# Table of Contents

1  Document History .............................................................. 5  

2  Getting Started ............................................................... 6  
2.1  About this guide ............................................................ 6  
2.2  Before you begin ........................................................... 6  
2.2.1  Customizing SAP BusinessObjects Business Intelligence platform . 7  
2.2.2  Provisioning tenants in the SAP BusinessObjects Business Intelligence platform . 9  
2.2.3  Customizing SAP Crystal Reports .................................. 9  

3  Business Intelligence Platform Installer Customization .................. 11  
3.1  Introduction ............................................................... 11  
3.2  Quick start for the Business Intelligence platform (Windows) .......... 11  
3.3  Quick start for the Business Intelligence platform (Unix or Linux) .... 12  
3.4  To download the installation program .................................. 13  
3.5  Planning the customization process .................................... 14  
3.5.1  Best practices ......................................................... 14  
3.6  Creating the configuration file .......................................... 15  
3.6.1  Configuration file overview ....................................... 16  
3.6.2  Renaming the product ................................................ 17  
3.6.3  Customizing user input .............................................. 24  
3.6.4  Removing installation screens ..................................... 25  
3.6.5  Embedding a keycode ................................................ 26  
3.6.6  Removing features .................................................... 26  
3.6.7  Preventing prerequisite checks .................................... 27  
3.6.8  Removing language packs ......................................... 27  
3.6.9  Preventing the WDeploy tool from running ....................... 28  
3.6.10  Removing the default database ................................... 28  
3.6.11  Changing resources ................................................ 29  
3.6.12  Removing items from the Collaterals folder ...................... 32  
3.7  Running the tool .......................................................... 33  
3.7.1  Command line parameters .......................................... 34  
3.8  Customizing update installation programs ............................. 35  
3.8.1  Frequently asked questions about update installation programs .... 35  
3.8.2  Quick start for update installation programs .................... 37  
3.8.3  How to customize update installation programs .................. 37  
3.9  IDs and codes for BI Platform customization ......................... 39  
3.9.1  Feature IDs .......................................................... 40  
3.9.2  Shortcut deployment unit IDs (Windows only) .................... 43  
3.9.3  String IDs ........................................................... 44  

© 2014 SAP AG or an SAP affiliate company. All rights reserved.
4 Web Application Customization .................................................. 54
4.1 Introduction ........................................................................... 54
4.1.1 Key concepts ..................................................................... 54
4.1.2 Testing your customizations ............................................. 56
4.2 Quick start ............................................................................. 56
4.3 Customizing BI launch pad .................................................... 57
4.3.1 To customize the favicon image ....................................... 58
4.3.2 To customize logos .......................................................... 58
4.3.3 Customizing other user interface elements ....................... 58
4.3.4 Working with BI workspaces and compound modules ....... 65
4.3.5 To change the name of BI launch pad ............................... 66
4.4 Customizing OpenDocument ................................................ 67
4.5 Customizing the Crystal Reports JavaScript viewer .............. 68
4.5.1 Customizing the viewer .................................................. 69

5 Business Intelligence Platform Multitenancy Management Tool ................................................. 71
5.1 Introduction ........................................................................... 71
5.2 Quick start ............................................................................. 72
5.2.1 Installation prerequisites .................................................. 73
5.2.2 Create your tenant template ............................................ 74
5.2.3 Configure your tenant definition file ................................. 77
5.2.4 Run the tool ..................................................................... 78
5.3 Setting up a tenant template .................................................. 80
5.3.1 Folders ............................................................................ 80
5.3.2 User groups ..................................................................... 81
5.3.3 Event folders .................................................................... 81
5.3.4 Categories ....................................................................... 81
5.3.5 Profiles ............................................................................ 82
5.3.6 Security settings ............................................................ 82
5.3.7 SAP Crystal Reports 2011 ............................................. 83
5.3.8 SAP Crystal Reports for Enterprise ................................. 85
5.3.9 Universes and connections ............................................. 87
5.4 Configuring the tenant configuration file ............................... 92
5.5 Running the multitenancy management tool ......................... 98
5.6 Troubleshooting ................................................................... 99
5.6.1 Multitenancy management tool errors ............................ 100
5.6.2 To configure tracing for the Multitenancy Management Tool 105

6 SAP Crystal Reports 2011 Customization ................................................. 108
6.1 Introduction .......................................................................... 108
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2</td>
<td>Quick start for Crystal Reports</td>
<td>108</td>
</tr>
<tr>
<td>6.3</td>
<td>To download the installation program</td>
<td>109</td>
</tr>
<tr>
<td>6.4</td>
<td>Planning the customization process</td>
<td>109</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Best practices</td>
<td>110</td>
</tr>
<tr>
<td>6.5</td>
<td>Creating the configuration file</td>
<td>111</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Configuration file overview</td>
<td>112</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Renaming the product</td>
<td>113</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Customizing default user input</td>
<td>120</td>
</tr>
<tr>
<td>6.5.4</td>
<td>Removing installation screens</td>
<td>121</td>
</tr>
<tr>
<td>6.5.5</td>
<td>Embedding a keycode</td>
<td>122</td>
</tr>
<tr>
<td>6.5.6</td>
<td>Removing features</td>
<td>122</td>
</tr>
<tr>
<td>6.5.7</td>
<td>Preventing prerequisite checks</td>
<td>123</td>
</tr>
<tr>
<td>6.5.8</td>
<td>Removing language packs</td>
<td>123</td>
</tr>
<tr>
<td>6.5.9</td>
<td>Changing resources</td>
<td>123</td>
</tr>
<tr>
<td>6.5.10</td>
<td>Removing items from the Collaterals folder</td>
<td>126</td>
</tr>
<tr>
<td>6.6</td>
<td>Customizing the report designer</td>
<td>127</td>
</tr>
<tr>
<td>6.6.1</td>
<td>Customizing the splash screen</td>
<td>128</td>
</tr>
<tr>
<td>6.6.2</td>
<td>Customizing the start page</td>
<td>128</td>
</tr>
<tr>
<td>6.6.3</td>
<td>Customizing menu strings</td>
<td>129</td>
</tr>
<tr>
<td>6.6.4</td>
<td>Deploying the OEM customization file</td>
<td>130</td>
</tr>
<tr>
<td>6.7</td>
<td>Running the tool</td>
<td>131</td>
</tr>
<tr>
<td>6.7.1</td>
<td>Command line parameters</td>
<td>132</td>
</tr>
<tr>
<td>6.8</td>
<td>Customizing update installation programs</td>
<td>133</td>
</tr>
<tr>
<td>6.8.1</td>
<td>Frequently asked questions about update installation programs</td>
<td>134</td>
</tr>
<tr>
<td>6.8.2</td>
<td>Quick start for update installation programs</td>
<td>135</td>
</tr>
<tr>
<td>6.8.3</td>
<td>How to customize update installation programs</td>
<td>136</td>
</tr>
<tr>
<td>6.9</td>
<td>IDs and codes for Crystal Reports customization</td>
<td>138</td>
</tr>
<tr>
<td>6.9.1</td>
<td>Feature IDs</td>
<td>138</td>
</tr>
<tr>
<td>6.9.2</td>
<td>Shortcut deployment unit IDs</td>
<td>141</td>
</tr>
<tr>
<td>6.9.3</td>
<td>String IDs</td>
<td>141</td>
</tr>
<tr>
<td>6.9.4</td>
<td>Language codes</td>
<td>142</td>
</tr>
<tr>
<td>6.9.5</td>
<td>Installation screen and property IDs</td>
<td>143</td>
</tr>
</tbody>
</table>
1 Document History

The following table provides an overview of the enhancements made to this document.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP BusinessObjects Business Intelligence Suite 4.0 Feature Pack 3</td>
<td>March, 2012</td>
<td>First release of this document.</td>
</tr>
<tr>
<td>SAP BusinessObjects Business Intelligence Suite 4.0 Support Package 5</td>
<td>November, 2012</td>
<td>Business Intelligence Platform Installer Customization:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● The new <code>baselinePath</code> parameter replaces the <code>baselinePackages</code> parameter for customizing Patch or Support Package installation programs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For details on the new behavior and examples, see <a href="#">How to customize Support Packages and Patches</a> and <a href="#">Command line parameters</a>.</td>
</tr>
<tr>
<td>SAP BusinessObjects Business Intelligence Suite 4.0 Support Package 9</td>
<td>February, 2014</td>
<td>Updated the following sections to explain that you cannot remove every data access feature, otherwise the connection server will not function:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Removing features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Feature IDs</td>
</tr>
</tbody>
</table>
2  Getting Started

2.1  About this guide

The SAP BusinessObjects Business Intelligence Suite provides a set of tools and templates that allow Original Equipment Manufacturer (OEM) partners to customize the SAP BusinessObjects Business Intelligence platform and the SAP Crystal Reports Designer. This guide shows you how to use these tools and templates to create your desired customizations.

Depending on the needs of your customers, you can remove features and language packs to reduce the size of the installation program and the installed product. And if you want to differentiate your OEM system and apply your own unique corporate branding, you can personalize the appearance of your products, including product name, logos, colors, and other elements of the user interface. Your customization can be as simple as a logo change, or as detailed as a complete re-skinning.

The best thing is that your customizations are supported throughout the life cycle of the products. It is easy to maintain your changes during future upgrades and updates.

This guide is meant for OEM partners who are customizing SAP BusinessObjects Business Intelligence Suite products, and it assumes some knowledge of the OEM process. You won’t need to read the entire document; the Before you begin [page 6] section describes the relevant workflows for each major area of product customization and tells you where to find the information you need.

Guide conventions

The following variables are used throughout this guide.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;INSTALLDIR&gt;</td>
<td>The file path where the BI platform is installed. On a Windows machine, the default file path is C:\Program Files (x86)\SAP BusinessObjects\</td>
</tr>
</tbody>
</table>

2.2  Before you begin

This guide covers the different types of OEM customization for the different products in the SAP BusinessObjects Business Intelligence suite. You need to read only those sections that cover the products you are planning to customize.
2.2.1 Customizing SAP BusinessObjects Business Intelligence platform

You can customize many aspects of the Business Intelligence platform deployment:

- **Customize the installation program.**
  You can remove features, language packs, and resources to reduce the size of the installed product, rename the product, change images, hide unwanted installation screens, embed a keycode, and pre-populate user input.
  See the *Introduction* for “Business Intelligence Platform Installer Customization” in this document.

- **Customize BI launch pad and OpenDocument web applications.**
  You can change the titles and the URLs that are used to access web applications. You can change the appearance and branding of these applications using custom images and Cascading Style Sheets (CSS).
  See the *Introduction* for “Web Application Customization” in this document.

- **Customize the Crystal Reports JavaScript API report viewer.**
  You can change the logo and customize the visual style of the viewer using custom images and Cascading Style Sheets (CSS). You can add your own event and action listeners to the existing JavaScript API, or add your own external JavaScript files.
  See *Customizing the Crystal Reports JavaScript viewer* in this document.

You can customize the web applications, the installation program, or both. The following diagram illustrates the workflow where all types of customizations are performed:
To customize the installation program:

Download the installation program for the BI platform, plus the installation programs for any desired Support Packages or Patches.

Customize the installation program(s).

Result: Customized installation program(s) with template.zip file.

Add template.zip to the customized installation program.

Repackage customizations into template.zip.

Extract branding bundles from template.zip and customize.

Result: template.zip file containing customizations.

Ship customized installation program(s) to customers.

To customize:
- BI Launch Pad
- OpenDocument
- Crystal Reports
- Javascript Viewer
2.2.2 Provisioning tenants in the SAP BusinessObjects Business Intelligence platform

Many OEM partners use the SAP BusinessObjects Business Intelligence platform in their Software as a Service (SAAS) environment. These environments host multiple customers, or tenants, in the same system. Tenant environments are typically very similar with a few key customizations. The multitenancy management tool allows SAAS providers to quickly create a unique tenant environment based on a common template.

See the Introduction for the BI platform multitenancy management tool in this document.

2.2.3 Customizing SAP Crystal Reports

There are many customizations you can perform to enhance and personalize the design and customer experience for your SAP Crystal Reports users:

- Install and run the SAP BusinessObjects customization tool. For details, see Quick start for Crystal Reports.

- If you want to customize the SAP Crystal Reports installation program, you can change its appearance, hide unwanted screens from users in the wizard, and remove unused features to reduce the installed product size on client machines.
  See the Introduction for “SAP Crystal Reports 2011 Customization” in this document.

- If you want to customize the report designer, you can change the default splash screen or start page. You can also customize the product name, menus, and other assets of the report designer.
  See Customizing the report designer in this document.

The following diagram illustrates the workflow where all types of customizations are performed:
To customize the installation program:

Download the installation program for Crystal Reports, plus the installation programs for any desired Support Packages or Patches.

Customize the installation program.

Result: Customized installation program

Result: Customized installation program with template.zip file

Add template.zip to the customized installation program

Package customizations into template.zip

Customize the report designer:
- Splash screen
- Start page
- Menu strings

Result: template.zip file containing customizations

Ship customized installation program to customers
3 Business Intelligence Platform Installer Customization

3.1 Introduction

The SAP BusinessObjects Business Intelligence platform can be repackaged and sold by partners. You can customize the installed product and the installation program in order to target a specific customer base, or to resell it as part of your own product. The SAP BusinessObjects customization tool customizes the SAP BusinessObjects Business Intelligence platform and its installation program with changes such as the following:

- Reducing the product size
- Renaming the product
- Changing default properties in the installation program
- Hiding screens in the installation program

To make customizations, you write a configuration file to specify the changes then run the SAP BusinessObjects customization tool to create a customized installation program. Customers can use this installation program to install a customized version of the product.

The customization tool is available for Windows and Unix. It can be used to customize a full installation program, a Support Package installation program, and a Patch installation program.

Note

This tool does not perform customizations on the SAP BusinessObjects Business Intelligence platform Client Tools.

3.2 Quick start for the Business Intelligence platform (Windows)

This section shows you how to run the customization tool to create a customized installation program for the SAP BusinessObjects Business Intelligence platform (BI Platform). It uses the sample configuration file that is provided with this tool. When you are finished this tutorial, you can run your customized installation package and install a customized version of the BI Platform.

The customizations include changing the default installation type, removing features, hard-coding the product keycode, changing the default installation folder, renaming the product, and changing the Windows Start menu shortcut for the Central Configuration Manager feature. They are described in more detail in the configuration file.

1. Set up the customization tool.
   a) Create a working folder on your development machine, for example C:\SAPCustomTool\packages.
   b) Copy the contents of the BI Platform installation package to C:\SAPCustomTool\packages.

   The installation package contains the folders Collaterals, dunit, langs, and setup.engine in addition to other binaries. See To download the installation program [page 13] for instructions.
c) (Optional). Add your keycode to the sample configuration file.

In an XML editor, open the file C:\SAPCustomTool\packages\Collaterals\Tools \CustomizationTool\example_customization_win_boe.xml and replace the phrase PutYourKeyCodehere with your BI Platform keycode. If you do not enter your keycode into the configuration file, you can use the Central Management Console to enter it after installing the customized BI Platform.

d) Create the folder C:\SAPCustomTool\output.

This folder must be empty.

e) Run the following command from the command prompt: cd C:\SAPCustomTool\packages \Collaterals\Tools\CustomizationTool

The folder CustomizationTool contains the executable customizationtool.exe and the sample configuration file example_customization_win_boe.xml.

2. Run the following command from the command prompt:

customizationtool.exe xml=example_customization_win_boe.xml packageDir=C:\SAPCustomTool\packages outputDir=C:\SAPCustomTool\output logDetail=error > C:\oemlog.log

Verify that the customized installation program was created at C:\SAPCustomTool\output. Ensure no errors were reported in the log file oemlog.log.

i Note

The customization tool may take several minutes to complete. You can check its progress by viewing the log file.

3. Use C:\SAPCustomTool\output\setup.exe to run the customized SAP BusinessObjects Business Intelligence platform installation program.

The BI Platform is installed with the customizations described in the configuration file.

3.3 Quick start for the Business Intelligence platform (Unix or Linux)

This section shows you how to run the customization tool to create a customized installation program for the SAP BusinessObjects Business Intelligence platform (BI Platform). It uses the sample configuration file that is provided with this tool. When you are finished this tutorial, you can run your customized installation package and install a customized version of the BI Platform.

The customizations include changing the default installation type, removing features, hard-coding the product keycode, changing the default installation folder, and renaming the product. They are described in more detail in the configuration file.

1. Set up the customization tool.

   a) Create a working folder on your development machine, for example /usr/jdoe/bip/package.

   b) Copy the contents of the BI Platform installation package to /usr/jdoe/bip/package.

   The installation package contains the folders Collaterals, dunit, langs, and setup.engine in addition to other binaries. See To download the installation program [page 13] for instructions.
c) (Optional). Add your keycode to the sample configuration file.

In an XML editor, open the file /usr/jdoe/bip/package/Collaterals/Tools/CustomizationTool/example_customization_linux_boe.xml and replace the phrase PutYourKeyCodehere with your BI Platform keycode. If you do not enter your keycode into the configuration file, you can use the Central Management Console to enter it after installing the customized BI Platform.

d) Create the folder /usr/jdoe/bip/output. This folder must be empty.

e) Change to the folder /usr/jdoe/bip/package/Collaterals/Tools/CustomizationTool. This folder contains the executable customizationtool.sh and the sample configuration file example_customization_linux_boe.xml.

2. Run the following command from the command prompt:

./customizationtool.sh xml=example_customization_linux_boe.xml packageDir=/usr/jdoe/bip/package outputDir=/usr/jdoe/bip/output logDetail=error &> custombip.log

The customizations that you see in the installation program and in the installed product are described in the configuration file /usr/jdoe/bip/package/Collaterals/Tools/CustomizationTool/example_customization_linux_boe.xml.

Verify that the customized installation program was created at /usr/jdoe/bip/output. Ensure no errors were reported in the log file custombip.log.

i) Note

The customization tool may take several minutes to complete. You can check its progress by viewing the log file.

3. From the command prompt, use /usr/jdoe/bip/output/setup.sh to run the customized BI Platform installation program.

The BI Platform is installed with the customizations described in the configuration file.

3.4 To download the installation program

2. On the Find your software tab, under the A–Z Index, select Installations and Upgrades.
3. Select B > SBOP BI platform (former SBOP Enterprise) > SBOP BI Platform 4.0.
4. Select Installation and Upgrade and then select your platform.
5. Select all of the objects titled SBOP BI PLATFORM <version> SERVER plus any additional add-on products you require, then follow the instructions on the website to download and extract the objects.

The software may take a long time to download, and you may need to contact the system administrator to ensure your company’s firewall will not terminate the download process.

Support Packages and Patches are installation programs that contain updates to BI platform software. You can download them from https://service.sap.com/bosap-support > Software Downloads. On the Find your software tab, under the A–Z Index, click Support Packages and Patches. For more information on installing Support Packages and Patches, see the SAP BusinessObjects Business Intelligence Platform Update guides.
3.5 Planning the customization process

To use the SAP BusinessObjects customization tool:

1. Download the installation program. See To download the installation program [page 13].
2. Decide what customizations are required. See Creating the configuration file [page 15].
3. Write the configuration file to specify the customizations.
4. Run the customization tool to create a customized installation program.
5. Run the customized installation program to install a customized version of SAP BusinessObjects Business Intelligence platform.

3.5.1 Best practices

This section provides recommendations for creating a customized installation program.

Validate the configuration file

You may want to validate the configuration file before running the tool. Use the validate command-line parameter.

Reduce product size

Customers prefer a smaller installation program and a smaller installed product. To keep the product as small as possible:

- Remove any language packs that are not required.
- Remove any features that are not required.
- Remove any items from the Collaterals folder that are not required.
- Remove the default database if it is not required.

Apply customized names consistently

The product name and version number appear in several places in the installation program and in the installed product. Ensure you verify customizations in the following locations:

- Product name, product version, and product major version
- Windows Start menu entry and all feature shortcuts
- Windows Add Remove Program utility
• Default installation folder

**Consider name change in all languages**

It is good practice to consider how the customized name appears in all supported languages.

**Modify patch installation programs to be consistent with the main installation program**

You must apply the same customizations to Support Packages and Patches as you applied to the main release. If you release a customized main installation program, then try to release a Support Package or Patch installation program with different customizations, you might see unpredictable results which might not be repairable using standard rollback procedures.

**Test rollback, modify, and repair installations for Support Packages and Patches**

Rollback, modify, and repair are supported for customized Support Packages and Patches, provided they have been customized in a manner consistent with the main installation package. It is recommended to test these scenarios.

**Related Information**

*Command line parameters* [page 34]

**3.6 Creating the configuration file**

The following section describes the customizations you can make to the installation program by editing the configuration file:

• Renaming the product
  ○ Customizing the product name and version number
  ○ Customizing the Windows *Start* menu shortcuts
  ○ Customizing the Windows *Add Remove Program* utility
  ○ Customizing the installation folder
3.6.1 Configuration file overview

The SAP BusinessObjects customization tool uses information in the configuration file to perform the customizations. The configuration file is an XML document, and you use XML elements to describe your customizations. The sample configuration file is contained in this folder in the installation program:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Location of sample configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>Collaterals\Tools\CustomizationTool\example_customization_win_boe.xml</td>
</tr>
<tr>
<td>Unix or Linux</td>
<td>Collaterals/Tools/CustomizationTool/example_customization_linux_boe.xml</td>
</tr>
</tbody>
</table>

The file must have this format:

```
<oem name="<Any name>">
  <cloneProduct sourceId="product.businessobjects64-4.0-core-32">
    ...
  </cloneProduct>
</oem>
```

The configuration file for the full installation program can have any name, for example, oem.xml.

The configuration file for the Support Package installation program is described in the section How to customize Support Packages and Patches.

**Note**

The configuration file must be written in correct XML syntax. Use an XML editor to create and edit the file, and verify the format is correct before running the tool.

**Example**

This example file specifies these customizations:
- Change the product long name to *Custom Company Server* for all languages.
- Change the product short name to *Custom CS* for all languages.
- Remove the installation screen titled *Choose Installation Type* and set the installation type to *Custom*.
- Specify that the only language packs included in the installation package are English, French, German, Italian, and Chinese.

```xml
<oem name="CustomCompanyServer">
  <cloneProduct sourceId="product.businessobjects64-4.0-core-32">
    <replaceString id="product.boe64_name" value="Custom Company Server" lang="all"/>
    <replaceString id="product.boe64_shortname" value="Custom CS" lang="all"/>
    <replaceProperty id="InstallType" defaultValue="custom"/>
    <removeDialog id="ChooseInstallType.dialog"/>
    <languageIncludeList value="en;fr;de;it;zh_CN"/>
  </cloneProduct>
</oem>
```

### 3.6.2 Renaming the product

You can rename the product in the following ways:

- Customize the product name and version number.
- Customize the Windows *Add or Remove Programs* entry. (Windows only)
- Customize the *Start* menu entry for feature shortcuts. (Windows only)
- Customize the default installation folder.

The following sections explains these steps.

#### 3.6.2.1 Customizing the product name and version number

You can customize the product name and version number. Use the `replaceString` element with the desired string ID:

```xml
<replaceString id="<string id>" value="<new value>" lang="<language list>"/>
```

There are four strings that represent the product name and version number: the product long name, the product short name, the product version number, and the product major version number. The full product name is composed of the product long name and the version number. The product short name and product major version are used in the Windows shortcut menu.

<table>
<thead>
<tr>
<th>String description</th>
<th>String ID</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product long name</td>
<td>product.boe64_name</td>
<td>SAP BusinessObjects BI platform</td>
</tr>
</tbody>
</table>
### String description | String ID | Default value
---|---|---
Product short name | `product.boe64_shortname` | BI platform server
Product version | `product_version` | 4.1
Product major version | `product_majorversion` | 4

### Note
You should customize the product version and product major version together. For example, if you change product version to 1.0 you should also customize product major version to 1. Otherwise the version number in the menus will not match the version number in the product.

You can specify a new name for each language.

### Example
Change the product long name to *Sales Insight Platform* and the product short name to *Sales Platform* for English. Change the product long name to *Sales Insight Platform (French)* and the product short name to *Sales Platform (French)* for French. Change the product version to 1.0 and the product major version to 1 for both French and English. The product name and version number in languages other than English and French will remain as the default value.

```xml
<replaceString id="product.boe64_name" value="Sales Insight Platform" lang="en"/>
<replaceString id="product.boe64_shortname" value="Sales Platform" lang="en"/>
<replaceString id="product.boe64_name" value="Sales Insight Platform (French)" lang="fr"/>
<replaceString id="product.boe64_shortname" value="Sales Platform (French)" lang="fr"/>
<replaceString id="product_version" value="1.0" lang="en;fr"/>
<replaceString id="product_majorversion" value="1" lang="en;fr"/>
```

The customization appears below. Notice the version number “FP3” is not removed:

![Sales Insight Platform 1.0 FP3 setup](image)

### To remove instances of “FP3” from the installation program

When you run the installation program, you may see instances of “FP3” in the product name. To remove “FP3”, modify the lines in the following files:
You must modify one file for every language that the installation program supports. For a list of language codes, see [Language codes](page 44). When you run the customization tool and then run the installation program, all instances of "FP3" will be removed. This process will be simplified in a future release.

**Example**

To remove "FP3" from the English installation program, modify the following files:

- `product.lang_en.uitext.xml`
- `setup.ui.framework.lang_en.uitext.xml`

The customization appears below:

![Image of Sales Insight Platform 1.0 setup](image-url)
Customizing the Windows Start menu shortcuts (Windows only)

The Windows Start menu contains shortcuts for features such as the Central Management Console and BI launch pad. You can customize the name, location, and tooltip for each shortcut. Any shortcut that you do not customize will be grouped under the default Start menu, **SAP BusinessObjects BI platform 4**.

The default Start menu in English installations looks like this:

![Start menu screenshot](image)

Use the **shortcut** element to customize the location, shortcut name, and tooltip for each feature:

```
<shortcut duSourceId="<shortcut deployment unit ID>">
  <arg id="linkFullPath" value="<full path to shortcut link>" lang="<language list>">
  </arg>
  <arg id="description" value="<tooltip string>" lang="<language list>" />
</shortcut>
```

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>duSourceId</td>
<td>The shortcut deployment unit ID that you want to modify. Typical values include:</td>
</tr>
<tr>
<td></td>
<td>• product.businessobjects64.shortcut.ccm­4.0­core Central Configuration Manager</td>
</tr>
<tr>
<td></td>
<td>• product.businessobjects64.shortcut.infoview­4.0­core BI launch pad</td>
</tr>
<tr>
<td></td>
<td>• product.businessobjects64.shortcut.cmc­4.0­core Central Management Console</td>
</tr>
<tr>
<td></td>
<td>For a complete list of sourceId values, see Shortcut deployment unit IDs.</td>
</tr>
<tr>
<td>linkFullPath</td>
<td>The full path to the shortcut link. Be sure to add .lnk to shortcut link or the link will not be created. You can put the shortcut link on the Start menu or you can put it on the desktop. The SAP BusinessObjects customization tool will create the links correctly.</td>
</tr>
</tbody>
</table>
### Attribute | Value
--- | ---
You can specify one link for each language. For a list of language codes, see Language Codes. | 
**description** | The tooltip string to display when the user hovers the mouse over the shortcut. You can specify one tooltip for each language. |

### Note
You can customize the link, but not the tooltip, for the following shortcuts:
- BI Launchpad (formerly InfoView)
- Online documentation
- WACs stored in InfoView
- Web Application Container Server

This will be resolved in a future release.

### Example
This example customizes the name of the Central Management Console shortcut to Sales Insight Manager for English and Sales Insight Manager (French) for French, and places the shortcuts under the Start menu entry called Sales Insight Platform 1. It also customizes the tooltip to Launch Sales Manager for English and Launch Sales Manager (French) for French. The shortcut name and tooltip will remain unchanged for all other languages.

```xml
<shortcut duSourceId="product.businessobjects64shortcut.cmc-4.0-core"
         <arg id="linkFullPath" value="[programmenufolder]\Sales Insight Platform 1\Sales Insight Manager.lnk" lang="en"/>
        <arg id="linkFullPath" value="[programmenufolder]\Sales Insight Platform 1 (French)\Sales Insight Manager (French).lnk" lang="fr"/>
        <arg id="description" value="Launch Sales Manager" lang="en"/>
        <arg id="description" value="Launch Sales Manager (French)" lang="fr"/>
     </shortcut>
```

The customization appears below:
Modifying the tomcat shortcut

There are two links for the Tomcat shortcut: **Tomcat Administration** and **Tomcat Configuration**, shown below:

You must take extra steps to customize this shortcut. Use this shortcut element to customize the **Tomcat Administration** link. Note the `pathToTarget` element.

```xml
<shortcut duSourceId="product.businessobjects64.shortcut.tomcat-4.0-core"
pathToTarget="http://localhost:[TomcatConnectionPort]/manager/html">
  <arg id="linkFullPath" value="<full path to shortcut link>" lang="<language list>">
  </arg>
  <arg id="description" value="<tooltip string>" lang="<language list>">
  </arg>
</shortcut>
```

Use this shortcut element to customize the **Tomcat Configuration** link. Note the `pathToTarget` element.

```xml
<shortcut duSourceId="product.businessobjects64.shortcut.tomcat-4.0-core"
pathToTarget="[INSTALLDIR]Tomcat6\bin\tomcat6w.exe">
  <arg id="linkFullPath" value="<full path to shortcut link>" lang="<language list>">
  </arg>
  <arg id="description" value="<tooltip string>" lang="<language list>">
  </arg>
</shortcut>
```

**Example**

This example customizes the name of the **Tomcat Administration** shortcut to **tomcat(english and french) shortcut1** for English and French installations, and **tomcat (German) shortcut1** for German installations. It places the shortcuts in the **Start** menu entry called **Company Programs**. It customizes the tooltip to **tomcat(english and french) shortcut1** for English and French, **tomcat (all others) shortcut1** for all other languages.

```xml
<shortcut duSourceId="product.businessobjects64.shortcut.tomcat-4.0-core"
pathToTarget="http://localhost:[TomcatConnectionPort]/manager/html">
  <arg id="linkFullPath" value="[programmenufolder]\Company Programs\tomcat (english and french) shortcut1.lnk" lang="en;fr"/>
  <arg id="linkFullPath" value="[programmenufolder]\Company Programs\tomcat (German) shortcut1.lnk" lang="de"/>
  <arg id="linkFullPath" value="[programmenufolder]\Company Programs\tomcat (all others) shortcut1.lnk" lang="it;zh_cn"/>
  <arg id="description" value="tomcat(english and french) shortcut1" lang="en;fr"/>
  <arg id="description" value="tomcat (German) shortcut1" lang="de"/>
  <arg id="description" value="tomcat (all others) shortcut1" lang="it;zh_cn"/>
</shortcut>
```
3.6.2.3 Customizing the Windows Add Remove Program utility (Windows only)

You can customize the display name, the publisher, and the icon in the Windows Add Remove Program (ARP) utility. You cannot customize the version number. Use the following element:

```xml
<arp duSourceId="product.businessobjects64.arp-4.0-core">
  <arg id="publisher" value="<publisher name/>
  <arg id="display_name" value="<product name>"  lang="<language list>
  <arg id="display_icon" value="<full path to icon>
</arp>
```

Icons displayed in the Windows Add Remove Program utility are typically 16x16. Refer to Windows documentation for complete information on creating the icon.

**Example**

Change the product name in the Windows ARP utility to Sales Insight Platform. This change will only affect English installations. Change the publisher to Data Excellence Corp. Replace the display icon with the icon located at C:\SAPCustomTool\DEC_logo.ico.

```
<arp duSourceId="product.businessobjects64.arp-4.0-core">
  <arg id="publisher" value="Data Excellence Corp"/>
  <arg id="display_name" value="Sales Insight Platform" lang="en"/>
  <arg id="display_icon" value="C:\SAPCustomTool\DEC_logo.ico"/>
</arp>
```

The customization appears below:
3.6.2.4 Customizing the installation folder

You can customize the default installation folder. Use the \texttt{replaceProperty} element with \texttt{id="InstallDir"}:

\begin{verbatim}
<replaceProperty id="InstallDir" defaultValue="<default installation folder>"/>
\end{verbatim}

Use this element for both Windows and Unix installations.

\textbf{Example}

Change the default installation folder to 
\texttt{C:\Program Files (x86)\SalesDataInsight}.

\begin{verbatim}
<replaceProperty id="InstallDir" defaultValue="C:\Program Files (x86)\SalesDataInsight"/>
\end{verbatim}

3.6.3 Customizing user input

You can customize the default value of the user input that is collected by the installation program. Use the \texttt{replaceProperty} element with \texttt{id="<property id>"} and the new default value:

\begin{verbatim}
<replaceProperty id="<property id>" defaultValue="<value to use as default value>"/>
\end{verbatim}

For a list of property IDs, see \textit{Installation screen and property IDs}.

The Windows installation program collects user input using dialog boxes, radio buttons, and other user interface elements. The Unix and Linux installation program collects user input using the console entry. Both installation programs are customized in the same way.

\textbf{Example}

On the installation screen called \textit{Choose Install Type}, the default install type is \textit{Full}. This example changes the default install type to \textit{Custom/Expand}.

\begin{verbatim}
<replaceProperty id="InstallType" defaultValue="custom"/>
\end{verbatim}

The customization appears below:
3.6.4 Removing installation screens

You can remove installation screens from the installation program. Use the `removeDialog` element with the installation screen ID:

```
<removeDialog id="<installation screen ID>"/>
```

For a list of installation screen IDs, see *Installation screen and property IDs*.

**Example**

This example shows how to remove the installation screen titled *Select Java Web Application Server*.

```
<removeDialog id="ChooseWebAppServer.dialog"/>
```
3.6.5  Embedding a keycode

You can embed a keycode in the installation program so the customer does not need to enter one. This task involves:

- Providing a default value for the keycode
- Removing the installation screen in which the user enters a keycode

Example

Use the `replaceProperty` element with `id="ProductKey"` to provide a default keycode. Keycodes must have the format `XXXXX-XXXXXXX-XXXXXXX-XXXXXXX-XX`.

Use the `removeDialog` element with `id="EnterProductKey.dialog"` to remove the installation screen for the license key.

```xml
<replaceProperty id="ProductKey" defaultValue="XXXXX-XXXXXXX-XXXXXXX-XXXXXXX-XX"/>
<removeDialog id="EnterProductKey.dialog"/>
```

Related Information

- *Installation screen and property IDs* [page 45]
- *Customizing user input* [page 24]
- *Removing installation screens* [page 25]

3.6.6  Removing features

The BI platform is composed of many optional features. You can remove a feature from the installation program. Use the `removeFeature` element with `id="<feature id>"`:

```xml
<removeFeature id="<Feature ID>"/>
```

For a list of feature IDs, see *Feature IDs*.

When you specify that a feature will be removed, the SAP BusinessObjects customization tool removes all executables, installation screens, and other files that belong to that feature. Removing unnecessary features is a good way to reduce the size of the customized product.

Note

Do not remove every database access component. You must leave at least one database access component in order for the connection server to start and function correctly.
Example

Remove the Crystal Reports feature. This removes all Crystal Reports servers, files, and resources.

<removeFeature id="CrystalReportsServers"/>

Related Information

Feature IDs [page 40]

3.6.7 Preventing prerequisite checks

Prerequisites are conditions that must exist on the host machine in order for the installation program to succeed. The installation program verifies the existence of these prerequisites before starting, and displays the results in the Prerequisite check screen. Removing the Prerequisite check screen prevents prerequisite checks from being performed. Use the removeDialog element with id="CheckPreRequisites.dialog".

Note

It is recommended that you remove this installation screen only if you are performing the prerequisite checks by some other means. If the prerequisites are not met, the installation program will fail.

Example

This example removes the Prerequisite check screen and prevents prerequisite checks from being performed.

<removeDialog id="CheckPreRequisites.dialog"/>

3.6.8 Removing language packs

The installation program allows the user to select which language packs to install. A language pack contains translated versions of all the strings that are used by the installed product. By default, all possible language packs are included in the installation program. You can specify which language packs to include. Use the languageIncludeList element with a list of language codes:

<languageIncludeList value="<list of language codes>"/>

For a list of language codes, see Language Codes.

Note

Language packs can be large. The installation program will be smaller if fewer language packs are included.
Include English, French, and German language packs in the installation program. The user can select from this list during installation.

```xml
<languageIncludeList value="en;fr;de"/>
```

### 3.6.9 Preventing the WDeploy tool from running

If the user installs a web application server other than the default one, the WDeploy tool will run when the installation is finished. On Windows platforms, WDeploy is a GUI tool while on Unix and Linux platforms, it is a script.

You can turn this feature off. Use with the `<replaceProperty>` element with `defaultValue="0"`.

```xml
<replaceProperty id="LaunchWDeploy" defaultValue="0"/>
```

### 3.6.10 Removing the default database

The default database is included with the installation program, and customers can choose to use it as the system database. The default database is Sybase SQL Anywhere.

If the default database is not required you can remove it and force customers to choose another. Removing the default database is a good way to reduce the size of the installation program.

**To remove the default database**

Use the `<removeFeature>` element with `id="PlatformServers.IntegratedDB.SQLAnywhere"`. You may also want to remove the installation screen titled *Select Default or Existing Database* and set the user input property to *Use an existing database*.

```xml
<removeFeature id="PlatformServers.IntegratedDB.SQLAnywhere"/>
<removeDialog id="<SelectDataSource.dialog/>
<replaceProperty id="SelectIntegratedDatabase" defaultValue="0"/>
```

Example

This example removes the default database. It also removes the installation screen titled *Select Default or Existing Database*, and sets the user input property to *Use an existing database*.

```xml
<removeFeature id="PlatformServers.IntegratedDB.SQLAnywhere"/>
<removeDialog id="<SelectDataSource.dialog/>
<replaceProperty id="SelectIntegratedDatabase" defaultValue="0"/>
```
3.6.11 Changing resources

The installation program stores image and text files as resources in this folder:
\dunit\product.businessobjects64-4.0-core-32\setup.ui.framework\resources

You can customize the resources in this folder. Resources that are commonly customized include:
- Images in the installation program
- License agreement in the installation program

To customize a resource:

1. Create a custom resources folder, for example (on Windows): C:\SAPCustomTool\MyResources. The file can have any name, but will be visible to customers. Use the same folder for all resources that you customize.
2. Create a new resource with the same name and filepath as the original resource, and place it into the custom resources folder. See the related topics section for specific examples.
3. Add the `<resources>` element to the configuration file to specify the location of the custom resources folder, for example:
   `<resources cleanTarget="no" sourcePath="C:\SAPCustomTool\MyResources"/>

   **cleanTarget attribute**

   If you set cleanTarget = 'yes', the customization tool will delete the original resources folder and use only those resources included in the custom resources folder. This option is not recommended.

Related Information

*Customizing the images in the installation program* [page 29]
*Customizing the license agreement* [page 31]

3.6.11.1 Customizing the images in the installation program

You can customize the images in the installation program including the welcome screen, the top image for all screens, and the billboard for the progress dialog. Images are stored as files in the resources folder:
\dunit\product.businessobjects64-4.0-core-32\setup.ui.framework\resources
### Table 2: Image files in the `resources` folder

<table>
<thead>
<tr>
<th>Image name</th>
<th>File name</th>
<th>Size (W x H)</th>
<th>Default image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome screen</td>
<td>dialogFull.bmp</td>
<td>500 x 400 px</td>
<td></td>
</tr>
<tr>
<td>Top image for all screens</td>
<td>dialogTop.bmp</td>
<td>500 x 83 px</td>
<td></td>
</tr>
<tr>
<td>Billboard for progress dialog</td>
<td>billboard.bmp</td>
<td>500 x 193 px</td>
<td></td>
</tr>
</tbody>
</table>

You customize an image by creating a new image file, putting the file in the custom resources folder, and adding the `resources` element to the configuration file.

**Example**

**Customizing the image in the welcome screen on Windows platforms**

1. Create a folder called `MyResources` in the location `C:\SAPCustomTool`
2. Create a new image file called `dialogFull.bmp` and place it in the `C:\SAPCustomTool\MyResources` folder
3. Ensure the `resources` element exists in the configuration file as follows:

   ```xml
   <resources cleanTarget="no" sourcePath="C:\SAPCustomTool\MyResources"/>
   ```
### 3.6.11.2 Customizing the license agreement

You can customize the license agreement that is presented to the user during installation. License agreements are stored as text files in the resources folder:

```plaintext
dunit\product.businessobjects64-4.0-core-32\setup.ui.framework\resources\<language code>
```

For example, on Windows platforms, the English license agreement is located here:

```plaintext
dunit\product.businessobjects64-4.0-core-32\setup.ui.framework\resources\en\license_en.rtf
```

On Unix and Linux platforms, the English license agreement is located here:

```plaintext
dunit\product.businessobjects64-4.0-core-32\setup.ui.framework\resources\en/\license_en.txt
```

For a list of language codes, see Language Codes.

You customize the license agreement by creating a new license file, putting the file in the custom resources folder, and adding the `resources` element to the configuration file.

#### Example

**Customize the Japanese license agreement on Windows platforms**

The Japanese license agreement is stored here:

```plaintext
dunit\product.businessobjects64-4.0-core-32\setup.ui.framework\resources\ja\license_ja.rtf
```

To customize the Japanese license agreement:

1. Create a folder called `ja` in the location `C:\SAPCustomTool\MyResources`.
2. Create a new license agreement file called `license_ja.rtf` and place it in the `C:\SAPCustomTool\MyResources\ja` folder.
3. Ensure the `resources` element exists in the configuration file as follows:
   ```xml
   <resources cleanTarget="no" sourcePath="C:\SAPCustomTool\MyResources"/>
   ```

---

**Related Information**

*Changing resources* [page 29]
3.6.12 Removing items from the Collaterals folder

The SAP BusinessObjects Business Intelligence platform installation program stores tools, samples, and documentation in the Collaterals folder of the installation program. By default, a customized installation program will contain the default Collaterals folder with the default contents. You can remove unnecessary items from the Collaterals folder in order to reduce the size of your customized installation program. Use the collaters element with cleanTarget="yes" and sourcePath="<full path to custom Collaterals folder>":

```xml
<collaters cleanTarget="yes" sourcePath="<full path to custom Collaterals folder/>
```

**Note**

You must set the cleanTarget attribute to yes so the customization tool will replace the original folder with the new folder.

**To remove items from the Collaterals folder**

1. Copy the contents of the existing Collaterals folder to a new location, for example (on Windows) C: \SAPCustomTool\Utilities.
2. Remove any items from C:\SAPCustomTool\Utilities that are not required by your customized installation program. See below for more information.
3. Add the <collaters> element to the configuration file to specify the location of the custom collaters folder, for example:

```xml
<collaters cleanTarget="yes" sourcePath="C:\SAPCustomTool\Utilities"/>
```

<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
<th>When to remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaterals &gt; Add-Ons</td>
<td>Provides connectivity to SAP systems.</td>
<td>Remove if there is no need to connect to SAP systems.</td>
</tr>
<tr>
<td>SAP</td>
<td>Subversion is the default version control system that is used by Lifecycle Management (LCM).</td>
<td>Remove if the LCM feature is removed.</td>
</tr>
<tr>
<td>Subversion</td>
<td>The server monitoring feature can integrate with IBM Tivoli, and this item provides the connectivity.</td>
<td>Remove if integration with IBM Tivoli is not required.</td>
</tr>
<tr>
<td>Tivoli Agent</td>
<td>Required template files.</td>
<td>Do not remove this folder.</td>
</tr>
<tr>
<td>Customization Template</td>
<td>SAP Solution Manager Diagnostics (SMD) agent. SMD is used by SAP Sup-</td>
<td>Remove if the SMD feature is removed.</td>
</tr>
<tr>
<td>Folder</td>
<td>Description</td>
<td>When to remove</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Collaterals &gt; Docs</td>
<td>port tools to troubleshoot installed product.</td>
<td></td>
</tr>
<tr>
<td>Collaterals &gt; Docs</td>
<td>Documentation in every language that SAP BusinessObjects Business Intelligence platform supports.</td>
<td>Remove any languages that are not included in the customized installation program. For a list of language codes, see Language Codes.</td>
</tr>
<tr>
<td>Collaterals &gt; Tools &gt; CustomizationTool</td>
<td>The SAP BusinessObjects customization tool.</td>
<td>Remove this folder if the customers do not need to customize their own installation programs.</td>
</tr>
<tr>
<td>Collaterals &gt; Tools &gt; LCM command line tool</td>
<td>Command-line utility for Lifecycle Management (LCM).</td>
<td>Remove if the LCM feature is removed.</td>
</tr>
<tr>
<td>Collaterals &gt; Tools &gt; wdeploy</td>
<td>WDeploy is used to deploy web applications to web application servers other than Tomcat.</td>
<td>Not recommended to remove. Remove only if customers will use Tomcat exclusively.</td>
</tr>
</tbody>
</table>

3.7 Running the tool

The SAP BusinessObjects customization tool is included with the SAP BusinessObjects Business Intelligence platform installation package in this location:

Collaterals\Tools\CustomizationTool

On Windows platforms, the tool is named customizationtool.exe. On Unix and Linux platforms, the tool is named customizationtool.sh

This section explains the command line parameters.

Note

The customization tool may take several minutes to complete. You can check its progress by viewing the log file.

Example

This example runs the customization tool on a Windows platform. To use this example you must:

- Create a configuration file called oem.xml in the location C:\SAPCustomTool.
- Download the SAP BusinessObjects Business Intelligence platform installation package to the folder C:\SAPCustomTool\packages. See To download the installation program [page 13].
- Create a folder called output in the location C:\SAPCustomTool.

C:\SAPCustomTool\packages\Collaterals\Tools\CustomizationTool\customizationtool.exe xml=C:\SAPCustomTool\oem.xml packageDir=C:\SAPCustomTool\packages outputDir=C:\SAPCustomTool\output logDetail=error > C:\oemlog.log
### 3.7.1 Command line parameters

#### Table 4: Required parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example (Windows)</th>
</tr>
</thead>
<tbody>
<tr>
<td>xml</td>
<td>Full path to the configuration file.</td>
<td>xml=example_customization_win_boe.xml</td>
</tr>
<tr>
<td>packageDir</td>
<td>Full path to the folder that contains the installation program you are modifying. The installation program is downloaded from SAP Service Marketplace in order to start the installation of SAP BusinessObjects Business Intelligence platform. It contains the folders Collaterals, dunit, langs, and setup.engine in addition to other binaries.</td>
<td>packageDir=C:\SAPCustomTool\packages</td>
</tr>
<tr>
<td>outputDir</td>
<td>Full path to the folder where the customized installation program will be created. Must be empty before running the tool.</td>
<td>outputDir=C:\SAPCustomTool\output</td>
</tr>
</tbody>
</table>

#### Table 5: Optional parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example (Windows)</th>
</tr>
</thead>
</table>
| baselinePath | Full path to a root folder containing the original, non-customized versions of all previous full and update installation programs you have customized. Use a semicolon (; - Windows) or colon (:) - Unix) to separate multiple root folders. | Assume you want to customize SAP BusinessObjects Business Intelligence platform 4.0 Support Package 5 and you customized the previous programs: 4.0 SP2 (Full install), 4.0 SP4. Customize 4.0 Support Package 5, and provide the root folder path to the non-customized packages for the 4.0 SP2 full installation and SP4 update installation. For example, if the non-customized packages are contained in the following directory structure: 

```
C:\productUpdates\4.0\SP2 Full\SP4\n```

set the value to baselinePath=C:\productUpdates\4.0\ See [Customizing Support Packages and Patches](#) for more information and examples of the baselinePath parameter. |
<p>| logDetail | The level of logging detail. Default value is info. Accepted values: | logDetail=warn |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example (Windows)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• error&lt;br&gt;• warn&lt;br&gt;• info&lt;br&gt;• debug&lt;br&gt;• trace</td>
<td></td>
</tr>
<tr>
<td>action</td>
<td>The tool mode. Accepted values are: &lt;br&gt;• generate (default value) &lt;br&gt;The tool performs the specified customizations. &lt;br&gt;• validate &lt;br&gt;The tool validates the configuration file but does not perform any customizations.</td>
<td>action=validate</td>
</tr>
</tbody>
</table>

Related Information

Quick start for the Business Intelligence platform (Windows) [page 11]
Quick start for the Business Intelligence platform (Unix or Linux) [page 12]

3.8 Customizing update installation programs

Update installation programs are minor releases. Support Packages or Patches that contain updates to your existing BI platform software. Support Packages contain more updates than Patches but are released less frequently. You can use the SAP BusinessObjects customization tool to customize these update installation programs, but some modifications to the command line and configuration file are required.

3.8.1 Frequently asked questions about update installation programs

Where do I find Support Packages and Patches?

2. On the Find your software tab, under the A–Z Index, click Support Packages and Patches.
3. Select SBOP BI platform (former SBOP Enterprise) SBOP BI PLATFORM (ENTERPRISE) SBOP BI PLATFORM 4.0 Comprised Software Component Versions SBOP BI PLATFORM SERVERS 4.0 <platform>. 
4. Select your Support Package or Patch, then follow the instructions on the website to download and extract the objects.

What parts of update installation programs can I customize?

You can customize the same aspects of update installation programs as you did in the main installation program. Because minor release, Support Package and Patch updates contain fewer installation screens, not all of the customization steps apply. It is recommended to run the minor release, Support Package or Patch before customizing it to determine what customizations you require.

How do I customize update installation programs?

Update installation programs use the same architecture as a main installation program for the BI platform (full installation), so you can use the customization tool as described in Creating the configuration file and Running the tool, with some modifications to the command line and the configuration file. See How to customize Support Packages and Patches in this section.

Is it necessary to customize and install all minor release, Support Package and Patch updates?

No. As with non-customized versions of the BI platform, you only need to install the updates that you want. This may be a minor release, Support Package, a Patch, or any valid combination of the three updates.

Can I install a non-customized update on a customized BI platform installation?

Yes. Both customized and non-customized updates may be applied to your customized installation. However, non-customized minor release, Support Package, or Patch installation programs will not display your branding or installation customizations (such as removed features or shortcut changes) you created for the main installation program.

I have delivered a customized version of BI platform to customers but I want to modify the customizations in an update installation program. Is this possible?

This scenario is not supported. The customizations that you make to update installation programs must be consistent with the original customizations.
3.8.2 Quick start for update installation programs

Ensure you have customized and installed the main installation program, such as SAP BusinessObjects Business Intelligence platform Feature Pack 3 (full installation) using the instructions in *Quick start for the Business Intelligence platform (Windows)*, and that the non-customized installation program is located in C:\SAPCustomTool\packages.

This section shows you how to run the SAP BusinessObjects customization tool to customize the installation program for a Support Package (update installation). It uses the sample configuration file provided with the customization tool. Notice that the sample configuration file contains the <cloneProduct> element for the main installation program as well as the <clonePatchProduct> element for a Support Package update installation program.

**Note**

You can run this example only when a Support Package is available on [https://service.sap.com/bosap-support](https://service.sap.com/bosap-support).

1. Download the installation program for the BI Platform 4.0 Support Package to the folder C:\SAPCustomTool\SupportPackage.
2. Ensure the *product_version* for the <clonePatchProduct> element in the configuration file matches the version number of the Support Package that you downloaded. See *Customizing the product name and version number*.
3. Customize the BI Platform 4.0 Support Package and place the customized installation program in C:\SAPCustomTool\output\SupportPackage. Use the following command:
   ```
customizationtool.exe xml=example_customization_win_boe.xml packageDir=C:\SAPCustomTool\SupportPackage baselinePath=C:\SAPCustomTool\packages outputDir=C:\SAPCustomTool\output\SupportPackage logDetail=error > C:\oemlog_SP04.log
   ```
4. Use C:\SAPCustomTool\output\SupportPackage\setup.exe to run the customized installation program for the BI Platform 4.0 Support Package.

3.8.3 How to customize update installation programs

Use the configuration tool as described in *Creating the configuration file* [page 15] and *Running the Tool* to customize update installation programs for minor releases, Support Packages, and Patches, with the following differences:

- The configuration file must use the clonePatchProduct element (with the correct product ID), instead of the cloneProduct element.
- The configuration file must contain the complete <cloneProduct> element for the main installation package that you are updating. If it does not, it may cause unpredictable results, especially when customizations involve removing features.
- The configuration file cannot contain more than one clonePatchProduct. If you are customizing both a Support Package and a Patch for example, you must create two configuration files: one file containing cloneProduct and clonePatchProduct for the Support Package, and the other file containing cloneProduct and clonePatchProduct for the Patch.
Refer to all prerequisite installation programs using the `baselinePath` parameter.

All configuration file elements and command-line parameters can be used to customize update installation programs, but not all of them are applicable to every minor release, Support Package, or Patch. Run the installation program for the update first to determine what you need to customize, then use the information in Creating the configuration file [page 15] and IDs and codes for BI Platform customization [page 39] to create the customization file.

To specify the product version in the configuration file

The configuration file for update installation programs must contain the `product version` in the `clonePatchProduct` element as shown below:

```xml
<oem name="<any name>">
  <clonePatchProduct sourceId="<product version>">
    ...
  </clonePatchProduct>
</oem>
```

The `product version` in the configuration file must match the version number of the installation program that you are customizing. To find the version number, look in the `dunit` folder for a folder with a name in this format:

`product.boe64.patch-4.x.x.x-core-32`

Use the name of this folder as the `product version`.

Example

This example configuration file customizes the SAP BusinessObjects Business Intelligence platform 4.1 Patch 1, which has the product version `product.boe64.patch-4.1.0.1-core-32`. The configuration file customizes the product long name to `Custom Company Server` and the product short name to `Custom CS`.

```xml
<oem name="Custom Patch Tool">
  <clonePatchProduct sourceId="product.boe64.patch-4.1.0.1-core-32">
    ...
  </clonePatchProduct>
</oem>
```

To use the `baselinePath` parameter

Use the command line parameter `baselinePath` to refer to a root folder containing the original, non-customized versions of all previous full or update installation programs you have customized. This means you must keep the original installation packages.

Note

This parameter replaces the `baselinePackages` parameter introduced in 4.0 Feature Pack 3.
To simplify the `baselinePath` parameter value, reference a single root folder - the customization tool will ignore unneeded files and folders. Otherwise, use a semicolon (`;` - Windows) or colon (`:` - Unix) in the `baselinePath` value to specify multiple root folders. Consider the following examples on Windows.

**Example**

**Customizing 4.0 SP5 Patch 2**

Assume you are customizing BI platform 4.0 Support Package 5 Patch 2 and you customized the previous programs: 4.0 SP2 (Full install), 4.0 SP4, 4.0 SP5, 4.0 SP5 Patch 1. Assume the non-customized installation programs are located in the following directory structure:

```
C:\productUpdates\4.0\SP2 Full\SP4\SP5\SP5 Patch 1\``

Set the `baselinePath` parameter to the root folder:

```
baselinePath=C:\productUpdates\4.0\``

**Example**

**Customizing 4.1 SP 1**

Assume you are customizing BI platform 4.1 Support Package 1 and you customized the previous programs: 4.0 SP2 (Full install), 4.0 SP4, 4.0 SP5, 4.1. Assume the non-customized installation programs are located in the following directory structure:

```
C:\productUpdates\4.0\SP2 Full\SP4\SP5\4.1\Full\``

Set the `baselinePath` parameter to the root folder:

```
baselinePath=C:\productUpdates\``

### 3.9 IDs and codes for BI Platform customization

The following section contains a list of all the IDs and codes you can use to customize the installation program:

- Feature IDs
- Shortcut deployment unit IDs (Windows only)
- String IDs
- Language codes
- Installation screen and property IDs
3.9.1 Feature IDs

Use these IDs in the `removeFeature` element to remove features and their components from the installation program and the installed product.

For example, this ID will remove all the web tier components, including `JavaWebApps1` and `tomcat60`:

```xml
<removeFeature id="WebTier"/>
```

- **root**: remove all features
  - **WebTier**: (remove all web tier components listed below)
    - `JavaWebApps1` (Java Web Applications)
    - `tomcat60` (Tomcat 6.0)
  - **Servers**: (remove all server components listed below)
    - `PlatformServers`: (remove all platform servers listed below)
      - `CMS` (Central Management Server)
      - `FRS` (File Repository Servers)
      - `PlatformServers.IntegratedDB.SQLAnywhere` (bundled Sybase SQL Anywhere database server)
      - `PlatformServers.EventServer`
      - `PlatformServers.SystemLandscapeSupplier` (SLD)
      - `PlatformServers.WebAppContainerService` (WACS)
      - `AdaptiveProcessingServer` (platform processing)
      - `AdaptiveJobServer` (scheduling)
      - `Platform.RestWebService`
      - `Platform.Action_Framework.backend`: Insight to Action Framework
      - `Subversion` (Subversion version control system)
    - `ConnectionServices`: (remove all connectivity components listed below)
      - `ConnectionProcService`
    - `DataFederatorServices`: (remove all data federation components listed below)
      - `DataFederatorQueryService`
    - `AdvancedAnalysisServices`: (remove all Analysis components listed below)
      - `MultidimensionalAnalysisServices` (MDAS)
      - `BExWebApplicationsService`
    - `CrystalReportsServers`: (remove all SAP Crystal Reports components listed below)
      - `CrystalReportsProcServices` (SAP Crystal Reports Processing)
○ CrystalReportSchedulingServices
○ CrystalReport2011ProcServices (SAP Crystal Reports 2011 Processing)
○ CrystalReport2011SchedulingServices (SAP Crystal Reports 2011 Scheduling)
○ WebIServers: (remove all Web Intelligence components listed below)
  ○ WebIProcServer (Web Intelligence processing)
  ○ WebISchedulingServices (Web Intelligence scheduling)
○ XcelsiusServers (Dashboards)
○ MobileServices (remove all mobile services listed below)
  ○ MobileServers
  ○ MobileAddon (CMS plugin for Mobile)
○ IntegrationServers remove all integration components listed below
  ○ BWPublisherServer (SAP BW authentication and SAP BW Publisher support)
○ MultitenancyManager
○ AdministratorTools (remove all administrator tools listed below)
  ○ UpgradeManager (upgrade management tool)
• DeveloperTools (remove developer tool components listed below)
  ○ BOE64bitNETSDK (64-bit SAP BusinessObjects Business Intelligence platform .NET SDK)
• DataAccess (remove database access components listed below)

  ○ DataAccess.DataFederator
  ○ DataAccess.HPNeoView
  ○ DataAccess.MySQL
  ○ DataAccess.GenericJDBC
  ○ DataAccess.GenericODBC
  ○ DataAccess.GenericOLEDB
  ○ DataAccess.OptionalDataDirectODBC
  ○ DataAccess.MaxDB
  ○ DataAccess.SAPHANA
  ○ DataAccess.Salesforce (Salesforce.com)
  ○ DataAccess.Netezza
  ○ DataAccess.Microsoft_AnalyticalServices
  ○ DataAccess.MicrosoftExchange
  ○ DataAccess.MicrosoftOutlook
  ○ DataAccess.Microsoft_SQLServer
  ○ DataAccess.Microsoft_Access
  ○ DataAccess.Ingres
  ○ DataAccess.Greenplum

**Note**

Do not remove every database access component. You must leave at least one database access component in order for the connection server to start and function correctly.
○ DataAccess.IBMDB2
○ DataAccess.Informix
○ DataAccess.ProgressOpenEdge
○ DataAccess.Oracle
○ DataAccess.Sybase
○ DataAccess.Teradata
○ DataAccess.SAPBW
○ DataAccess.SAPERP
○ DataAccess.OData
○ DataAccess.Excel
○ DataAccess.XMLWebServices
○ DataAccess.SAP (security and data access for SAP BW and R/3 systems)
○ DataAccess.PersonalFiles
○ DataAccess.JavaBean
○ DataAccess.OpenConnectivity
○ DataAccess.HSQLDB
○ DataAccess.Derby
○ DataAccess.HadoopHive
○ DataAccess.Essbase
○ DataAccess.Peoplesoft (PeopleSoft Enterprise)
○ DataAccess.JDEdwards (JD Edwards EnterpriseOne)
○ DataAccess.Siebel (Siebel Enterprise server)
○ DataAccess.OracleEBS (Oracle E-Business Suite)
○ DataAccess.Universe (SAP BusinessObjects universe)
○ DataAccess.MyCube (OLAP cube)
○ DataAccess.XML
○ DataAccess.ADO.NET
○ DataAccess.COMData
○ DataAccess.DataSet (Dataset Consumer)
○ DataAccess.SymantecACT
○ DataAccess.BDE (IDAPI database DLL)
○ DataAccess.CDO (Crystal data objects)
○ DataAccess.FieldDefinitions
○ DataAccess.FileSystem
○ DataAccess.NTEventLog
○ DataAccess.WebActivityLog
○ DataAccess.Btrieve (Pervasive database driver)
○ DataAccess.dBase
○ DataAccess.UWSC (Universal Web Services Connector (UWSC))

● Samples (remove sample reports and data sources)
3.9.2 Shortcut deployment unit IDs (Windows only)

Use the deployment unit IDs in the shortcut element to change the location and name of the program shortcuts in the Windows Start menu.

Table 6: Shortcut deployment unit IDs

<table>
<thead>
<tr>
<th>Shortcut deployment unit ID</th>
<th>Shortcut target</th>
</tr>
</thead>
<tbody>
<tr>
<td>product.businessobjects64.shortcut.wdeploy-4.0-core</td>
<td>WDeploy</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.ccm-4.0-core</td>
<td>Central Configuration Manager</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.cmc-4.0-core</td>
<td>Central Management Console</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.infoview-4.0-core</td>
<td>BI launch pad (InfoView)</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.odbc-4.0-core</td>
<td>32-bit Data Source Administrator</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.online-doc-4.0-core</td>
<td>Online documentation</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.tomcat-4.0-core</td>
<td>Apache Tomcat. See for additional instructions.</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.upgrade-4.0-core</td>
<td>Upgrade management tool</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.wacs.infoview-4.0-core</td>
<td>WACs stored in InfoView</td>
</tr>
<tr>
<td>product.businessobjects64.shortcut.wacs-4.0-core</td>
<td>Web Application Container Server</td>
</tr>
</tbody>
</table>
3.9.3 String IDs

You can change the value of all strings in the installation program. You can replace a string for all languages or for a specific language. Use the `replaceString` element, for example:

```xml
<replaceString id="productname" value="Sales Data Insight lang="all"/>
```

### Table 7: Commonly changed strings

<table>
<thead>
<tr>
<th>String ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>product.boe64_name</td>
<td>Product long name</td>
</tr>
<tr>
<td>product.boe64_shortname</td>
<td>Product short name</td>
</tr>
<tr>
<td>product_version</td>
<td>Product version</td>
</tr>
<tr>
<td>product_majorversion</td>
<td>Product major version</td>
</tr>
</tbody>
</table>

**Related Information**

*Customizing the product name and version number* [page 17]

3.9.4 Language codes

The SAP BusinessObjects customization tool uses these language codes to represent supported languages:

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>EN</td>
</tr>
<tr>
<td>Czech</td>
<td>CS</td>
</tr>
<tr>
<td>Danish</td>
<td>DA</td>
</tr>
<tr>
<td>Dutch</td>
<td>NL</td>
</tr>
<tr>
<td>Finnish</td>
<td>FI</td>
</tr>
<tr>
<td>French</td>
<td>FR</td>
</tr>
<tr>
<td>German</td>
<td>DE</td>
</tr>
<tr>
<td>Hungarian</td>
<td>HU</td>
</tr>
<tr>
<td>Italian</td>
<td>IT</td>
</tr>
<tr>
<td>Japanese</td>
<td>JA</td>
</tr>
<tr>
<td>Korean</td>
<td>KO</td>
</tr>
<tr>
<td>Norwegian Bokmal</td>
<td>NB</td>
</tr>
<tr>
<td>Polish</td>
<td>PL</td>
</tr>
</tbody>
</table>
3.9.5 Installation screen and property IDs

Use the installation screen IDs in the `removeDialog` element to remove screens from the installation program. For example, use this element to remove the `User Information` screen:

```xml
<removeDialog id="EnterProductKey.dialog"/>
```

Use the properties and the property values to prepopulate user input. For example, use this element to set the default installation type to `custom`:

```xml
<replaceProperty id="InstallType" defaultValue="custom"/>
```

**Note**

Property values are case-sensitive.
<table>
<thead>
<tr>
<th>Title of installation screen</th>
<th>Installation screen ID</th>
<th>Property ID(s)</th>
<th>Allowed property value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prerequisite check</strong></td>
<td>CheckPreRequisites.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Please choose a setup language</strong></td>
<td>SelectUILanguage.dialog</td>
<td>SortedAvailableSetupLanguages</td>
<td>Set of language codes that the installation program can be run in, for example &quot;en;ja&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetupUILanguage</td>
<td>Single language code describing the language that the installation program will be run in, for example &quot;en&quot;</td>
</tr>
<tr>
<td><strong>Welcome to the installation wizard ...</strong></td>
<td>ShowWelcomeScreen.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>License Agreement</strong></td>
<td>ShowLicenseAgreement.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>User Information</strong></td>
<td>EnterProductKey.dialog</td>
<td>RegisteredUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RegisteredCompany</td>
<td>Company name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProductKey</td>
<td>Product keycode</td>
</tr>
<tr>
<td><strong>Choose Language Packs</strong></td>
<td>SelectLanguagePack.dialog</td>
<td>SelectedLanguagePacks</td>
<td>The set of language packs to be installed, for example &quot;en;ja&quot; For a list of language codes, see Language Codes.</td>
</tr>
</tbody>
</table>
| **Choose Install Type**      | ChooseInstallType.dialog | InstallType | • default (Full)  
• custom  
• webtier |
| **Specify the Destination Folder** | ChooseInstallDir.dialog | InstallDir | Installation folder       |
| **Select Default or Existing Database** | SelectDataSource.dialog | SelectIntegratedDatabase | • 0 (Use an existing database)  
• 1 (Install and use the default database) |
| **Expand Installation**      | ExpandInstallMessage    | Not applicable | Not applicable            |
| **Select Java Web Application Server** | ChooseWebAppServer.dialog | WebAppServerType | • tomcat  
• manual  
• wacs |
| **Select Features**          | SelectFeatures.dialog  | Not applicable | Not applicable            |
| **Configure Version Management** | SelectLCM.dialog | NewOrExistingLCM | • existing  
• new |
<table>
<thead>
<tr>
<th>Title of installation screen</th>
<th>Installation screen ID</th>
<th>Property ID(s)</th>
<th>Allowed property value(s)</th>
</tr>
</thead>
</table>
| **Expand Installation**      | ChooseExpandInstall.dia-  | NewOrExpandIn- | ● new  
|                              | log                     | stall           | ● expand                  |
| **Configure Subversion**     | SetLCMConfig.dia-       | LCMName         | Repository name           |
|                              | log                     | LCMPort         | Repository port           |
|                              |                         | LCMUserName     | Repository user           |
|                              |                         | LCMPassword     | Repository password       |
|                              |                         | LCMPasswordCon-  | Confirm password          |
|                              |                         | firm            |                           |
| **Configure Server Intelli-  | GetSIAInfo.dialog       | SIAPort         | SIA port                  |
| gence Agent (SIA)            |                         | SIAName         | Node name                 |
| **Configure Central Manage-  | GetCMSInfo.dialog       | CMSPort         | Any valid port number     |
| ment Server (CMS)            |                         |                 |                           |
| **Configure CMS Account**    | GetCMSPassword.dia-     | CMSPassword     | The CMS password          |
|                              | log                     | CMSPasswordCon- | The CMS password          |
|                              |                         | firm            |                           |
|                              |                         | ClusterKey      | The CMS cluster key       |
|                              |                         | ClusterKeyCon-  | The CMS cluster key       |
|                              |                         | firm            |                           |
| **Configure Sybase SQL Any-  | GetSQLAnywhereInfo.dia- | SQLAnywhereSer- | The SQL Anywhere server n- |
| where**                      | log                     | verName         | ame (Unix and Linux only) |
|                              |                         | SQLAnywherePor- | The SQL Anywhere port     |
|                              |                         | t                     |                           |
|                              |                         | SQLAnywhereAd-   | The SQLAnywhere administra- |
|                              |                         | minPassword      | tor password (username is dba). |
|                              |                         |                 |                           |
| **Configure Microsoft SQL    | GetSQLExpressInfo.dia-  | SQLExpressAdmin- | The SQL administrator pa- |
| Server 2008 Express**       | log                     | Password         | ssword                    |
|                              |                         | SQLExpressBOEU-  | The SAP BusinessObjects Bi |
|                              |                         | serName         | platform username         |
|                              |                         | SQLExpressBOEU-  | The SAP BusinessObjects Bi |
|                              |                         | serPassword     | platform password         |
| **Choose to start or stop** | ChooseToEnableServer-   | EnableServers   | ● 0 (Stop servers upon installation)  
<p>| servers**                    | s.dialog               |                 | ● 1 (Start servers upon installation) |
| <strong>Configure Tomcat</strong>         | ShowTomcatInfo.dia-     | TomcatConne-    | Connection port           |
|                              | log                     | ctionPort       |                           |</p>
<table>
<thead>
<tr>
<th>Title of installation screen</th>
<th>Installation screen ID</th>
<th>Property ID(s)</th>
<th>Allowed property value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>● GetTomcatInfo.dialog</td>
<td>TomcatShutdownPort</td>
<td>Shutdown port</td>
<td></td>
</tr>
<tr>
<td>Both dialog IDs must be included in the configuration file in order to remove the Configure Tomcat screen. That is, you must include two removeDialog elements in your configuration file.</td>
<td>TomcatRedirectPort</td>
<td>Redirect port</td>
<td></td>
</tr>
</tbody>
</table>

| Configure Connectivity Solution Manager Diagnostics (SMD) Agent | SelectSMDIntegrate.dialog | ChooseSMDIntegrate | ● nointegrate (Do not integrate)  
● integrate (Integrate) |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Connectivity to SMD Agent</td>
<td>ConfigureSMDAgent.dialog</td>
<td>SMDAgent_HOST</td>
<td>SMD agent host</td>
</tr>
<tr>
<td></td>
<td>SMDAgent_PORT</td>
<td>SMD agent port</td>
<td></td>
</tr>
</tbody>
</table>
| Introscope Integration | SelectIntroscopeIntegrate.dialog | ChooseIntroscopeIntegration | ● nointegrate (do not integrate)  
● integrate (integrate) |
|                                                                       | Introscope_ENT_HOST      | Introscope host name  |
|                                                                       | Introscope_ENT_PORT      | Introscope port number  |
| Configure Connectivity to Introscope Enterprise Manager | ConfigureIntroscope.dialog | Introscope_ENT_HOST | Enterprise manager host  |
|                                                                 | Introscope_ENT_PORT      | Enterprise manager port  |
|                                                                 | Introscope_ENT_INSTRUCTION | Set to true to indicate that you configured this installation screen  |
| Configure HTTP Listening port | GetWACSPort.dialog       | WACSPort           | Port number for the web application container service  |
| Select existing Auditing Database Type | SelectAuditDatabase.dialog | UsingAuditDBType | ● sybase  
● db2  
● oracle  
● mysql  
● mssql  
● maxdb  
● none |
<table>
<thead>
<tr>
<th>Title of installation screen</th>
<th>Installation screen ID</th>
<th>Property ID(s)</th>
<th>Allowed property value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select existing CMS Database Type</td>
<td>SelectCMSDatabase.dialog</td>
<td>UsingCMSDBType</td>
<td>sybase, db2, oracle, mysql, mssql, maxdb</td>
</tr>
<tr>
<td>Existing CMS Deployment Information</td>
<td>SetRemoteCMSInfo.dialog</td>
<td>RemoteCMSName</td>
<td>Name of the existing CMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoteCMSPort</td>
<td>Port number for the existing CMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoteCMSAdminName</td>
<td>Administrator’s username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RemoteCMSAdminPassword</td>
<td>Administrator’s password</td>
</tr>
<tr>
<td>SAP BusinessObjects BI platform has been successfully installed</td>
<td>ShowInstallCompleteLaunchWDeploy.dialog</td>
<td>LaunchWDeploy</td>
<td>0 (Do not launch WDeploy tool after install), 1 (Automatically launch WDeploy tool after install)</td>
</tr>
<tr>
<td>Configure Auditing Database - DB2</td>
<td>ExistingAuditDB2.dialog</td>
<td>ExistingAuditingDBServer</td>
<td>DB2 alias name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBPassword</td>
<td>Password</td>
</tr>
<tr>
<td>Configure Auditing Database - MaxDB</td>
<td>ExistingAuditMaxDB.dialog</td>
<td>ExistingAuditingDBDatabase</td>
<td>Name of existing auditing database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBDatabaseUser</td>
<td>Username for existing database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBPassword</td>
<td>User’s password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBPort</td>
<td>Port number for existing database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBServer</td>
<td>MaxDB server name</td>
</tr>
<tr>
<td>Configure Auditing Database - Oracle</td>
<td>ExistingAuditOracle.dialog</td>
<td>ExistingAuditingDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBPassword</td>
<td>Password</td>
</tr>
<tr>
<td>Title of installation screen</td>
<td>Installation screen ID</td>
<td>Property ID(s)</td>
<td>Allowed property value(s)</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------</td>
<td>----------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Configure Auditing Database - SQL Server (ODBC)</td>
<td>ExistingAuditditMSSQL.dialog</td>
<td>ExistingAuditingDBDatabase</td>
<td>SQL database name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBServer</td>
<td>SQL server name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBPassword</td>
<td>Password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBUseTrustedConnection</td>
<td>Use trusted connection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBDSN</td>
<td>Data source name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBShowSysDB</td>
<td>Show system database</td>
</tr>
<tr>
<td>Configure Auditing Database - MySQL</td>
<td>ExistingAuditMySQL.dialog</td>
<td>ExistingAuditingDBDatabase</td>
<td>Auditing database name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBPassword</td>
<td>Password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBPort</td>
<td>MySQL Port</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditingDBServer</td>
<td>MySQL Server</td>
</tr>
<tr>
<td>Configure Auditing Database - SQL Anywhere (ODBC)</td>
<td>ExistingAuditSQLAnywhere.dialog</td>
<td>ExistingAuditDBUser</td>
<td>Username for existing database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditDBPassword</td>
<td>User’s password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditDBDatabase</td>
<td>Name of existing auditing database</td>
</tr>
<tr>
<td>Configure Auditing Database - Sybase</td>
<td>ExistingAuditSybase.dialog</td>
<td>ExistingAuditDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingAuditDBPassword</td>
<td>Password</td>
</tr>
<tr>
<td>Title of installation screen</td>
<td>Installation screen ID</td>
<td>Property ID(s)</td>
<td>Allowed property value(s)</td>
</tr>
<tr>
<td>-----------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>----------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>Configure CMS Repository Database - SQL Anywhere (ODBC)</td>
<td>ExistingCMSSQLAnywhere.dialog</td>
<td>ExistingAUDitingDBServer</td>
<td>Sybase service name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBDSN</td>
<td>Data source name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBUser</td>
<td>Username for existing database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBPassword</td>
<td>User’s password</td>
</tr>
<tr>
<td>Configure CMS Repository Database - DB2</td>
<td>ExistingCMSDB2.dialog</td>
<td>ExistingCMSDBServer</td>
<td>DB2 Alias Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBPassword</td>
<td>Password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBReset</td>
<td>• 0 (Do not reset existing database)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 1 (Reset existing database)</td>
</tr>
<tr>
<td>Configure CMS Repository Database - MaxDB</td>
<td>ExistingCMSMaxDB.dialog</td>
<td>ExistingCMSDBServer</td>
<td>CMS Database Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBPassword</td>
<td>Password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBReset</td>
<td>• 0 (Do not reset existing database)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 1 (Reset existing database)</td>
</tr>
<tr>
<td>Configure CMS Repository Database - SQL Server</td>
<td>ExistingCMSMSSQL.dialog</td>
<td>ExistingCMSDBServer</td>
<td>Existing server name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBPassword</td>
<td>Password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBReset</td>
<td>• 0 (Do not reset existing database)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 1 (Reset existing database)</td>
</tr>
<tr>
<td>Title of installation screen</td>
<td>Installation screen ID</td>
<td>Property ID(s)</td>
<td>Allowed property value(s)</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------</td>
<td>----------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBDatabase</td>
<td>CMS Database Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBUseTrustedConnection</td>
<td>Use trusted connection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBDSN</td>
<td>Data source name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBShowSysDB</td>
<td>Show system database</td>
</tr>
<tr>
<td><strong>Configure CMS Repository Database - MySQL</strong></td>
<td>ExistingCMSMySQL.dialog</td>
<td>ExistingCMSDBServer</td>
<td>MySQL Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBUser</td>
<td>Username</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExistingCMSDBPassword</td>
<td>Password</td>
</tr>
</tbody>
</table>
|                              |                         | ExistingCMSDBReset    | ● 0  (Do not reset existing database)  
|                              |                         |                           | ● 1  (Reset existing database)      |
|                              |                         | ExistingCMSDBDatabase | CMS Database Name          |
|                              |                         | ExistingCMSDBPort     | MySQL Port                 |
| **Configure CMS Repository Database - Oracle** | ExistingCMSOracle.dialog | ExistingCMSDBServer | Oracle TNSNAME              |
|                              |                         | ExistingCMSDBUser     | Username                   |
|                              |                         | ExistingCMSDBPassword | Password                   |
|                              |                         | ExistingCMSDBReset    | ● 0  (Do not reset existing database)  
|                              |                         |                           | ● 1  (Reset existing database)      |
| **Configure CMS Repository Database - Sybase** | ExistingCMSSybase.dialog | ExistingCMSDBServer | Sybase service name         |
|                              |                         | ExistingCMSDBUser     | Username                   |
|                              |                         | ExistingCMSDBPassword | Password                   |
|                              |                         | ExistingCMSDBReset    | Reset existing database    |
### Title of installation screen

<table>
<thead>
<tr>
<th>Installation screen ID</th>
<th>Property ID(s)</th>
<th>Allowed property value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configure Subversion</strong></td>
<td>SetLCMConfig.dialog</td>
<td>LCMName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCMPort</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCMUserName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCMPassword</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCMPasswordConfirm</td>
</tr>
<tr>
<td><strong>SAP BusinessObjects BI platform 4.0 FP3 has been successfully installed</strong></td>
<td>ShowInstallComplete.dialog</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Start Installation</strong></td>
<td>ShowInstallSummary.dialog</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Post Installation Steps</strong></td>
<td>ShowPostInstall.dialog</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Uninstall Confirmation</strong></td>
<td>VerifyToRemove.dialog</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>SAP BusinessObjects BI platform 4.0 FP3 has been successfully uninstalled</strong></td>
<td>ShowUninstallComplete.dialog</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

**Related Information**

- *Customizing user input* [page 24]
- *Removing installation screens* [page 25]
4 Web Application Customization

4.1 Introduction

You can apply your own branding (or “skinning”) to the BI launch pad, OpenDocument, and Crystal Reports JavaScript viewer web applications. For example, you can customize your OEM systems by applying your own corporate identity elements.

You can customize the following web and graphic elements:

- Favicon (the icon displayed in the browser URL bar)
- Logos
- Certain background patterns and colors
- Certain animated gifs (for example, the progress indicator)
- Certain CSS styles (borders, padding, margins, and so on)
- JavaScript files for the Crystal Reports JavaScript viewer

You can customize many aspects of the BI platform web applications, and you may choose to customize only a subset of these options.

Who should use this information?

This section is intended for web application designers, developers, and system administrators who are customizing BI platform web applications. Familiarity with the basics of CSS design and Java web application archives is required. If you are deploying customizations, then you should also be familiar with the methods to install and deploy BI platform web applications to an application server.

For information on installing the SAP BusinessObjects Business Intelligence platform, see the Business Intelligence Platform Installation Guide.

For information on deploying BI platform web applications using the WDeploy tool, see the Business Intelligence Platform Web Application Deployment Guide.

4.1.1 Key concepts

To make and deploy your customizations, you should understand the following concepts:

Installation package

The installation package is the set of binaries that are downloaded from SAP Service Marketplace in order to start the installation of SAP BusinessObjects Business Intelligence platform. It contains the folders Collaterals, dunit.langs, and setup.engine in addition to other binaries.
Customization template

The template.zip file is located in the Collaterals\CustomizationTemplate folder of your installation package, and contains the branding bundles (JAR files) to customize before installing SAP BusinessObjects Business Intelligence platform. This file is the starting point for customizing your web applications.

BOE WAR file

BOE.war is the primary web application archive for the BI platform. BI launch pad, OpenDocument, the Crystal Reports JavaScript viewer, and the changes you make in template.zip to their respective branding bundles are included by the installation program in BOE.war. To deploy your customizations and make these applications available to your customers, you must deploy BOE.war to your Java application server either during or after the installation process.

Branding bundles

A branding bundle is a JAR file that contains the custom resources (CSS, icons, images, JavaScripts) that you include in the installation program within template.zip. The following branding bundles are included:

- **com.businessobjects.webpath.InfoViewBranding.jar** (BI launch pad)
  This branding bundle consists of two main folders: a css folder containing a custom CSS file, and an images folder containing a favicon and theme subfolder with custom logos, images, and animated GIFs.

  ```
  \com.businessobjects.webpath.InfoViewBranding\web
  \css
  \images
  favicon.ico
  \theme
  *\.png, *\.gif
  ```

  This branding bundle consists of two main folders: a css folder containing a custom CSS file, and an images folder containing a theme subfolder with custom logos and images.

  ```
  \com.businessobjects.webpath.InfoViewBranding\web
  \service
  \css
  \images
  \theme
  *\.png
  ```

- **com.businessobjects.webpath.CrystalReports_oem.jar** (Crystal Reports JavaScript viewer)
  This branding bundle consists of two main resources: a JavaScript file with custom listeners where you can define new behavior for certain viewer events, and a JSON properties file to reference any custom JavaScript files or images you choose to use for the viewer.

  ```
  \com.businessobjects.webpath.CrystalReports_oem\web
  CustomListener.js
  ```
Web application deployment

The SAP BusinessObjects Business Intelligence platform installation program can deploy BOE.war only to the bundled Tomcat web application server. Other supported web application servers require that you deploy the web applications after the installation is complete. It is recommended that you use the WDeploy tool.

4.1.2 Testing your customizations

Before performing customizations on your production systems, it is good practice to test your customizations first on a test installation. In a default installation that uses the bundled Tomcat server, you can instantly see the effects of your changes by making temporary modifications to the webpath.InfoViewBranding, webpath.OpenDocumentBranding, and webpath.CrystalReports_oem folders in the Tomcat work directory: C:\SAP BusinessObjects\Tomcat6\work\Catalina\localhost\BOE\eclipse\plugins \webpath.OpenDocumentBranding\web\service. These folders have the same structure as the branding resources contained in template.zip.

*Note*
The Tomcat work directory is not permanent and your temporary changes are deleted after a Tomcat restart.

4.2 Quick start

Before you start, back up \Collaterals\Tools\CustomizationTemplate\template.zip from your installation package.

This section shows you the basic steps required to customize and deploy one of the BI platform web applications: BI launch pad. The steps shown are also applicable to OpenDocument and the Crystal Reports JavaScript viewer.

*Note*
This quick start describes the end-to-end customization, including performing a full installation of SAP BusinessObjects Business Intelligence platform and the deployment of BOE.war to an application server. These steps may take considerable time.

1. Locate template.zip in your SAP BusinessObjects Business Intelligence platform installation package at: \Collaterals\Tools\CustomizationTemplate.
2. Extract the contents of template.zip to a working folder.
3. Unpackage the BI launch pad branding bundle.
   `com.businessobjects.webpath.InfoViewBranding.jar`
   ```bash
   jar xf com.businessobjects.webpath.InfoViewBranding.jar
   ```

4. Customize the default favicon that is displayed in the browser URL bar for BI launch pad.
   The BI launch pad branding bundle contains a sample favicon. Copy `web\sample\images\favicon.ico` up one level to `web\images\favicon.ico`.

5. Repackage `com.businessobjects.webpath.InfoViewBranding.jar` containing your new favicon, and include it in `template.zip`.
   To repackage the `web` and `META-INF` folder contents back into
   `com.businessobjects.webpath.InfoViewBranding.jar`:
   ```bash
   jar cf com.businessobjects.webpath.InfoViewBranding.jar web META-INF
   ```

6. Create a subfolder called `OEMZips` at:
   ```shell
   \dunit\product.businessobjects64.oemzips-4.0-core-nu
   ```

7. Add `template.zip` to the `OEMZips` folder.
   Your customized zip file is now at:
   ```shell
   \dunit\product.businessobjects64.oemzips-4.0-core-nu\OEMZips\template.zip
   ```

8. Install and deploy `BOE.war` to your Java application server using one of the following options:
   ![Option Table]
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the bundled Tomcat server</td>
<td>Selected during the installation process.</td>
</tr>
<tr>
<td>Use your own supported Java application server</td>
<td>Performed after the installation program completes. Use the WDeploy tool.</td>
</tr>
</tbody>
</table>

   Use `setup.exe` (Windows) or `setup.sh` (Unix) to start the installation process.

9. After successful installation and deployment, test your changes by accessing BI launch pad: `http://<web server name>:<port>/BOE/BI`

   You can see the new favicon in your browser URL bar:
   ![New Favicon]

### 4.3 Customizing BI launch pad

For BI launch pad, you can customize the favicon, logo, backgrounds, styles, and more. Most of these customizations involve changing the CSS rules in the `customize.css` file. All customizations must be made available in the `web` folder of `com.businessobjects.webpath.InfoViewBranding.jar` to take effect, as follows:

```bash
\web
  \css
    customize.css
  \images
    favicon.ico
```
4.3.1 To customize the favicon image

The favicon is the small icon displayed in a browser’s address bar when viewing the BI launch pad.

Replace the favicon.ico file stored in the web\images folder with your own favicon.ico image.

4.3.2 To customize logos

Logos used in the BI launch pad can be customized by editing the CSS rules in the web\css\customize.css file. If you are using custom images and referring to them in the customize.css file, make sure you place them in the web\images\theme folder.

4.3.3 Customizing other user interface elements

Logos, background, styles, and other user interface elements of the BI launch pad can be customized by editing the CSS rules in the web\css\customize.css file. If you are using custom images and referring to them in the customize.css file, make sure you place them in the web\images\theme folder.

The following diagrams show the elements customized in the sample branding bundle, for reference purposes. The numbers in the balloons refer to sections in the bundle’s customize.css file.
1. (0.1) customize the background of pages and sub-pages (inside frames)
2. (0.2) input text field
3. (0.4) password field
4. (1.1) authentication fields container
5. (1.2) container of everything
6. (1.3) horizontal rules
7. (1.5) “Log On” button
8. (1.6) banner background pattern
9. (1.7) logo

10. (2.1.1) top banner area (background pattern)
11. (2.1.2) banner logo
12. (2.2.1) tab container
13. (2.2.2) active tab
14. (2.2.3) inactive tab
15. (2.3.1) BI launch pad icon
16. (2.2.4) tab buttons
17. (0.7) spinner
18. (3.1.1) accordion inactive header
19. (3.1.2) accordion active header
20. (3.1.3) accordion drawer/tree background
21. (3.3.1) toolbar background
22. (3.4) footer background
23. (3.5.2) selected unfocused row in both the list pane and the tree view (on the left)
24. (3.6) accordion-list pane resize bar
25. (3.2.1) list pane container
26. (3.2.2) list pane heading
27. (3.2.3) list pane rows
28. (3.5.1) selected focused row in both the list pane and tree view (on the left)
29. (5.1.2, 5.1.3, 5.1.4) resize handle and knob
30. (0.8.1) context menu container
31. (0.8.2) context menu body
32. (0.8.3) context menu item
33. (0.8.4) context menu selected item
34. (0.9) tooltip
35. (4.1) details container
36. (5.2) details header
37. (6.1) simple dialog container
38. (6.2.1) simple dialog header
39. (6.2.2) close button dialog header
40. (6.3) simple dialog body
41. (6.4) simple dialog footer
42. (6.5) simple dialog text field (overrides general text field)
43. (6.6) simple dialog buttons

44. (3.3.2, 3.3.3) toolbar button hover/press
45. (3.3.4) toolbar menu item
46. (3.3.5) toolbar menu item hovered
47. (3.3.6) toolbar menu separator
48. (3.3.7) toolbar menu frame
49. (3.3.8) toolbar menu refresh icon

50. (7.1.1) large dialog header
51. (7.1.2) large dialog header – maximize button (hover)
4.3.4  Working with BI workspaces and compound modules

You can also use a BI workspace or a compound module as your BI launch pad home page. You can customize the workspace or compound module to match the OEM style of your BI launch pad.

Note

The customized OEM style is reflected only on the home page. If the same workspace or compound module is opened outside of the home page (in regular view), the standard style will be used.

The following diagrams show the elements customized in the sample branding bundle, for reference purposes. The numbers in the balloons refer to sections in the bundle’s customize.css file.

For the default home page or module

The following settings can be used to customize the default home page, or any BI workspace or compound module that has been set as the home page.
1. (8.1.2) module title background
2. (8.1.3) module border
3. (8.2.1) background of BI launch pad module
4. (8.2.2) color of See More text

For the regular view of a BI workspace

The following settings can be used to customize the appearance of a BI workspace in regular view.
1. (8.3.1) customized top tab container
2. (8.3.2) customized subtab container
3. (8.3.4) active top tab
4. (8.3.5) inactive top tab
5. (8.3.6) subtab

4.3.4.1 To match the style of a BI workspace to the style of BI launch pad

1. Open the BI workspace for editing.
2. From the first tab of the workspace, click Properties. The Properties dialog box appears.
3. Select the icon of the option immediately before the (last) Default style option.
4. Click OK.

4.3.4.2 To match the style of a compound module to the style of BI launch pad

1. On the BI launch pad home page, click Preferences. The Preferences – Administrator dialog box appears.
2. Select BI workspaces from the Preferences list.
3. From the list in the main pane, select BI Launch Pad.

4.3.5 To change the name of BI launch pad

You may want to change the name of BI launch pad so that the application blends in with an existing set of applications that your company uses.
Note
You do not need to change the branding bundles in order to change the name of BI launch pad, but you may want to also change related images, which will require changes in the branding bundles.

1. Copy the BIlaunchpad.properties file from

\<INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\warfiles\webapps\BOE\WEB-INF\config\default

to

\<INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\warfiles\webapps\BOE\WEB-INF\config\custom

Note
Do not change files in the default folder. You should always make your changes to copies of the files stored in the custom folder.

2. Modify the following properties:

    app.name=BI launch pad
    app.name.greeting=BusinessObjects
    app.name.short=BI launch pad
    app.url.name=/BI

3. Re-deploy BOE.war to your Java application server.

4.4 Customizing OpenDocument

For OpenDocument, you can customize the logo, backgrounds, and styles of the logon page. Most of these customizations involve changing the CSS rules in the customize.css file. All customizations must be made available in the web\service folder of com.businessobjects.webpath.OpenDocumentBranding to take effect, as follows:

    \web
    \service
    \css
    customize.css
    \images
    \theme
    *.*

Note
Sample customizations are provided in the JAR file. For example, when you open com.businessobjects.webpath.OpenDocumentBranding, there is a web\sample folder which includes a sample CSS file, sample images, and a README file.

The following diagram shows the elements customized in the sample branding bundle, for reference purposes. The numbers in the balloons refer to sections in the bundle's customize.css file.
1. (0.1) customize the background of pages and sub-pages (inside frames)
2. (0.2) input text field
3. (0.3) password field
4. (1.1) authentication fields container
5. (1.2) container of everything
6. (1.3) horizontal rules
7. (1.5) "Log On" button
8. (1.6) banner background pattern
9. (1.7) logo

### 4.5 Customizing the Crystal Reports JavaScript viewer

This section shows how to customize the report viewer included in your SAP BusinessObjects Business Intelligence platform OEM deployment.

The viewer can be customized by adding the following:

- A custom logo
- SAP Crystal Reports JavaScript API event and action listeners
- CSS files
- External JavaScript files or libraries
4.5.1 Customizing the viewer

You can repackage the `template.zip` file with the files you are using to customize the report viewer.

The workflow is as follows:

1. **Extract the contents of the** `template.zip` **file.**
2. **Modify the** `com.businessobjects.webpath.CrystalReports_oem.jar` **file.**
3. **Enable custom viewer behavior, by setting the** `crystal_enable_jsapi` **property to** `true` **in the** SAP BusinessObjectsEnterprise XI 4.0\warfiles\webapps\config\custom \CrystalReports.properties file.**
4. **Recreate the** `template.zip` **file.**

**Note**

It is recommended to make a backup copy of the `template.zip` file before modifying its contents.

Within the `template.zip` file, the following file must be modified:

<table>
<thead>
<tr>
<th><code>template.zip</code></th>
<th>Modify</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP_BusinessObjectsEnterprise XI 4.0\warfiles\webapps\BOE\WEB-INF\eclipse \plugins \com.businessobjects.webpath.CrystalReports_oem.jar</td>
<td>Unzip and modify.</td>
</tr>
</tbody>
</table>

Within the `com.businessobjects.webpath.CrystalReports_oem.jar` file, the following may be modified:

<table>
<thead>
<tr>
<th><code>CrystalReports_oem.jar</code></th>
<th>Modify</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\web</code></td>
<td>Add custom or external JavaScript and CSS files to this folder.</td>
</tr>
</tbody>
</table>
| `\web\CustomListener.js` | Add SAP Crystal Reports JavaScript API event listeners to the `OnViewerInit` and `OnViewerFail` functions in the `CustomListener.js` file.  
For more information, see the *SAP Crystal Reports JavaScript API Guide*. |
| `WEB-INF\classes\JSAPI-properties.json` | Add a relative path to all images, JavaScript files, and CSS files added to the `\web` folder. You can also change the logo that will be displayed by the viewer.  
In the following example, a logo, a JavaScript file, a folder and its JavaScript contents, and a CSS file are added: 

```json
{
  "logo": {
    "img": "images/logo.gif",
    "tooltip": "SAP Crystal Reports",
  },
  "customContent": {
    "folder": {
      "prefix": "custom",
      "files": [
        "CustomListener.js",
      ]
    },
    "css": {
      "prefix": "custom",
      "files": [
        "custom.css"
      ]
    }
  }
}
``` |
<table>
<thead>
<tr>
<th>CrystalReports_oem.jar</th>
<th>Modify</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;url&quot;: &quot;<a href="http://www.businessobjects.com/ipl/default.asp?destination=ViewerLogoLink&amp;product=crystalreports&amp;version=14%2E0">http://www.businessobjects.com/ipl/default.asp?destination=ViewerLogoLink&amp;product=crystalreports&amp;version=14%2E0</a>&quot;, &quot;scripts&quot;: [{ CustomListener.js \CustomFiles*.js }], &quot;styles&quot;: [{ \CustomStyle.css }</td>
</tr>
</tbody>
</table>

**Note**

All files referenced in the JSAPI-properties.json file must be included in the \web folder.
5 Business Intelligence Platform Multitenancy Management Tool

5.1 Introduction

The SAP BusinessObjects Business Intelligence platform Multitenancy Management Tool is a Java-based program that allows SAP OEM partners to standardize and automate the creation of objects and settings for their new customers in a multitenant BI platform deployment.

In a multitenant deployment, multiple tenants are hosted on a single instance of BI platform. Each tenant has a separate view of its own data that cannot be seen by other tenants. There are many examples of tenant deployments, including:

- Tenants that share the same universe and database
- Tenants that use separate universes and databases

This is typically accomplished by manually configuring the BI platform for each tenant. For example, you may:

1. Create separate user groups for each tenant
2. Create separate folders for tenant documents
3. Restrict universe data views for each tenant
4. Apply appropriate security settings to isolate each tenant’s experience

The Multitenancy Management Tool automates these and other steps to ease the process of creating new tenants.
Who should use this information?

This section is intended for content and system administrators who are responsible for configuring, managing, and maintaining a multitenant BI platform installation. Familiarity with the basic concepts and tools used to manage a BI platform installation is required. An understanding of designing universes for reports and analytics may also be required depending on your tenant deployment requirements. However, to assist all levels of administrative experience, this section aims to provide sufficient background and conceptual information to clarify all administrative tasks and features.

For information on setting up the security and server infrastructure of BI platform, see the Business Intelligence Platform Administrator Guide.

For information on managing, scheduling, and distributing BI content in the repository, see the Business Intelligence Platform User Guide.

For information on designing classic universes (.unv files), see the Universe Design Tool User Guide.

Conventions

The following terms are used throughout this section:

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>tenant</td>
<td>A customer of an SAP OEM partner who shares a single instance of the BI platform infrastructure and services with other customers, while their data and user experience are kept separate.</td>
</tr>
<tr>
<td>tenant template</td>
<td>A collection of repository objects, rights, and settings that serve as the structural basis to create new tenants in a multitenant BI platform installation.</td>
</tr>
<tr>
<td>template token</td>
<td>A string used to identify a tenant template object or setting to be replicated by the tool when creating a new tenant.</td>
</tr>
<tr>
<td>tenant configuration file</td>
<td>A java properties file (tenant_template_def.properties) that allows you to set options before running the Multitenancy Management Tool.</td>
</tr>
</tbody>
</table>

5.2 Quick start

This quick start provides a series of tasks to get you up and running with the BI platform Multitenancy Management Tool. The tasks are designed to help you become familiar with the essential steps required to create new tenants based on a template. Links to more detailed instructions and information on the essential concepts are provided where necessary.

To use the tool:
1. The Multitenancy Management Tool must be installed (default installation or selected during a custom installation).

2. You must create a collection of objects and settings in your BI platform installation that serve as a tenant template.

3. You must configure a tenant definition file for each new tenant.

4. You must run the tool to create a tenant. The tool uses the tenant template and settings defined in the tenant definition file to create the tenant.

This quick start first shows you how to create a new tenant template that consists of a single BI platform user group, two public folders, and granted rights on a folder. You are then shown how to setup a tenant definition file and run the tool using these settings to provision a new tenant during onboarding.

5.2.1 Installation prerequisites

To use the multitenancy management tool, you must have the following software installed:

- JRE 1.6
- SAP BusinessObjects Business Intelligence platform 4.0 Feature Pack 3 or later

The multitenancy management tool is installed by default with the BI platform and located in the `\java\apps\` folder:

- **Windows:** `<INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\java\apps\multitenancyManager\jars\`
- **Unix:** `<INSTALLDIR>/sap_bobj/enterprise_xi40/java/apps/multitenancyManager/jars/`

If you performed a custom installation, you can add the tool by modifying your installation and selecting the following feature:

- **Windows:** `Servers Multi-tenancy Manager`
- **Unix:** `Servers MultitenancyManager`

For detailed instructions on how to modify the installed features in your BI platform deployment, see the *Business Intelligence Platform Installation Guide*. 
5.2.2 Create your tenant template

For this quick start example, let’s create the following template objects and settings using the Central Management Console (CMC):

- A user group named "$TemplateToken$"
- A public folder named $TemplateToken$ at the root level
- A public folder structure tenants/$TemplateToken$ _temp at the root level
- Assigned rights to the $TemplateToken$ folder

To get started, start the CMC and log on with appropriate credentials to create the objects. By default, you can start the CMC by going to http://<webservername>:8080/BOE/CMC.

Related Information

Setting up a tenant template [page 80]

5.2.2.1 To create a new user group

1. In the "Users and Groups" management area of the CMC, click Manage > New > New Group. The Create New User Group dialog box appears.
2. Type the group name $TemplateToken$ and a description.
3. Click OK.

The template user group is created.
5.2.2.2 To create a new folder

1. In the Folders management area of the CMC, navigate to the root public folder.
2. Click Manage > New > Folder.
3. Type $TemplateToken$ as the name of your new folder.
4. Click OK.
5. Repeat steps 2 to 5 and create a folder called _tenants with a subfolder called $TemplateToken$_temp inside it.

The new folders appear in the list of folders and objects.
5.2.2.3 To set rights

1. In the Folders management area of the CMC, select the $TemplateToken$ folder.
2. Click Manage > User Security. The User Security dialog box appears and displays the access control list.
3. Click Add Principals. The Add Principals dialog box appears.
4. Move the template user group "$TemplateToken$" from the Available users/groups list to the Selected users/groups list.
5. Click Add and Assign Security.
6. Select the access levels you want to grant the $TemplateToken$ user group. For example, View.
7. Choose whether to enable or disable folder or group inheritance.
8. Repeat steps 1 to 7 and assign rights for the template user group to the $TemplateToken$ _temp folder.

The template user group now has rights assigned to the template folders.
5.2.3 Configure your tenant definition file

Before you configure your tenant definition file, go to the multitenancyManager folder and back up a copy of the original tenant_template_def.properties file. The multitenancyManager folder is located at:

- Windows: `<INSTALLDIR>`\SAP BusinessObjects Enterprise XI 4.0\java\apps\multitenancyManager\jars\`
- Unix: `<INSTALLDIR>`/sap_bobj/enterprise_xi40/java/apps/multitenancyManager/jars/

In this task, you set options in your tenant definition file (tenant_template_def.properties) that will allow the tool to log onto your Central Management Server (CMS), identify the template objects by the template token string, and create new copies of these objects with a specific tenant name.

1. Open `tenant_template_def.properties` for editing.
2. Change the mandatory `cms`, `auth`, `user`, and `password` options to the authentication details used to log onto the CMS.

   ```
cms=mycmsdomainname:port
auth=secEnterprise
user=Administrator
pwd=mypassword
```

Potential values for the `auth` option include: `secEnterprise`, `secLDAP`, `secWinAD`, or `secSAPR3`.

3. Change the mandatory `tenantName` option to the name of the new tenant to be created.

   ```
tenANTName=abc_customer
```

4. Change the mandatory `templateToken` option to the string used to identify your template folders and user group.

   In this example, use the string "$TemplateToken$" as you did in the previous quick start task “Create your tenant template”.

   ```
templateToken=$TemplateToken$
```
5. Change the mandatory `templateContentFolder` option to specify the top-level template folders.

   In this quick start, you’ve created two template folders that have the template token in their name. Separate any multiple values with a semicolon and include the full path under the public root folder.

   ```
   templateContentFolder=${TemplateToken$};_tenants/${TemplateToken$}_temp
   ```

6. Save and close `tenant_template_def.properties`.

Related Information

*Configuring the tenant definition file* [page 92]

### 5.2.4 Run the tool

To run the tool, navigate to the folder where the Multitenancy Management Tool is located, open a command prompt, and run `multitenancymanager.jar` with your tenant definition file passed into the `-configFile` option:

```
java -jar multitenancymanager.jar -configFile tenant_template_def.properties
```

**Note**

The multitenancy management tool is installed by default with the BI platform and located in the `\java\apps\` folder:

- **Windows:** `<INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\java\apps\multitenancyManager\jars\`
- **Unix:** `<INSTALLDIR>/sap_bobj/enterprise_xi40/java/apps/multitenancyManager/jars/`

After the program completes successfully, log on to the Central Management Console (CMC) and view the newly created public folders, user group, and security settings for the "abc_customer" tenant.
By default, a log file named `multitenancymanager<yyyy><mm><dd><time>.csv` is created in the following folder:

- **Windows**: `<INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\logging\`
- **Unix**: `<INSTALLDIR>/sap_bobj/logging/`

**Next steps**

This quick start showed you how to setup a basic tenant template and create a new tenant with a few simple objects and security settings. The tool allows you to work with more complex scenarios however, and replicate
many other objects, including universes and report database connection information. Refer to the remaining
sections of this guide for information on setting up all types of template objects, and details on the different
optional configurations you can make in the tenant definition file.

Related Information

Running the tenant provisioning tool [page 98]
Troubleshooting [page 99]

5.3 Setting up a tenant template

A tenant template is a collection of repository objects, rights, and settings that serve as the structural basis to
create new tenants in a multi-tenant BI platform installation. For example, tenants in your deployment may have
common traits, such as structurally identical user groups and public folder structures.

Rather than re-creating these similar objects and settings every time a new tenant has to be created, you can
create them once in the BI platform as a template, and then run the multitenancy management tool to create new
tenant instances based on that template.

The different parts of a tenant template are identified by the use of a template token. The template token is a
string that identifies a particular object or setting to be replicated by the tool. For example:

1. Choose the string "$TemplateToken$" as your token.
2. Create a user group via the Central Management Console (CMC) with "$TemplateToken$" in its name, such
   as "$TemplateToken$", or "$TemplateToken$_usergroup".

After running the tool, a new user group is created for that tenant with the identical structure and settings of the
template user group. The token string, "$TemplateToken$", is replaced by the actual name defined in the tenant
configuration file.

The following sections provide details on how to identify the different components of your tenant template using
your template token, and how the tool functions with respect to each component.

i Note

For detailed instructions on how to create new objects such as user groups and folders in the CMC, see the
Business Intelligence Platform Administrator Guide.

5.3.1 Folders

To include public folders in your template:

1. In the CMC, create a folder with the template token in its name.
2. Specify this folder using the templateContentFolder option in the tenant configuration file with a
   semicolon-separated list.
You need to specify only the top-level folders in your template when running the tool; all subfolders and objects contained in the folders will also be copied automatically. In the example below, you would specify the $TemplateToken$ and $TemplateToken$_temp folders in your configuration file. The Documents folder and the Crystal report will be copied automatically as children of the top-level $TemplateToken$ folder.

<table>
<thead>
<tr>
<th>Public Folders</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TemplateToken$</td>
</tr>
<tr>
<td>Documents</td>
</tr>
<tr>
<td>Tenant_report.rpt</td>
</tr>
<tr>
<td>Reports Samples</td>
</tr>
<tr>
<td>$TemplateToken$_temp</td>
</tr>
</tbody>
</table>

### 5.3.2 User groups

To include a user group in your template, in the CMC, create new groups with the template token in their name—for example, "$TemplateToken$" or "$TemplateToken$_usergroup".

It is recommended that template user groups not contain user accounts since they will be made members of all new tenant user groups and create potential permission conflicts. Tenants should not share users. Create empty user groups with titles and descriptions to use as templates.

*Note*

User groups are automatically searched for by the tool and cannot be excluded from the tenant template.

### 5.3.3 Event folders

To include event folders in your template:

1. In the CMC, create an event folder with the template token in its name.
2. Set `optionIncludeEvents` to `true` in the tenant configuration file.
3. Specify the event folder using the `templateEventFolder` option with a semicolon-separated list.

You need to specify only the top-level event folders in your template when running the tool; all subfolders and events contained in the folders will also be copied.

### 5.3.4 Categories

To include categories in your template:

1. In the CMC, create categories with the template token in their names.
2. Set `optionIncludeCategories` to `true` in the tenant configuration file.
3. Specify the categories with the `templateCategoryFolder` option with a semicolon-separated list.
5.3.5 Profiles

To include profiles in your template:

1. In the CMC, create a profile with the template token in its name.
2. Set `optionIncludeProfiles` to `true` in the tenant configuration file.

All users, user groups, profile values, and global profile targets are copied to the new tenant profile.

5.3.6 Security settings

Access levels (groups of rights) can be part of your tenant template. To include access levels in your template:

1. In the CMC, create an access level with the template token in its name.
2. Set `optionIncludeLevels` to `true` in the tenant configuration file.

All included rights are copied to the new tenant access level.

In addition, any rights or access levels granted on template objects to principals are carried over to the new tenant object. If the target tenant object already exists before running the tool (for example, when running the tool multiple times for the same tenant), use the `optionImportSecMode` option to specify the exact manner in which existing principal rights on the tenant object are handled:

- `optionImportSecMode=0` (Merge mode): Merge the template principals and rights with the existing tenant object. Preserves the original rights granted to any principals on the existing tenant object.

  **Note**

  In the event of a conflict, the template setting takes precedence during the merge. For example, if the template object specifically grants a right to a principal while the existing tenant object specifically denies that same right.

- `optionImportSecMode=1` (Principal-level overwrite): Replace the rights of identical principals with the template rights. Add unique principals and rights from the template and preserve unique principals and rights on the tenant object.

- `optionImportSecMode=2` (Object-level overwrite): Remove all existing principals and rights on the existing tenant object and replace with the principals and rights granted on the template object.

**Example**

To illustrate, consider an example. Assume you have a template folder (`$TemplateToken$`) and an existing tenant folder (`ABC`) with the access levels granted to principals as follows:

<table>
<thead>
<tr>
<th>Principal</th>
<th>Access Level</th>
<th>Principal</th>
<th>Access Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>View</td>
<td>User 1</td>
<td>Full Control</td>
</tr>
<tr>
<td>User 2</td>
<td>View</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

© 2014 SAP AG or an SAP affiliate company. All rights reserved. SAP BusinessObjects OEM Customization Guide Business Intelligence Platform Multitenancy Management Tool
After running the tool, the resulting permissions granted on the tenant folder ABC based on the `optionImportSecMode` setting are as follows:

<table>
<thead>
<tr>
<th>Tenant Folder = “ABC” optionImportSecMode=0</th>
<th>Tenant Folder = “ABC” optionImportSecMode=1</th>
<th>Tenant Folder = “ABC” optionImportSecMode=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Principal</td>
<td>Access Level</td>
<td>Principal</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>User 1</td>
<td>Full Control; View</td>
<td>User 1</td>
</tr>
<tr>
<td>User 2</td>
<td>View</td>
<td>User 2</td>
</tr>
<tr>
<td>User 3</td>
<td>View</td>
<td>User 3</td>
</tr>
</tbody>
</table>

The behavior also applies to individually-added rights, not only access levels.

### 5.3.7 SAP Crystal Reports 2011

The multitenancy management tool allows you to map direct-to-data connection information for any SAP Crystal Reports 2011 template reports to new tenant connection settings.

To map tenant data source connection information for Crystal reports:

1. Create and upload a set of reports to a template folder in the BI platform. Template folders are specified in the tenant configuration file using the `templateContentFolder` option.
2. Specify the template’s DSN information using the `crystalreport.templatedb1` option in the tenant configuration file. The format of the value is `<database server>;<database name>;<data source type>;<username>;<password>`. For example:
   ```plaintext
crystalreport.templatedb1=MyTemplateDSN;MyTemplateDatabase;odbc;administrator;password
   ```

   **Tip**

   To find the correct values for `<database server>;<database name>`, right-click on the report in the CMC and select `Database Configuration`. The `server` and `database` fields can be copied to the tenant configuration file.
3. Specify the tenant’s new DSN information using the `crystalreport.tenantdb1` option in the tenant configuration file.
The format of the value is <database server>;<database name>;<data source type>;<username>;<password>. For example:

crystalreport.tenantdb1=MyTenantDSN;MyTenantDatabase;odbc;tenantname;tenantpwd

After running the tool, the newly copied Crystal reports in the tenant’s own folders are mapped to their own data source connection. You can specify additional template/tenant pairs of settings by adding crystalreport.templatedb<n> and crystalreport.tenantdb<n> option pairings to the tenant configuration file. For example, crystalreport.templatedb2 and crystalreport.tenantdb2.

Consider the following before-and-after snapshots of the CMS, assuming a tenant named “ABC” is created:

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Folders</td>
<td>Public Folders</td>
</tr>
<tr>
<td>$TemplateToken$_folder</td>
<td>$TemplateToken$_folder</td>
</tr>
<tr>
<td>$TemplateToken$_report1.rpt</td>
<td>$TemplateToken$_report1.rpt</td>
</tr>
<tr>
<td>$TemplateToken$_report1.rpt</td>
<td>$TemplateToken$_report1.rpt</td>
</tr>
<tr>
<td>$TemplateToken$_report2.rpt</td>
<td>$TemplateToken$_report2.rpt</td>
</tr>
<tr>
<td>$TemplateToken$_report2.rpt</td>
<td>$TemplateToken$_report2.rpt</td>
</tr>
<tr>
<td>● $TemplateToken$_report1 and $TemplateToken$_report1 both use crystalreport.templatedb1 DSN settings.</td>
<td>● ABC_report1.rpt and ABC_report1.rpt both use crystalreport.tenantdb1 DSN settings.</td>
</tr>
<tr>
<td>● $TemplateToken$_report2 uses crystalreport.templatedb2 DSN settings.</td>
<td>● ABC_report2.rpt uses crystalreport.tenantdb2 DSN settings.</td>
</tr>
</tbody>
</table>

**Mapping table prefixes**

The template database may use different table prefixes than the new tenant database you are repointing the reports to. If this is the case, use the crystalreport.templatetableprefixes<n> and crystalreport.tenanttableprefixes<n> options to specify how tables should be mapped. Multiple prefixes are separated by semicolons, and the list must end in a semicolon so that empty strings (“”) can be identified. Consider the following example:

templatetableprefixes1=templateprefixa;templateprefixb;;
tenanttableprefixes1=;tenantprefixb;tenantprefixc;

This results in the following mapping:

<table>
<thead>
<tr>
<th>Template prefix</th>
<th>Maps to tenant prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>templateprefixa</td>
<td>(empty string)</td>
</tr>
<tr>
<td>templateprefixb</td>
<td>tenantprefixb</td>
</tr>
<tr>
<td>(empty string)</td>
<td>tenantprefixc</td>
</tr>
</tbody>
</table>

**Tip**

To find the name of a table prefix in a template report, right-click on the report in the CMC and select *Database Configuration*. The *Table Prefix* field provides the name.
Supported data source types

The multitenancy management tool supports changing the following direct-to-data connection types. Use these values in the `crystalreport.templatedb<n>` and `crystalreport.tenantdb<n>` options:

<table>
<thead>
<tr>
<th>Data source type</th>
<th>odBC</th>
<th>oracle</th>
<th>db2</th>
<th>sybase</th>
<th>informix</th>
<th>crdb_xml</th>
</tr>
</thead>
</table>

Table 11: SAP Crystal Reports 2011 data source type

For information on the specific versions of these databases supported by SAP Crystal Reports 2011, see the *Product Availability Matrix (Supported Platforms/PAR)* in the SAP BusinessObjects section of the SAP Support Portal at [https://service.sap.com/bosap-support](https://service.sap.com/bosap-support).

Related Information

*Folders* [page 80]

5.3.8 SAP Crystal Reports for Enterprise

The multitenancy management tool allows you to map any SAP Crystal Reports for Enterprise report that configures a direct-to-data connection through a `CCIS.DataConnection` connection object created with the SAP BusinessObjects Information Design Tool. These connection objects are hosted on the SAP BusinessObjects Business Intelligence platform and can be viewed in the *Connections* folder using the Central Management Console (CMC). Template reports are mapped to new tenant connection settings.

To map tenant data source connection information for reports that use a connection object, configure the following option in the tenant configuration file:

- `ccis.dataconnection.dbcredentials<n>=<template_CCIS.CONN_CUID>;<datasourcename>;<database name>;<username>;<password>`

  - `template_CCIS.CONN_CUID` is the CUID of a template connection object.

**Tip**

To find the connection object associated with the template report, right-click on the report in the CMC and select `Tools > Check Relationships`. A *Relational Connection* object is listed in the results.
If you have more than one template connection object, add additional `ccis.dataconnection.dbcredentials` options. For example:

- `ccis.dataconnection.dbcredentials2`, `ccis.dataconnection.dbcredentials3`, ...
- `ccis.dataconnection.dbcredentialsn`

After running the tool, the newly copied Crystal reports in the tenant's own folders are mapped to their own data source connection.

Consider the following before-and-after snapshots of the CMS, assuming a tenant named "ABC" is created, and the tenant configuration file is configured as follows:

```
ccis.dataconnection.dbcredentials1=ZZZZZZZZZZ
```

### Mapping table prefixes

The template database may use different table prefixes than the new tenant database you are repointing the reports to. If this is the case, configure the following options:

- `crystalreport.ccis.dataconnection.templatedb<n>`

The CUID of the template connection object
• crystalreport.ccis.dataconnection.templatetableprefixes
  The table prefixes of the template data source
• crystalreport.ccis.dataconnection.tenanttableprefixes
  The table prefixes of the tenant data source to map to

Multiple prefixes are separated by semicolons, and the list must end in a semicolon so that empty strings (""") can be identified. Consider the following example:

```
crystalreport.ccis.dataconnection.templatedb1=ZZZZZZZZZZ
crystalreport.ccis.dataconnection.templatetableprefixes1=templateprefixa;templateprefixb;;
crystalreport.ccis.dataconnection.tenanttableprefixes1=;tenantprefixb;tenantprefixc;
```

This results in the following mapping:

<table>
<thead>
<tr>
<th>Template prefix</th>
<th>Maps to tenant prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>templateprefixa</td>
<td>(empty string)</td>
</tr>
<tr>
<td>templateprefixb</td>
<td>tenantprefixb</td>
</tr>
<tr>
<td>(empty string)</td>
<td>tenantprefixc</td>
</tr>
</tbody>
</table>

➤ Tip

To find the name of a table prefix in a template report, view the details of the direct-to-data connection object using the SAP BusinessObjects Information Design Tool.

Related Information

Universes [page 87]
Folders [page 80]

5.3.9 Universes and connections

The multitenancy management tool allows you to manage the way your tenant’s universes and connection information are handled by the system. This topic outlines the different scenarios to consider in this release.

Note

This release supports only classic universes (.unv files); it does not support .unx files.
Unshared universes and connections

In this scenario, reports and analytics connect to different universes (with separate underlying connection objects). This allows you to automate the distribution of template reports while isolating the accessible data for each tenant. To provide different, unshared universes and connection objects for each tenant:

1. Create a connection with the template token in its name. Save to a template connection folder that uses the template token string in its name.
2. Create a universe with the template token in its name and that uses the template connection in step 1. Save to a universe folder that uses the template token string in its name.
3. Create and upload a set of reports or analytics that use the universe in step 2 to a template folder in the BI platform.
4. Set the following options in the tenant configuration file:

<table>
<thead>
<tr>
<th>Option Required</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>optionIncludeUniverses</td>
<td>Set to true so that template universes are copied to each tenant.</td>
</tr>
<tr>
<td>optionIncludeConnections</td>
<td>Set to true so that template connections are copied to each tenant.</td>
</tr>
<tr>
<td>templateUniverseFolder</td>
<td>Set to the folder path where your universe templates are located—for example, $TemplateToken$_unshared.</td>
</tr>
<tr>
<td></td>
<td>The path is relative to the root Universes folder.</td>
</tr>
<tr>
<td>templateConnectionFolder</td>
<td>Set to the folder path where your universe templates are located—for example, $TemplateToken$_unshared.</td>
</tr>
<tr>
<td></td>
<td>The path is relative to the root Connections folder.</td>
</tr>
<tr>
<td>ccis.dataconnection.dbcredentials1</td>
<td>Set to the details of the template connection object you want to replicate for each tenant—for example, <code>&lt;CUID&gt;</code>;&lt;data source name&gt;;&lt;database name&gt;;&lt;user name&gt;;&lt;password&gt;.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;CUID&gt;</code> is the CUID of a template connection object. Specify the DSN information (&lt;data source name&gt;, &lt;database name&gt;) for the new tenant connection that will be created.</td>
</tr>
<tr>
<td></td>
<td>If you have more than one template connection object, add additional ccis.dataconnection.dbcredentials options—for example, ccis.dataconnection.dbcredentials2, ccis.dataconnection.dbcredentials3,...ccis.dataconnection.dbcredentialsn.</td>
</tr>
</tbody>
</table>
### Note

These options are in addition to any other options, such as folder templates, that you must set to run the tool.

After running the tool, the new tenant has their own instances of the reports/analytics, universes, and connection objects. To illustrate, consider the following before-and-after snapshots of the CMS, assuming the tool is run twice to create tenants named "ABC" and "DEF". In this example, the CUID in the `ccis.dataconnection.dbcredentials1` option is set to `ZZZZZZZZZZ`:

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folders and reports</td>
<td>$TemplateToken$_unshared</td>
<td>$TemplateToken$_unshared</td>
</tr>
<tr>
<td></td>
<td>$TemplateToken$_sales.wid</td>
<td>$TemplateToken$_sales.wid</td>
</tr>
<tr>
<td>$TemplateToken$_sales.wid:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CUID=XXXXXXXXXX</td>
<td>• CUID=XXXXXXXXXX</td>
</tr>
<tr>
<td></td>
<td>• Universe=$TemplateToken$_ODBCUniverse.unv</td>
<td>• Universe=$TemplateToken$_ODBCUniverse.unv</td>
</tr>
<tr>
<td>Universes</td>
<td>$TemplateToken$_unshared</td>
<td>$TemplateToken$_unshared</td>
</tr>
<tr>
<td></td>
<td>$TemplateToken$_ODBCUniverse.unv</td>
<td>$TemplateToken$_ODBCUniverse.unv</td>
</tr>
<tr>
<td>$TemplateToken$_ODBCUniverse.unv:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CUID=YYYYYYYYYYY</td>
<td>• CUID=YYYYYYYYYY</td>
</tr>
<tr>
<td></td>
<td>• Connection=$TemplateToken$_ODBCConnection</td>
<td>• Connection=$TemplateToken$_ODBCConnection</td>
</tr>
</tbody>
</table>

© 2014 SAP AG or an SAP affiliate company. All rights reserved.
### Shared universes and connections

In this scenario, reports and analytics connect to the same universes and connection objects. To specify a shared environment:

1. Create a connection (must not have the template token in its name). Save to a connection folder that is shared to all tenants.
2. Create a universe that uses the template connection in step 1 (must not have the template token in its name). Save to a universe folder that is shared to all tenants.
3. Create and upload a set of reports or analytics that use the universe in step 2 to a template folder in the BI platform.
4. Set the following options in the tenant configuration file:

<table>
<thead>
<tr>
<th>Option Required</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>optionUseSharedUniverses</td>
<td>Set to true.</td>
</tr>
<tr>
<td>optionUseSharedConnections</td>
<td>Set to true.</td>
</tr>
<tr>
<td>sharedUniverseFolder</td>
<td>Set to the folder path where your shared universe is located—for example, SharedUniverses. The path is relative to the root Universes folder.</td>
</tr>
</tbody>
</table>
### Option Required

<table>
<thead>
<tr>
<th>sharedConnectionFolder</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set the folder path where your universe templates are located—for example, <code>SharedConnections</code>. The path is relative to the root <code>Connections</code> folder.</td>
</tr>
</tbody>
</table>

### Note

These options are in addition to any other options, such as folder templates, that you must set to run the tool.

After running the tool, the new tenant has their own instances of the reports/analytics that use a common universe and connection object. To illustrate, consider the following before-and-after snapshots of the CMS, assuming the tool is run twice to create tenants named "ABC" and "DEF":

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
</table>
| Folders and reports | Public Folders
  $TemplateToken$ _unshared
  $TemplateToken$ _sales.wid
  $TemplateToken$ _sales.wid:
    - CUID=XXXXXXXXXX
    - Universe=SharedODBCUniverse.unv | Public Folders
  $TemplateToken$ _unshared
  $TemplateToken$ _sales.wid
  ABC_unshared
  ABC_sales.wid
  DEF_unshared
  DEF_sales.wid:
    - CUID=XXXXXXXXXX
    - Universe=SharedODBCUniverse.unv |
| Universes         | Unicverses
  SharedUniverses
  SharedODBCUniverse.unv         | Unicverses
  SharedUniverses
  SharedODBCUniverse.unv         |
| Connections       | Connections
  SharedConnections
  SharedODBCConnection          | Connections
  SharedConnections
  SharedODBCConnection          |
Restricting data access for shared universes

When sharing universes between tenants, you may want to restrict the data within the universe that each tenant can view. This is accomplished by assigning rights on objects, rows, queries, and other aspects of a universe to certain user groups. This is done during the design-time of the universe, using the Manage Access Restrictions dialog box of the SAP BusinessObjects universe design tool. When the universe is accessed by a member of that user group, the query generated at run-time will return only the data that the user group has access to.

The multitenancy management tool helps you automate this task. Instead of setting up the access restriction for a specific user group, set it for the template user group (containing the template token string). Ensure that the name of the restriction you create also contains the template token string. After running the tool, the access restriction will be replaced with the proper tenant name, specified by the tenantName option of the tenant configuration file.

Note

You must ensure that the user group is also a template group (ex: $TemplateToken$) so that conflicting restrictions are not applied to the same tenant user group. You cannot apply more than one restriction to the same user group. For detailed instructions on how to set access restrictions in classic universes (.unv files), see the Universe Design Tool User Guide.

5.4 Configuring the tenant configuration file

The following section outlines the options you can set in the tenant configuration file (tenant_template_def.properties).

Table 12: Mandatory tenant provisioning options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>tenantName</td>
<td>The name of the tenant customer to be added. This string replaces the templateToken string wherever found. For example, if tenantName=abc, templateToken=$TemplateToken$ and you’ve created a user group template called “$TemplateToken$$_usergroup”, the tool will create a new user group called “abc$_usergroup”.</td>
<td>Yes</td>
</tr>
<tr>
<td>templateToken</td>
<td>The string used to identify a tenant template object or setting to be replicated by the tool when creating a new tenant. The templateToken string is replaced by the tenantName string when the tool runs.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Option | Description | Required?
--- | --- | ---
**Note** | Choose a unique string that is not commonly found in existing object names within your BI platform installation. | 

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>cms</td>
<td>The name or IP and port of the Central Management Server (CMS) to connect to. If no port is specified, 6400 is used by default.</td>
<td>Yes</td>
</tr>
<tr>
<td>auth</td>
<td>The authentication type to use for logging on. Valid values include secEnterprise, secLDAP, secWinAD, and secSAPR3.</td>
<td>Yes</td>
</tr>
<tr>
<td>user</td>
<td>The user name of the account used to log on to the CMS and run the tool.</td>
<td>Yes</td>
</tr>
<tr>
<td>pwd</td>
<td>The password of the user.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Table 13: Optional tenant provisioning options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Required?</th>
</tr>
</thead>
</table>
| statusLog | The folder where the log file is created. For troubleshooting purposes, the log file contains a trace of the program and any errors that occurred. For example: `statusLog="C:\TenantLogs\"` The log file is automatically named based on the tenantName value and the date and time stamp when the program is run—for example, multitenancymanager<yyyy><mm><dd><time>.csv If this option is not specified, the default location is:  
- Windows: `<INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\logging\`  
- Unix: `<INSTALLDIR>/sap_bobj/logging/` | No |
| abortANDRollback | Set to `true` to rollback all changes if one of the new objects fails to commit successfully to the CMS. If this option is not specified, the default setting is `true`. **Note** If some objects fail to commit to the CMS, and `abortANDRollback=false` your tenant is only partially created with the objects that were successfully committed. Use the log file to troubleshoot and correct any errors, then rerun the tool for that tenant. | No |
| optionImportSecMode | Indicates how to handle the importation of security settings (rights) on objects created from the template objects. Valid values include: | No |
Table 14: Object template options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>templateContentFolder</td>
<td>A semicolon-separated list of folder paths that defines all top-level template folders in the repository. Template folders contain the template token string in their name.</td>
<td>No</td>
</tr>
<tr>
<td>optionIncludeUniverses</td>
<td>Indicates whether the program will search for universes using the template token string and create a new tenant version. If this option is not specified, the default setting is true.</td>
<td>No</td>
</tr>
<tr>
<td>optionIncludeConnections</td>
<td>Indicates whether the program will search for universe connections using the template token string and create a new tenant version. If this option is not specified, the default setting is true.</td>
<td>No</td>
</tr>
<tr>
<td>optionIncludeCategories</td>
<td>Indicates whether the program will search for BI platform categories using the template token string and create a new tenant version. If this option is not specified, the default setting is true.</td>
<td>No</td>
</tr>
<tr>
<td>optionIncludeProfiles</td>
<td>Indicates whether the program will search for BI platform profiles using the template token string and create a new tenant version. If this option is not specified, the default setting is true.</td>
<td>No</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Required?</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>optionIncludeEvents</td>
<td>Indicates whether the program will search for BI platform event folders using the template token string and create a new tenant version. If this option is not specified, the default setting is true.</td>
<td>No</td>
</tr>
<tr>
<td>optionIncludeAccessLevels</td>
<td>Indicates whether the program will search for BI platform access levels (groups of rights) using the template token string and create a new tenant version. If this option is not specified, the default setting is true.</td>
<td>No</td>
</tr>
<tr>
<td>optionUseSharedUniverses</td>
<td>Indicates whether template reports and analytics use shared universes. Use in conjunction with sharedUniversesFolder. If this option is not specified, the default setting is true.</td>
<td>No</td>
</tr>
<tr>
<td>optionUseSharedConnections</td>
<td>Indicates whether template reports and analytics use shared connections. Use in conjunction with sharedConnectionFolder. If this option is not specified, the default setting is true.</td>
<td>No</td>
</tr>
<tr>
<td>templateUniverseFolder</td>
<td>A semicolon-separated list of universe folder paths that defines all top-level template universe folders in the repository. Template universe folders contain the template token string in their name.</td>
<td>No</td>
</tr>
<tr>
<td>templateConnectionFolder</td>
<td>A semicolon-separated list of connection folder paths that defines all top-level template universe folders in the repository. Template connection folders contain the template token string in their name.</td>
<td>No</td>
</tr>
<tr>
<td>templateCategoryFolder</td>
<td>A semicolon-separated list of BI platform categories. Template categories contain the template token string in their name.</td>
<td>No</td>
</tr>
<tr>
<td>templateEventFolder</td>
<td>A semicolon-separated list of event folders. Template event folders contain the template token string in their name.</td>
<td>No</td>
</tr>
<tr>
<td>sharedUniversesFolder</td>
<td>A semicolon-separated list of universe folder paths where shared universes are located.</td>
<td>No</td>
</tr>
<tr>
<td>sharedConnectionFolder</td>
<td>A semicolon-separated list of connection folder paths where shared connections are located.</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 15: Data source options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>crystalreport.templatedb&lt;n&gt;</td>
<td>SAP Crystal Reports 2011 only. The template report’s DSN information. The format of the value is <code>&lt;database server&gt;;&lt;database name&gt;;&lt;data source type&gt;;&lt;network layer&gt;;&lt;user name&gt;;&lt;password&gt;</code></td>
<td>No</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Required?</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>crystalreport.tenantdb&lt;n&gt;</td>
<td>SAP Crystal Reports 2011 only. The tenant's new DSN information. The format</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>of the value is `&lt;database server&gt;;&lt;database name&gt;;&lt;data source type&gt;;&lt;net-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>work layer&gt;;&lt;user name&gt;;&lt;password&gt;`.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can specify additional templates and their new tenant settings by</td>
<td></td>
</tr>
<tr>
<td></td>
<td>adding crystalreport.templatedb&lt;n&gt; and crystalreport.tenantdb&lt;n&gt; option</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pairings to the tenant configuration file—for example,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>crystalreport.templatedb2 and crystalreport.tenantdb2.</td>
<td></td>
</tr>
<tr>
<td>crystalreport.template-</td>
<td>SAP Crystal Reports 2011 only. A semicolon-separated list of table prefixes</td>
<td>No</td>
</tr>
<tr>
<td>tableprefixes&lt;n&gt;</td>
<td>in the template database. Maps to prefixes listed in the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>crystalreport.tenanttableprefixes&lt;n&gt; option.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can specify additional templates and their new tenant settings by</td>
<td></td>
</tr>
<tr>
<td></td>
<td>adding crystalreport.templatetableprefixes&lt;n&gt; and crystalreport.tenant-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tableprefixes&lt;n&gt; option pairings to the tenant configuration file—for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>example, crystalreport.templatetableprefixes2 and crystalreport.tenant-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tableprefixes2.</td>
<td></td>
</tr>
<tr>
<td>crystalreport.tenant-</td>
<td>SAP Crystal Reports 2011 only. A semicolon-separated list of table prefixes</td>
<td>No</td>
</tr>
<tr>
<td>tableprefixes&lt;n&gt;</td>
<td>in the target tenant database. Maps to prefixes listed in the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>crystalreport.templatetableprefixes&lt;n&gt; option.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can specify additional templates and their new tenant settings by</td>
<td></td>
</tr>
<tr>
<td></td>
<td>adding crystalreport.templatetableprefixes&lt;n&gt; and crystalreport.tenant-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tableprefixes&lt;n&gt; option pairings to the tenant configuration file—for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>example, crystalreport.templatetableprefixes2 and crystalreport.tenant-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tableprefixes2.</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Required?</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>crystalreport.ccis.dataconnection.templatedb&lt;n&gt;</td>
<td>SAP Crystal Reports for Enterprise only. The DSN details of the template connection object you want to replicate for each tenant—for example, <code>&lt;CUID&gt;; &lt;data source name&gt;; &lt;database name&gt;; &lt;user name&gt;; &lt;password&gt;</code>. <code>&lt;CUID&gt;</code> is the CUID of a template connection object. If you have more than one template connection object, add additional <code>crystalreport.ccis.dataconnection.templatedb</code> options—for example, <code>crystalreport.ccis.dataconnection.templatedb2</code>, <code>crystalreport.ccis.dataconnection.templatedb3</code>, ... <code>crystalreport.ccis.dataconnection.templatedb&lt;n&gt;</code>.</td>
<td>No</td>
</tr>
<tr>
<td>crystalreport.ccis.dataconnection.templatetableprefixes&lt;n&gt;</td>
<td>SAP Crystal Reports for Enterprise only. A semicolon-separated list of table prefixes in the template database. Maps to prefixes listed in the <code>crystalreport.ccis.dataconnection.templatetableprefixes&lt;n&gt;</code> option. You can specify additional templates and their new tenant settings by adding <code>crystalreport.ccis.dataconnection.templatetableprefixes&lt;n&gt;</code> and <code>crystalreport.ccis.dataconnection.templatetableprefixes&lt;n&gt;</code> option pairings to the tenant configuration file—for example, <code>crystalreport.ccis.dataconnection.templatetableprefixes2</code> and <code>crystalreport.ccis.dataconnection.templatetableprefixes2</code>.</td>
<td>No</td>
</tr>
<tr>
<td>crystalreport.ccis.dataconnection.tenanttableprefixes&lt;n&gt;</td>
<td>SAP Crystal Reports for Enterprise only. A semicolon-separated list of table prefixes in the target tenant database. Maps to prefixes listed in the <code>crystalreport.ccis.dataconnection.tenanttableprefixes&lt;n&gt;</code> option. You can specify additional templates and their new tenant settings by adding <code>crystalreport.ccis.dataconnection.tenanttableprefixes&lt;n&gt;</code> and <code>crystalreport.ccis.dataconnection.tenanttableprefixes&lt;n&gt;</code> option pairings to the tenant configuration file—for example, <code>crystalreport.ccis.dataconnection.tenanttableprefixes2</code> and <code>crystalreport.ccis.dataconnection.tenanttableprefixes2</code>.</td>
<td>No</td>
</tr>
<tr>
<td>ccis.dataconnection.dbcredentials&lt;n&gt;</td>
<td>The DSN details of the template connection object you want to replicate for each tenant—for example, <code>&lt;CUID&gt;; &lt;data source name&gt;; &lt;database name&gt;; &lt;user name&gt;; &lt;password&gt;</code>. <code>&lt;CUID&gt;</code> is the CUID of a template connection object. Specify the DSN information (<code>&lt;data source name&gt;</code>, <code>&lt;database name&gt;</code>) for the new tenant connection that will be created. For JDBC and OLEDB connections, the <code>&lt;data source name&gt;</code>, <code>&lt;database name&gt;</code> settings refer to:</td>
<td>No</td>
</tr>
</tbody>
</table>
5.5 Running the multitenancy management tool

To run the tool, navigate to the folder where the multitenancy management tool is located, open a command prompt, and run `multitenancymanager.jar` with your tenant definition file passed to the `configFile` option:

```
java -jar multitenancymanager.jar -configFile tenant_template_def.properties
```

### Note

The multitenancy management tool is installed by default with the BI platform and located in the `\java\apps\` folder:

- **Windows:** `<INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\java\apps\multitenancyManager\jars\`
- **Unix:** `<INSTALLDIR>/sap_bobj/enterprise_xi40/java/apps/multitenancyManager/jars/`

### Caution

The multitenancy management tool creates and commits new objects, such as user groups, folders, documents, and universes, to your CMS repository. It is recommended to keep the `abortANDRollback` option set to `true` (this is the default). The tool will then rollback all changes if one of the new objects fails to commit successfully to the CMS.

### Passing command-line options

You can pass in options directly into the command line. Any options set in the command line will overwrite the setting specified in the tenant configuration file. For example, you could override the tenant’s name with the `tenantName` option as follows:

```
java -jar multitenancymanager.jar -configFile tenant_template_def.properties -tenantName=xyz_customer
```
This allows you to use a single configuration file to process a batch job of multiple tenants, differing only by the name passed into the tenantName option. All options in the configuration file can be overridden at the command line.

5.6 Troubleshooting

Consider the following best practices when troubleshooting errors.

Review the tool log file (.csv)

Program messages displayed on the command line as the multitenancy management tool runs are saved to a log file that you can review after the program completes, with additional details. By default, a log file named multitenancymanager<yyyy><mm><dd><time>.csv is created in the following folder:

- **Windows:** <INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\logging\n- **Unix:** <INSTALLDIR>/sap_bobj/logging/

The location of this log file can be changed using the statusLog option in the tenant configuration file.

Enable and review the trace log file (.glf)

To help diagnose problems, it is recommended to enable tracing. System-level messages generated by BI platform servers and applications can be traced and written to log files. These log files contain more detailed information for diagnosing problems than the command line output or the .csv log file. You can configure tracing for the multitenancy management tool using the BO_trace.ini configuration file and determine the type and verbosity of the information traced and sent to the log file. See also “To configure tracing for the multitenancy management tool”.

Trace messages are collected in log files saved under the generic log file (.glf) extension. The .glf file name is formatted as a combination of shorthand identifier, and number reference—for example, multitenancymanager_trace.000001.glf. A new trace log file is created once the log file approaches a pre-configured size. Traces for the multitenancy management tool are created in the following folder:

- **Windows:** <INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\logging\n- **Unix:** <INSTALLDIR>/sap_bobj/enterprise_xi40/logging/

Determine where the error originated

Errors thrown specifically by the multitenancy management tool are often caused by incorrect settings in the tenant configuration file (tenant_template_def.properties) or by problems local to the machine where
multitenancymanager.jar is run. For common errors and possible actions to take, see also “Multitenancy management tool errors”.

Errors may also originate from the BI platform servers or security framework, and are accompanied by an error code (3-letter string followed by a 5-digit number). Common error codes include FWB and FWM. For example:

Enterprise authentication could not log you on. Please make sure your logon information is correct. (FWB 00008)

For a list of possible error codes and solutions, see the SAP BusinessObjects Business Intelligence Suite Error Message Guide.

Related Information

To configure tracing for the multitenancy management tool [page 105]
Multitenancy management tool errors [page 100]

5.6.1 Multitenancy management tool errors

The following table provides a list of common errors displayed by the multitenancy management tool and recorded in the .csv log file.

<table>
<thead>
<tr>
<th>Error</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid application configuration file, error parsing <code>&lt;config_option&gt;</code></td>
<td>The value provided for <code>&lt;config_option&gt;</code> cannot be processed by the multitenancy management tool. Often this is the result of mandatory options that are left blank.</td>
<td>Open your configuration file (<code>tenant_template_def.properties</code>) and ensure that the value for <code>&lt;config_option&gt;</code> is set correctly. Refer to the documentation and comments in the configuration file for examples. <code>&lt;config_option&gt;</code> is the name of the option, such as <code>templateContentFolder</code>.</td>
</tr>
<tr>
<td>An unexpected error has occurred.</td>
<td>An unspecified exception occurred when running the multitenancy management tool.</td>
<td>Check the trace log file (<code>multitenancymanager_trace.00000&lt;n&gt;.glf</code>) to perform further analysis.</td>
</tr>
<tr>
<td>No matching objects are found using query <code>&lt;query&gt;</code></td>
<td>No template objects can be retrieved based on the template paths specified.</td>
<td>Open your configuration file (<code>tenant_template_def.properties</code>) and ensure that any template paths specified are correct. Check the following options:</td>
</tr>
</tbody>
</table>

Note
A .csv log file will not be created when this error occurs.
<table>
<thead>
<tr>
<th>Error</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to fetch object(s).</td>
<td>An error occurred when fetching objects from the Central Management Server (CMS).</td>
<td>Check that you can connect to the CMS. If the CMS can be connected to, check the .csv log file for the object CUIDs and ensure that they exist in the CMS. If the CMS can be connected to and the objects exist, check the trace log file (multitenancymanager_trace.00000&lt;n&gt;.glf) to perform further analysis.</td>
</tr>
<tr>
<td>Failed to map template object(s) to tenant object(s).</td>
<td>An error occurred when mapping a template object to a tenant object or when the multitenancy management tool CMS application object is retrieved.</td>
<td>Check that you have enough disk space on the home directory of the operating system user running multitenancymanager.jar. Ensure that the home directory is writable. If there is adequate disk space, check the trace log file (multitenancymanager_trace.00000&lt;n&gt;.glf) to perform further analysis.</td>
</tr>
<tr>
<td>Failed to perform backup.</td>
<td>An error occurred while creating a backup of existing tenant objects. This error may occur when running multitenancymanager.jar a second time (or more) on the same tenant. Since the tenant already exists, existing tenant objects are backed up locally by the tool to a BIAR file and this error may be thrown during this process.</td>
<td>Check that you have enough disk space on the home directory of the operating system user running multitenancymanager.jar. Ensure that the home directory is writable. If there is adequate disk space, check the trace log file (multitenancymanager_trace.00000&lt;n&gt;.glf) to perform further analysis.</td>
</tr>
</tbody>
</table>

**Note**

You may not have templates for some object types. It is best practice to set the appropriate optionInclude option to false in these cases—for example, optionIncludeCategories=false..
<table>
<thead>
<tr>
<th>Error</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to restore backup.</td>
<td>An error occurred while restoring the backup of tenant objects to their original state. This error may occur when running <code>multitenancymanager.jar</code> a second time (or more) on the same tenant and if another error triggers a rollback (<code>abortAndRollback=true</code>). When rolling back changes, the tool attempts to restore the original objects and settings of a tenant from the local backup and this error may be thrown during this process.</td>
<td>Check that you have enough disk space on the home directory of the operating system user running <code>multitenancymanager.jar</code>. Ensure that the home directory is writable. If there is adequate disk space, check the trace log file (<code>multitenancymanager_trace.00000&lt;n&gt;.glf</code>) to perform further analysis.</td>
</tr>
</tbody>
</table>
| Failed to refresh tenant object(s) or update shared object(s). | An error occurred while attempting to update tenant documents with a tenant-specific data sources. | Check for a more specific error thrown earlier in the `.csv` log file. Often the result of incorrect or mismatched configuration file options such as:  
  - `crystalreport.templatedb<n>`  
  - `crystalreport.tenantdb<n>`  
  - `crystalreport.templatableprefixes<n>`  
  - `crystalreport.tenanttableprefixes<n>`  
  - `crystalreport.ccis.dataconnection.templatedb<n>`  
  - `crystalreport.ccis.dataconnection.templatableprefixes<n>`  
  - `crystalreport.ccis.dataconnection.tenanttableprefixes<n>`  
  - `ccis.dataconnection.dbcredentials<n>`  
  The CUID and name of the report or document is recorded. If your configuration file settings are correct, check the trace log file (`multitenancymanager_trace.00000<n>.glf`) to perform further analysis. |
<p>| Failed to load document refresh plugin for kind <code>&lt;SI_KIND&gt;</code>. | An error occurred while loading a JAR file from the <code>docRefreshPlugins</code> folder. <code>&lt;SI_KIND&gt;</code> refers to the object type, for example | By default, the <code>docRefreshPlugins</code> folder is located at: |</p>
<table>
<thead>
<tr>
<th>Error</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed to refresh document for <code>&lt;document_details&gt;</code></td>
<td>An error occurred while attempting to update a tenant document with a tenant-specific data source.</td>
<td>Check the trace log file (<code>multitenancyManager_trace.00000&lt;n&gt;.glf</code>) to perform further analysis.</td>
</tr>
<tr>
<td>Failed to prepare tenant for importing.</td>
<td>An error occurred while retrieving a tenant's objects that are to be imported to the CMS.</td>
<td>Check the trace log file (<code>multitenancyManager_trace.00000&lt;n&gt;.glf</code>) to perform further analysis.</td>
</tr>
</tbody>
</table>
| Missing template database configuration `<crystalreport.templatedb<n>` or tenant database configuration `<crystalreport.tenantdb<n>` | A template to tenant mapping for SAP Crystal Reports 2011 direct-to-data connections is missing or not configured correctly. | Open your configuration file (`tenant_template_def.properties`) and ensure that every `<crystalreport.templatedb<n>` option has a corresponding `<crystalreport.tenantdb<n>` option.  
Check the trace log file (`multitenancyManager_trace.00000<n>.glf`) to perform further analysis. |
| Invalid template database configuration, error parsing `<config_option>` | The template database configuration for an SAP Crystal Reports 2011 report does not have the right format. | Open your configuration file (`tenant_template_def.properties`) and ensure that all `<crystalreport.templatedb<n>` options are set to valid values.  
For information about setting tenant options for SAP Crystal Reports 2011 direct-to-data connections, see “SAP Crystal Reports 2011”. |
<p>| Invalid tenant database configuration, error parsing crystalreport.tenantdb&lt;n&gt; | The tenant database configuration for an SAP Crystal Reports 2011 report does not have the right format. | Open your configuration file (<code>tenant_template_def.properties</code>) and ensure that all <code>&lt;crystalreport.tenantdb&lt;n&gt;</code> options are set to valid values. |</p>
<table>
<thead>
<tr>
<th>Error</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
</table>
| Invalid database server type `<dbserver_type>` | The specified data source type (`<dbserver_type>`) for a SAP Crystal Reports 2011 direct-to-data connection is not supported. | Open your configuration file (`tenant_template_def.properties`) and ensure you are using supported data source types for both template and tenant reports. Check the following options:  
  - `crystalreport.templatedb<n>`  
  - `crystalreport.tenantdb<n>`  
  For a list of supported data source types, see “SAP Crystal Reports 2011”. |
| Table prefixes mismatch for `<template_prefix>` and `<tenant_prefix>`, either one of them does not exist, or number of prefixes do not match. | The number of template table prefixes does not equal the number of tenant table prefixes. | Open your configuration file (`tenant_template_def.properties`) and ensure that every template table prefix option has a corresponding mapping to a tenant table prefix option with an equal number of prefixes. Check the following Crystal Reports options:  
  - `crystalreport.templatetableprefixes<n>`  
  - `crystalreport.tenanttableprefixes<n>`  
  - `crystalreport.ccis.dataconnection.templatetableprefixes`  
  - `crystalreport.ccis.dataconnection.tenanttableprefixes` |
| Database configuration is invalid, please see previous item(s) for details. | The database configuration for a tenant’s Crystal report is invalid. Error can be encountered for both SAP Crystal Reports 2011 and SAP Crystal Reports for Enterprise. | Check for a data source update error thrown earlier in the `.csv` log file.  
 Check the trace log file (`multitenancymanager_trace.00000<n>.glf`) to perform further analysis. |
Unexpected internal error has occurred while updating a crystal report document.

An unexpected error occurred while updating a SAP Crystal Reports for Enterprise report with a tenant-specific data source.

Check that the Crystal Reports Processing Server is enabled and running. If the server is running, check the trace log file (multitenancymanager_trace.00000<n>.glf) to perform further analysis.

### Related Information

*SAP Crystal Reports 2011* [page 83]

### 5.6.2 To configure tracing for the Multitenancy Management Tool

1. Open the `BO_trace.ini` file.
   - The default location on Windows is `<INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\conf`.
   - The default location on Unix is `<INSTALLDIR>/sap_bobj/enterprise_xi40/conf`.
2. Uncomment the required lines under the *Trace Syntax and Setting* section.
3. Add an IF statement to specify tracing settings for the Multitenancy Management Tool.
   
   For example:
   ```
   if (process == "multitenancymanager") {
       active = true;
       importance = xs;
       alert = true;
       severity = 'S';
       keep = false;
       size = 100 * 1000;
   }
   ```

   **Tip**

   The process must be specified as `multitenancymanager` for the tracing setting to apply to the Multitenancy Management Tool.

   The table below lists all the available parameters for configuring tracing.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Possible values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>active</code></td>
<td><code>false, true</code></td>
<td>Enables tracing for the current process or server if set to <code>true</code>. Default value is <code>false</code>.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Possible values</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>importance</td>
<td>'&lt;&lt;', '&lt;=', '==', '&gt;', '&gt;&gt;', 'xs', 's', 'm', 'l', 'xl'</td>
<td>Specifies the threshold for tracing messages. All messages beyond the threshold will be traced. Default value is m (medium).</td>
</tr>
<tr>
<td>alert</td>
<td>false, true</td>
<td>Specifies to automatically enable trace for severe system events. Default value is true.</td>
</tr>
<tr>
<td>severity</td>
<td>'S', 'W', 'E', 'A', 'F', success, warning, error, assert, fatal</td>
<td>Specifies the threshold severity over which massages can be traced. 'S' consumes the most disk space. Default value is 'E'.</td>
</tr>
<tr>
<td>size</td>
<td>Possible values are integers &gt;= 1000</td>
<td>Specifies the number of messages in a trace log file before a new one is created. Default value is 100000.</td>
</tr>
<tr>
<td>keep</td>
<td>false, true</td>
<td>Specifies whether or not to keep the old log file after a new file is created. Default value is false.</td>
</tr>
<tr>
<td>administrator</td>
<td>Strings or integers</td>
<td>Specifies an annotation to use in the output log file. For example, if administrator = &quot;hello&quot; this string will be inserted into the log file.</td>
</tr>
<tr>
<td>log_dir</td>
<td></td>
<td>Specifies the output log file directory. By default log files are stored in the Logging folder.</td>
</tr>
<tr>
<td>always_close</td>
<td>on, off</td>
<td>Specifies if the log file should be closed after a trace is written to the log file. Default value is off.</td>
</tr>
</tbody>
</table>

4. Save and close the BO_trace.ini file.

The next time you run the Multitenancy Management Tool, a trace log file (multitenancymanager_trace.00000<n>.glf) is created.

Instead of modifying the default BO_trace.ini file, you can also create a copy of this file specifically for the Multitenancy Management Tool and output the trace log file to a different location. For example, to use C:\my_BO_trace.ini for the trace log settings and output the trace log file to C:\myLogging, edit the following logging options in the multitenancymanagerSystem.properties file:

```xml
<!-- logging -->
<entry key="mtm.systemVar.trace.logDir">C:\myLogging</entry>
<entry key="mtm.systemVar.trace.iniDir">C:\</entry>
<entry key="mtm.systemVar.trace.iniFile">my_BO_trace.ini</entry>
```
**Note**

This will also change the default output location of the .csv log file (multitenancymanager<yyyy><mm><dd><time>.csv).

For information on configuring additional tracing for BI platform servers and other applications, see “Managing and Configuring Logs” in the *Business Intelligence Platform Administrator Guide*. 
6 SAP Crystal Reports 2011 Customization

6.1 Introduction

SAP Crystal Reports 2011 can be repackaged and sold by partners. You can customize the installed product and the installation program to create a seamless experience for customers. The SAP BusinessObjects customization tool customizes the installation program and the installed product with changes such as the following:

- Reducing the product size
- Renaming the product
- Changing default properties in the installation program
- Hiding screens in the installation program

To make customizations, you write a configuration file to specify the customizations then run the SAP BusinessObjects customization tool to create a customized installation program. Customers can then use this installation program to install a customized version of the product.

The customization tool can be used to customize a full installation program, a Support Package installation program, and a Patch installation program.

6.2 Quick start for Crystal Reports

This section shows you how to run the customization tool to create a customized installation program for SAP Crystal Reports. It uses the sample configuration file that is provided with this tool. When you finish this tutorial, you can run your customized installation package and install a customized version of Crystal Reports.

The customizations include changing the default installation type, removing features, hard-coding the product keycode, changing the default installation folder, renaming the product, and changing the Windows Start menu shortcut. These customizations are described in more detail in the configuration file.

1. Set up the SAP BusinessObjects customization tool.
   a) Create a working folder on your development machine, for example: C:\SAPCustomTool\packages.
   b) Copy the contents of the Crystal Reports installation package to C:\SAPCustomTool\packages.
      The installation package contains the folders Collaterals, dunit, langs, and setup.engine in addition to other binaries. See To download the installation program [page 109] for instructions.
   c) (Optional) Add your keycode to the sample configuration file.
      In an XML editor, open the file C:\SAPCustomTool\packages\Collaterals\Tools\CustomizationTool\example_customization_win_cr.xml and replace the phrase PLEASE SET in <replaceProperty id="ProductKey" defaultValue="PLEASE SET" /> with your Crystal Reports keycode.
   d) Create the folder C:\SAPCustomTool\output.

   Note
   This folder must be empty.
e) Run the following command from the command prompt:
```
cd C:\SAPCustomTool\packages\Collaterals\Tools\CustomizationTool
```
The folder `CustomizationTool` contains the executable `customizationtool.exe` and the sample configuration file `example_customization_win_cr.xml`.

2. Run the following command from the command prompt:
```
customizationtool.exe xml=example_customization_win_cr.xml packageDir=C:\SAPCustomTool\packages outputDir=C:\SAPCustomTool\output logDetail=error > C:\oemlog.log
```
Verify that the customized installation package was created at `C:\SAPCustomTool\output`. Ensure no errors were reported in the log file `C:\oemlog.log`.

Note

The customization tool may take several minutes to complete. You can check its progress by viewing the log file.

3. Use `C:\SAPCustomTool\output\setup.exe` to run the customized Crystal Reports installation program.

Crystal Reports is installed with the customizations described in the configuration file `C:\SAPCustomTool\packages\Collaterals\Tools\CustomizationTool\example_customization_win_cr.xml`.

6.3 To download the installation program

2. On the Find your software tab, under the A–Z Index, select Installations and Upgrades.
4. Select Installation and Upgrade > WINDOWS.
5. Select the object titled SAP Crystal Reports 2011 <version> Windows (32B), and then follow the instructions on the website to download and extract the objects.

The software may take a long time to download, and you may need to contact the system administrator to ensure that your company’s firewall will not terminate the download process.

Support Packages and Patches are installation programs that contain updates to SAP Crystal Reports. You can download them from `https://service.sap.com/bosap-support`. On the Find your software tab, under the A–Z Index, click Support Packages and Patches. For more information on installing Support Packages and Patches, see Customizing Support Packages and Patches.

6.4 Planning the customization process

To use the SAP BusinessObjects customization tool:
1. Download the installation program. See To download the installation program [page 109].
2. Decide what customizations are required. See Creating the configuration file [page 111].
3. Write the configuration file to specify the customizations.
4. Run the customization tool to create a customized installation program.
5. Run the customized installation program to install a customized version of SAP Crystal Reports.

6.4.1 Best practices

This section provides recommendations for creating a customized installation program.

Validate the configuration file

You may want to validate the configuration file before running the tool. Use the validate command-line parameter.

Reduce product size

Customers prefer a smaller installation program and a smaller installed product. To keep the product as small as possible:

- Remove any language packs that are not required.
- Remove any features that are not required.
- Remove any items from the Collaterals folder that are not required.

Apply customized names consistently

The product name and version number appear in several places in the installation program and in the installed product. Ensure you verify customizations in the following locations:

- Product name, product version, and product major version
- Windows Start menu entry and all feature shortcuts
- Windows Add Remove Program utility
- Default installation folder

Consider name changes in all languages

It is good practice to consider how the customized name appears in all supported languages.
Modify patch installation programs to be consistent with the main installation program

You must apply the same customizations to Support Packages and Patches that you applied to the main release. If you release a customized main installation program and then try to release a Support Package or Patch installation program with different customizations, you might see unpredictable results which might not be repairable using standard rollback procedures.

Test rollback, modify, and repair installations for Support Packages and Patches

Rollback, modify, and repair installations are supported for customized Support Packages and Patches, provided they have been customized in a manner consistent with the main installation package. It is recommended to test these scenarios.

Related Information

Command line parameters [page 132]

6.5 Creating the configuration file

The following section describes the customizations you can make to the installation program by editing the configuration file:

- Renaming the product
  - Customizing the product name and version number
  - Customizing the Windows Start menu shortcuts
  - Customizing the Windows Add Remove Program utility
  - Customizing the installation folder
- Customizing default user input
- Removing installation screens
- Embedding a keycode
- Removing features
- Preventing prerequisite checks
- Removing language packs
- Changing resources
  - Customizing the images in the installation program
  - Customizing the license agreement
Removing items from the Collaterals folder

6.5.1 Configuration file overview

The SAP BusinessObjects customization tool uses information in the configuration file to perform the customizations. The configuration file is an XML document, and you use XML elements to describe your customizations. The sample configuration file is contained in this folder in the installation program:

Collaterals\Tools\CustomizationTool\example_customization_win_cr.xml

The file must have this format:

```xml
<oem name="<Any name>">
  <cloneProduct sourceId="product.crystalreports-4.0-core-32">
    ...
  </cloneProduct>
</oem>
```

The configuration file for the full installation program can have any name, for example, oem.xml.

The configuration file for the Support Package installation program is described in the section How to customize Support Packages and Patches.

**Note**

The configuration file must be written in correct XML syntax. Use an XML editor to create and edit the file, and verify that the format is correct before running the tool.

**Example**

This example specifies the following customizations:

- Change the product’s long name to “Custom Company Crystal Reports” for all languages.
- Change the product’s short name to “Custom CR” for all languages.
- Change the publisher and product name for the Windows Add Remove Program entry.
- Remove the installation screen titled Choose Installation Type and set the installation type to Custom.
- Specify that the only language packs that are included in the installation package are English, French, German, Italian, and Chinese.

```xml
<oem name="CustomCompanyCrystalReports">
  <cloneProduct sourceId="product.crystalreports-4.0-core-32">
    <replaceString id="product.cr_name" value="Custom Company Crystal Reports" lang="all"/>
    <replaceString id="product.cr_shortname" value="Custom CR" lang="all"/>
    <arp duSourceId="product.crystalreports.arp-4.0-core">
      <arg id="publisher" value="Custom Company"/>
      <arg id="display_name" value="Custom Company Crystal Reports"/>
    </arp>
    <replaceProperty id="InstallType" defaultValue="custom"/>
    <removeDialog id="ChooseInstallType2.dialog"/>
  </cloneProduct>
</oem>
```
6.5.2 Renaming the product

You can rename the product by customizing the following:

- The product name and version number
- The Windows Add or Remove Programs entry
- The Start menu entry for feature shortcuts
- The default installation folder

The following sections explain these customizations.

6.5.2.1 Customizing the product name and version number

You can customize the product name and version number. Use the replaceString element with the desired string ID:

```
<replaceString id="<string id>" value="<new value>" lang="<language list>"/>
```

There are four strings that represent the product name and version number: the product long name, the product short name, the product version number, and the product major version number. The full product name is composed of the product long name and the version number. The product short name and product major version are used in the Windows shortcut menu.

<table>
<thead>
<tr>
<th>String description</th>
<th>String ID</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product long name</td>
<td>product.cr_name</td>
<td>Crystal Reports</td>
</tr>
<tr>
<td>Product short name</td>
<td>product.cr_shortname</td>
<td>Crystal Reports</td>
</tr>
<tr>
<td>Product version</td>
<td>product_cr_version</td>
<td>2011</td>
</tr>
<tr>
<td>Product major version</td>
<td>product_cr_majorversion</td>
<td>2011</td>
</tr>
</tbody>
</table>

**Note**

You should customize the product version and product major version together. For example, if you change product version to “1.0” you should also customize product major version to “1”. Otherwise the version number in the menus will not match the version number in the product.

You can specify a new name for each language. For a list of language codes, see Language Codes.
Example

This example makes the following customizations:

- Change the product long name to “Custom Company Crystal Reports” and the product short name to “Custom CR” for English.
- Change the product long name to “Custom Company Crystal Reports (French)” and the product short name to “Custom CR (French)” for French.
- Change the product version to “1.0” and the product major version to “1” for all languages.

The product name in languages other than English and French will remain as the default value, but the product version and major version will be changed for all languages.

The result of the customization appears below. Notice the version number “FP3” is not removed:

```xml
<replaceString id="product.cr_name" value="Custom Company Crystal Reports" lang="en"/>
<replaceString id="product.cr_shortname" value="Custom CR" lang="en"/>
<replaceString id="product.cr_name" value="Custom Company Crystal Reports (French)" lang="fr"/>
<replaceString id="product.cr_shortname" value="Custom CR (French)" lang="fr"/>
<replaceString id="product_cr_version" value="1.0" lang="all"/>
<replaceString id="product_cr_majorversion" value="1" lang="all"/>
```

To remove instances of “FP3” from the installation program

When you run the installation program, you may see instances of “FP3” in the product name. To remove “FP3”, modify the lines in the following files:

<table>
<thead>
<tr>
<th>File name</th>
<th>Original line</th>
<th>Modified line</th>
</tr>
</thead>
<tbody>
<tr>
<td>dunit\product.crystalreports-4.0-core-32\setup.ui.framework \uitext\CrystalReports \product.lang_&lt;language code&gt;.uitext.xml</td>
<td>&lt;string id=&quot;productname_patch&quot; value=&quot; FP3&quot;/&gt;</td>
<td>&lt;string id=&quot;productname_patch&quot; value=&quot;&quot;/&gt;</td>
</tr>
</tbody>
</table>
You must modify one file for every language that the installation program supports. For a list of language codes, see Language codes. When you run the customization tool, and then run the installation program, all instances of “FP3” will be removed. This process will be simplified in a future release.

Example

To remove “FP3” from the English installation program, modify the following files:

- product.lang_en.uitext.xml
- setup.ui.framework.lang_en.uitext

The result of the customization appears below:

6.5.2.2 Customizing the Windows Start menu shortcuts

The Windows Start menu contains shortcuts for features such as the ODBC Data Source Administrator. You can customize the name, location, and tooltip for each shortcut. Any shortcut that you do not customize will be grouped under the default Start menu entry, Crystal Reports 2011.

The default Start menu in English installations looks like this:
Use the `<shortcut>` element to customize the location, shortcut name, and tooltip for each feature:

```xml
<shortcut duSourceId="<shortcut deployment unit ID>">
  <arg id="linkFullPath" value="<full path to shortcut link>" lang="<language list>"/>
  <arg id="description" value="<tooltip string>" lang="<language list>"/>
</shortcut>
```

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>duSourceId</td>
<td>The shortcut deployment unit ID that you want to modify. Typical values include:</td>
</tr>
<tr>
<td></td>
<td>● product.crystalreports.shortcut.crw-4.0-core</td>
</tr>
<tr>
<td></td>
<td>Crystal Reports 2011</td>
</tr>
<tr>
<td></td>
<td>● product.crystalreports.shortcut.odbc-4.0-core</td>
</tr>
<tr>
<td></td>
<td>ODBC Data Source Administrator</td>
</tr>
<tr>
<td></td>
<td>● product.crystalreports.shortcut.rptpubwiz-4.0-core</td>
</tr>
<tr>
<td></td>
<td>Report Upload Wizard</td>
</tr>
<tr>
<td></td>
<td>For a complete list of duSourceId values, see <a href="#">Shortcut deployment unit IDs</a>.</td>
</tr>
<tr>
<td>linkFullPath</td>
<td>The full path to the shortcut link. Be sure to add .lnk to the shortcut link or the link will not be created. You can put the link in the Start menu or you can put it on the desktop. The SAP BusinessObjects customization tool will create the links correctly. You can specify one link for each language. For a list of language codes, see <a href="#">Language Codes</a>.</td>
</tr>
<tr>
<td>description</td>
<td>The tooltip string to display when the user hovers the mouse over the shortcut. You can specify one tooltip for each language.</td>
</tr>
</tbody>
</table>

**Example**

This example makes the following customizations:

- Change the name of the Crystal Reports 2011 shortcut to “Custom Company CR” for English.
- Customize the Crystal Reports 2011 tooltip to “Launch Custom Company CR” for English.
- Change the name of the “ODBC Data Source Administrator” shortcut to “Custom ODBC” for English.
- Customize the “ODBC Data Source Administrator” tooltip to “Custom ODBC” for English.
- Place the “Custom Company CR” and “Custom ODBC” shortcuts under the **Start** menu entry called “Company Programs”.
- Change the name of the “Report Upload Wizard” shortcut to “Custom Wizard” for English.
- Place the “Custom Wizard” shortcut under the **Start** menu entry called “Custom Wizard”.
- Customize the “Custom Wizard” tooltip to “Launch Custom Wizard” for English.

The shortcut name and tooltip will remain unchanged for all other languages.

**Note**

To use this example you must create the following links and folders:
- Custom Company CR.lnk
- Custom ODBC.lnk
- Custom Wizard.lnk
- Company Programs
- Custom Wizard

Place Custom Company CR.lnk and Custom ODBC.lnk in the folder Company Programs and place Custom Wizard.lnk in the folder Custom Wizard. Place these folders in the same location you plan to redirect the installation folder to.

In this example, the installation folder has been redirected to the c:\ drive. See Customizing the installation folder for more information.

The result of the customization appears below:
Example

This example keeps the default name of the “Report Upload Wizard” feature shortcut, but places it under the Start menu entry called “Custom Wizard” for all languages. It also changes the “Custom Wizard” tooltip to “Launch Custom Wizard” for all languages.

Note

To use this example you must place the Report Upload Wizard.lnk in the folder Custom Wizard. Place this folder in the same location as the installation folder.

In this example, the installation folder has been redirected to the C:\ drive.

<shortcut duSourceId="product.crystalreports.shortcut.rptpubwiz-4.0-core">
  <arg id="linkFullPath" value="[programmenufolder]\Custom Wizard\Report Upload Wizard.lnk" lang="all"/>
  <arg id="description" value="Launch Custom Wizard" lang="all"/>
</shortcut>

6.5.2.3 Customizing the Windows Add Remove Program utility

You can customize the display name, the publisher, and the icon in the Windows Add Remove Program (ARP) utility. Use the following element:

<arp duSourceId="product.crystalreports.arp-4.0-core">
  <arg id="publisher" value="<publisher name/>">
  <arg id="display_name" value="<product name>" lang="<language list>"/>
  <arg id="display_icon" value="<full path to icon>"/>
</arp>
The display name must have a \texttt{lang} tag to specify a different display name for each language. Multiple languages using the same display name must be separated by a semi-colon. Any unspecified languages will use the default value.

For a list of language codes, see \textit{Language Codes}.

You must take additional steps in order for the customized value of the publisher to display correctly in the ARP. Follow these steps:

1. Open the file \texttt{dunit\product.crystalreports.arp-4.0-core-32\seed.xml}.
2. Find the element \texttt{<action id="AddARPEntry"}. You will see multiple nested \texttt{<arg>} elements.
3. Add the following line under \texttt{<arg name="DISPLAY_NAME" value="[ARP.DISPLAYNAME]" />}:
   \begin{verbatim}
   <arg name="PUBLISHER" value="SAP" />
   \end{verbatim}
4. Save the file.

After you customize the publisher name in the configuration file, run the customization tool, and then install the customized product, your customized value for the publisher displays in the ARP. This process will be simplified in a future release.

Icons displayed in the Windows ARP utility are typically 16x16. Refer to Windows documentation for complete information on creating the icon.

This example makes the following customizations in the Windows ARP utility:

- Change the product name to “Custom Company Crystal Reports Patch 1” for English and French.
- Change the product name to “Custom Company Crystal Reports (German)” for German.
- Change the publisher to “Custom Company”.
- Replace the display icon with the icon \texttt{C:\SAPCustomTool\CC_logo.ico}.

To use this example you must put an icon called \texttt{CC_logo.ico} in the location \texttt{C:\SAPCustomTool}.

The result of the customization appears below:

\begin{verbatim}
<arp duSourceId="product.crystalreports.arp-4.0-core"/>
  <arg id="publisher" value="Custom Company"/>
  <arg id="display_name" value="Custom Company Crystal Reports Patch 1" lang="en;fr"/>
  <arg id="display_name" value="Custom Company Crystal Reports (German)" lang="de"/>
  <arg id="display_icon" value="C:\SAPCustomTool\CC_logo.ico"/>
</arp>
\end{verbatim}
6.5.2.4  Customizing the installation folder

You can customize the location of the default installation folder. Use the `replaceProperty` element with id="<installation folder file path>":

```xml
<replaceProperty id="InstallDir" defaultValue="<default installation folder>"/>
```

Example

Change the default install folder to `C:\MyInstallDIR\CustomCompanyCrystalReports`.

```xml
<replaceProperty id="InstallDir" defaultValue="C:\MyInstallDIR\CustomCompanyCrystalReports"/>
```

6.5.3  Customizing default user input

You can customize the default value of the user input that is collected by the installation program. Use the `replaceProperty` element with id="<property id>" and the new default value:

```xml
<replaceProperty id="<property id>" defaultValue="<value to use as default value>"/>
```

For a list of property IDs, see Installation screen and property IDs.

The Windows installation program collects user input using dialog boxes, radio buttons, and other user interface elements.

Example

On the installation screen called Choose Install Type, the default install type is Typical. This example changes the default install type to Custom.

```xml
<replaceProperty id="InstallType" defaultValue="custom"/>
```

The result of the customization appears below:
6.5.4 Removing installation screens

You can remove installation screens from the installation program. Use the `removeDialog` element with the installation screen ID:

```xml
<removeDialog id="<installation screen ID>"/>
```

For a list of installation screen IDs, see *Installation screen and property IDs*.

**Example**

This example shows how to remove the installation screen titled *Select Features*.

```xml
<removeDialog id="SelectFeatures.dialog"/>
```
6.5.5 Embedding a keycode

You can embed a keycode in the installation program so the customer does not need to enter one. This task involves:

- Providing a default value for the keycode
- Removing the installation screen in which the user enters a keycode

**Example**

Use the `replaceProperty` element with `id="ProductKey"` to provide a default keycode. Keycodes must have the format `XXXXX­XXXXXXX­XXXXXXX­XXXXXXX­XX`.

Use the `removeDialog` element with `id="CREnterProductKey.dialog"` to remove the installation screen for the license key.

```xml
<replaceProperty id="ProductKey" defaultValue="XXXXX­XXXXXXX­XXXXXXX­XXXXXXX­XX"/>
<removeDialog id="CREnterProductKey.dialog"/>
```

**Related Information**

- Installation screen and property IDs [page 143]
- Customizing default user input [page 120]
- Removing installation screens [page 121]

6.5.6 Removing features

SAP Crystal Reports includes many optional features. You can remove a feature from the installation program. Use the `removeFeature` element with `id="<feature id>"`:

```xml
<removeFeature id="<Feature ID>"/>
```

For a list of feature IDs, see Feature IDs.

When you specify a feature to be removed, the SAP BusinessObjects customization tool removes all executables, installation screens, and other files that belong to that feature. Removing unnecessary features is a good way to reduce the size of the customized product.

**Example**

Remove the geographic mapping feature. This ID will remove the program's ability to display relationships between data and geographic regions:

```xml
<removeFeature id="Mapping"/>
```
6.5.7 Preventing prerequisite checks

Prerequisites are conditions that must exist on the host machine in order for the installation program to succeed. The installation program verifies the existence of these prerequisites before starting, and displays the results in the Prerequisite check screen. Removing the Prerequisite check screen prevents prerequisite checks from being performed. Use the removeDialog element with id="CheckPreRequisites.dialog"

i Note
It is recommended that you remove this installation screen only if you are performing the prerequisite checks by some other means. If the prerequisites are not met, the installation program will fail.

Example
This example removes the Prerequisite check screen and prevents prerequisite checks from being performed.

<removeDialog id="CheckPreRequisites.dialog"/>

6.5.8 Removing language packs

The installation program allows the user to select which language packs to install. A language pack contains translated versions of all the strings that are used by the installed product. By default, all possible language packs are included in the installation program. You can specify which language packs to include. Use the languageIncludeList element with a list of language codes:

/languageIncludeList value="<list of language codes>"/>

For a list of language codes, see Language codes.

i Note
Language packs can be large. The installation program will be smaller if fewer language packs are included.

Example
Include English, French, and German language packs in the installation program. The user can select from this list during installation.

/languageIncludeList value="en,fr,de"/>

6.5.9 Changing resources

The installation program stores image and text files as resources in this folder:
You can customize the resources in this folder. Resources that are commonly customized include:

- Images in the installation program
- License agreement in the installation program

To customize a resource:

1. Create a custom resources folder, for example `C:\MyResources`. The folder can have any name, but note that it will be visible to customers. Use the same folder for all resources that you customize.
2. Create a new resource with the same name and filepath as the original resource, and place it into the custom resources folder. See the related topics section for specific examples.
3. Add the `<resources>` element to the configuration file to specify the location of the custom resources folder, for example:
   ```xml
   <resources cleanTarget="yes" sourcePath="C:\MyResources"/>
   ```

   **cleanTarget attribute**

   If you set `cleanTarget='yes'`, the customization tool will delete the original `resources` folder and use only those resources in the custom resources folder. This option is not recommended.

**Related Information**

*Customizing the images in the installation program* [page 124]
*Customizing the license agreement* [page 126]

**6.5.9.1 Customizing the images in the installation program**

You can customize the images in the installation program including the welcome screen, the top image for all screens, and the billboard for the progress dialog. Images are stored as files in the resources folder:
You customize an image by creating a new image file, putting the file in the custom resources folder, and adding the `resources` element to the configuration file.

**Example**

**Customizing the image in the welcome screen**

1. Create a folder called `MyResources` in the `C:\` drive.
2. Create a new image file called `dialogFull.bmp` and place it in the `C:\MyResources` folder.
3. Ensure that the `resources` element exists in the configuration file as follows:

   ```xml
   <resources cleanTarget="no" sourcePath="C:\MyResources"/>
   ```

**Related Information**

[Changing resources](page 123)
6.5.9.2 Customizing the license agreement

You can customize the license agreement that is presented to the user during installation. License agreements are stored as text files in the resources folder:

dunit\product.crystalreports-4.0-core-32\setup.ui.framework\resources\<language code>

For example, the English license agreement is located here:

dunit\product.crystalreports-4.0-core-32\setup.ui.framework\resources\en\license_en.rtf

For a list of language codes, see Language codes.

You customize the license agreement by creating a new license file, putting the file in the custom resources folder, and then adding the resources element to the configuration file.

**Example**

Customize the English license agreement

The English license agreement is stored here:

dunit\product.crystalreports-4.0-core-32\setup.ui.framework\resources\en\license_en.rtf

To customize the English license agreement:

1. Create a folder called MyResources in the C: \ drive.
2. Create a folder called en and place it in the C: \ MyResources folder.
3. Create a new license agreement file called license_en.rtf and place it in the C: \ MyResources \ en folder.
4. Ensure that the resources element exists in the configuration file as follows:

   `<resources cleanTarget="no" sourcePath="C: \ MyResources"/>

Related Information

Changing resources [page 123]

6.5.10 Removing items from the Collaterals folder

The SAP Crystal Reports installation program stores tools, samples, and documentation in the Collaterals folder of the installation program. By default, a customized installation program that is delivered to customers will also contain the Collaterals folder with the same contents. You can remove unwanted items from the Collaterals folder in order to reduce the size of your customized installation program. Use the collaterals element with cleanTarget="yes" and sourcePath="<full path to custom Collaterals folder>":

   `<collaterals cleanTarget="yes" sourcePath="<full path to custom Collaterals folder>"/>`
To remove items from the Collaterals folder

1. Copy the contents of the existing Collaterals folder to a new location, for example C:\MyCollaterals.
2. Remove any items from C:\SAPCustomTool\Collaterals that are not required by your customized installation program.
3. Add the `<collaterals>` element to the configuration file to specify the location of the custom collaterals folder, for example:

   `<collaterals cleanTarget="yes" sourcePath="C:\MyCollaterals"/>

Table 18: Description of items in the Collaterals folder

<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
<th>When to remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaterals &gt; Add-Ons &gt; SAP</td>
<td>Provides connectivity to SAP systems. Remove this folder if there is no need to connect to SAP systems.</td>
<td></td>
</tr>
<tr>
<td>Collaterals &gt; CustomizationTemplate</td>
<td>Contains the sample template.zip file for customizations to the report designer. Remove this folder if the customers do not need to provide a sample template.zip file.</td>
<td></td>
</tr>
<tr>
<td>Collaterals &gt; Docs</td>
<td>Documentation in every language that Crystal Reports supports. Remove any languages that are not included in the customized installation program. For a list of language codes, see <a href="#">Language codes</a>.</td>
<td></td>
</tr>
<tr>
<td>Collaterals &gt; Tools &gt; CustomizationTool</td>
<td>The SAP BusinessObjects customization tool. Remove this folder if the customers do not need to customize their own installation programs.</td>
<td></td>
</tr>
</tbody>
</table>

6.6 Customizing the report designer

You can customize the following properties of the report designer:

- Splash screen
- Start page
- String values on the menu

You can make these customizations after installing the program. You can also deploy your customizations to your customized installation package, so that the customizations are applied when users install the program.
6.6.1 Customizing the splash screen

When Crystal Reports is run, a splash screen loads. You can replace this splash screen with your own bitmap.

**Note**
The following steps assume you already installed Crystal Reports. If you want to deploy the splash screen into your customized installation program, rename the bitmap you want to use for the splash screen to splash.bmp and follow the instructions in Deploying the OEM customization file.

1. Rename the bitmap you want to use for the splash screen to splash.bmp.
   **Note**
   The bitmap must be a valid .bmp file and can be of any size.

2. Place splash.bmp in the same folder as crw32.exe.
   By default, crw32.exe is found in the following location:
   ```
   C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86
   ```
   When Crystal Reports is run, splash.bmp should load. If it does not load, then the default splash screen loads instead.

6.6.2 Customizing the start page

You can modify the content of the start page with your own HTML file. Most of your customizations will affect the top part of the start page. You can also remove the bottom part, which contains links to SAP Crystal Reports web pages.

**Note**
The following steps assume you already installed Crystal Reports. If you want to deploy the start page into your customized installation program, rename the HTML file you want to use for the start page to start.html and follow the instructions in Deploying the OEM customization file.

1. Rename the HTML file you want to use for the start page to start.html.

2. Place start.html in the Start Page\<language code> sub-folders, depending on the languages you want to support.
   **Note**
   By default, the file path of the sub-folder is:
   ```
   C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86\Start Page\<language code>
   ```
   For a list of all language codes, see Language codes.
Tip

If `start.html` uses images, place them in the following location:

```
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86\Start Page\image
```

Your HTML file should use relative paths to point to this folder.

When Crystal Reports is run, the start page will display the customizations you made in `start.html`.

### 6.6.3 Customizing menu strings

You can modify string values on the menu that contain the product name SAP Crystal Reports. Such values are:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
<th>Location</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductName</td>
<td>Product name</td>
<td>Window title</td>
<td>SAP Crystal Reports</td>
</tr>
<tr>
<td>CrystalReportHelp</td>
<td>Product help</td>
<td>Help menu</td>
<td>SAP Crystal Reports Help</td>
</tr>
<tr>
<td>AboutCrystalReport</td>
<td>About product help</td>
<td>Help menu</td>
<td>About SAP Crystal Reports</td>
</tr>
</tbody>
</table>

To customize these strings, an XML file is required. The XML file name must have the following format:

```
crw_oem_res_<language code>.xml
```

For example, the English XML file name is:

```
crw_oem_res_en.xml
```

For a list of language codes, see [Language codes](#).

Note

If you have already installed Crystal Reports, then the XML file should be placed in the same folder as `crw32.exe`. By default, this is found in:

```
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86
```

When Crystal Reports is run in a specific language, the corresponding language XML file is loaded.

Note

If you want to deploy the customized strings into your customized installation program, follow the instructions in [Deploying the OEM customization file](#).

Example

This example makes the following customizations:
• Change the `ProductName` value to Custom CR
• Change the `CrystalReportHelp` value to Custom CR help
• Change the `AboutCrystalReport` value to About Custom CR

```
<Root>
  <ProductName>Custom CR</ProductName>
  <MainFrameMenu>
    <Help>
      <CrystalReportHelp>Custom CR help</CrystalReportHelp>
      <AboutCrystalReport>About Custom CR</AboutCrystalReport>
    </Help>
  </MainFrameMenu>
</Root>
```

**Note**

- To support multiple languages, the attribution encoding should be UTF-8: `<?xml version="1.0" encoding="UTF-8"?>`. In addition, when the XML file is saved with a text editor, select **UTF-8** from the `Encoding` menu.
- Keep the property name and value in the same line. For example the following is acceptable:

  ```
  <ProductName>Custom CR</ProductName>
  ```

The following is not acceptable. There will be unrecognizable characters in the modified strings when Crystal Reports is run:

  ```
  <ProductName>
  Custom CR
  </ProductName>
  ```

### 6.6.4 Deploying the OEM customization file

After you prepare your customized files (splash images, start page, and menu strings), you can deploy your customizations into the installation package. Place the customized files in a zip file.

1. Create a zip file named `template.zip`.
2. Place the customized files into the zip file.

**Note**

The folder structure within the zip file must match the structure of the folder where you want the files placed, relative to the installation folder. Files must be placed in the following location in `template.zip`:

`SAP BusinessObjects Enterprise XI 4.0\win32_x86`.

For example, the following customized files are placed in these locations in `template.zip`:

<table>
<thead>
<tr>
<th>Customized file</th>
<th>Location in <code>template.zip</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>splash.bmp</td>
<td><code>SAP BusinessObjects Enterprise XI 4.0\win32_x86</code></td>
</tr>
</tbody>
</table>
3. Copy the zip file to the following location:
   dunit\product.crystalreports.oemzips-4.0-core-nu\OEMZips

   i  Note
   The OEMZips folder may need to be created manually.

4. Run the installer.

   The contents of `template.zip` will be unzipped to the installation folder.

   i  Note
   In the install package of SAP Crystal Reports, there is a sample zip file located in:
   Collaterals\CustomizationTemplate\template.zip

### 6.7 Running the tool

The SAP BusinessObjects customization tool `customizationtool.exe` is included with the SAP Crystal Reports installation package in this location:
Collaterals\Tools\CustomizationTool

This section explains the command line parameters used for the tool.

i  Note
The SAP BusinessObjects customization tool may take several minutes to complete. You can check its progress by viewing the log file.

**Example**

This example runs the customization tool and creates a log file located in the C:\ drive. To use this example, you must do the following:
- Create a configuration file called `oem.xml` in the location C:\SAPCustomTool.
- Download the Crystal Reports installation package to the location C:\SAPCustomTool\packages. See To download the installation program [page 109].
- Create a folder called `output` in the location C:\SAPCustomTool.
Run the following command from the command prompt:
```
cd C:\SAPCustomTool\packages\Collaterals\Tools\CustomizationTool
customizationtool.exe xml=C:\SAPCustomTool\oem.xml packageDir=C:\SAPCustomTool\packages
outputDir=C:\SAPCustomTool\output logDetail=error > C:\oemlog.log
```

For more information on how to run the SAP BusinessObjects customization tool, see *Quick start for Crystal Reports*.

### 6.7.1 Command line parameters

#### Table 19: Required parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>xml</td>
<td>Full path to the configuration file.</td>
<td>xml=C:\SAPCustomTool\oem.xml</td>
</tr>
<tr>
<td></td>
<td>The configuration file for the full installation program can have any name.</td>
<td></td>
</tr>
<tr>
<td>packageDir</td>
<td>Full path to the folder that contains the installation program you are modifying.</td>
<td>packageDir=C:\SAPCustomTool\packages</td>
</tr>
<tr>
<td></td>
<td>The installation program is downloaded from SAP Service Marketplace in order to start the installation of SAP Crystal Reports. It contains the folders Collaterals, dunit, langs, and setup.engine in addition to other binaries.</td>
<td></td>
</tr>
<tr>
<td>outputDir</td>
<td>Full path to the folder where the customized installation program will be created. Must be empty before running the tool.</td>
<td>outputDir=C:\SAPCustomTool\output</td>
</tr>
</tbody>
</table>

#### Table 20: Optional parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>baselinePath</td>
<td>Full path to a root folder containing the original, non-customized versions of all previous full and update installation programs you have customized. Use a semicolon (;) to separate root folders.</td>
<td>Assume you want to customize SAP Crystal Reports 2011 Support Package 5 and you customized the previous programs: 2011 (Full install), 2011 SP4. Customize 2011 Support Package 5, and provide the root folder path to the non-customized packages for the major 2011 release and Support Package 4 release. For example, if the non-customized pack-</td>
</tr>
</tbody>
</table>
## Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ages are contained in the following directory structure:</td>
<td></td>
<td>C:\productUpdates\2011\2011 Full\SP4\</td>
</tr>
<tr>
<td>set the value to baselinePath=C:\productUpdates\2011\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>See <a href="#">Customizing Support Packages and Patches</a> for more information and examples of the baselinePath parameter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>logDetail</td>
<td>The level of detail tracked in the log file. The default value is info. The following are the accepted values:</td>
<td>logDetail=warn</td>
</tr>
<tr>
<td>• error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• warn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• info</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• debug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• trace</td>
<td></td>
<td></td>
</tr>
<tr>
<td>action</td>
<td>The tool mode. The following are the accepted values:</td>
<td>action=validate</td>
</tr>
<tr>
<td>• generate (default value) The tool performs the specified customizations.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• validate The tool validates the configuration file but does not perform any customizations.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Related Information

[Quick start for Crystal Reports](#) [page 108]

### 6.8 Customizing update installation programs

Update installation programs are Support Packages or Patches that contain updates to your existing SAP Crystal Reports software. Support Packages contain more updates than Patches but are released less frequently. You can use the SAP BusinessObjects customization tool to customize these installation programs, but some modifications to the command line and configuration file are required.
6.8.1 Frequently asked questions about update installation programs

Where do I find Support Packages and Patches?

2. On the Find your software tab, under the A–Z Index, click Support Packages and Patches.
4. Select your Support Package or Patch, then follow the instructions on the website to download and extract the objects.

What parts of update installation programs can I customize?

You can customize the same aspects of update installation programs as you did in the main installation program. Because Support Package and Patch updates contain fewer installation screens, not all of the customization steps apply. It is recommended to run the Support Package or Patch before customizing it to determine what customizations you require.

How do I customize update installation programs?

Update installation programs use the same architecture as a main installation program for Crystal Reports (full installation), so you can use the customization tool as described in Creating the configuration file and Running the tool, with some modifications to the command line and the configuration file. For more information, see Customizing Support Packages and Patches in this section.

Is it necessary to customize and install all Support Package and Patch updates?

No. As with non-customized versions of Crystal Reports, you only need to install the updates that you want. This may be a Support Package, a Patch, or both.
Can I install a non-customized update on a customized Crystal Reports installation?

Yes. Both customized and non-customized Support Packages or Patches may be applied to your customized installation. However, non-customized Support Package or Patch installation programs will not display your branding or installation customizations you created for the main installation program.

I have delivered a customized version of Crystal Reports to customers but I want to modify the customizations in an update installation program. Is this possible?

This scenario is not supported. The customizations that you make to Support Packages and Patches must be consistent with the original customizations.

6.8.2 Quick start for update installation programs

Ensure that you have customized and installed the main (full) installation program (SAP Crystal Reports) using the instructions in Quick start for Crystal Reports, and that the non-customized installation package is located in C:\SAPCustomTool\packages.

This section shows you how to run the SAP BusinessObjects customization tool to customize the installation program for a Support Package. It uses the sample configuration file provided with the customization tool. Notice that the sample configuration file contains the <cloneProduct> element for the main installation program as well as the <clonePatchProduct> element for a Support Package installation program.

**Note**

You can run this example only when a Support Package is available on https://service.sap.com/bosap-support.

1. Download the installation program for the Crystal Reports Support Package to the folder C:\SAPCustomTool\SupportPackage.
2. Ensure the product_cr_version for the <clonePatchProduct> element in the configuration file matches the version number of the Support Package that you downloaded. See Customizing the product name and version number.
3. Run the following command from the command prompt: cd C:\SAPCustomTool\packages\Collaterals\Tools\CustomizationTool
4. Customize the Crystal Reports Support Package and place the customized installation program in C:\SAPCustomTool\output\SupportPackage by using the following command:
customizationtool.exe xml=example_customization_win_cr.xml packageDir=C:\SAPCustomTool\SupportPackage baselinePath=C:\SAPCustomTool\packages outputDir=C:\SAPCustomTool\output\SupportPackage logDetail=error > C:\oemlog_SP02.log
5. Use `C:\SAPCustomTool\output\SupportPackage\setup.exe` to run the customized installation program for the Crystal Reports Support Package.

### 6.8.3 How to customize update installation programs

Use the configuration tool as described in *Creating the configuration file* and *Running the tool* to customize update installation programs for Support Packages and Patches, with the following differences:

- The configuration file must use the `clonePatchProduct` element (with the correct product ID), instead of the `cloneProduct` element.
- The configuration file must contain the complete `<cloneProduct>` element for the main installation package that you are updating. If it does not, it may cause unpredictable results, especially when customizations involve removing features.
- The configuration file cannot contain more than one `clonePatchProduct`. If you are customizing both a Support Package and a Patch, you must create two configuration files: one file containing `cloneProduct` and `clonePatchProduct` for the Support Package, and the other file containing `cloneProduct` and `clonePatchProduct` for the Patch.
- Refer to all prerequisite installation programs using the `baselinePath` command.

All configuration file elements and command-line parameters can be used to customize update installation programs, but not all of them are applicable to every Support Package or Patch. Run the installation program for the Support Package or Patch first to determine what you need to customize, then use the information in *Creating the configuration file* and *IDs and codes for Crystal Reports customization* to create the customization file.

#### To specify the product version in the configuration file

The configuration file for update installation programs must contain the product version in the `clonePatchProduct` element as shown below:

```xml
<oem name="<any name>">
  <clonePatchProduct sourceId="<product version>">
    ...
  </clonePatchProduct>
</oem>
```

The `product version` in the configuration file must match the version number of the installation program that you are customizing. To find the version number, look in the `dunit` folder for a folder with a name in this format:

`product.cr.patch-4.x.x.x-core-32`

You can use the name of this folder as the `product version`.

**Example**

This example configuration file customizes SAP Crystal Reports 2011 Patch 1, which has the product version `product.cr.patch-4.1.0.1-core-32`. The configuration file customizes the product long name to *Custom Company Crystal Reports* and the product short name to *Custom CR*.

```xml
<oem name="Custom Patch Tool">
  <clonePatchProduct sourceId="product.cr.patch-4.1.0.1-core-32">
    ...
  </clonePatchProduct>
</oem>
```
To use the baselinePath parameter

Use the command line parameter `baselinePath` to refer to a root folder containing the original, non-customized versions of all previous full or update installation programs you have customized. This means you must keep the original installation packages.

**Note**

This parameter replaces the `baselinePackages` parameter introduced in 2011 Feature Pack 3.

To simplify the `baselinePath` parameter value, reference a single root folder - the customization tool will ignore unneeded files and folders. Otherwise, use a semicolon (:) in the `baselinePath` value to specify multiple root folders. Consider the following examples.

**Example**

**Customizing Crystal Reports 2011 SP5**

Assume you are customizing Crystal Reports 2011 Support Package 5 and you customized the previous programs: 2011 (Full install), 2011 SP4. Assume the non-customized installation programs are located in the following directory structure:

```
C:\productUpdates\2011\2011 Full\SP4\n```

Set the `baselinePath` parameter to:

```
baselinePath=C:\productUpdates\2011\n```

**Example**

**Customizing Crystal Reports 2011 SP5 Patch 2**

Assume you are customizing Crystal Reports 2011 Support Package 5 Patch 2 and you customized the previous programs: 2011 (Full install), 2011 SP4, 2011 SP5, 2011 SP5 Patch 1. Assume the non-customized installation programs are located in the following directory structure:

```
C:\productUpdates\2011\2011 Full\SP4\SP5\SP5 Patch 1\n```

Set the `baselinePath` parameter to:

```
baselinePath=C:\productUpdates\2011\n```
6.9 IDs and codes for Crystal Reports customization

The following section contains a list of all the IDs and codes you can use to customize the installation program:

- Feature IDs
- Shortcut deployment unit IDs
- String IDs
- Language codes
- Installation screen and property IDs

6.9.1 Feature IDs

Use feature IDs in the `removeFeature` element to remove features and their components from the installation program and the installed product.

For example, this ID will remove support for displaying relationships between data and geographic regions:

```xml
<removeFeature id="Mapping"/>
```

You can remove features for the following components:

- Data access
- Enterprise system integration
- Export support
- Other

Table 21: Data access

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataAccess</td>
<td>Data access</td>
</tr>
<tr>
<td>Access</td>
<td>Microsoft Access</td>
</tr>
<tr>
<td>ADO.NET</td>
<td>ADO.NET</td>
</tr>
<tr>
<td>BDE</td>
<td>IDAPI Database DLL</td>
</tr>
<tr>
<td>Btrieve</td>
<td>Pervasive Database Driver (Btrieve)</td>
</tr>
<tr>
<td>COMData</td>
<td>COM Data Provider</td>
</tr>
<tr>
<td>Comm_Rep</td>
<td>Command in repository</td>
</tr>
<tr>
<td>DB2</td>
<td>IBM DB2</td>
</tr>
<tr>
<td>dBase</td>
<td>dBase</td>
</tr>
<tr>
<td>Feature ID</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Exchange</td>
<td>Microsoft Exchange</td>
</tr>
<tr>
<td>FieldDefinitions</td>
<td>Field Definitions</td>
</tr>
<tr>
<td>FileSystem</td>
<td>File System</td>
</tr>
<tr>
<td>HPNeoview</td>
<td>HP Neoview</td>
</tr>
<tr>
<td>Informix</td>
<td>Informix</td>
</tr>
<tr>
<td>JavaData</td>
<td>Java Data Provider</td>
</tr>
<tr>
<td>JDBC</td>
<td>JDBC (JNDI) Data Driver</td>
</tr>
<tr>
<td>MicrosoftOutlook</td>
<td>Microsoft Outlook</td>
</tr>
<tr>
<td>MyCube</td>
<td>OLAP Cube</td>
</tr>
<tr>
<td>MYSQL_DataAccess</td>
<td>MySQL</td>
</tr>
<tr>
<td>NCRTeradata</td>
<td>NCRTeradata</td>
</tr>
<tr>
<td>NETEZZA</td>
<td>NETEZZA</td>
</tr>
<tr>
<td>NTEventLog</td>
<td>NT Event Log</td>
</tr>
<tr>
<td>OLE_DB_Data</td>
<td>OLE DB Data</td>
</tr>
<tr>
<td>OptionalDataDirectODBC</td>
<td>DataDirect ODBC</td>
</tr>
<tr>
<td>Oracle</td>
<td>Oracle</td>
</tr>
<tr>
<td>Progress.OpenEdge</td>
<td>Progress OpenEdge</td>
</tr>
<tr>
<td>RDO</td>
<td>ODBC RDO</td>
</tr>
<tr>
<td>SFORCE</td>
<td>Salesforce.com Driver</td>
</tr>
<tr>
<td>SIEBEL</td>
<td>Siebel</td>
</tr>
<tr>
<td>Sybase</td>
<td>Sybase</td>
</tr>
<tr>
<td>SymantecACT</td>
<td>ACT!</td>
</tr>
<tr>
<td>Universe</td>
<td>BusinessObjects Universe</td>
</tr>
<tr>
<td>UWSC</td>
<td>Universal Web Services Connector</td>
</tr>
<tr>
<td>WebActivityLog</td>
<td>Web Activity Log</td>
</tr>
<tr>
<td>XML</td>
<td>XML Driver</td>
</tr>
</tbody>
</table>

Table 22: Enterprise system integration

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IntegrationOptions</td>
<td>Integration options</td>
</tr>
<tr>
<td>EBS</td>
<td>Oracle E-Business Suite</td>
</tr>
<tr>
<td>JDE</td>
<td>JD Edwards EnterpriseOne</td>
</tr>
<tr>
<td>PSFT</td>
<td>Peoplesoft Enterprise</td>
</tr>
<tr>
<td>Feature ID</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>SAP</td>
<td>SAP Solutions</td>
</tr>
<tr>
<td>SIEBEL</td>
<td>Siebel</td>
</tr>
</tbody>
</table>

Table 23: Export support

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Export to an application</td>
</tr>
<tr>
<td>CharacterSeparated</td>
<td>Character Separated format</td>
</tr>
<tr>
<td>CrystalReports</td>
<td>Crystal Reports format</td>
</tr>
<tr>
<td>DiskFile</td>
<td>Export to a file</td>
</tr>
<tr>
<td>Excel</td>
<td>Microsoft Excel 97-2003 format</td>
</tr>
<tr>
<td>ExchangeFolders</td>
<td>Exchange Folder</td>
</tr>
<tr>
<td>HTML</td>
<td>HTML 3.2 and HTML 4.0 (DHTML) formats</td>
</tr>
<tr>
<td>LegacyXMLExport</td>
<td>Legacy XML format</td>
</tr>
<tr>
<td>LotusNotes</td>
<td>Lotus Notes document</td>
</tr>
<tr>
<td>LotusNotesMail</td>
<td>Lotus Domino</td>
</tr>
<tr>
<td>ODBC</td>
<td>Export to any of your installed ODBC formats</td>
</tr>
<tr>
<td>PDF</td>
<td>PDF format</td>
</tr>
<tr>
<td>Record</td>
<td>Record Style format</td>
</tr>
<tr>
<td>ReportDefinition</td>
<td>Report Definition format</td>
</tr>
<tr>
<td>RichTextFormat</td>
<td>Rich Text Format</td>
</tr>
<tr>
<td>Text</td>
<td>Text formats</td>
</tr>
<tr>
<td>WordforWindows</td>
<td>Microsoft Word 97-2003 format</td>
</tr>
<tr>
<td>XMLExport</td>
<td>XML document</td>
</tr>
</tbody>
</table>

Table 24: Other

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CrystalReportsRoot</td>
<td>Crystal Reports 2011</td>
</tr>
<tr>
<td>crw</td>
<td>Crystal Reports Designer</td>
</tr>
<tr>
<td>Mapping</td>
<td>Geographic mapping</td>
</tr>
<tr>
<td>MicrosoftMail</td>
<td>Microsoft Mail Destination</td>
</tr>
<tr>
<td>PGEditor</td>
<td>Custom charting</td>
</tr>
<tr>
<td>UploadWizard</td>
<td>Report Upload Wizard</td>
</tr>
</tbody>
</table>
Related Information

Removing features [page 122]

6.9.2 Shortcut deployment unit IDs

Use the deployment unit IDs in the shortcut element to change the location and name of the program shortcuts in the Windows Start menu.

Table 25: Shortcut deployment unit IDs

<table>
<thead>
<tr>
<th>Shortcut deployment unit ID</th>
<th>Shortcut target</th>
</tr>
</thead>
<tbody>
<tr>
<td>product.crystalreports.shortcut.crw-4.0-core</td>
<td>Crystal Reports 2011</td>
</tr>
<tr>
<td>product.crystalreports.shortcut.odbc-4.0-core</td>
<td>ODBC Data Source Administrator</td>
</tr>
<tr>
<td>product.crystalreports.shortcut.rptpub-wiz-4.0-core</td>
<td>Report Upload Wizard</td>
</tr>
</tbody>
</table>

Related Information

Customizing the Windows Start menu shortcuts [page 115]

6.9.3 String IDs

You can change the value of all strings in the installation program. You can replace a string for all languages or for a specific language. Use the replaceString element, for example:

<replaceString id="product.cr_name" value="Custom Company Crystal Reports lang="all"/>

Table 26: Commonly changed strings

<table>
<thead>
<tr>
<th>String ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>product.cr_name</td>
<td>Product long name</td>
</tr>
<tr>
<td>product.cr_shortname</td>
<td>Product short name</td>
</tr>
<tr>
<td>product_cr_version</td>
<td>Product version</td>
</tr>
<tr>
<td>product_cr_majorversion</td>
<td>Product major version</td>
</tr>
</tbody>
</table>
Related Information

Customizing the product name and version number [page 113]

### 6.9.4 Language codes

The SAP BusinessObjects customization tool uses these language codes to represent supported languages:

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>EN</td>
</tr>
<tr>
<td>Czech</td>
<td>CS</td>
</tr>
<tr>
<td>Danish</td>
<td>DA</td>
</tr>
<tr>
<td>Dutch</td>
<td>NL</td>
</tr>
<tr>
<td>Finnish</td>
<td>FI</td>
</tr>
<tr>
<td>French</td>
<td>FR</td>
</tr>
<tr>
<td>German</td>
<td>DE</td>
</tr>
<tr>
<td>Hungarian</td>
<td>HU</td>
</tr>
<tr>
<td>Italian</td>
<td>IT</td>
</tr>
<tr>
<td>Japanese</td>
<td>JA</td>
</tr>
<tr>
<td>Korean</td>
<td>KO</td>
</tr>
<tr>
<td>Norwegian Bokmal</td>
<td>NB</td>
</tr>
<tr>
<td>Polish</td>
<td>PL</td>
</tr>
<tr>
<td>Portuguese</td>
<td>PT</td>
</tr>
<tr>
<td>Romanian</td>
<td>RO</td>
</tr>
<tr>
<td>Russian</td>
<td>RU</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>zh_CN</td>
</tr>
<tr>
<td>Slovak</td>
<td>SK</td>
</tr>
<tr>
<td>Spanish</td>
<td>ES</td>
</tr>
<tr>
<td>Swedish</td>
<td>SV</td>
</tr>
<tr>
<td>Thai</td>
<td>TH</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>zh_TW</td>
</tr>
<tr>
<td>Turkish</td>
<td>TR</td>
</tr>
</tbody>
</table>
6.9.5 Installation screen and property IDs

Use the installation screen IDs in the `removeDialog` element to remove screens from the installation program. For example, use this element to remove the `Select Features` screen:

```
<removeDialog id="SelectFeatures.dialog"/>
```

Use the property IDs and the property values to prepopulate user input. For example, use this element to set the default installation type to `custom`:

```
<replaceProperty id="InstallType" defaultValue="custom"/>
```

<table>
<thead>
<tr>
<th>Title of installation screen</th>
<th>Installation screen ID</th>
<th>Property IDs in installation screen</th>
<th>Property values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please choose a setup language</td>
<td>SelectUILanguage.dialog</td>
<td>SortedAvailableSetupLanguages</td>
<td>Set of language codes representing supported setup languages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetupUILanguage</td>
<td>Single language code representing the setup language</td>
</tr>
<tr>
<td>Install cannot proceed</td>
<td>SharedAlwaysFailure.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Choose Install Type</td>
<td>ChooseInstallType2.dialog</td>
<td>InstallType</td>
<td>• default (Typical) • custom</td>
</tr>
<tr>
<td>Prerequisite check</td>
<td>CheckPreRequisites.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Welcome to the installation wizard....</td>
<td>ShowWelcomeScreen.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>License Agreement</td>
<td>ShowLicenseAgreement.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>User Information</td>
<td>CREnterProductKey.dialog</td>
<td>RegisteredUser</td>
<td>Your Username</td>
</tr>
<tr>
<td>Title of installation screen</td>
<td>Installation screen ID</td>
<td>Property IDs in installation screen</td>
<td>Property values</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------</td>
<td>-------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RegisteredCompany</td>
<td>Your Company name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProductKey</td>
<td>Your Product key-code</td>
</tr>
<tr>
<td>Specify the Destination Folder</td>
<td>ChooseInstallDir.dialog</td>
<td>InstallDir</td>
<td>Filepath of the installation folder</td>
</tr>
<tr>
<td>Choose Language Packs</td>
<td>SelectLanguagePack.dialog</td>
<td>SelectedLanguage-Packs</td>
<td>Array of language codes</td>
</tr>
<tr>
<td>Select Features</td>
<td>SelectFeatures.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>SAP Crystal Reports 2011 has been successfully installed</td>
<td>ShowInstallComplete.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>SAP Crystal Reports 2011 has been successfully installed</td>
<td>ShowInstallComplete_PatchUpdate.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Start Installation</td>
<td>ShowInstallSummary.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Uninstall Confirmation</td>
<td>VerifyToRemove.dialog</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>
| Web Update Service Option  | ShowPrivacyStatement.dialog | DisableWebUpdateService             | ● 0 (Enable Web Update Service)  
● 1 (Disable Web Update Service) |
| SAP Crystal Reports 2011 has been successfully uninstalled | ShowUninstallComplete.dialog | Not applicable                      | Not applicable |