Customizing the Pentaho BI Suite

Help and Support Resources

If you have questions that are not covered in this guide, or if you would like to report errors in the documentation, please contact your Pentaho technical support representative.

Support-related questions should be submitted through the Pentaho Customer Support Portal at http://support.pentaho.com.

For information about how to purchase support or enable an additional named support contact, please contact your sales representative, or send an email to sales@pentaho.com.

For information about instructor-led training on the topics covered in this guide, visit http://www.pentaho.com/training.

Limits of Liability and Disclaimer of Warranty

The author(s) of this document have used their best efforts in preparing the content and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, express or implied, with regard to these programs or the documentation contained in this book.

The author(s) and Pentaho shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims.

Trademarks

Pentaho (TM) and the Pentaho logo are registered trademarks of Pentaho Corporation. All other trademarks are the property of their respective owners. Trademarked names may appear throughout this document. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, Pentaho states that it is using the names for editorial purposes only and to the benefit of the trademark owner, with no intention of infringing upon that trademark.

Company Information

Pentaho Corporation
Citadel International, Suite 340
5950 Hazeltine National Drive
Orlando, FL 32822
Phone: +1 407 812-OPEN (6736)
Fax: +1 407 517-4575
http://www.pentaho.com

E-mail: communityconnection@pentaho.com

Sales Inquiries: sales@pentaho.com

Documentation Suggestions: documentation@pentaho.com

Sign-up for our newsletter: http://community.pentaho.com/newsletter/

Contents

Customizing the Pentaho User Console

The Pentaho User Console is the standard Web interface for the Pentaho BI Server, and includes interactive elements of Pentaho Reporting and Analysis. The Pentaho Enterprise Console is the central framework through which your BI Server is configured and managed. You can customlize the look and feel of the User Console by editing its configuration files, graphics, and CSS3 style sheets manually; and both the User Console and Enterprise Console can be translated into other languages by replacing plain-text message bundles.

This document provides guidance and instructions for customizing the Pentaho User Console and Pentaho Enterprise Console; it is designed for Pentaho consultants, partners, and customers who have in-house localization resources or Web development and/or design skills.

Overview of Customizable Configuration Files

Note: Stop the Pentaho application server before editing anything inside of the Pentaho WAR file.

MantleSettings.properties

Location: /pentaho/server/biserver-ee/tomcat/webapps/pentaho/WEB-INF/lib/mantle-3.9.0-GA.jar

This file is stored in the above-listed JAR file in the Pentaho WAR. You must open the JAR with a ZIP archive utility, then navigate to the `/org/pentaho/mantle/server/` directory inside of it. No matter how you decide to extract and edit the file, it must be put back in the JAR when you are done. Pentaho recommends using **7-Zip** for this process because it can go into the JAR, launch your preferred editing program, then do an in-place update of the archive.

The MantleSettings.properties file contains settings for the menubar, main toolbar, and the logo panel. If you decide to turn off the menubar or toolbar, the logo panel will also be turned off because of UI constraints. Look for settings in the file such as **show-menu-bar=true**. To toggle the behavior, simply change the **true** to **false**.

MantleStyle.css

Location: /pentaho/server/biserver-ee/tomcat/webapps/pentaho/mantle/MantleStyle.css

This is the structural cascading style sheet for the Pentaho User Console. It inherits some elements from the **Widgets.css** file in the same directory, so you may need to look at that as well.

Note: Modifying these styles could have dramatic impact on Pentaho User Console rendering. This is probably not the style sheet you are looking for.

Theme-specific local and global style sheets for PUC

Note: The Slate theme is included in the BI Server, but disabled by default because it has rendering issues in Internet Explorer. However, you may find it useful for development and testing purposes.

Location: /pentaho-solutions/system/common-ui/resources/themes/onyx/globalOnyx.css

This is the main structural, theme-specific style sheet for the default theme for the Pentaho User Console (Onyx).

Location: /pentaho-solutions/system/common-ui/resources/themes/onyx/globalSlate.css

This is the main structural, theme-specific style sheet for an alternate theme for the Pentaho User Console (Slate).

Location: /pentaho/mantle/themes/onyx/mantleOnyx.css

This is the customizable presentation portion of the theme style sheet for the default theme for the Pentaho User Console (Onyx).

Location: /pentaho/mantle/themes/slate/mantleSlate.css

This is the customizable presentation portion of the theme style sheet for an alternate theme for the Pentaho User Console (Onyx).

Product-specific theme settings for Analyzer, Dashboard Designer, and Interactive Reporting

These directories contain style sheets and other theme materials for each BI Server client tool.

Analyzer location: /biserver-ee/pentaho-solutions/system/analyzer/styles/themes/

Interactive Reporting location: /biserver-ee/pentaho-solutions/system/pentaho-interactive-reporting/resources/web/themes/

Dashboard Designer location: /biserver-ee/pentaho-solutions/system/dashboards/themes/

Replacing the Default Icon Set

The icons used throughout the Pentaho User Console are consolidated and stored in PNG files called **image bundles**. These files only encompass a single HTTP request from the client to the server for the entire set of images, thereby saving bandwidth and improving performance. The image bundles all have machine-generated filenames with PNG extensions, and are stored in the `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/mantle` directory. An example of a likely image bundle name is: **EAC2E33CD4B780B5A88A44308412F1CB.cache.png**.

If you edit an image bundle, you must be sure to preserve its location and size. The Pentaho Web application uses CSS to chop the image bundle up during runtime, so you must respect the size, position, and location of what is currently there. Because of these constraints, Pentaho recommends that you do not alter the images themselves, but instead shift the color palette to fit your brand specifications. This can be done with most modern graphic design tools.

Changing the Default Logo

The Pentaho logo displayed in the top right corner of the Pentaho User Console may be replaced or disabled to suit your needs. To turn off the logo, refer to the **MantleSettings.properties** explanation above.

The default logo is: `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/mantle/themes/onyx/images/logo.png` (change the name of the theme from **onyx** to **slate**, or whatever your custom theme is, if you aren't using Onyx). The default Pentaho logo has a transparent background, so you will see through it to the background, whatever it may be. If you want to modify or replace the logo, you must maintain the file name and image size, which is **152x75** pixels.

If you must use a different size graphic for the logo, you can change the **.puc-logo-spacer** class in **mantleOnyx.css** (or the Slate equivalent, if that's what you're using) to use different dimensions:

```
.puc-logo-spacer {
. width: 152px;
. height: 75px;
}
```

Changing the logo link

You can make the logo go to a given URL when clicked. In its default state, the Pentaho logo doesn't go to a URL, but the functionality to make it clickable is in place. You can set the destination URL via the **logoPanelWebsite** entry in the `/pentaho/mantle/messages/mantleMessages.properties` file. This properties file is inside of the Pentaho WAR, so you will have to stop the BI Server before making any modifications.

Changing the Workspace Background

You can change the background image that appears in the content pane in the Pentaho User Console by modifying or replacing this file: `/pentaho/mantle/style/core/images/bg_pentaho_default.png`. The dimensions and filename must be identical.

Switching PUC Themes

If you've created an alternate theme and/or localized message bundle, you can switch to it through the **View** menu in the Pentaho User Console.

Note: You can prevent themes from appearing in this menu by adding a **hidden="true"** property to the theme node.

Testing Your Design Changes

The information in this section assists developers and designers in testing changes to themes, graphics, and other custom modifications to the BI Server.

Using the PUC UI Widget Test Page

Pentaho provides a hidden test page to assist theme designers and interface developers with testing (change the hostname, port number, and context to match your deployment):

```
http://localhost:8080/pentaho/content/common-ui/resources/web/test/style-samples.html
```

From this page, you can safely click on all of the buttons and dialogues that are included in your themes without the overhead of logging into the Pentaho User Console and navigating through various content pages and client tools to get to each widget.

Using Firebug to Test CSS Changes

The style sheet files explained earlier in this guide contain many style definitions, which makes it difficult to map the CSS classes to what you see in the application at runtime. To make this task easier, Pentaho recommends that you use the Firebug extension for Firefox to inspect the UI elements you want to style. Once you've selected an element, you can view all of the CSS properties that apply to it, then change the CSS and watch your changes take immediate effect. This allows you to quickly and easily find the appropriate CSS to edit and gives you the ability to preview your changes in real time.
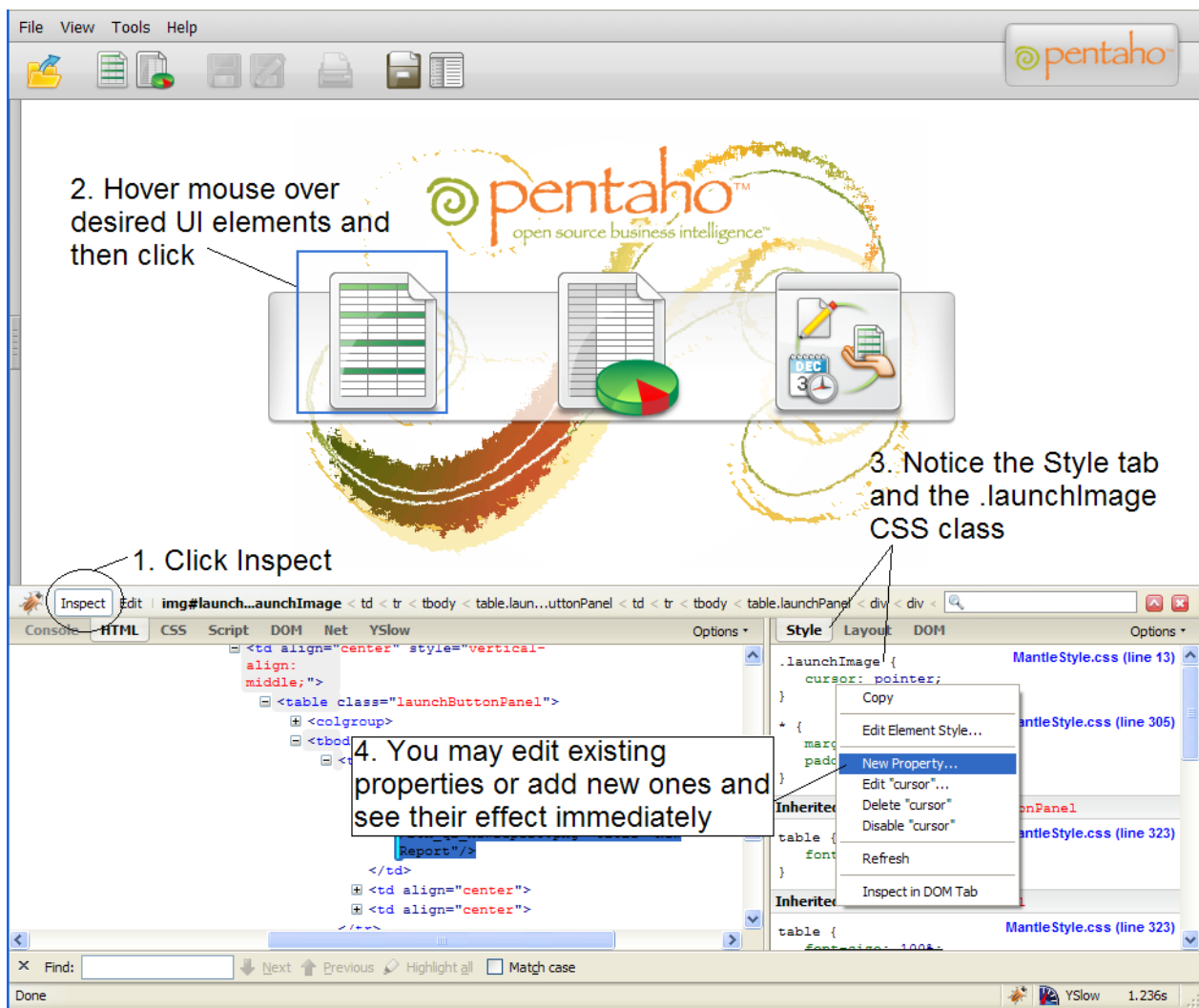
Note: Both Chrome and Safari come with built-in developer tools that may be just as effective as Firebug for this purpose.

Removing or Changing Support Messages

The information below explains how to remove the warning messages at the bottom of the ad hoc Reporting and JPivot client tools.

Removing the Support Warning Message From JPivot

Follow this process to remove the following message from the bottom of the JPivot screen: **"JPivot has been replaced by Pentaho Analyzer. It is provided as a convenience but will no longer be enhanced or officially supported by Pentaho."**

1. Shut down the BI Server if it is currently running.
2. Edit the `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/jsp/pivot.jsp` file.
3. Search for **deprecatedWarning** to locate the warning message entry.
4. Put SGML comment tags around the deprecatedWarning div tag as shown in the example below.
5. Save and close the file.
6. Start the BI Server.

The warning message will no longer appear at the bottom of the JPivot screen.

```
<!-- <div id="deprecatedWarning" style="margin: auto; width:
 100%">
  <table width="580px" align="center" style="background-color:
 #fffdd5; border-style: solid; border-color: #dcb114; border-
width= 1px; font: normal .85em Tahoma, 'Trebuchet MS', Arial">
    <tr>
      <td>
        <img src="./jpivot/navi/warning.png"/>
      </td>
      <td>
        JPivot has been replaced by Pentaho Analyzer.<br/>
        It is provided as a convenience but will no longer be
 enhanced or offically supported by Pentaho.
      </td>
    </tr>
  </table>
</div> -->
```

Removing the Support Warning Message From WAQR

Follow this process to remove the following message from the bottom of the ad hoc Reporting screen:
"Web Ad Hoc Query and Reporting has been replaced by the new Interactive Reporting client. It is provided as a convenience but will no longer be enhanced or officially supported by Pentaho."

1. Shut down the BI Server if it is currently running.
2. Edit the `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/adhoc/waqr.html` file.
3. Search for **waqrDeprecatedAlert** to locate the warning message entry.
4. Put SGML comment tags around the waqrDeprecatedAlert td tag as shown in the example below.
5. Save and close the file.
6. Start the BI Server.

The warning message will no longer appear at the bottom of the ad hoc Reporting screen.

```
<!-- <td valign="middle" id="waqrDeprecatedAlert" style="padding-
right: 5px">
        <table width="580px" style="background-color: #fffdd5;
 border-style: solid; border-color: #dcb114; border-width= 1px;
 font: normal .85em Tahoma, 'Trebuchet MS', Arial">
          <tr>
            <td>
              <img src="images/warning.png"/>
            </td>
            <td>
            Web Ad Hoc Query and Reporting has been replaced by
 the new Interactive Reporting client.<br/>
              It is provided as a convenience but will no longer be
 enhanced or offically supported by Pentaho.
            </td>
          </tr>
        </table>
</td> -->
```

Switching to a Production Login Screen

The default Pentaho User Console login screen contains information and instructions for evaluators. Pentaho allows this extra information to be hidden via configuration; follow the directions below to hide the extra information.

1. Shut down the BI Server and Enterprise Console if they are currently running.
2. Open a terminal or file browser window and navigate to the `/pentaho-solutions/system/` directory.

   If you installed via the archive package or the graphical installer, the full path is `/pentaho/server/biserver-ee/pentaho-solutions/system/`.
3. Edit the **pentaho.xml** file.
4. Find the **login-show-sample-users-hint** node and change its value to **false**.
5. Start the BI Server and Pentaho Enterprise Console.

You now have a login screen that has been scrubbed of evaluation content.

Internationalization of Message Bundles

This section explains how to localize all Pentaho applications that abstract their integrated textual content to message bundles. Pentaho supports dynamic message bundle switching according to browser locale, so you can create localized message bundles for every language you want to support, and let each individual user's system language settings determine which one is loaded.

Replacing Or Translating BI Server Message Bundles

Note: Stop the Pentaho application server before editing anything inside of the Pentaho WAR file.

You can internationalize the Pentaho User Console, Pentaho Analyzer, and Dashboard Designer by creating locale- and language-specific message bundles within the Pentaho Web application.

For brevity's sake, only the default Pentaho User Console files will be explained in detail. The file naming convention is identical among all message bundles in the Pentaho BI Suite. The following files are located in the /mantle/messages/ directory:

- **mantleMessages.properties**: The default message bundle for the Pentaho User Console. In its initial condition, it is a copy of **mantleMessages_en.properties**. If you want to change the default language and dialect, copy your preferred message bundle file over this one.
- **mantleMessages_en.properties**: The English-language version of the standard message bundle. This is an identical copy of **mantleMessages.properties**.
- **mantleMessages_fr.properties**: The French-language version of the standard message bundle.
- **mantleMessages_de.properties**: The German-language version of the standard message bundle.
- **mantleMessages_supported_languages.properties**: Contains a list of localized message bundles and the native language names they correspond to; this relieves the BI Server of the burden of having to discover them on its own. A supported_languages.properties file should be created for every message bundle that you intend to localize.

Example of mantleMessages_supported_languages.properties

```
en=English
de=Deutsch
fr=Français
```

New files are created in the following format: **mantleMessages_xx_YY.properties** where **xx** represents a lowercase two-letter language code, and **YY** represents a two-letter locale code, where applicable. So, for instance, U.S. and British English could have two separate message bundles if you wanted to draw a distinction between the two dialects:

- mantleMessages_en_US.properties
- mantleMessages_en_GB.properties

The language and country codes must be in standard ISO format. You can look up both sets of codes on these pages:

- *http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes*
- *http://www.iso.org/iso/english_country_names_and_code_elements*

You can edit the default message bundle directly if you need to make surgical changes to the content inside of it. If you plan to translate it into another language, it makes more sense to copy the file and change the name appropriately, then translate it line by line. Be sure to update supported_languages.properties with the new country and dialect code and the native language name that it corresponds to.

Note: All message bundles must be UTF-8 encoded.

Pentaho User Console Message Bundles

The message bundle locations for the Pentaho User Console and Dashboard Designer are:

- `/pentaho.war/mantle/messages/` for the Pentaho User Console
- `/pentaho/server/biserver-ee/pentaho-solutions/system/dashboards/resources/messages/` for Dashboard Designer
- `/pentaho/server/biserver-ee/pentaho-solutions/system/dashboards/resources/gwt/messages/` also for Dashboard Designer

Pentaho Analyzer and JPivot require a special set of instructions for localizing their analysis schema message bundles. See *Localization/Internationalization of Analysis Schemas* on page 11 for more information.

👉 **Note:** Dashboard Designer requires both sets of message bundles, and both are different, so you cannot simply copy one set of messages from one directory to the other.

Localization/Internationalization of Analysis Schemas

You can create internationalized message bundles for your analysis schemas and deploy them with the Pentaho Web application. This enables Pentaho Analyzer and JPivot to access localized schemas.

1. Edit your analysis schema and tokenize all values that you want to localize.
   Typically you would create variables for all caption and description values.

```
<Schema measuresCaption="%{foodmart.measures.caption}">
    <Dimension name="Store" caption="%{foodmart.dimension.store.caption}"
description="%{foodmart.dimension.store.description}">
        <Hierarchy hasAll="true" allMemberName="All Stores"
allMemberCaption="%{foodmart.dimension.store.allmember.caption =All
Stores}" primaryKey="store_id"
caption="%{foodmart.hierarchy.store.country.caption}"
description="%{foodmart.hierararchy.store.country.description}>
            <Table name="store"/>
            <Level name="Store Country" column="store_country"
uniqueMembers="true" caption="%{foodmart.dimension.store.country.caption}"
description="%{foodmart.dimension.store.country.description}"/>
```

2. Stop the BI Server.
3. Edit the **mondrian.properties** file in the `/pentaho/server/biserver-ee/pentaho-solutions/system/mondrian/` directory and add this line (or modify it if it's already there):

```
mondrian.rolap.localePropFile=com.pentaho.messages.MondrianMessages
```

4. Save and close the file.
5. Edit your `/pentaho/server/biserver-ee/pentaho-solutions/system/olap/datasources.xml` file and modify the **Provider** line to declare the default locale, include the i18n dynamic schema processor (this enables message token replacement), and refresh the schema when Analyzer detects that the checksum is different than the cached version:

```
Provider=mondrian; Locale=en_US; DynamicSchemaProcessor=
mondrian.i18n.LocalizingDynamicSchemaProcessor; UseContentChecksum=true;
Jdbc= jdbc:odbc:MondrianFoodMart; Catalog= /WEB-INF/FoodMart.xml
```

6. Save and close the file.
7. Create localized **MondrianMessages.properties** files in the `/WEB-INF/classes/com/pentaho/messages/` directory inside of the Pentaho WAR, and define each token you used in the analysis schema.

👉 **Note:** JBoss users will have to delete the unpacked Pentaho WAR directory if it exists, then unpack the pentaho.war file with an archive utility, create the message bundles in the proper location, then repack it into a WAR again.

If you need further assistance in creating localized message bundles, refer to *Replacing Or Translating BI Server Message Bundles*.

```
foodmart.measures.caption=Measures
```

```
foodmart.dimension.store.country.caption=Store Country
foodmart.dimension.store.name.property_type.column=store_type
foodmart.dimension.store.country.member.caption=store_country
foodmart.dimension.store.name.property_type.caption=Store Type
foodmart.dimension.store.name.caption=Store Name
foodmart.dimension.store.state.caption=Store State
foodmart.dimension.store.name.property_manager.caption=Store Manager
foodmart.dimension.store.name.property_storesqft.caption=Store Sq. Ft.
foodmart.dimension.store.allmember.caption=All Stores
foodmart.dimension.store.caption=Store
foodmart.cube.sales.caption=Sales
foodmart.dimension.store.city.caption=Store City
foodmart.cube.sales.measure.unitsales=Unit Sales
```

8. Start the BI Server.

Your analysis schemas will now be localized to whatever language is currently selected in the Pentaho User Console, if a message bundle for that locale was copied to the proper directory as explained above.

Switching PUC Message Bundles

If you've created a localized message bundle, you can switch to it through the **View** menu in the Pentaho User Console.

Pentaho Enterprise Console Message Bundles

The Pentaho Enterprise Console message bundles are stored in the `/pentaho/server/enterprise-console/www/com.pentaho.pac.PentahoProAdminConsole/messages/` directory. There are two files that you must modify to localize the Enterprise Console:

- messages.properties
- promessages.properties

> **Note:** The same language and locale two-letter naming conventions apply to promessages.properties.

Action Sequence Message Bundles

The message bundle for action sequence error messages is `/pentaho/WEB-INF/classes/org/pentaho/platform/engine/services/messages/messages.properties`.

Changing the Default Character Set Encoding for the Pentaho Enterprise Console

You can use your own character set encoding by editing the **console.properties** file located under `...\enterprise-console\resource\config`. The property name is **content.characterset** and it is currently set to **UTF-8**. If you have the platform set to any other character encoding, you must change this value for the Pentaho Enterprise Console to work correctly.

In addition to editing the console.properties file, you must pass a JVM parameter in the start script, (`start-pec.bat/.sh`), of the Pentaho Enterprise Console, (`-Dfile.encoding="UTF-8"`). This must match the platform's character encoding.

Creating Design Tool Templates

The following Pentaho products have customizable templates:

- Report Designer (Report Design Wizard)
- Interactive Reporting
- Dashboard Designer
- Community Dashboard Framework (CDF)

This section explains how to modify the default templates and, where applicable, how to create your own.

Creating a Dashboard Designer Template

Follow the directions below to create a new Pentaho Dashboard Designer template.

Note: You will have to migrate this template by hand if/when you upgrade Dashboard Designer, since the template files will be stored in the Dashboard Designer plugin directory. The upgrade procedure that Pentaho provides does not typically cover customizations like this one, except to mention that they must be migrated by hand.

1. Stop the BI Server.
2. Navigate to the `/pentaho/server/biserver-ee/pentaho-solutions/system/dashboards/templates/xul/` directory.
3. If you want to remove all templates that you know will never be used, you can safely delete their corresponding XUL files now. When you are done, also remove the corresponding files from the **html** sibling directory.
4. Copy the existing XUL file that most closely resembles the template layout you want to create, giving the new XUL file a name that starts with a two-digit number that represents the template's order in the hierarchy, followed by a short description of its dimensions.
   All Dashboard Designer templates follow this naming convention.
5. Create a **.properties** file that corresponds to the one you just copied in the previous step, and put one item in it: **name=Description here**, where "Description here" represents the display name of this template.
6. Create a thumbnail graphic that fits the same dimensions as the other PNG thumbnails in this directory, and give it the same name as the previous two files, with a PNG extension.
   You should now have three new files, all with the same name, with three different extensions: .xul, .properties, and .png.
7. Edit the new **.xul** file and change the **box** attributes to match your template specifications.

   A **vbox** node creates a column; an **hbox** node creates a row; a **box** element defines an individual panel in each row.

   height and **width** define static widths in pixels; the **flex** size attribute defines a percentage of the total width of the dashboard. If you'd like more extensive definitions of XUL elements, refer to the official XUL documentation: *http://developer.mozilla.org/en/XUL_Reference*.

   Note: Ensure that each box, vbox, and hbox node has its own unique **id**.

8. Save and close all open files, then start the BI Server.
9. Test your new template and adjust its configuration accordingly.

You now have a custom Dashboard Designer template deployed to your BI Server.

You must copy the template files by hand if you upgrade the BI Server or Dashboard Designer in the future. You may want to back up your custom templates to a safe location right now just in case you forget to copy them over during a future upgrade.

Reporting Templates

Both the Report Design Wizard and Interactive Reporting use templates for rapid report creation. The template creation process is similar for both RDW and PIR -- in both cases, you use Pentaho Report Designer to create a report with certain template options selected, then deploy it to the client tools that will use it. The instructions below explain how to create templates, how reports are generated from templates, and how to display templates within RDW and PIR.

Report Design Wizard Template Design Guidelines

Methodology

To create a new RDW template, you must use Report Designer to create a report with certain template-specific properties enabled. The resultant PRPT file is then deployed to the Report Designer and/or Pentaho Data Integration template directory.

Requirements

The **generated-content-marker** attribute in the **wizard** group is the flag that turns a normal report into a template. This attribute can only be applied to a band (group header, group footer, details header, details footer, details, or sub-band).

RDW will insert its auto-generated content into the first band with the **generated-content-marker** set to true. This applies to the group header, group footer, details header, details footer, and the details bands. In the event there are more groups defined in the Report Design Wizard than defined in the template, it repeats the last defined group header and footer in the template.

Formatting Inheritance

Formatting styles are inherited, so any formatting applied to a band will also be applied to the elements used within it. Formatting is applied in three ways and in the following order:

1. Through the template via band inheritance
2. Through query data where it is defined in Pentaho Metadata
3. As defined by RDW users through the RDW interface

Inheriting Styles from the Data Query

The **query-metadata** section of the Attributes tab contains options that determine whether formatting styles can come from the data query and be applied to the detail header, details, or detail footer band. This must be set directly on the detail header, detail footer, or details band; and the **style-format** option must be set to **true** for it to work. You must also disable any individual formatting styles (enable-style-*=true) that you don't want to come from the query.

Padding and Grid Lines

Since the Details band is dynamically generated, you have to specify grid line and padding settings in the template definition. This is done through the **wizard** attribute group for the band that has the **generated-content-marker** enabled.

Updating

An RDW template is only a set of initial defaults for a report, so if a template is updated, completed reports that were based on that template will not be affected; there is no connection between the template and the report once the report is saved. If you want to update an RDW-based report to reflect template changes, you can edit the report with Report Design Wizard, make any necessary selections, and re-save it.

> **Note:** Interactive Reporting templates have the opposite behavior because the report links itself to the template; when an IR template is changed, all reports based on that template will automatically inherit the updated template.

Interactive Reporting Template Design Guidelines

Methodology

To create a new Interactive Reporting template, you must use Report Designer to create a report with certain template-specific properties enabled. The resultant PRPT file is then deployed to the IR plugin's template directory.

Requirements

The **generated-content-marker** attribute in the **wizard** group is the flag that turns a normal report into a template. This attribute can only be applied to a band (group header, group footer, details header, details footer, details, or sub-band).

RDW will insert its auto-generated content into the first band with the **generated-content-marker** set to true. This applies to the group header, group footer, details header, details footer, and the details bands. In the event there are more groups defined in the Report Design Wizard than defined in the template, it repeats the last defined group header and footer in the template.

Formatting Inheritance

Formatting styles are inherited, so any formatting applied to a band will also be applied to the elements used within it. Formatting is applied in three ways and in the following order:

1. Through the template via band inheritance
2. Through query data where it is defined in Pentaho Metadata
3. As defined by IR users through the Interactive Reporting interface

Inheriting Styles from the Data Query

The **query-metadata** section of the Attributes tab contains options that determine whether formatting styles can come from the data query and be applied to the detail header, details, or detail footer band. This must be set directly on the detail header, detail footer, or details band; and the **style-format** option must be set to **true** for it to work. You must also disable any individual formatting styles (enable-style-*=true) that you don't want to come from the query.

Padding and Grid Lines

Since the Details band is dynamically generated, you have to specify grid line and padding settings in the template definition. This is done through the **wizard** attribute group for the band that has the **generated-content-marker** enabled.

Updating

A template is not just a set of initial defaults for a report -- it is the basis for that report. So if a template is updated, completed reports that were based on that template will also change.

> **Note:** Report Design Wizard templates have the opposite behavior; when an RDW template is changed, none of the reports based on that template will be automatically be updated with those changes. Instead, you will have to edit each report, apply the new template, and save it.

Dynamic Element Positioning in Templates

Use the following methods to accommodate for multiple page sizes in reports:

Percentages

Express the height, width, x position, and y position in percentages.

Block, inline, or row

Switch the band's layout mode from canvas to either block, inline or row:

layout-mode	value
canvas	Uses the x and y position to place the element in the band.
block	Stacks elements vertically according to the layer order in a band; width is set to 100%.
inline	Stacks elements horizontally according to the layer order in a band; width is determined by the length of the text in the field, and wraps within the band.
row	Stacks elements horizontally in one row according to the layer order in the band.

Dynamic height message elements

Set **dynamic-height=true** on message elements. This will allow the element size to grow according to line height. Also, setting the following Reporting engine configuration option will allow the element size to grow according to the font size used:

```
org.pentaho.reporting.engine.classic.core.layout.fontrenderer.UseMaxCharBounds
 = true
```

Proportional column widths

To support varying window sizes in a Web browser, enable the use of proportional column widths so that the resulting table will have a width of 100% and the columns will have the proportional equivalent of their static widths as width. The relevant Reporting engine option to set this globally is:

```
org.pentaho.reporting.engine.classic.core.modules.
output.table.html.ProportionalColumnWidths = true
```

Template Properties

The following band properties affect RDW and PIR report templates. Notice that many properties have a different effect in RDW than in PIR.

Band	Report Design Wizard	Interactive Reporting
page header	None	Edit message elements
report header	None	Edit message elements
group headers	Inserts a message element with the value of: **fieldname: $(fieldname)**	Edits any message elements within the band. Inserts a message element with the value of: **fieldname: $(fieldname)**
detail header	**1:** Inserts a column header label for every field used. Default value for the label is the fieldname. **2:** Edits any label within the band.	**1:** Inserts a column header label for every field used. Default value for the label is the fieldname. **2:** Edits any label within the band.
details	Inserts the appropriate fields defined	Inserts the appropriate fields defined
detail footer	Insert a numeric field in the same x-position and width of the details field that	None

Band	Report Design Wizard	Interactive Reporting
	the summary calculation is applied.	
group footer	None	**1:** Inserts a numeric field in the same x-position and width of the details field that the summary calculation is applied. **2:** Inserts a message element in the x-position of the details field with no summary calculations applied.
report footer	None	**1:** Inserts a numeric field in the same x-position and width of the details field that the summary calculation is applied. **2:** Inserts a message element in the x-position of the details field with no summary calculations applied. **3:** Edit any message elements within the band
page header	None	Allows the modification of message elements

Deploying a Template to Report Design Wizard

Once you've created a template for RDW, you must follow the below process to deploy it to Report Design Wizard in Report Designer and Pentaho Data Integration.

1. Shut down Report Designer and Pentaho Data Integration if either of them are currently running.
2. Create an icon for your template, in PNG format, with the same name as the template file.
   The size of the icon doesn't matter; RDW will scale it to fit the correct dimensions. However, you can avoid unusual scaling issues by creating a square-shaped (equal width and height) graphic. If you'd like further guidance, take a look at the default template icons that Pentaho provides in the templates directory.
3. Copy the icon and the PRPT template files to the following directories:
   - /pentaho/design-tools/report-designer/templates/
   - /pentaho/design-tools/data-integration/plugins/spoon/agile-bi/templates/

Your template is now deployed to Report Design Wizard and will be available when you next start Report Designer or Data Integration.

Deploying a Template to Interactive Reporting

Once you've created a template for PIR, you must follow the below process to deploy it to the Interactive Reporting plugin.

1. Shut down the BI Server if it is currently running.
2. Create an icon for your template, in PNG format, with the same name as the template file.
   The size of the icon doesn't matter; PIR will scale it to fit the correct dimensions. However, you can avoid unusual scaling issues by creating a square-shaped (equal width and height) graphic. If you'd like further guidance, take a look at the default template icons that Pentaho provides in the templates directory.

3. Copy the icon and the PRPT template files to the `/pentaho-solutions/system/pentaho-interactive-reporting/resources/templates/` directory.

4. Edit the `/pentaho-solutions/system/pentaho-interactive-reporting/resources/messages.properties` file and add a new line for your template with the **template_** prefix, the name of your template file, and a friendly name for the template as you'd like it to appear in the PIR interface, as in the following example (given a template filename of **template_demo.prpt**):

```
template_template_demo=Template Demo
```

Your template is now deployed to Pentaho Interactive Reporting.

Setting the Default Interactive Reporting Template

To change the default template for Interactive Reporting, edit the `/pentaho-solutions/system/pentaho-interactive-reporting/settings.xml` file and change the value of the **<default-template>** node. You do not have to provide a path to the template PRPT file -- just the filename.

```
<!-- default template -->
<default-template>1_jade_1_left_aligned.prpt</default-template>
```