundles such as language packs and extensions are merged. To avoid overriding files, respect the following organization of the extension bundle folders:

```
web
    webiApplet
    webiDHTML
        viewer
    …
    extension
        <Provider>
            <ExtensionName>
    WEB-INF
        lib
```

Table 10:

Folder	Description
`web`	Top folder of all webpath bundles
`extension`	Subfolders that belong to an extension bundle
`<Provider>`	The name of bundle provider. The provider can be a vendor or a company name.
`<ExtensionName>`	The extension. It should reflect the functionality provided by the extension.
`WEB-INF/lib`	The folder where you can deploy libraries. Mainly JAR files.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

The `web/extension/<Provider>/<ExtensionName>` root path of the extension bundle is the folder from where the resource files are deployed. The bundle resources can be for example HTML, JavaScript, JSP, or images files. All URLs to resources must be related to the root path.

> **Remember**
>
> Add folders and JAR files to the CLASSPATH so that they can be found and loaded by the class loader.

7.4 Where to Find the Bundle Host

The `webpath.AnalyticalReporting` bundle host is installed with SAP BusinessObjects BI platform server at `C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\warfiles\webapps\BOE\WEB-INF\eclipse\plugins`.

7.5 Prerequisites

- Before creating an extension, you must have the SAP BusinessObjects BI platform servers installed on your development machine.
- SAP recommends that you build your development environment on Eclipse 3.6 or higher.

> **Note**
>
> To learn how to build your development environment, see the *Creating Extensions to SAP BusinessObjects Web Intelligence* document on the SAP Community Network.

7.6 To Import the Bundle Host

The `webpath.AnalyticalReporting` bundle host is the master bundle, which the extension bundles will refer to.

1. Open Eclipse, select *Window* > *Open Perspective* > *Other* and select *Plug-in Development* as your work perspective.
2. Select *Window* > *Preferences* to set the Java/Installed runtimes to JDK 1.6.0.x.
3. Select *File* > *Import*.
4. In the *Import* dialog box, select *Plug-in Development* > *Plug-ins and Fragments* and click *Next*.
5. Select the *Directory* option in *Import From*.

   The directory must be `C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\warfiles\webapps\BOE\WEB-INF\eclipse\plugins`.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

6. Select *com.businessobjects.webpath.AnalyticalReporting (1.0.0)* bundle from the list of plug-ins and fragments, click *Add* and *Finish*.

7.7 To Create an Extension Bundle

1. Select *File* > *New* > *Project*.
2. In the *New Project* dialog box, select *Plug-in Development* > *Fragment Project* and click *Next*.
3. Fill in the *New Fragment Project* dialog box as below and click *Next*.
   a. Enter the project name: **com.businessobjects.webpath.AnalyticalReporting_<Provider>_<ExtensionName>**.

      `<Provider>` and `<ExtensionName>` must be the same as those provided by the extension attributes. They must contain only alphanumeric characters. SAP does not recommend that you use space characters, but underscores instead.
   b. Select *Use default location*.
   c. Select *Create a Java project* with (Source folder: **src**, and Output folder: **bin**).
   d. Select the *Equinox* OSGi framework. and click *Next*.

      The Equinox Registry contains a list of all extensions and extension points belonging to the loaded and resolved bundles during the Apache Tomcat startup.
4. Fill in the *Fragment Content* as below:
   a. ID: **com.businessobjects.webpath.AnalyticalReporting_<Provider>_<ExtensionName>**
   b. Version: **1.0.0**
   c. Name: **<ExtensionName>**
   d. Provider: **<Provider>**
   e. Execution Environment: **JavaSE-1.6**
   f. Plug-in ID: **com.businessobjects.webpath.AnalyticalReporting**
5. Click *Finish*.

7.8 To Declare the Extension Bundle Contribution

The extension bundle is created to contribute to the `webpath.AnalyticalReporting` bundle host. You declare the extension in the `MANIFEST.MF` file.

1. Open the `MANIFEST.MF` file and select the *Overview* tab.
2. Click *Extensions* in *Extension/Extension Point Content* area.
3. In *Extensions* tab, click *Add* to display the list of available extension points.
4. Select the `com.businessobjects.webpath.AnalyticalReporting.webiApplication` extension point and click *Finish*.
5. Select the created extension *com.businessobjects.webpath.AnalyticalReporting_`<Provider>`_`<ExtensionName>`* to define attributes.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

a. id: **com.<Provider>.<ExtensionName>**

   The id attribute can take the bundle or package name.

b. class: **com.<Provider>.<ExtensionName>.Extension**

   The class attribute refers to the Java class that implements the `IExtension` interface.

c. name: **<ExtensionName>**

d. provider: **<Provider>**

e. version: **1.0.0**

6. In the *Runtime* tab, click *Add* to add the extension class path and select the `bin` folder.
7. In the *Overview* tab, check that the minimum execution environment is JavaSE-1.6.
8. Save the changes.

7.9 To Implement the IExtension Interface

The extension bundle must implement the `IExtension` interface to contribute to the bundle host. A Java class is created in the `com.sap.webi.toolkit.extension` package, which must be exported to make the interface available for the extensions.

1. Create the `com.<Provider>.<ExtensionName>` package.

   The package name must contain alphanumeric characters and a dot character as separator. It should be only in lower case.

2. Create a folder tree for extension files under the project.

```
web
    extension
        <Provider>
            <ExtensionName>
                assets
                    css
                    img
                    js
    WEB-INF
        lib
```

3. Select the created package and right-click it to create the `Extension` Java class.

   The Java class must implement the `IExtension` interface.

4. Implement the `getExtensionProperties` and `getContribution` methods.

7.9.1 getExtensionProperties

The `getExtensionProperties(String lang)` method returns an instance of the `ExtensionProperties` class in the required language. The `lang` parameter takes the current user interface language as value.

The instance is provided by the extension and must contain the following properties:

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Table 11:

Property	Description
Title	The extension title displayed in Extension Management UI in the CMC.
Description	A short description of the feature provided by the extension.

7.9.2 getContribution

The `getContribution(String lang)` method returns a list of UIElement objects. The `lang` parameter takes the current user interface language as value.

In SAP BusinessObjects Web Intelligence, a UIElement is a graphical element such as a button, drop-down list, or text field.

> **Remember**
>
> In the present release, the left-side pane and the status bar areas can accept contributions.

Contribution to the Left-Side Pane

A contribution to the left-side pane can only be a Button widget. There can be more than one contribution.

Each contribution is an instance of the `SidepaneButton` class. The class is a UIElement that contains a set of properties of a button on the left-side pane:

Table 12:

Property	Description
Name	Button identifier. It must be unique within the extension.
Title	Text displayed on the drop-down list of available buttons in the side pane. It also displays as tooltip of the button.
Description	Button description. It is only displayed on the Java Web Intelligence application.
IconURL	URL related to the icon button. SAP recommends to use the image type as "png" and the size of 24 * 24.
TargetPage	URL related to the main page. The content of this page is displayed on the Side panel frame. The page type could be any type that can be displayed by your Java application server.
Perspectives	List of perspectives where the added UIElement is visible. The list of available perspectives is listed in the `Perspective` class.

Contribution to the Status Bar

A contribution to the status bar can be a Button or a Toggle Button widget. There can be more than one contribution.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Each contribution is an instance of the `StatusbarButton` or `StatusbarToggleButton` class. The class is a UIElement that contains a set of properties of a button on the status bar:

Table 13:

Property	Description
Name	Button identifier. It must be unique within the extension.
Title	Text displayed on the drop-down list of available buttons in the status bar. It also displays as tooltip of the button.
Description	Button description. It is only displayed on Java Web Intelligence application.
IconURL	URL related to the icon button. SAP recommends you use the image type as "png" and the size of 16 * 16.
TargetPage	URL related to the main page. The content of this page is displayed on the Side panel frame. The page type could be any type that can be displayed by your Java application server.
Text	Text to display in the button.

7.10 To Append Contribution Files

Once you have created the Java class that implements the `IExtension` interface, you need to add an image that represents the extension on the user interface, and a target file of the functionality provided by the extension.

A target page can be any page type supported by Java application servers (HTML, JSP, Servlet, and so on.).

When adding a button to Web Intelligence, the icon size must be 24x24 pixels.

1. Add your contribution icon to the `web\extension\<Provider>\<ExtensionName>\assets\img` folder.
2. To create an HTML page, right-click the `<ExtensionName>` root folder of the extension and select *New* > *Other*.
3. Select *Web* > *HTML File* and click *Next*.
4. Select the parent folder, enter the HTML file name and click *Finish*.

7.11 To Develop with the JavaScript APIs

Once you have created the target page, you need to develop the functionality provided by the extension with the Web Intelligence Application and Service JavaScript APIs.

1. To access the Web Intelligence Application APIs, include the following files into your target pages:
   - `webi.application.js`
   - `webi.application.sidepane.js`
   - `webi.application.dialogbox.js`
   - `webi.application.bar.js`

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

- For a left-side pane:

```
<script type="text/javascript" src="../../../js/extension/
webi.application.js"></script>
<script type="text/javascript" src="../../../js/extension/
webi.application.sidepane.js"></script>
```

- For a dialog box:

```
<script type="text/javascript" src="../../../js/extension/
webi.application.js"></script>
<script type="text/javascript" src="../../../js/extension/
webi.application.dialogbox.js"></script>
```

- For a status bar button:

```
<script type="text/javascript" src="../../../js/extension/
webi.application.js"></script>
<script type="text/javascript" src="../../../js/extension/
webi.application.bar.js"></script>
```

2. To access the Web Intelligence Service APIs, include the `webi.services.js` file into your target pages as follows:

```
<script type="text/javascript" src="../../../js/extension/webi.services.js"></
script>
```

3. Develop the functionality of the extension.

> **Caution**
>
> The XMLHttpRequest specification has deprecated the requests to servers in synchronous mode. If you use the Application and Service JavaScript APIs in this mode, the end-user may receive an `InvalidAccessError` exception when using the extension.
>
> SAP recommends you use the asynchronous mode. The APIs are not modified. You simply use a callback function as additional argument of any Services APIs and of the `WebiApplication.loadReport` API. This function will receive the response asynchronously.
>
> The Web Intelligence Applet, DHTML, and Rich Client interfaces support both synchronous and asynchronous modes. For example:
>
> **Synchronous Mode:**

```
function getConnections()
 {
    var res = WebiServices.getConnections();

    if (!res.error)
    {
       var cnx = res.connections.connection;
       if (!isArray(cnx))
          cnx = [ cnx ];

       var table = document.getElementById("connections-table");
       table.options.length = 0;
       for (var i = 0; i < cnx.length; i++)
       {
          var entry = cnx[i];
          table.options.add(new Option(entry.name, JSON.stringify(entry)));
       }
    }
```


© 2016 SAP SE or an SAP affiliate company. All rights reserved.

```
    else
       alert(res.error.message);
 }
...
var res = WebiServices.document.addDataProvider(
 {
    dataprovider:
      {
        name: xlsEntry.name,
        dataSourceId: xlsEntry.id,
        properties: properties
      }
 });
```

Asynchronous Mode:

```
function getConnections()
 {
    WebiServices.getConnections(function(res) {
       if (!res.error)
       {
          var cnx = res.connections.connection;
          if (!isArray(cnx))
             cnx = [ cnx ];

          var table = document.getElementById("connections-table");
          table.options.length = 0;
          for (var i = 0; i < cnx.length; i++)
          {
             var entry = cnx[i];
             table.options.add(new Option(entry.name, JSON.stringify(entry)));
          }
       }
       else
          alert(res.error.message);
    });
 }
...
function createExcelDataProvider()
 {
    ...
    var arg =
    {
       dataprovider:
         {
           name: xlsEntry.name,
           dataSourceId: xlsEntry.id,
           properties: properties
         }
    };

    WebiServices.document.addDataProvider(arg, function(res) {
       if (res.success)
       {
           WebiServices.document.refreshDocument(function(response) {
             var docContext = WebiApplication.getContext();
             WebiApplication.loadReport({
                 reportId:docContext.selectedReportId},function()
                   {
                     alert("Report loaded!");
                   });
           });
        }
        else
           alert(res.error.message);
    });
```


PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

```
}
```

Related Information

https://xhr.spec.whatwg.org/

7.12 To Make the Extension Visible on the Interface

The BOE Equinox must load the extension to make it visible in the BI launch pad.

> **Remember**
>
> You must perform this task in your development environment. This is done automatically in a production environment.

1. Under the BOE project, open the `WebContent\WEBI-INF\eclipse\configuration\config.ini` file for editing.
2. Add a reference to your extension as follows:

```
#Eclipse Runtime Configuration File
osgi.bundles= \
    org.eclipse.equinox.common@2:start, \
    org.eclipse.update.configurator@start, \
    org.eclipse.equinox.ds@start, \
    com.businessobjects.servletbridge.core@start, \
    reference\:file\:C\:/workshop/
com.businessobjects.webpath.AnalyticalReporting_<Provider>_<ExtensionName>
osgi.bundles.defaultStartLevel=4
```

3. Save the file.
4. In the Server view, start the Apache Tomcat server.
5. In the Console view, type **`ss`** and press Enter to check that the extension bundle has been loaded properly.

   You should see the following information on the fifth line:

```
5 RESOLVED
com.businessobjects.webpath.AnalyticalReporting_<Provider>_<ExtensionName>_1.0.0
```

7.13 To Test the Extension Bundle

To test the extension in your development environment, you need to select it as a Web Intelligence customization in the CMC.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1. Log on to the CMC (`http://<server-name>:8090/BOE/CMC`) and click *Users and Groups* on the home page.

   > **Remember**
   >
   > Since the extension has not been deployed at this point, you must use the port configured for the Tomcat server of Eclipse (8090) to see the extension in the CMC.

2. Click *Group List* on the left pane to display all available groups.
3. Right-click on a group name you want to customize and click *Customization*.
   A customization dialog page is displayed.
4. Click *Add Folders* to select the document folders for which you want to enable the extension point.
   The folder appears in the list of customized folders.
5. Select the *Extensions* tab to display all installed extensions.
6. Check the extension you want to validate and make available for the customized folder of the users of the selected group.

   If you check the extension "`<ExtensionName>` 1.0.0", it will be added as an icon to the application user interface.
7. Click *Save* to save your selection.
8. Log on to the BI launch pad (`http://<server-name>:8090/BOE/BI`) and open a Web Intelligence document.

   You must use the port configured for the Tomcat server of Eclipse.

   > **Remember**
   >
   > To be able to see the extension icon:
   > - The test User must be a member of the customized *User Group*.
   > - The document must belong to the customized folder of the test User.

You should see the extension icon on the application interface.

7.14 To Build the Extension Bundle

You build the extension bundle to create a deployable JAR file. In our use case we only create a build binary.

In the out-of-the-box installation, Tomcat 6 is used as an application server. The bundles are not deployed as JAR files but as subfolders under the `<tomcat-dir>\webapps\BOE\WEB-INF\eclipse\plugins` folder.

1. In *Project Explorer* or *Package explorer*, double-click the `build.properties` file to open it.
2. Select the following folders in *Binary Build*:
   - *META-INF*
   - *bin*
   - *fragment.xml*
   - *web*
3. Click *File* > *Export*.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

4. In the *Export* dialog box, select *Deployable plug-ins and fragments* and click *Next*.
5. Select the extension bundle project in *Available Plug-ins and Fragments* and specify in the *Destination* tab the folder where the JAR file is generated.

   If you select the project folder, then a new folder with name "plugins" is created and the JAR file is copied to this folder.

7.15 To Deploy the Extension Bundle in Production

You deploy the extension in your production environment.

1. Stop Apache Tomcat.
2. Copy the extension JAR file to the following folders:
   - `<bip-install-dir>\warfiles\webapps\BOE\WEB-INF\eclipse\plugins`
   - `<tomcat-dir>\webapps\BOE\WEB-INF\eclipse\plugins`
   - `<tomcat-dir>\work\Catalina\localhost\BOE\eclipse\plugins`
3. Start Apache Tomcat.

> **Remember**
>
> Perform the additional step below on a client machine to use the extension bundle with the Java interface of Web Intelligence. You must have Java 7 installed. You do not have to perform this step when deploying the extension points for Web Intelligence Rich Client.

4. Copy the `jfxrt.jar` file to the `ext` folder. How you do this depends on whether you are using a 32-bit or 64-bit internet browser on your client machine:
   a. For a 32-bit internet browser, copy the file `C:\Program Files\Java\jre7\lib\jfxrt.jar` to the folder: `C:\Program Files\Java\jre7\lib\ext`.
   b. For a 64-bit internet browser, copy the file `C:\Program Files (x86)\Java\jre7\lib\jfxrt.jar` to the folder: `C:\Program Files (x86)\Java\jre7\lib\ext`.

To use the extension in your production environment, make sure you selected the extension in the CMC. Use the port configured for the Tomcat server of the BI platform (8080). See To Test the Extension Bundle [page 38].

7.16 About the Web Intelligence UI Extension Point Sample

The sample is a ready-to-use extension that demonstrates the usage of the JavaScript APIs. It allows you to test the following features:

- Refresh reports periodically
- List document reports in JSON format

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

The sample is installed with the SAP BusinessObjects BI platform servers and deployed automatically at installation in the following directories:

- `<bip-install-dir>\warfiles\webapps\BOE\WEB-INF\eclipse\plugins\com.businessobjects.webpath.AnalyticalReporting_SAP_ExtensionSample_1.0.0.jar`
- `<tomcat-dir>\webapps\BOE\WEB-INF\eclipse\plugins`
- `<tomcat-dir>\work\Catalina\localhost\BOE\eclipse\plugins`

The JAR file content is extracted automatically on the Tomcat server.

7.16.1 To Use the Extension Sample

The `com.businessobjects.webpath.AnalyticalReporting_SAP_ExtensionSample_1.0.0.jar` sample has been deployed automatically on the BI platform and Tomcat servers at platform installation.

1. Logon to the CMC to make the extension visible on the Web Intelligence interface.
   See instructions To Test the Extension Bundle [page 38].
2. Logoff from the CMC and logon to the BI launch pad.
3. Open any Web Intelligence document.
   The Extension Sample pane displays in the Web Intelligence left-side panel.
4. Play with the extension:
   - Enter a refresh schedule time (in seconds) and click *Start*. Click *Stop* to stop the refresh.
   - Click *Display*. A dialog box opens that contains the list of reports as a JSON object.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

8 Exposing Web Intelligence Features with REST Web Services

The Web Intelligence RESTful Web Service SDK provides a series of REST APIs that allows you to expose the Web Intelligence functionalities into your analytics applications.

Since the SDK is provided with the BI platform, you have nothing to install on the developer machine or where your application is deployed. The major benefit of the SDK is that you can use the REST APIs with any programming language that support the HTTP protocol, so that end-users can access the broad range of Web Intelligence features in many ways. A web service performs CRUD (Create, Read, Update, Delete) operations on data over HTTP, sending requests and receiving responses either in XML or JSON format. The way you implement these services is at your convenience. For example, you can automate batch operations on Web Intelligence documents. You can develop, using Java, a query panel to be embedded in your own reporting application. You can also make documents and reports available into non-SAP web applications.

The REST APIs expose features that relate to all Web Intelligence functional domains:

- Creating documents and building queries
- Creating reports with tables, sections, and charts
- Refreshing documents to get data
- Formatting reports
- Saving and exporting documents and reports
- Scheduling documents

Some Java samples are also provided to help you understand the REST APIs. They are supplied in the archive `<bip-install-dir>\Samples\webi\RaylightRESTWS_Samples.zip`.

> **Note**
>
> Before using the APIs, you need to logon to the BI platform and access the document or universe folder via the BI platform RESTful Web Service SDK.

Table 14:

Documentation Deliverable	Description
See the *SAP BusinessObjects RESTful Web Service SDK User Guide for Web Intelligence and the BI Semantic Layer* on the help portal.	The official guide for developing with the Web Intelligence RESTful Web Service SDK. The 4.1 SP6 version of this document is the latest one for 4.1 BI platform releases.
See the *Business Intelligence platform RESTful Web Service Developer Guide* on the help portal.	The official guide for developing with the BI platform RESTful Web Service SDK. The 4.1 SP5 version of this document if the latest one for 4.1 BI platform releases.

> **Note**
>
> Access to these guides is restricted to partners.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

9 Consuming BI Semantic Layer Universes with REST Web Services

The BI Semantic Layer REST Web Service SDK provides a series of REST APIs that allow you to access relational universes, browse universe metadata, create and execute queries. It supports UNV universes created with the universe design tool as well as UNX universes created with the information design tool.

Since the SDK is provided with the BI platform, you have nothing to install on the developer machine or where your application is deployed. The major benefit of the SDK is that you can use the REST APIs with any programming language that support the HTTP protocol, so that end-users can access the broad range of information design tool features in many ways. A web service performs CRUD (Create, Read, Update, Delete) operations on data over HTTP, sending requests and receiving responses either in XML or JSON format. The way you implement these services is at your convenience. For example, you can write Java scripts to query universes automatically. You can develop your own application to retrieve universe data. Result sets will be returned using the OData protocol.

The REST APIs expose features that relate to the main functional domains of the SAP BusinessObjects design tools:

- Browsing universes and retrieving metadata
- Building queries on universes and retrieving data

Some Java samples are also provided to help you understand the REST APIs. They are supplied in the archive `<bip-install-dir>\SL SDK\SDK Samples\SLRESTWebService.zip`.

> **Note**
>
> Before using the APIs, you need to logon to the BI platform and access the universe folder via the BI platform RESTful Web Service SDK.

Table 15:

Documentation Deliverable	Description
See the *SAP BusinessObjects RESTful Web Service SDK User Guide for Web Intelligence and the BI Semantic Layer* on the help portal.	The official guide for developing with the BI Semantic Layer RESTful Web Service SDK. The 4.1 SP6 version of this document is the latest one for 4.1 BI platform releases.
See the *Business Intelligence platform RESTful Web Service Developer Guide* on the help portal.	The official guide for developing with the BI platform RESTful Web Service SDK. The 4.1 SP5 version of this document if the latest one for 4.1 BI platform releases.

> **Note**
>
> Access to these guides is restricted to partners.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

10 Developing Applications to Design and Administrate Universes

The BI Semantic Layer Java SDK gives access to the features of the information design tool in non-SAP client tools. You can develop Java applications to design the UNX universe resources (data foundations, business layers, and connections), to publish them in a CMS repository, and to configure security settings on published universes.

Some samples are also provided to help you understand the Java SDK APIs. They are supplied in the archive `<bip-install-dir>\SL SDK\SDK Samples\com.sap.sl.sdk.authoring.samples.source.jar`.

Similarly, the Universe Design Tool COM SDK gives you access to the features of the universe design tool. You can develop applications to design and manage UNV universes using the provided COM objects.

Table 16:

Documentation Deliverable	Description
SAP BusinessObjects BI Semantic Layer Java SDK Developer Guide	The official user guide for developing with the BI Semantic Layer Java SDK. The 4.1 SP6 version of this document is the latest one for 4.1 BI platform releases.
SAP BusinessObjects BI Semantic Layer Java API Reference	The reference for interfaces and methods of the Java APIs. This document is the latest version for 4.1 BI platform releases.
SAP BusinessObjects BI Semantic Layer Java Object Model Diagrams	The object model diagrams of the BI Semantic Layer Java SDK. The 4.1 SP6 version of this document is the latest one for 4.1 BI platform releases.
SAP BusinessObjects Universe Design Tool COM API Reference	The reference for the COM objects and methods. This document version is related to the 4.0 platform, but its content is also valid for the 4.1 releases.
SAP BusinessObjects Universe Design Tool Object Model Diagrams	The object model diagrams of the Universe Design Tool COM SDK. This document version is related to the 4.0 platform, but its content is also valid for the 4.1 releases.

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

11 Creating a Data Access Driver

A data access driver is a software component that runs with the data access service of the BI platform called Connection Server to perform requests to data sources and retrieve data for UNV and UNX universes. SAP BusinessObjects applications use a wide range of data access drivers to communicate with database middleware. In addition to the supplied data access drivers, you can use the Driver Development Kit to develop, using Java, data access drivers for data sources for which there are no drivers available.

Some JavaBean and Open driver samples are provided at `<bip-install-dir>\data access \connectionServer\DDK\examples` to help you develop your own Java driver:

- The Open driver sample illustrates how to develop a driver that accesses data stored in a comma-separated value (CSV) file.
- The JavaBean driver sample illustrates how to develop a JavaBean driver that accesses data stored in an Excel spreadsheet.

The driver sample codes are also available on the SCN.

Table 17:

Documentation Deliverable	Description
SAP BusinessObjects Data Access Driver Java SDK Developer Guide	The official guide for developing your own driver with the Driver Development Kit. This document version is related to the 4.0 platform, but its content is also valid for the 4.1 releases.
SAP BusinessObjects Data Access Driver Java API Reference	The reference for interfaces and methods of the Java APIs. This document is the latest version for 4.1 BI platform releases.

Related Information

Data Access Driver Samples

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of wilful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

PUBLIC
© 2016 SAP SE or an SAP affiliate company. All rights reserved.

go.sap.com/registration/contact.html

© 2016 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 http://www.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.