

WebCenter 25.03

Custom Reports

03 - 2025

Contents

1. Copyright Notice	5
2. Introduction to Custom Reports	7
2.1. What's new in 24.03	7
2.2. What's new in 21.11	7
2.3. What's new in 20	7
2.4. What's New in 18.1	7
2.5. What's new in 18	8
2.6. What's New in 16.1	8
2.7. What's New in 16	8
2.8. What's New in 14.1	8
2.9. What's New in 14	8
2.10. What's New in 12.1	8
3. Report Creation Process	10
4. Developing Custom Report using iReport	11
4.1. Install iReport	11
4.1.1. Java 1.6	11
4.1.2. iReport designer for JasperReports	11
4.1.3. iReport Documentation	12
4.2. iReport Configuration	12
4.2.1. Collect needed files	12
4.2.2. Classpath Settings	13
4.2.3. Add JDBC Driver	13
4.2.4. Connect to the WebCenter database	15
4.2.5. Jasper Reports Properties	18
4.3. iReport Basics	18
4.3.1. User Interface	18
4.3.2. Report Data Source	20
4.3.3. Report Bands	20
4.3.4. Report Elements	21
4.4. Create your first Custom Report	22
5. Developing Custom Report using Jaspersoft Studio	26
5.1. Install	26
5.2. Configuration	26
5.3. User Interface	30
5.4. New Features (Not Available in iReport)	31
5.5. Create your custom report using Jaspersoft Studio	32

6. Basic SQL queries	33
6.1. Reporting views.....	33
6.2. Simple SELECT query.....	34
6.2.1. WHERE clause.....	34
6.2.2. ORDER BY clause.....	34
6.3. Aggregate operators.....	35
6.4. Interesting to know.....	35
7. Report fields, Parameters and Variables	36
7.1. Report fields.....	36
7.1.1. Usage.....	36
7.2. Parameters.....	36
7.2.1. Usage.....	37
7.2.2. Built In System Parameters.....	37
7.2.3. Custom Report parameters.....	37
7.2.4. Example Parameters.....	38
7.3. Variables.....	39
7.3.1. Usage.....	40
7.3.2. Built-In Variables.....	40
7.3.3. Custom Variables.....	40
7.3.4. Examples for Variables.....	41
8. Custom Reports and WebCenter	42
8.1. Report Template Files.....	42
8.1.1. JRXML.....	42
8.1.2. ZIP.....	42
8.2. Report Parameters.....	43
8.2.1. WebCenter Report Parameters.....	43
8.2.2. Parameter Types.....	44
8.2.3. Additional Report Parameter Properties.....	44
8.2.4. Example: Linking Customer Parameter.....	46
9. Create your Own Custom Report Template	48
9.1. Blank Template.....	48
9.2. Using the WebCenter Report Template.....	48
9.3. Modifying Default Reports.....	49
10. Fine-tuning your Report	51
10.1. Date Formatting.....	51
10.2. Determine What to Display if no date is found.....	52
10.3. Field Properties.....	53
10.4. Custom JR Data Source.....	54
11. Advanced Topics	58
11.1. Dataset.....	58

11.2. Using Tables.....	59
11.3. Using Images.....	61
11.4. Report Styles.....	63
11.5. Sub Reports.....	64
11.6. Expressions.....	65
11.7. How to fill a text field with RTF Content.....	66
11.7.1. Reading BLOB's from the database.....	66
11.7.2. Markup text field property.....	67
11.7.3. Reading WebCenter RTF Attributes.....	67
12. Localization Support.....	68
12.1. Report Parameters labels in WebCenter UI.....	68
12.2. Localization Support for the generated Reports.....	68
13. Exporting Custom Fonts.....	69
14. Tips and Tricks.....	71
15. WebCenter Database Interface.....	73
16. Scriptlets.....	84
16.1. Project Scriptlet.....	85
16.2. Database Scriptlet.....	85
16.3. ImageGenerator Scriptlet.....	88
17. Chart Customizer.....	89
18. Custom JR Data Source Providers.....	90

1. Copyright Notice

© Copyright 2025 Esko Software BV, Gent, Belgium

All rights reserved. This material, information and instructions for use contained herein are the property of Esko Software BV. The material, information and instructions are provided on an AS IS basis without warranty of any kind. There are no warranties granted or extended by this document. Furthermore Esko Software BV does not warrant, guarantee or make any representations regarding the use, or the results of the use of the software or the information contained herein. Esko Software BV shall not be liable for any direct, indirect, consequential or incidental damages arising out of the use or inability to use the software or the information contained herein.

The information contained herein is subject to change without notice. Revisions may be issued from time to time to advise of such changes and/or additions.

No part of this document may be reproduced, stored in a data base or retrieval system, or published, in any form or in any way, electronically, mechanically, by print, photoprint, microfilm or any other means without prior written permission from Esko Software BV.

This document supersedes all previous dated versions.

PANTONE[®] Colors displayed in the software application or in the user documentation may not match PANTONE-identified standards. Consult current PANTONE Color Publications for accurate color. PANTONE[®] and other Pantone trademarks are the property of Pantone LLC. © Pantone LLC, 2015

Pantone is the copyright owner of color data and/or software which are licensed to Esko to distribute for use only in combination with Studio Visualizer. PANTONE Color Data and/or Software shall not be copied onto another disk or into memory unless as part of the execution of Studio Visualizer.

This software is based in part on the work of the Independent JPEG Group.

Portions of this software are copyright © 1996-2002 The FreeType Project (www.freetype.org). All rights reserved.

Portions of this software are copyright 2006 Feeling Software, copyright 2005-2006 Autodesk Media Entertainment.

Portions of this software are copyright © 1998-2003 Daniel Veillard. All rights reserved.

Portions of this software are copyright © 1999-2006 The Botan Project. All rights reserved.

Part of the software embedded in this product is gSOAP software. Portions created by gSOAP are Copyright © 2001-2004 Robert A. van Engelen, Genivia inc. All rights reserved.

Portions of this software are copyright © 1998-2008 The OpenSSL Project and © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Adobe Creative Suite, Illustrator, InDesign, PDF, Photoshop, PostScript, XMP and the Powered by XMP logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft and the Microsoft logo are registered trademarks of Microsoft Corporation in the United States and other countries.

SolidWorks is a registered trademark of SolidWorks Corporation.

Portions of this software are owned by Spatial Corp. 1986 2003. All Rights Reserved.

JDF and the JDF logo are trademarks of the CIP4 Organization. Copyright 2001 The International Cooperation for the Integration of Processes in Prepress, Press and Postpress (CIP4). All rights reserved.

The Esko software contains the RSA Data Security, Inc. MD5 Message-Digest Algorithm.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle corporation in the U.S. and other countries.

Part of this software uses technology by Best Color Technology (EFI). EFI and Bestcolor are registered trademarks of Electronics For Imaging GmbH in the U.S. Patent and Trademark Office.

Contains PowerNest library Copyrighted and Licensed by Alma, 2005 – 2007.

Part of this software uses technology by GlobalVision. ArtProof and ScanProof are registered trademarks of GlobalVision Inc.

Part of this software uses technology by Qoppa Software. Copyright © 2002-Present. All Rights Reserved.

All other product names are trademarks or registered trademarks of their respective owners.

Correspondence regarding this publication should be forwarded to:

Esko Software BV

Raymonde de Larochelaan 13

B – 9051 Gent

info.eur@esko.com

2. Introduction to Custom Reports

Building Custom reports has been made easy from WebCenter 12 onwards, as it is integrated with Jasper Reports, an open source reporting engine. WebCenter ships with two built-in reports:

- Approval Statistic report
- Task Management report
- Sample Attribute Category Excel
- Sample Attribute Category PDF



Note: If you are a WebCenter SaaS customer, you cannot generate your own custom reports. You can use Esko Reporting service to get your reports.

2.1. What's new in 24.03

- Jasper reports library upgraded to version 6.18.1
- XLS output format is not supported in this version (XLSX output format can be used instead)

2.2. What's new in 21.11

- **Noto Sans** font is provided as the default font for the attribute category PDF reports.

2.3. What's new in 20

- Reporting views extensions.

2.4. What's New in 18.1

- **DOCUMENTVER** New report parameter type linked to the WebCenter entity. With this parameter, it is possible to select from all visible documents in a project.
- **More output types supported:** PDF, XML, XLS, XLSX, CSV, TXT, RTF, HTML, DOC.
- **Run Report workflow node** Introduced.
- **RunWcReport.jsp** SDK call extended to allow specifying values for the report parameters.
- Reporting views extensions WebCenter Database Interface.

2.5. What's new in 18

Changes introduced in WebCenter 18

- **Jasper report library 6.4.1**
- **New JR Custom Data Source:** This will help to generate reports for which data source is a custom data source.
- **New report parameter type linked to the WebCenter entity – “DOCUMENT”:** With this parameter it is possible to select from all visible documents in a project.
- **Reporting views extensions** [WebCenter Database Interface](#) on page 73.
- **WebCenter built-in reports updated to use ‘Roboto’ as a default font** (this font is part of the WebCenter installation).
- **New built-in reports:** Sample Attribute Category Excel, Sample Attribute Category PDF.

2.6. What's New in 16.1

- **Localization files update:** the same set of languages is supported as for the WebCenter application.
- **Fixed logic to calculate ApprovalInfo_Duration_Days:** Column value in the ‘WcReports_DocApprovalQ’ reporting view (used in built-in approval reports).

2.7. What's New in 16

- Jasper report templates can be now built also using Jaspersoft Studio

2.8. What's New in 14.1

- **Basic HTML output format no longer supported** (due to an inconsistent output).
- **Built-in custom reports bug-fixes.**

2.9. What's New in 14

- **Reporting views extension** More details at [WebCenter Database Interface](#) on page 73
- **Built-in custom reports improvements**

2.10. What's New in 12.1

- **Jasper report library 5.0**

- **New 'built in' report parameters**
- **Reporting views extensions** [WebCenter Database Interface](#) on page 73
- **Project scriptlet extensions** [Scriptlets](#) on page 84

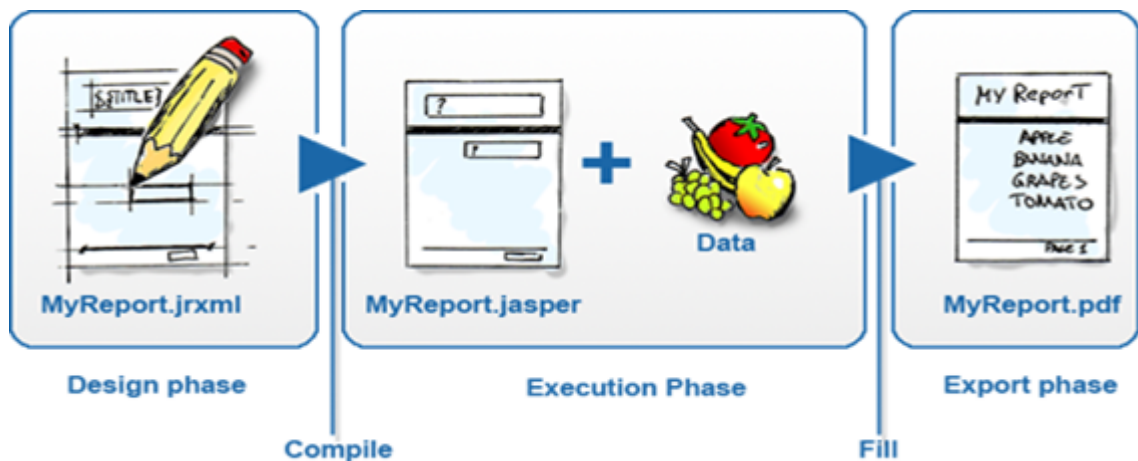
Some more examples to describe the new features:

- Add info about the user who ran the report
- Display project thumbnail
- Display document thumbnail
- How to fill a text field with RTF content
- What query to use to get the documents included in the project bill of material (BOM)?

3. Report Creation Process

The process to create custom reports consists of the following:

1. **Identify the reporting needs.** You need no prior knowledge of Jasper reporting to complete this step. However, you need to understand the data organization of WebCenter from a user's perspective.
 - a. What needs to appear in the report, and where is this information sitting in WebCenter. A classic problem is the expectation to show information which is not entered in WebCenter or only entered in a non-structural way.
 - b. When will the report be run and what parameters will you need to provide for this. This includes understanding time effects.
 - c. How the data in the report will be presented. This includes lay-out, terminology, translations
2. **Create a Report Definition** using either iReport or Jaspersoft studio tool. The report definition file is either a jrxml file or a zip file with a jrxml file and other files (images, fonts, ...)
3. **Upload this report definition to WebCenter** to define it as a custom report. You can now start using this report.



This document largely describes creating a report definition and further steps on uploading and using the reports are explained in the WebCenter Administrator Guide in detail.



Note: To use iReport or Jaspersoft Studio tool, you need to install and connect it to a WebCenter database.

4. Developing Custom Report using iReport



Note: iReport/Jaspersoft Studio Support Announcement: As of version 5.5.0, Jaspersoft Studio will be the official design client for Jasper Reports. iReport was a supported product in maintenance-only mode until December 31, 2015.

4.1. Install iReport

Jasper Reports is pre-installed in WebCenter. It knows what to put in a report by getting a description for this. This description is a *jrxml* file. This *jrxml* file can be created by hand (in a text editor) and uploaded to WebCenter but this may be prone to errors and cumbersome.

Therefore it is advised to use free report designers like iReport and Jaspersoft Studio available. As a report creator, you must install either of these tools on your computer.



Note: This document assumes you are using **iReport** or **Jaspersoft Studio** on a Windows computer.

4.1.1. Java 1.6

iReport is a Java program and therefore needs Java 1.6 or later. In case your computer does not have Java 1.6 or later, you must download this first from the Oracle/Sun website.

If you can use WebCenter's viewer successfully, then you can use iReport.

4.1.2. iReport designer for JasperReports

Jasper reports library 5.0 is used as of WebCenter 12.1. So for developing custom reports for WebCenter 12.1, you can use iReport 5.0.4.

You can download it from the following location:

<http://sourceforge.net/projects/ireport/files/iReport/iReport-5.0.4/>

Download the software and install the same.

If you encounter problems starting iReport, complaining about wrong Java version, you can manually configure iReport to use a specific version or installation of Java. This can be done by editing the *ireportpro.conf* file that can be found in the iReport installation directory (C:\Program Files (x86)\Jaspersoft\iReport-4.0.1\etc).

Add or modify the following line: `jdkhome="c:/Program Files (x86)/Java/jre7"`



Note: Jasper report library 4.0.1 is used in WebCenter 12.0. So for developing custom reports for WebCenter 12.0 you have to download iReport version 4.0.1. You can download it from the following location <http://sourceforge.net/projects/ireport/files/iReport/iReport-4.0.1/>

4.1.3. iReport Documentation

To know more about i Report or Jasper Reports these are some resources available online:

<http://community.jaspersoft.com/wiki/i-report-designer-tutorials-help>

You can download iReport Ultimate Guide for free from <http://community.jaspersoft.com/documentation> upon registration.

4.2. iReport Configuration

You will now configure iReport to use a WebCenter database. This can either be a WebCenter database on your own computer (if you have WebCenter installed on it) or you can connect to the database of a remote WebCenter. Make sure to connect to a test server, not to a running production server.

The following configuration needs to be done in iReport

- Set the class path
- Add a JDBC Driver
- Connect to the database

4.2.1. Collect needed files

During the configuration steps, you must browse through several files.

To access the required files, the best way is as follows:

1. Go to <http://help.esko.com/products/WebCenter>
2. Download WebCenter_Reports_Files.
3. Unzip the zip to a location like C:\Jasper or any location of your choice.

In case you are unable to access to help.esko.com, you may try the below option:

These files are also delivered with the WebCenter installation (see the below table). Make sure to not take them away from WebCenter, they are also used by WebCenter itself.

File	Location	Description
ojdbc6.jar	ApplicationServer\Jboss\server\default\lib\	Oracle JDBC driver
sqljdbc4.jar	ApplicationServer\Jboss\server\default\lib\	MS SQLServer JDBC driver
WebcenterReportTemplate.zip	ApplicationServer\DatabaseSchema\CustomReports\	Custom report template

File	Location	Description
war-ireport-(version number).jar	ApplicationServer\Jboss\server \default\deploy\ wcr-ear.ear\lib\	Scriptlets Chart customizer

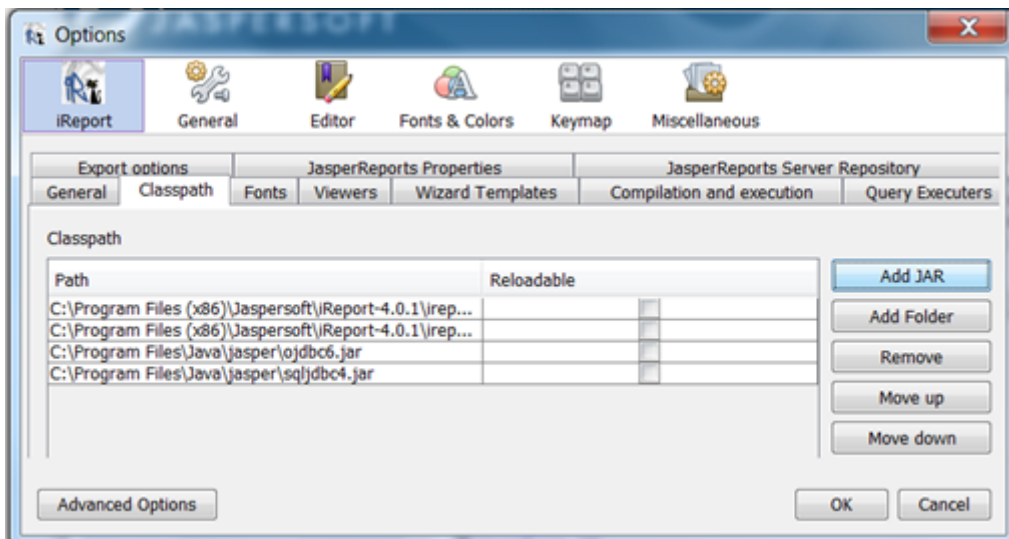
4.2.2. Classpath Settings

iReport needs a driver to connect to the WebCenter database. This driver is one of the above-mentioned files (*sqljdbc4.jar* for MS SQL and *ojdbc6.jar* for Oracle).

Start with **adding the driver JAR libraries to the classpath** of iReport. This is done with following steps:

1. Click on **Tools/Options**. This opens a dialog. Make sure iReport is selected at the top and click on the **Classpath** tab to show the classpath. (on a Mac, go to iReport/Preferences).
2. Click on **Add JAR** and browse to the *sqljdbc4.jar* (if you connect to a MS SQL database) or to *ojdbc6.jar* (if you connect to an Oracle database). This should show the dialog as below (with typically only 1 of the 2 new lines).

Ensure you point to the scriptlets jar file too here.

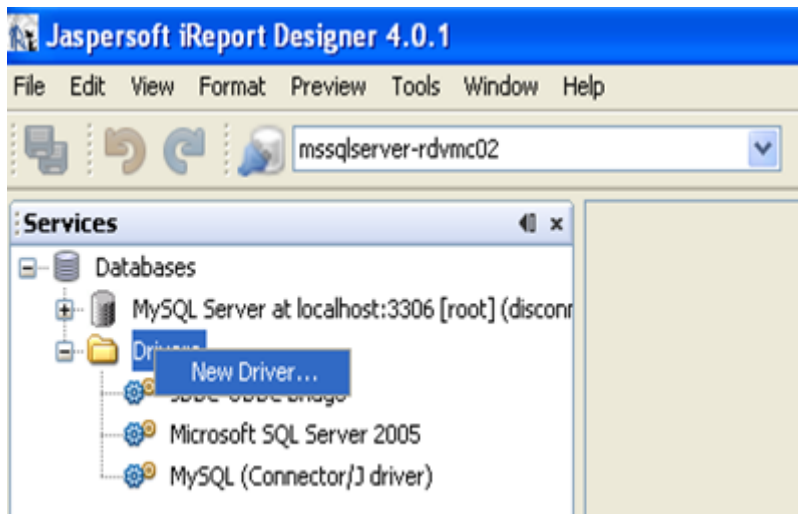


3. Click again **Add JAR** and browse to *wcr-ireport-(version number).jar*

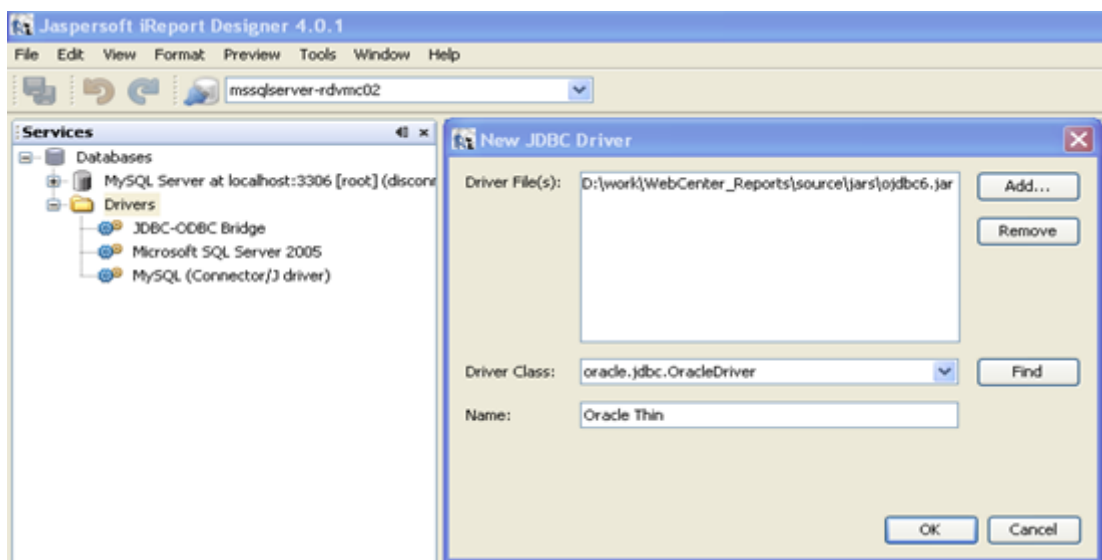
4.2.3. Add JDBC Driver

To install the driver

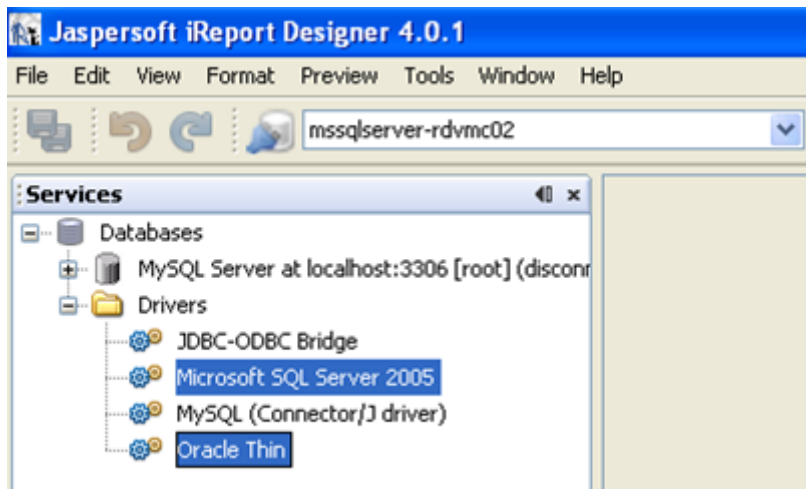
1. Click on **Window > Services**. Expand the **Databases** node. (the window might already be open, for example bottom left of your screen).
2. Right-click the **Drivers** node and choose **New Driver**.



3. In the dialog box, click **Add** and browse to either `sqljdbc4.jar` (if you connect to a MS SQL database) or to `ojdbc6.jar` (if you connect to an Oracle database). The other fields will be filled in automatically. Click **OK**.



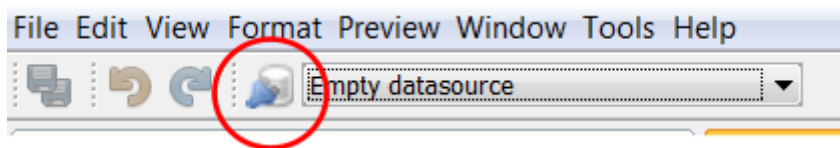
This will result in following (only 1 of the 2 needed).



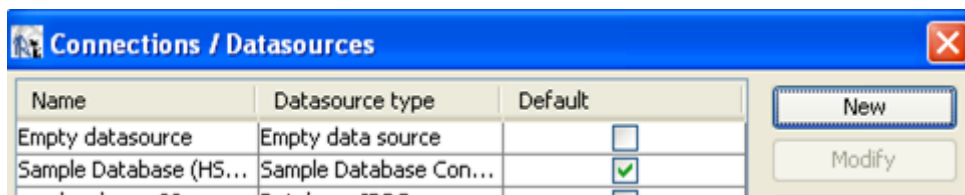
4.2.4. Connect to the WebCenter database

To setup a new database connection in iReport, do the following steps:

1. Click on the **Report Data Sources** icon from the toolbar.



2. In the popup window, select **New**.

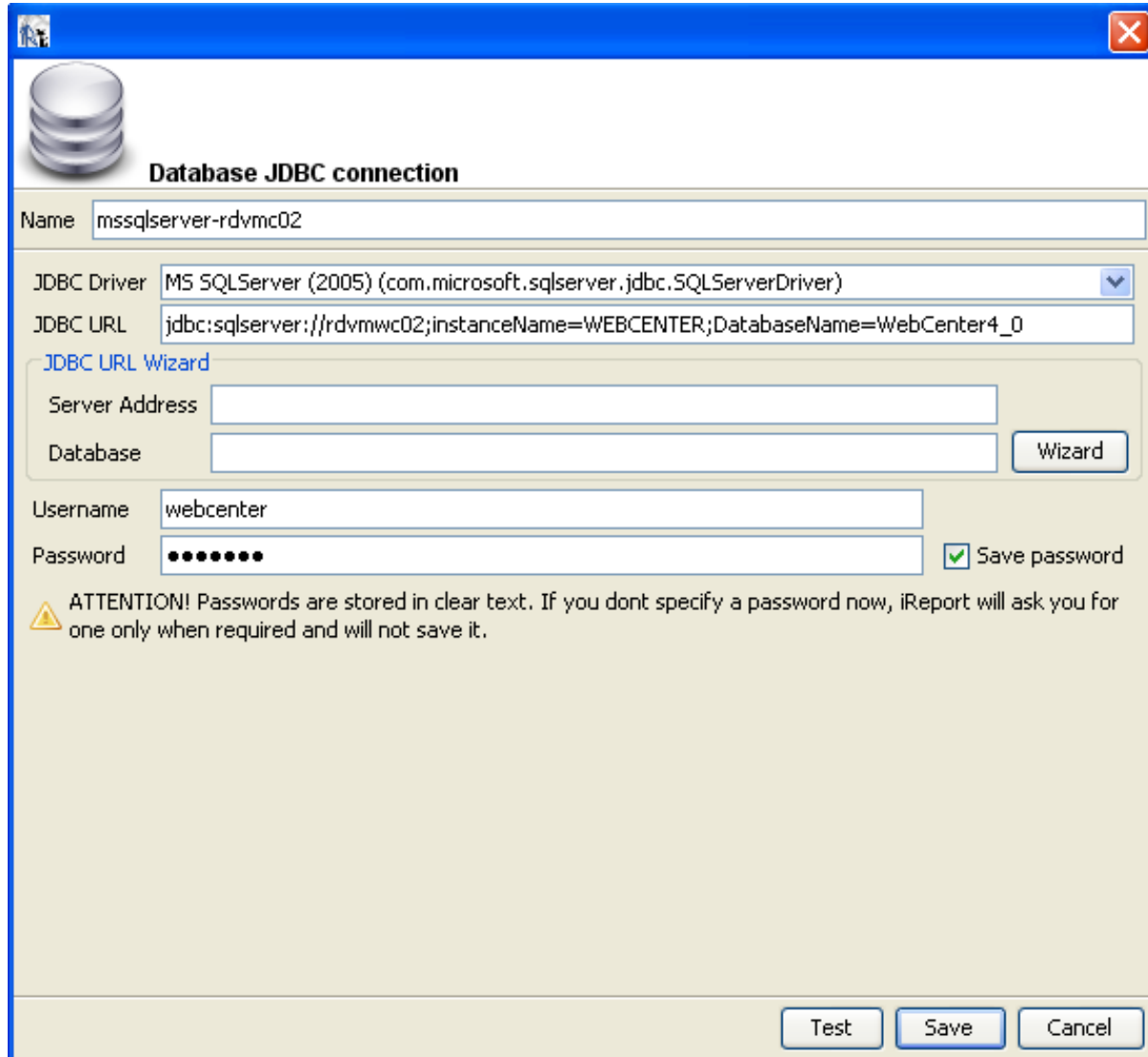


3. It opens another popup window, select **Database JDBC Connection** and click **Next** to enter the Database connection details.

a. For MS-SQL

- a. JDBC Driver: *MS SQLServer (2005) (com.microsoft.sqlserver.jdbc.SQLServerDriver)*.
- b. JDBC URL: **(this must really be correct!!!!)**`jdbc:sqlserver://rdvmwc10:instanceName=WEBCENTER; databaseName=WebCenter4_0.`
 - Replace rdvmwc10 with the name of the computer on which the database resides (remember this might be another computer than the app server).
 - Mostly the instance name is WEBCENTER but it is possible that people have used another one or the unnamed instance.
 - databaseName is always WebCenter4_0.

- c. Username: webcenter
- d. Password: *webcenter user database password*
- e. Click **Test** button and **Save**.



Database JDBC connection

Name: mssqlserver-rdvmc02

JDBC Driver: MS SQLServer (2005) (com.microsoft.sqlserver.jdbc.SQLServerDriver)

JDBC URL: jdbc:sqlserver://rdvmwc02;instanceName=WEBCENTER;DatabaseName=WebCenter4_0

JDBC URL Wizard

Server Address:

Database:

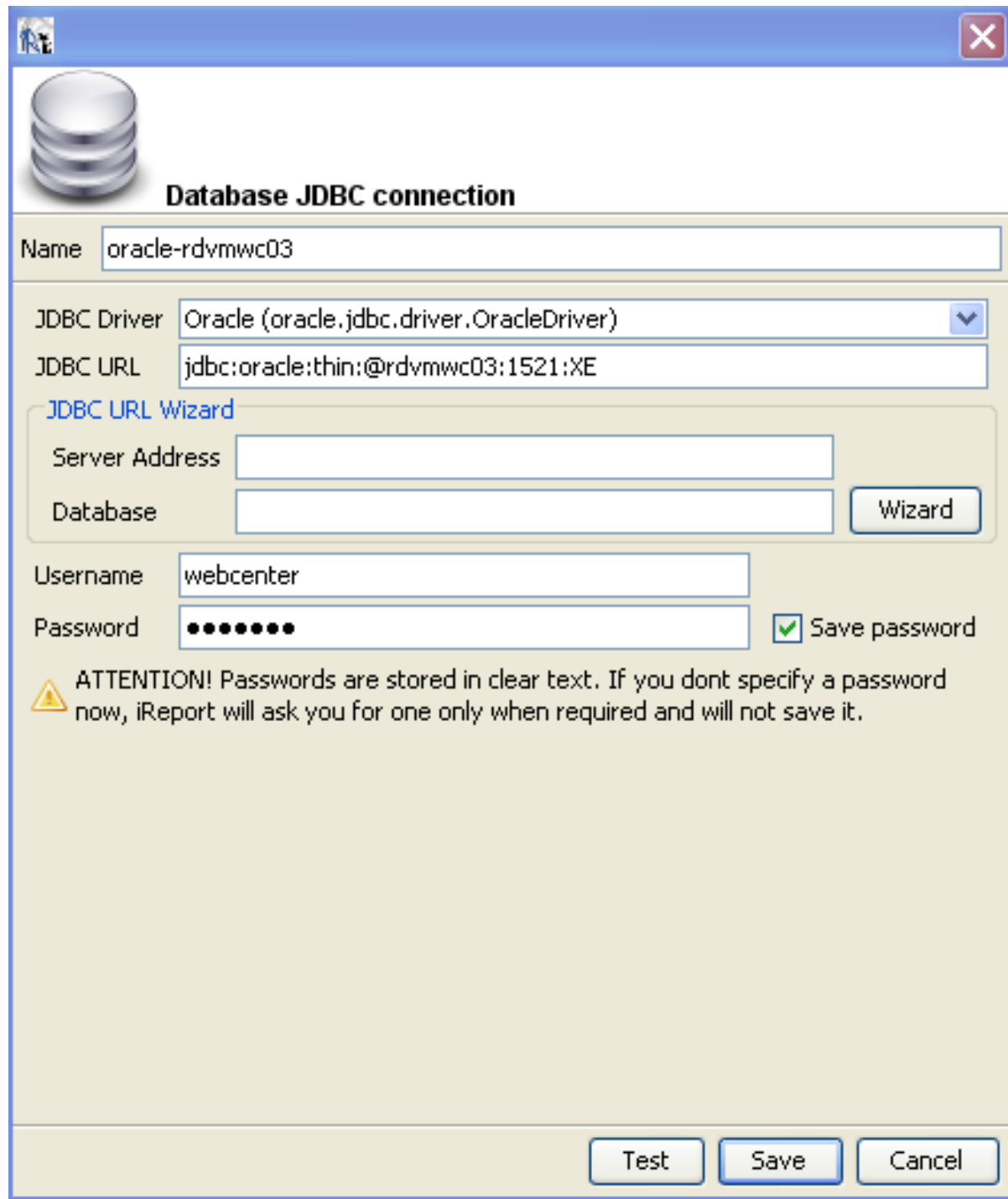
Username: webcenter

Password: Save password

ATTENTION! Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.

b. For Oracle

- a. JDBC Driver:
- b. Server address: the name of the computer having the database, for example rdvmwc10.
- c. Database: WebCenter_4.
- d. Then click wizard. This will update the JDBC URL.
- e. Username: WebCenter.
- f. Password: *webcenter user database password*
- g. Click the **Test** button
- h. Click **Save**.



Database JDBC connection

Name

JDBC Driver

JDBC URL


JDBC URL Wizard

Server Address

Database

Username

Password Save password

 **ATTENTION!** Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.

Troubleshooting:

If the test fails, try following in sequence:

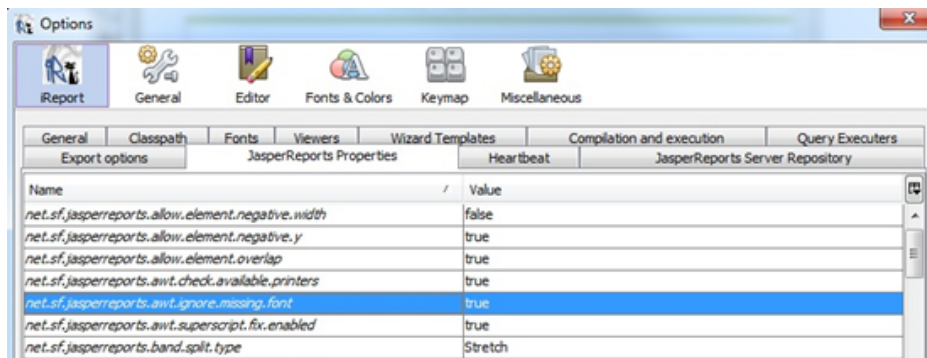
- Double check the URL for typing errors. Ensure every character is present.
- Retype the password (check the caps lock key).
- In case of MS-SQL, connect to the database from your computer using MS SQL Server Management Studio with the same credentials and double check the name of the instance.

- The file ApplicationServer\Jboss\server\default\deploy\wc5-ds.xml contains largely the same information.

4.2.5. Jasper Reports Properties

iReport designer has a section called **JasperReports Properties** at **Tools > Options > iReport > Jasper Report Properties**.

Keep these properties defaulted, and do not change any of the properties. These options/properties are not relevant for processing reports in WebCenter (WebCenter uses the default configuration and there is no GUI available to modify it).



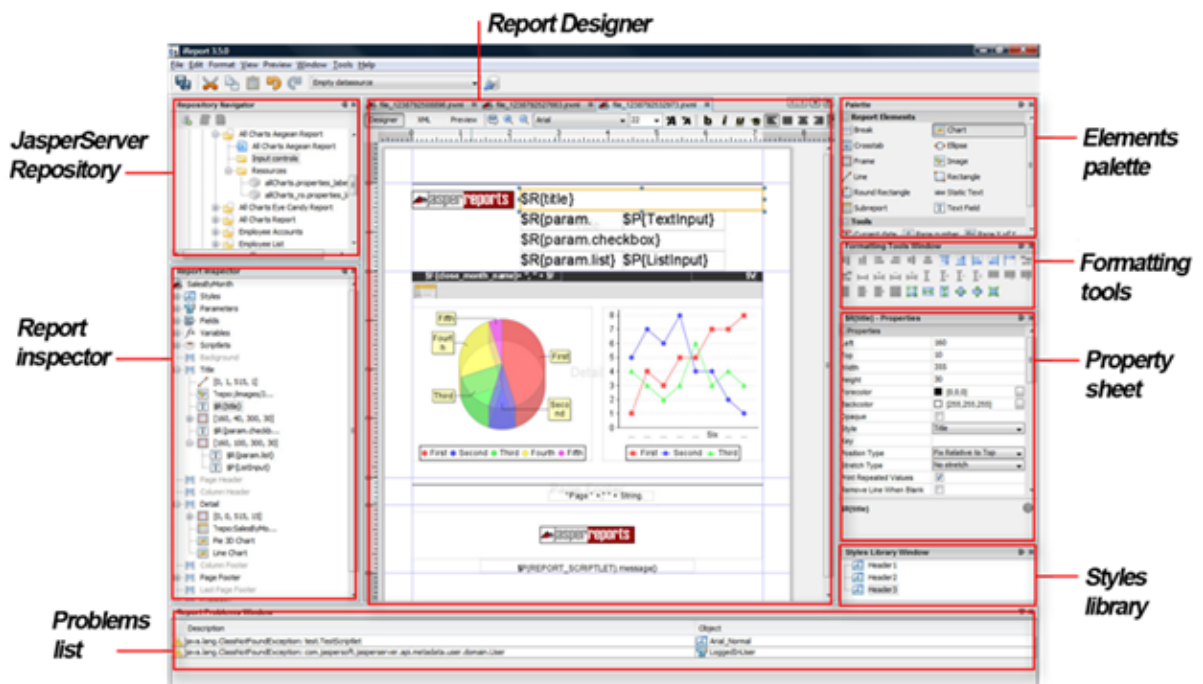
The only property you might have to change is to ignore missing font 'net.sf.jasperreports.awt.ignore.missing.font = true'.

4.3. iReport Basics

4.3.1. User Interface

The below picture shows the most important iReport user interface elements.

As you can see, the user interface consists out of several dockable sections. When you start iReport, it can look different from the below picture. For example, it is possible that the Elements palette is not displayed.



Following is an overview of the sections. You can activate them from the **Window** menu.

Section	What you do with it
JasperServer Repository	Can be ignored
Report Inspector	<ul style="list-style-type: none"> Shows the report in a structured (not graphical) way. Browse through your report (think of the browser in PackEdge). The structure is composed of many objects (such as fields, parameters and variables), bands and elements (such as text fields, images or charts). Right-clicking a report object invokes its context menu.
Problem list	When you preview a report, this shows errors and points of attention.
Report Designer	<p>Main working section where you layout your report.</p> <p>It has 3 tabs namely Designer, XML, Preview.</p> <p>You normally work in Designer and use Preview to try the report. You typically won't use the XML tab (this one shows the .jrxml file as XML code)</p>
Elements Palette	Contains the elements and tools that can be dragged inside a band to display your data.
Formatting tools	Tools to align elements and adapt their size to fit other elements.
Property Sheet	Used to edit the properties of all the components that make up the report (bands, elements, parameters, variables, fields and page format options etc).

Section	What you do with it
	When something is selected in the designer or in the Report Inspector view, the property sheet shows the options for the selected object.

4.3.2. Report Data Source

The WebCenter database is the data source for the WebCenter custom reports. It is defined by the report query.

- SQL select queries are used to fetch data from the WebCenter database. See [Create your Own Custom Report Template](#) on page 48.
- The database contains a set of views specially written to supply data for custom reports. See [Scriptlets](#) on page 84.

The main report allows only one SQL query to get report data. Sometimes you may want your report to show some other information in Tables or Charts or List elements. This can be achieved by defining dataset objects. For more details see chapter 13.1.

If you don't need any query for any report, just define a dummy query like

- **Oracle:** select 1 from dual
- **MSSql Server:** select 1 value

4.3.3. Report Bands

Jasper report is divided into different horizontal portions named **bands**. All of them are optional. You can choose the bands required for your report. All bands are visible by default. To remove some band from your report just select the band in the report inspector, do right click and select **Delete Band**. The deleted band is grayed out. You can always enable this band again by selecting **Add Band** from its context menu.

Following is an overview of all default report bands:

Band	Description
Title (mandatory)	Printed at the top of the first page of the report and is created only once. Put your report title information here.
Page Header	Repeated at the top of every page in the report. Page heading information can be placed in this band.
Column Header	Printed on each page that contains a detail band. If a page is split in one or more columns, this band is printed on each column.
Detail	Contains the body of the report. The detail band is repeated for each record in the data source. This is the only band that you can have more than one in your report. To add another detail band, select the default detail band and select Add Another Detail Band from its context menu.
Column Footer	Printed on each page that contains a detail band. If a page is split in one or more columns, this band is printed on each column.
Page Footer	Printed at the bottom of every page. It is a good place to insert page numbers, etc.

Band	Description
Summary	Printed only once at the end of the report. Your report summary information can be placed in this band. This band is often named 'report footer'.
Last Page Footer	Printed only in the last page. Use this band if you want to make the footer on the last page different from the other page footers.
No Data	Printed only when the report data source does not provide any data. It is only applicable if the report property 'When No Data' is set to 'No Data Section'.
Background	Used for setting a page background for all the pages in the report. It can have a maximum height equal to the page height and its content will appear on all the pages without being influenced by the page content defined in the other bands.

A band is always as wide as the page width (excluding the margins). Each band has its own configurable minimal height. The band height can change during the filling process.

You can conditionally show or hide any band at run time using the band 'Print when expression' property.

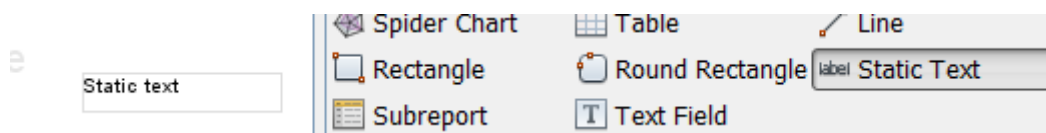
If the expression is empty or evaluates to true, the band is printed, otherwise it is not printed. A band becomes invisible if its height is set to zero too.

4.3.4. Report Elements

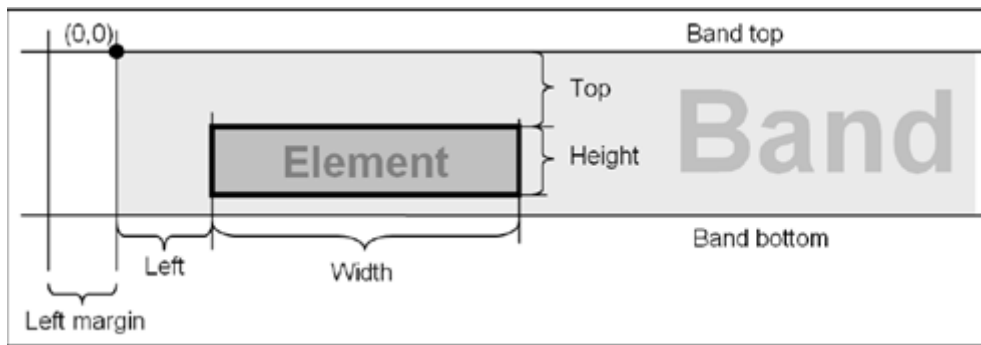
A report is designed by placing the desired report elements and report objects such as parameters, fields and variables in the appropriate bands (sections).

Working with iReport elements:

- To insert an element in a report, drag the element from the palette section into a report band. The new element will be created with a standard size and will appear in the Report Inspector. For example you can drag and drop the text field from the palette to the Title band.



- You can edit an element's properties by selecting the element (click on it in the report designer or select it in the report inspector) and then change the properties in the properties section. Select different elements to see how the section adapts. In our example, you can change the font and size of the static text field. Changing the text itself can be done by clicking in the field itself and start typing.
- All elements have a set of common properties, such as height, width, position, and the band to which they belong. Some properties are specific to the type of the element. Below picture explains the properties in a graphical way.



- Elements are inserted into bands, and every element is associated with its band. Element has to be completely contained within the band, otherwise design warning will be displayed in the Report Problems view and the report will not run.

Following are the basic element types:

Element type	Usage
Text elements	Used to print text strings such as labels or fields. <ul style="list-style-type: none"> • Static Text – Use this element to print any static text in the report. It is suitable for adding labels. • Text Field – Use this element for dynamic text content. Text field has an associated '<u>Expression</u>' that is evaluated at runtime to produce the text content that will be displayed. You can use parameters, variables, etc. in the expression.
Graphic elements	Used to create shapes and display images (they are line, rectangle, ellipse, image); Graphics elements have the pen and the fill properties.
Frame	Used to group a set of elements and optionally draw a border around them.
Table	Used to display data in a tabular format
Chart	Used to display various charts

4.4. Create your first Custom Report

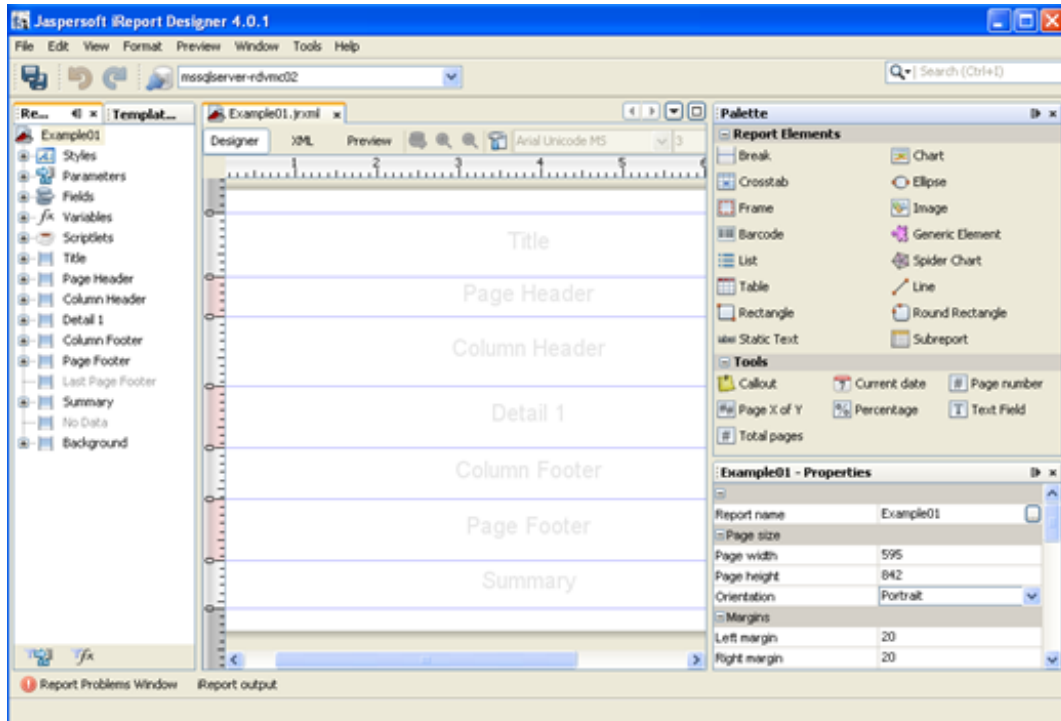
To create a new custom report, you can either

- start from a blank template
- use the template provided by Esko
- just modify the built-in reports.

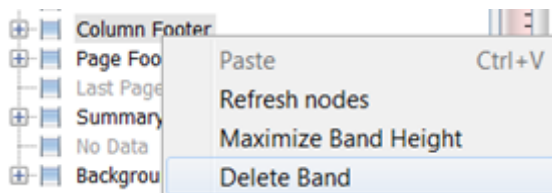
For our first exercise, we will create one from a blank template.

1. Select **File > New**.
2. In the **Report Creation Wizard** window, select **Report** on the left side.
3. Select **Blank A4** template and click on the **Open this Template** button.
4. In next screen, give the name 'Example01' and select the location where the report files will be kept. Click **Next** and **Finish** button.

- An empty report created using **Blank A4 template** will be visible in the **Report Designer** with default pre-defined bands (background, title, page header, column header, detail, column footer, page footer and summary).



- Using the report inspector, delete the Column Footer band (this band is not needed for this sample report).



- Drag a **Static Text** element from the Elements Palette and drop it in the Title band. Double click and enter **Esko Project Detail Report**.
- Set the font to 'Arial', font size to 20 and mark it bold. Set the font color blue from **Text Properties**.
In the report designer, resize the field to fit the width of the title band (drag to snap) and put the title center aligned (use the toolbar, you might have to drag the scrollbar to the right).
- If an element is not properly placed within the band display area, it will show a warning symbol. Typically, this means that the field is larger than the band or at least extends in one direction beyond the borders of the band.
- Drag **Text Field** element and drop it in the **Page Header** band. In the Text field expression of the text field, enter **List of Completed Projects** (the double quotes must be there, otherwise you get a compilation error when you run the report).
- The **Text field expression** property is available in the Properties View. You may also invoke the **Text Field Expression** dialog by pressing the F2 button while keeping the selection on the text field. Set the font size to 16, bold, green color and resize the text field.

- Right click on the report name (Example01) in the **Report Inspector** and choose **Edit Query** or click on the icon next to the **Preview** button. Enter following query by hand.

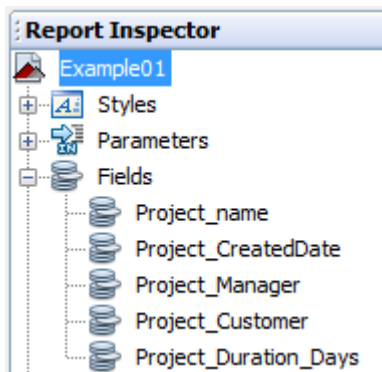
```
SELECT Project_name, Project_CreatedDate,
       Project_Manager, Project_Customer, Project_Duration_Days FROM
webcenter.WCReports_ProjectsQ WHERE Project_StatusName =
'Active'
```



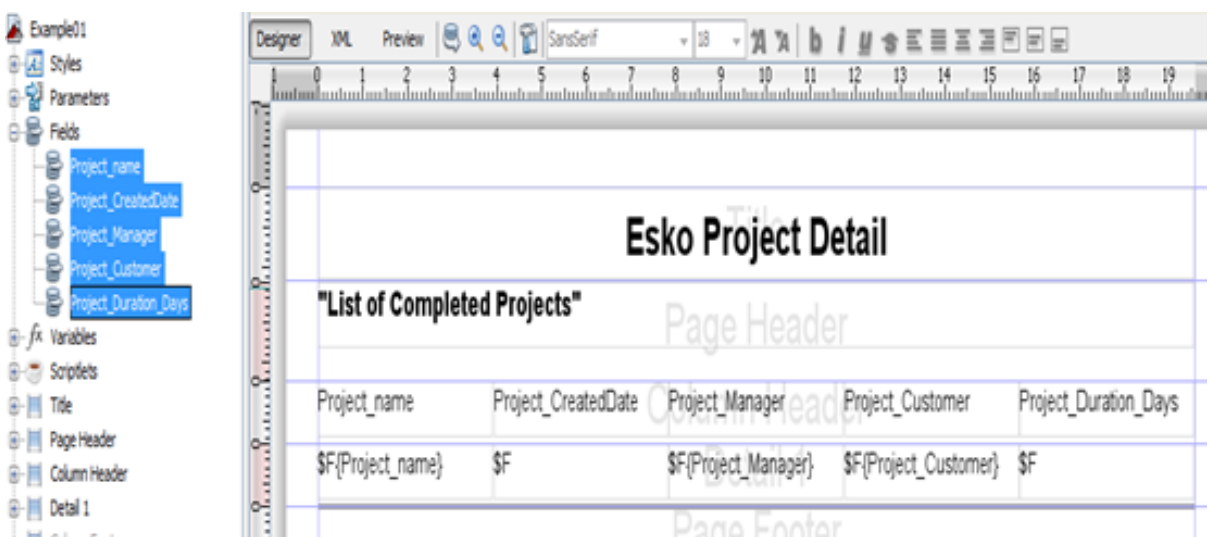
Note: Keep an eye on the bottom where errors (typically caused by typo's) are shown. **Error:SQL problems:Invalid column name 'Project_CreatedData'.**

- Make sure the 'Automatically Retrieve Fields' checkbox is checked, or click at 'Read Fields' button to retrieve the list of fields (columns) that are returned by the query. The available fields get listed in the bottom part of the dialog.
- You may Preview the query results and click **OK** on the right side.
- The columns mentioned in the query can now be found in **Fields** in the report inspector.

These fields need to be placed in the detail band to render the query results in the report.



- Select all the fields and drag and drop them in one go in the detail band (Detail1). Automatically, the Column Header band will be populated with the field title for every field. By default, it shows the field name. Double click and rename them if you want, and make it bold. Increase or decrease the field length. You may rearrange the fields by drag and drop.



17. Reduce the band height for both **Detail** and **Column Header** band to fit enough the data, otherwise there would be excessive space between rows.
18. You can now run your report for the first time. Click on Preview. This should show a simple report (make sure you actually have completed projects in your database). If you get a compilation error, it typically means one of the above steps was not done right (for example the quotes are missing from "List of Completed Projects).
19. Let's fill the page footer band with page number and date text fields from Tools section of the 'Report Elements' Palette. Drag 'Page X of Y' and 'Current date' one after other into the footer band and align them. These text fields come with pre-defined expressions. Play with the sizes and left/right alignment to get this to work fine.
20. Put a horizontal line above the page number, set the line height to 2 and fore-color to Grey.
21. Now let's add something to the summary band as well. Expand the Variables node in the 'Report Inspector', drag and drop the 'REPORT_COUNT' variable into the summary band. Edit the variable in the "Expression Editor" and change it like this:

```
There
      are " + $V{REPORT_COUNT} + " projects listed in the report
```

Change the Expression Class property to String in the Properties View. Align it centrally, set the font size to 12, red color.

22. Re-run your preview.

5. Developing Custom Report using Jaspersoft Studio

Jaspersoft Studio is the new Eclipse-based report designer for JasperReports and JasperReports Server. It is a full rewrite of iReport Designer, available as Eclipse plugin, and as a standalone application. Jaspersoft Studio allows you to create sophisticated layouts containing charts, images, sub reports, cross-tabs and much more. You can access your data through JDBC, TableModels, JavaBeans, XML, Hibernate, CSV, and custom sources, and then publish your reports as PDF, RTF, XML, XLS, CSV, HTML, text, DOCX, or OpenOffice.

Jaspersoft Studio's primary goal is to provide the features in the well-known Jaspersoft Report Editor, available as a port of iReport Designer. This is only the beginning - having its foundations on the Eclipse platform, Jaspersoft Studio will be a more complete solution allowing users to extend its capabilities and functionality.

The version compatible with WebCenter 24.03 is **Jaspersoft Studio 6.20.0**.

5.1. Install

As any other Java project Jaspersoft Studio require the Java Runtime Environment (JRE), but to compile the report scriptlets a full distribution of Java is required. You need to download the Java Development Kit (JDK) 1.6 or newer from oracle/sun website and download the latest version of the Jaspersoft Studio.

To download the latest version of Jasper Studio:

1. Click <https://community.jaspersoft.com/files/file/41-jaspersoft-community-edition/>
2. Navigate to **Window > Preferences and search for Compatibility**.
3. Select **JasperReports 6.13.0** as the source .jrxml Version.

Get the executable file from above link, execute and follow instruction. Be sure to download and install the correct version of Java and Jaspersoft Studio for your operative system.

5.2. Configuration

Now you need to configure the JasperSoft Studio to develop and execute the jasper report in your local machine. Following configuration need to be done in Jasper Soft Studio.

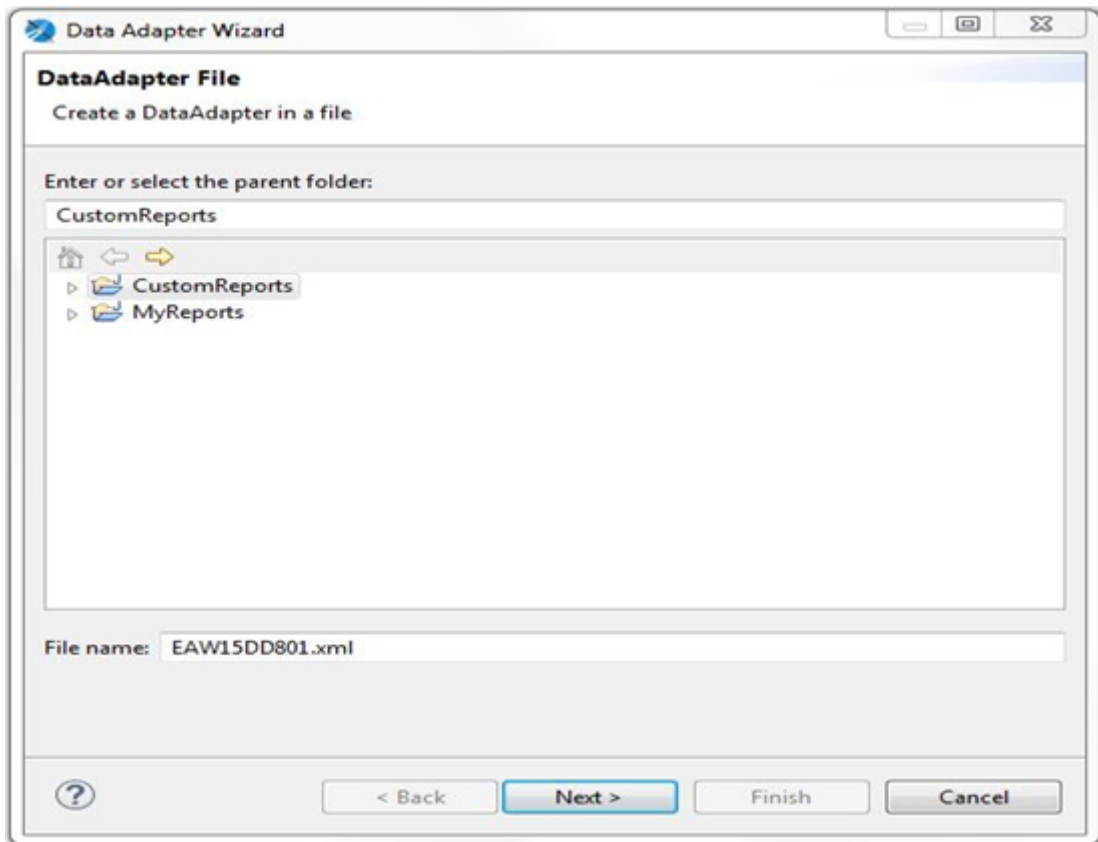
- Connect to the database.
- Add drivers to the class path.

Here are the steps to configure:

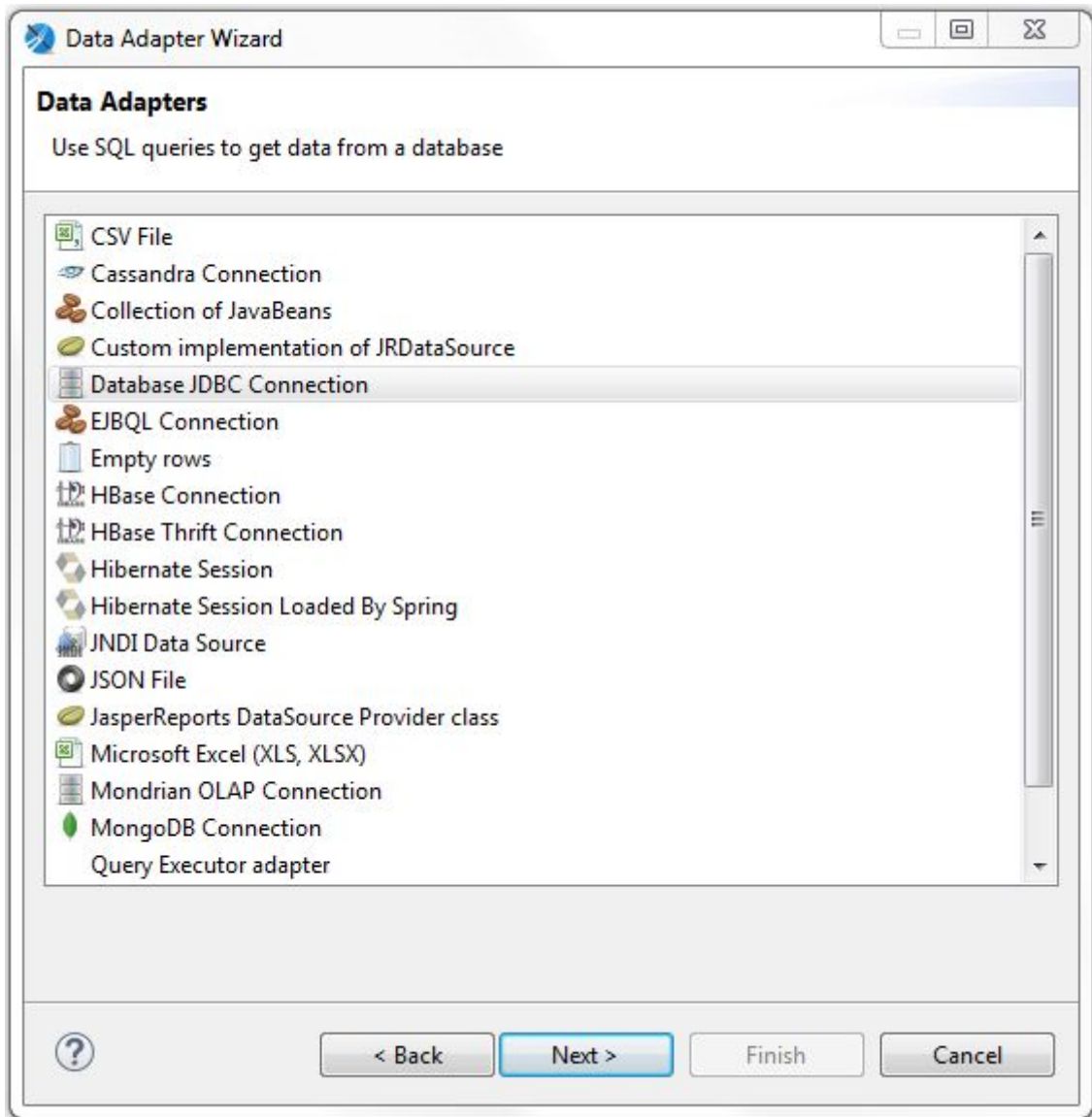
1. Click on the New Data Adapter in the tool bar.



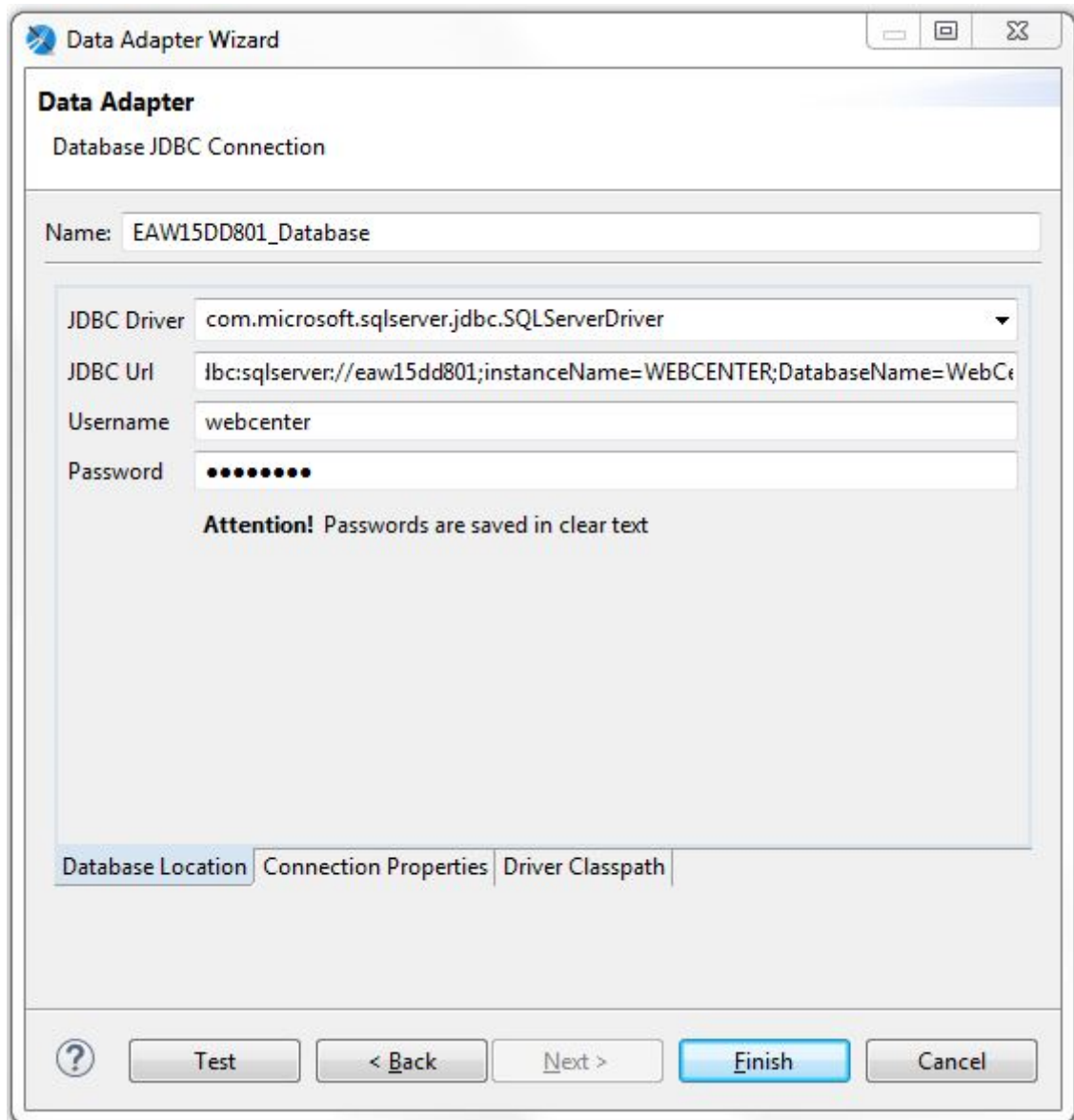
2. Data Adapter Wizard will show up, enter the folder name and click **Next**.



3. Select the proper required data adapter to fetch the data from database and click **Next**.



4. Fill the Database credentials. Detail of the database credentials is available in **iReport Configuration** section.



Data Adapter Wizard

Data Adapter
Database JDBC Connection

Name: EAW15DD801_Database

JDBC Driver: com.microsoft.sqlserver.jdbc.SQLServerDriver

JDBC Url: jdbc:sqlserver://eaw15dd801;instanceName=WEBCENTER;DatabaseName=WebCe

Username: webcenter

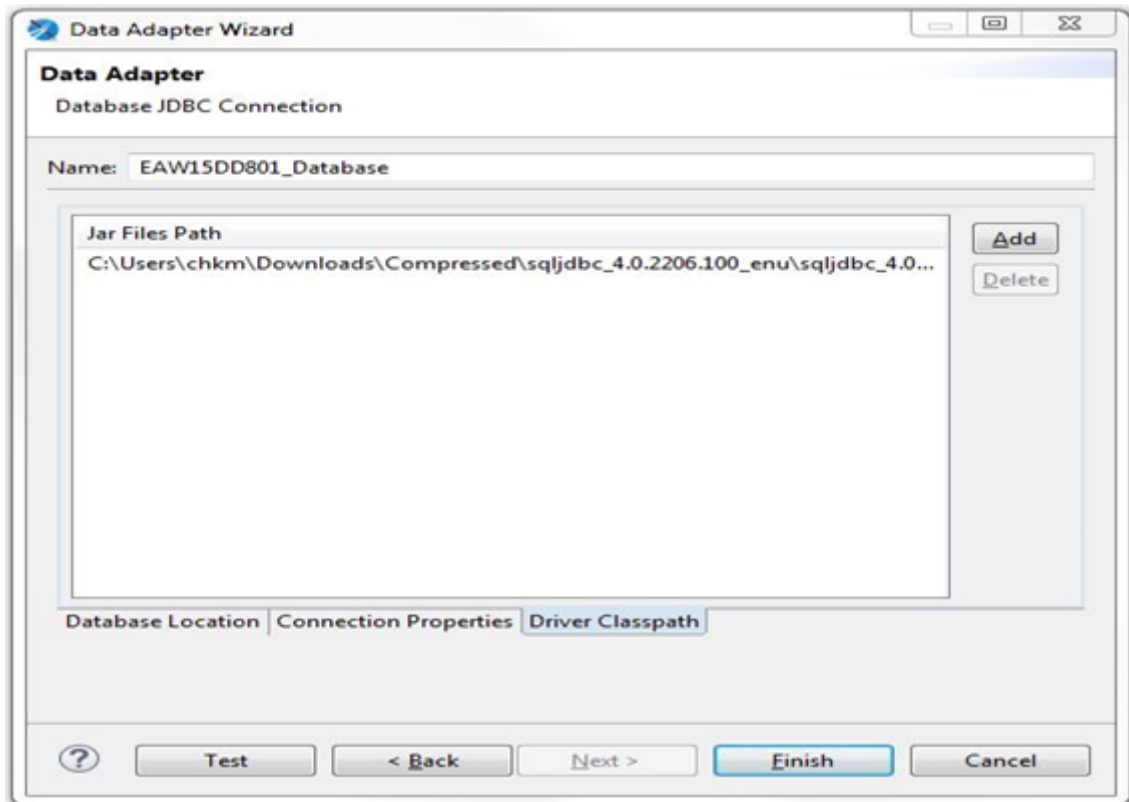
Password: ●●●●●●

Attention! Passwords are saved in clear text

Database Location | Connection Properties | Driver Classpath

? Test < Back Next > Finish Cancel

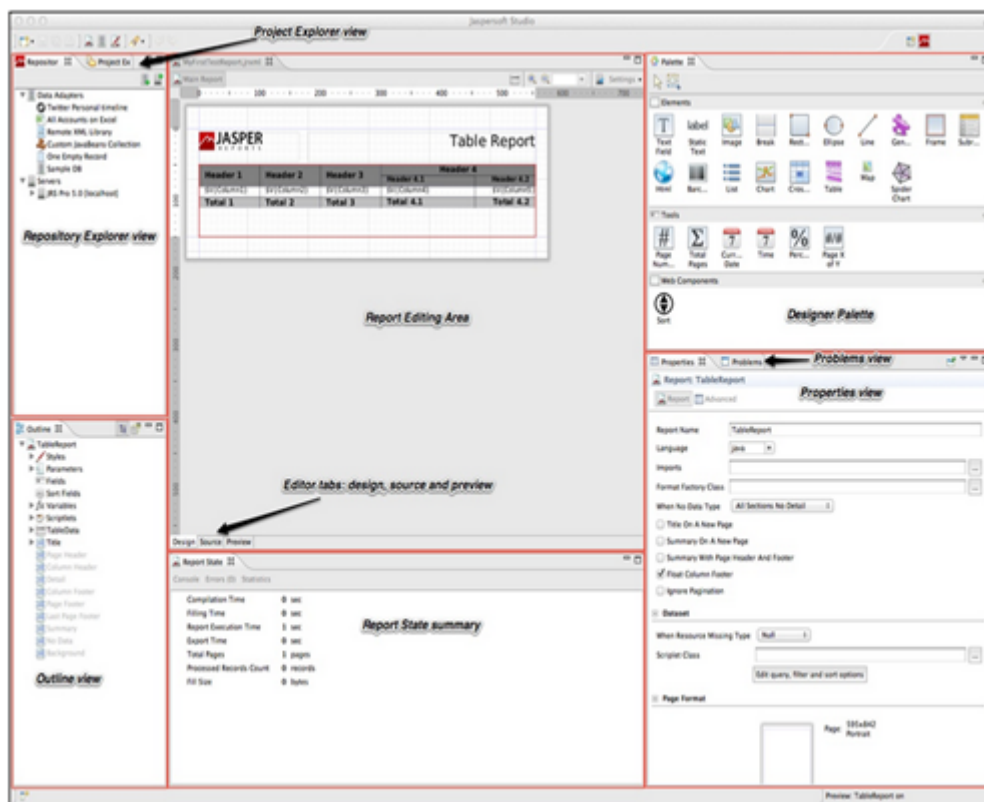
5. Add the proper database library. You can see the list of required jars in **iReport Configuration** section.



5.3. User Interface

The below picture shows different views of Jaspersoft Studio.

There are a number of views with which you can explore the data:



- The Repository Explorer view maintains the list of JasperServer connections and available data adapters.
- The Project Explorer view maintains the list of the projects in the current workspace, usually JasperReports projects.
- The Outline view shows the complete structure of the report in a tree-form way.
- The Properties view is usually a foundation one for any Eclipse based product/plugin. It usually gets populated with properties information of the actual selected element. That's way when you select a report element from the main design area (i.e: a text field) or from the Outline, the view shows information on it.

Some of these properties can be read-only, but most of them are editable and their modification will usually notify changes to the drawn elements (i.e: element width or height).

- The Problems view shows a list of problems and errors that can for example block the correct compilation of a report.

Finally, the **Report state summary** gives the user useful information about statistics on report compilation/filling/execution. Errors are shown here as well.

5.4. New Features (Not Available in iReport)

There are lot of new features in Jaspersoft Studio which are not available in iReport, some useful features are mentioned below.

- Better classpath management.
- Ability to provide size and position values by using measure units other than pixels (i.e. by typing '0.5 cm')
- Enhanced dataset & query dialog that allows to do more than the iReport "Report Query" dialog. I.e: You can directly play/change the data adapters.

- Reports are generated in an asynchronous way, allowing to preview almost immediately the first pages of very large reports and to cancel long running reports any time. Partially rendered pages are refreshed as the page changes during execution without having to wait until report has been fully generated.
- A new powerful expression editor with the ability of using built-in functions and creates new ones.
- More appealing properties. Usage of fancy dialogs and tabbed properties sheet instead of the "flat" iReport properties window similar to the new "Advanced" properties tab.

5.5. Create your custom report using Jaspersoft Studio

To create your first custom report please click on following link.

<http://community.jaspersoft.com/wiki/designing-report-jaspersoft-studio>

6. Basic SQL queries

From this point on wards, this document uses iReport reports designer to explain the remaining concepts involved in developing reports for WebCenter

As you have seen from the example in the section 3.4, you'll need to understand some basics about SQL. Your knowledge can be limited to understanding the SQL SELECT statement. An example select statement looks like:

```
SELECT Project_name, Project_CreatedDate, Project_Manager, Project_Customer
FROM webcenter.WCReports_ProjectsQ
WHERE Project_StatusName = 'Active'
```

Only the first two lines are obligatory. The first line says which information you want to get from the WebCenter database. The second line says in which table or view you'll find this information.

iReport can help you create the query with the query builder. However, it is still good to be able to read SQL SELECT statements.

6.1. Reporting views

To make your life easier, we have made special views in the WebCenter database with information pre-configured for easy reporting. These are all the views starting with webcenter.WCReports. The following views are available.

WcReports_UsersQ	WebCenter users details
WcReports_ProjectsQ	WebCenter projects information
WcReports_Project_AttrQ	WebCenter projects attributes
WcReports_Project_CharQ	WebCenter projects characteristics
WcReports_TasksQ	WebCenter tasks information
WcReports_TaskSpecificationQ	WebCenter tasks specifications
WcReports_DocVersionQ	WebCenter documents version information
WcReports_DocVersion_AttrQ	WebCenter documents version attributes
WcReports_Document_CharQ	WebCenter documents characteristics
WcReports_DocApprovalQ	WebCenter documents approval information
WcReports_UserHistoryQ	WebCenter user history information

It is recommended to use these Views, unless an expert in the internals of the WebCenter database is creating the report. Note that the views structure are maintained over the different WebCenter versions (the rest of the database is NOT guaranteed to remain the same structure).

If you need other information from the WebCenter database that is not available via those views, please contact the WebCenter core team.

Names of all WebCenter entities (projects, tasks, groups, ...) are case insensitive. Use 'UPPER' (or *DatabaseScriptlet.getFuncUpper*) when querying the database (not needed for IDs).

6.2. Simple SELECT query

The simplest SQL SELECT query is to select all columns (fields) from the required view.

```
SELECT * FROM webcenter.WCReports_ProjectsQ
```

The above query will fetch all the columns and all the rows from the view and hence there will be too much data. The query will perform slowly and your report will run into many pages. It is always a good idea to limit the number of columns and rows returned to the number really needed for your report.

```
SELECT Project_name, Project_CreatedDate, Project_Manager, Project_Customer
FROM webcenter.WCReports_ProjectsQ
```

The second example will only return the listed 4 columns, but still all projects.

6.2.1. WHERE clause

To limit the number of rows that is returned also, use the **'WHERE'** condition.

```
SELECT Project_name, Project_CreatedDate, Project_Manager, Project_Customer
FROM webcenter.WCReports_ProjectsQ
WHERE Project_StatusName = 'Active'
```

The third line (starting with WHERE) filters the results. You will only get a line in your report for projects with status Active. Mind the quotes around Active.

You can add more conditions in the WHERE clause using either 'AND' or 'OR' to further narrow down the result.

```
SELECT Project_name, Project_CreatedDate, Project_Manager, Project_Customer
FROM webcenter.WCReports_ProjectsQ
WHERE Project_StatusName = 'Active' AND Project_CustomerLegalName = 'Esko'
```

The above example will return all projects with status Active, but limited to the 'Esko' customer only.

6.2.2. ORDER BY clause

You can use 'ORDER BY' clause to get the results ordered ascending (ASC) or descending (DESC) by any column specified in the select clause.

```
SELECT Project_name, Project_CreatedDate, Project_Manager, Project_Customer
FROM webcenter.WCReports_ProjectsQ
WHERE Project_StatusName = 'Active' AND Project_CustomerLegalName = 'Esko'
ORDER BY Project_CreatedDate ASC
```

The returned list of projects will be ordered ascending by the project created date in the above example.

6.3. Aggregate operators

You can also use aggregate operators in the select clause such as SUM, COUNT, AVG, MIN, MAX, etc. Aggregate operators are normally applied to group of rows hence you have to use 'GROUP BY' clause when using such operators.

```
SELECT Task_Assignee, COUNT(*) Completed, SUM(Task_Duration_Days) Duration
FROM webcenter.WcReports_TasksQ
WHERE Task_StatusName = 'Completed'
GROUP BY Task_Assignee
ORDER BY Task_Assignee ASC
```

The above query is to get all the completed tasks grouped by task assignee and then apply aggregate operators on each group to get number of tasks completed by each assignee and how many days taken to complete those many tasks.

6.4. Interesting to know

The report SQL queries might be database dependent. If you need to create 'cross database' custom reports, please refer to [Database Scriptlets](#) section. Reports created by Esko people should be designed as database independent, to be able to easily reuse them for other use cases.

Varia:

- SQL is case insensitive
- Whereas it is good practice to put SELECT, FROM and WHERE each on a line, there is nothing in SQL making this necessary. You could have put all on 1 line or use more lines.
- All strings (for example Active in the above) must be between single quotes.
- You can use wildcards with LIKE and % in the WHERE clause, for example

```
WHERE Project_name LIKE '2013%'
```

This will only show projects which start with 2013.

7. Report fields, Parameters and Variables

In a report there are three groups of objects that can store values:

- the fields,
- the parameters
- the variables.

These objects can be used in the expressions, they can change their value during the print progression and they are typed. They have to be declared in the report template in order to be used.

7.1. Report fields

Fields are elements which get replaced on the fly during the creation of the report. For WebCenter reports, fields will typically contain information coming out of the WebCenter database. In the previous example we used 1 field for each of the project properties shown.

When you add a query to your report (as shown in the first example), the columns of the query will automatically become available as fields.

7.1.1. Usage

Use the syntax

`$F{FieldName}` to refer a field in your report.

Fields can be combined in report expressions. For example,

`$F{First Name} + " " + $F{Last_Name}`

7.2. Parameters

Parameters are named report objects which are used for passing simple input data to the report engine at the time of report execution.

When you look at the default reports in WebCenter, you'll see that when you try running them, you get a list of parameters which influence the content of the report. The most common one is probably to enter starting and ending dates on the content of the report. If you want to show all approvals done in March, then Mar 1 and 31 Mar are the start and end dates. Since you don't want to make a new report template each month (or each week, each day...), you want to give your report parameters.

Important to know:

- You have to define parameters (at design time) before using them.
- A report can declare any number of parameters.
- Parameters have types like Date, Integer or String.
- If you do not supply a value for a parameter, its value is considered to be null, unless you specify a default value expression for the parameter.

- iReport parses parameters in the same order in which they are declared, so if you want to use one parameter in the default value expression for another parameter, you should make sure it is defined above it.

7.2.1. Usage

Parameter can be referred in report expressions using the following syntax: **\$P{ParameterName}**

There are two possible ways of using parameters in the query:

1. Using **\$P{}** in the report query is for situations where the query is fixed at design time and you only wish to inject values into the query before it is executed.

```
SELECT * FROM Orders WHERE CustomerID = $P{Customer}
```

2. You can use parameters to dynamically modify portions of the SQL query or to pass the entire SQL query as a parameter to the report engine. In such a case, the syntax differs a bit. It indicates that the parameter does not have to be dealt as a value, but that it represents a little piece of the query.

```
SELECT * FROM Orders WHERE $P!{CustomerId_Condition}
```

We might pass "CustomerID = 5401" as the parameter value then.

7.2.2. Built In System Parameters

There are also built-in system parameters, ready to be used in expressions for example, **REPORT_PARAMETERS_MAP**, **REPORT_SCRIPTLET**.

They will not be discussed in this document.

7.2.3. Custom Report parameters

For custom parameters you can specify which parameters are internal and should be hidden for the user and which ones the user should be prompted to enter a value. This is controlled via **Use as a Prompt** parameter property.

Sometimes it is handy to create internal parameters to hold values (for example a part of a query) you want to use later in your report at more places and you don't want to repeat it each time or you need to send the value to a sub-report. In this case you should not check the parameter Use as a Prompt flag.

To add a custom report parameter:

1. Select Parameters node in the report inspector view.
2. Right click and select **Add Parameter**.
3. In the properties view specify the parameter properties as described in the table below.

The table below describes the report parameter properties.

Property	Description
Name	Name of the report parameter. It must be unique within the report. It is case sensitive.
Parameter Class	Parameters (and variables as well) are Java objects. Select the Java class from the pre-defined drop down list box. See the section 10.2.2 for the list of supported types by WebCenter.

Property	Description
Use as a Prompt	<p>This flag indicates whether the user should be prompted for the parameter value.</p> <ul style="list-style-type: none"> When clicking on Preview in iReport, it will ask you for a value of the parameter. When running the report from WebCenter, WebCenter will ask you for values via its user interface.
Default Value Expression	<p>Specifies the parameter default value. If you open the report details in WebCenter for the first time, this value will be pre-selected for the parameter there.</p> <p>Example: <code>new Integer(123)</code></p>
Description	<p>A short text description of the parameter. This is used as the parameter label in the WebCenter UI.</p>
Properties	<p>Optional. These are key/value pairs that can be defined for each parameter. There are some WebCenter specific properties described here.</p>

7.2.4. Example Parameters

Example: Use a parameter to change the title

In the report of exercise 1:

- Create a parameter Title:** In the report inspector, expand the parameter field and right-click **Parameters > Add Parameters**
- In the properties panel, change the name to *Title*, keep use as a prompt selected and give a default value expression ***Project Report*** (quotes are needed here). The parameter class (type) should stay `java.lang.String`
- Remove the static text field for the title and put a text field instead. In the text field expression, put

```
$P{Title}
```

Give it layout properties as you like.

When you now preview the report, iReport will ask for a value for the title. Enter a value and you'll see that whatever you have typed will appear as the title.

If you now upload this report to WebCenter, and you run the report, you will see that you are asked for a title. Again, this title will appear as you type it.

Example: Filter the results on status

In the report of exercise 1, we are going to add a parameter Status, which will show a report of project in a certain status (chosen by the WCR operator):

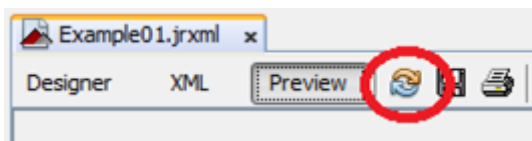
- Create a parameter Status:** In the report inspector, expand the parameter field and right-click **Parameters > Add Parameters**
- In the properties panel, change the name to *Status*, keep use as a prompt selected and give a default value expression ***Active*** (quotes are needed here). The parameter class (type) should stay `java.lang.String`.

- Right click on the report name in the report inspector, Edit Query. Change the WHERE clause to

```
WHERE Project_StatusName = ${Status}
```

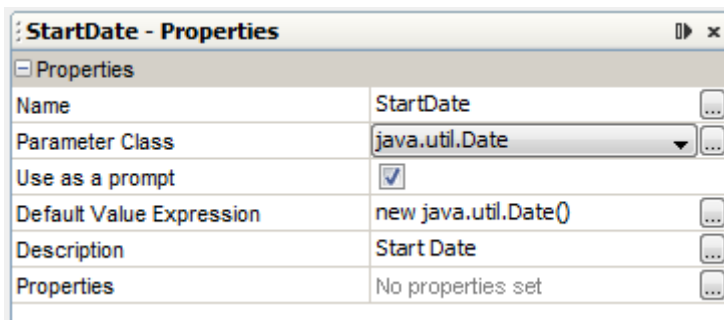
- Preview again, you will be asked for a title and a status. Choose **Use default > the default value (Active)** will be used in your report.
- Preview again but now give *Completed* as status, which shows all completed projects.
- You probably also want to use the parameter in the subtitle (to let it show the status in the subtitle). The expression should be "List of " + $\${Status}$ + " Projects"
- Upload this report to WebCenter and run it from WebCenter. You'll see that you can type now a title and a status.

iReport will not prompt you to enter the values for the parameters if you preview the report again without changing the template. To rerun the report with different parameters, click at the 'Run Again' button next to the 'Preview' button.



Example: Filter the results on date

In the same way, you can add a date parameter, for example the starting date. Remark that the parameter class must be set to `java.util.Date` now.



You can enter a description too. This description will be used in WebCenter aside from the parameter itself.

In the query, you now have to add an 'and' part to the WHERE clause

```
WHERE Project_StatusName = ${Status} and Project_CreatedDate > ${StartDate}
```

In the same way you can also add an end date.

7.3. Variables

Variables are objects used to store the results of calculation such as subtotals, sums, etc. There are two types of variables:

- Built-in

- User Defined Variables.

7.3.1. Usage

You can refer to a variable using the following syntax in an expression: `$V{VariableName}`

7.3.2. Built-In Variables

Jasper Reports provides some built-in variables. You can read these variables, but you cannot modify them.

The following table describes the built-in variables:

Variable	Description
PAGE_NUMBER	Contains the current page number
PAGE_COUNT	Current number of records that have been processed in the current page
COLUMN_NUMBER	Contains the current column number of your report
REPORT_COUNT	Current number of records that have been processed
COLUMN_COUNT	Current number of records that have been processed during the current column (of the current page) creation

We used the **REPORT_COUNT** variable in our first exercise to show the total number of projects in the report summary band.

7.3.3. Custom Variables

To add a variable:

1. Select Variables node in the report inspector view.
2. Right click and select **Add Variable**.
3. In the properties view specify the variable properties as described in the table below.

The table below describes the report variable properties.

Property	Description
Name	Name of the variable. It must be unique within the report.
Variable class	Java type of the variable. Select one from the drop down list.
Calculation	Can be either built-in calculations or you can build a custom expression. The default is Nothing (means don't perform any calculation). It performs the specified calculation by changing the variable's value for every new record that is read from the data source.
Reset type	Specify at which time the variable is reinitialized (using the initial value expression). The default value is Report which means that it is initialized only once at the beginning of the report.

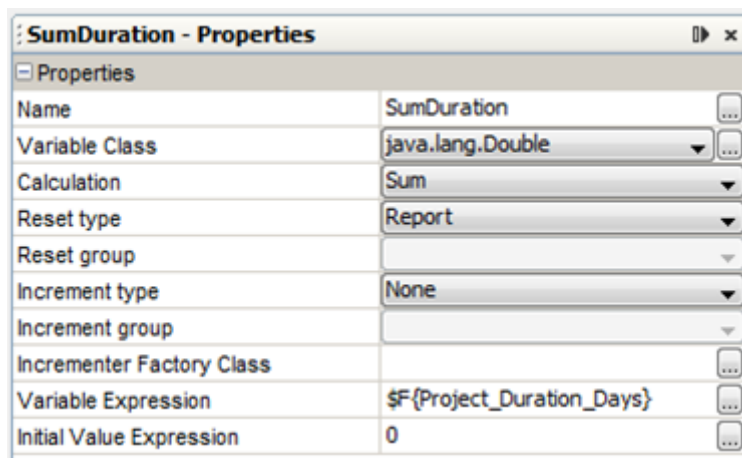
Property	Description
	For example, if we want to calculate the total price in each page, we would choose the page reset type.
Increment type	Specifies when a variable value has to be evaluated. By default, it is updated every time a record is fetched from the data source.
Variable expression	Expression to calculate the value.
Initial value expression	Expression used to set the initial value. If blank, the variable is initialized to null.

7.3.4. Examples for Variables

Make a total for a column

In the report of exercise 1, we are going to add a variable to calculate the sum of the projects duration and print an average project duration time in days.

1. Select Variables in the Report Inspector, right click and select **Add Variable**. In the variable properties window set the following fields:
2. Give it a name "SumDuration".
3. Select "java.lang.Double" for the Variable class.
4. Fill $\$F\{Project_Duration_Days\}$ in the Variable Expression and put initial value to 0.



5. Select "Sum" for the Calculation type.
6. Add another Text Field in the summary band (under the count of projects). Type in the following text field expression:

The average duration of a project is

" + (int)($\$V\{SumDuration\}/\$V\{REPORT_COUNT\}$) + " days

7. Preview the report both in iReport and WebCenter.

8. Custom Reports and WebCenter

Jasper report library is integrated in WebCenter. All the reports that you create and are able to run from iReport, you should be able to run from the WebCenter UI.

They are however some rules to follow to have the integration as smooth for the user as possible.

For more information about how to manage and run custom reports from WebCenter consult the WebCenter documentation.

8.1. Report Template Files

8.1.1. JRXML

The simplest reports only contain a .jrxml file. This is a file describing the layout of the report and where to get the data to fill it with. If you have a report template consisting of a single .jrxml file, you can upload it directly to WebCenter.

8.1.2. ZIP

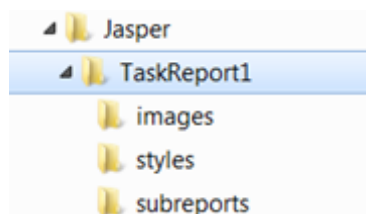
As soon as your report contains images, special styles or sub-reports, you will have more than just a .jrxml file. These extra files must be put in a specific folder structure on your computer and uploaded in a zip format in WebCenter.

Important: The main report jrxml file must sit in the zip root, you will get the following error when uploading the report template to WebCenter otherwise:

"Cannot locate the report definition file. Missing .jrxml file."

Each custom report should be placed in a separate folder at your computer and create the following 3 sub-folders: images, styles and sub-reports as shown in the picture below.

Following are the list of files that can be distributed for every complex report deployed in WebCenter:



File Name	Folder	Comments
Main Report (.jrxml)	<RootFolder>	The root folder contains only the master report template (.jrxml)
Sub Reports (.jrxml)	<RootFolder>/subreports	All the sub-report templates (.jrxml) go here.

File Name	Folder	Comments
Images *.jpg or *.png	<RootFolder>/images	All the images that are used in the report go here.
Styles *.jrtx	<RootFolder>/styles	Style templates

8.2. Report Parameters

As mentioned in the previous sections, the report parameters with "Use as a prompt" property checked will be available to the user in the WebCenter UI when running the report too. The parameter description will be used as the parameter label in the WebCenter UI.

The report parameters appear in the WebCenter UI in the order they are defined in the report template.

Overview of the basic parameter properties can be found in [Fine-tuning your Report](#) on page 51.

8.2.1. WebCenter Report Parameters

There is a common set of parameters each WebCenter custom report should define. The WebCenter report template already contains all the definitions. You can also easily copy and paste the parameters from one report into another within the iReport Report Inspector window.

There is a default value for each parameter, but some of those parameters will be filled in by real values when running from WebCenter (for example the ROOT_DIR). This is marked in the **WCR** column in the overview tables below:

Parameter Name	Default Value	WCR	Description
ROOT_DIR	."	Yes	Report location
STYLE_DIR	`\${ROOT_DIR}+ "/styles/"`	No	Style directory
STYLE_FILE_NAME	StyleTemplate.jrtx	No	Style file name
IMAGE_DIR	`\${ROOT_DIR}+ "/images/"`	No	Images directory path
SUBREPORT_DIR	`\${ROOT_DIR}+ "/subreports/"`	No	Subreports directory path
DbType	"SQLSERVER"	Yes	WebCenter database type (SQLSERVER, ORACLE)

The table below lists additional parameters supported as of WebCenter 12.1. You might need to change the default directory values based on your installation if you want to preview the report in iReport (the correct values can be found in the appconfig.xml).

Parameter Name	Default Value	WCR	Description
REPORT_USER_ID	"00001_0000000001"	Yes	ID of the user who run the report

Parameter Name	Default Value	WCR	Description
WCR_THUMBNAIL_DIR	"C:\\Esko\\Artios\\ \\WebCenter\\FileStore\\ \\thumbnails"	Yes	Location of project thumbnails
WCR_DOCUMENT_DIR	"C:\\Esko\\Artios\\ \\WebCenter\\FileStore\\ \\documents"	Yes	Location of document thumbnails

All the parameters should be of type String and the "Use as a prompt" property should not be checked (we don't want to expose those parameters to the user).

8.2.2. Parameter Types

Based on the parameter type a suitable UI field will be shown to the user to allow him to type in the parameter value.

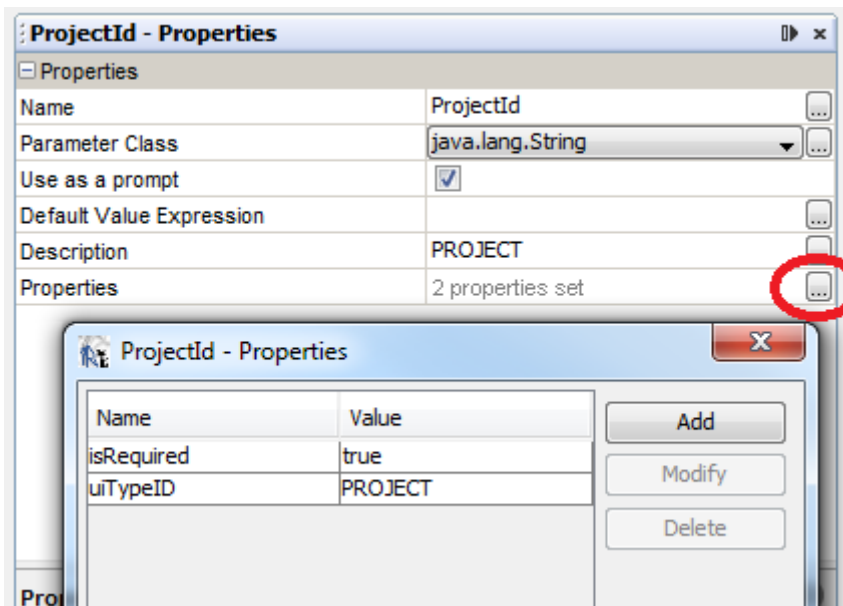
The following table describes the supported parameter types (classes):

Class	UI representation
java.lang.Short	Input with 'number' validation
java.lang.Integer	Input with 'number' validation
java.lang.Long	Input with 'number' validation
java.lang.Float	Input with 'float number' validation
java.lang.Double	Input with 'float number' validation
java.lang.String	Text input
java.lang.Boolean	Checkbox
java.util.Date	Date (and time) selector
java.sql.Timestamp	Date (and time) selector
java.sql.Time	Date (and time) selector

8.2.3. Additional Report Parameter Properties

It is possible to define some additional report parameter properties, for example to link the parameters with WebCenter values. This makes it easier for user to enter values.

To open the additional properties window, click at the expand button in the Properties row.



The following table describes the additional supported report parameter properties:

Name	Class	Value	Description
isRequired	all	true/false	The parameter is marked as mandatory in the UI
uiTypeID	String	See table below	The user is presented with a drop down with available values for the given WCR entity. Also the user selects the entity name, the corresponding WCR id is passed in the parameter.
uiTypeAttrName	String	WCR attribute name	Name or ID of an attribute. The UI representation depends on the corresponding WebCenter attribute type (input, date selector, dropdown with a list of available values for restricted set). Attributes of "Rich Text" type are not supported.
uiTypeAttrID	String	WCR attribute ID	

Only one of the 'uiType' properties should be defined per parameter. If more properties are present, they are taken in the priority as listed in the table above (so 'uiTypeID' has the highest priority).

The following table lists the supported values for the uiTypeID parameter property:

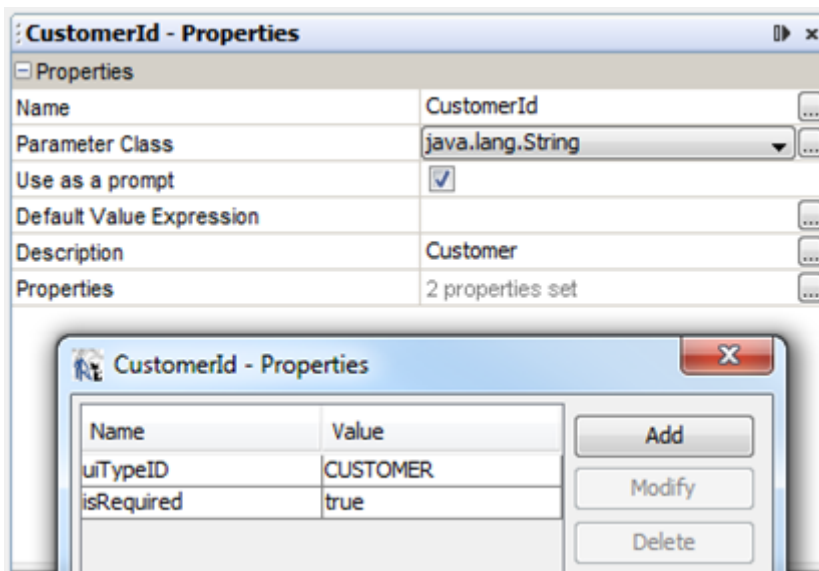
Value	UI representation
USER	Dropdown with a list of users
GROUP	Dropdown with a list of groups
PROJECT	Dropdown with a list of projects
DOCUMENT DOCUMENTVER	Dropdown with a list of project documents. It should be defined together with a parameter of the 'PROJECT' type as the list of documents will be linked to the project selection.
CUSTOMER COMPANY	Dropdown with a list of companies

Value	UI representation
LOCATION	Dropdown with a list of company+location pairs
ATTR	Dropdown with a list of attributes
TASKSPEC	Dropdown with a list of task specifications
TASKTYPE	Dropdown with a list of task types

8.2.4. Example: Linking Customer Parameter

If you would like to make a report per customer, you could make 1 report template for each customer and use the customer name in the WHERE clause. This can be a lot of work. So you can set the customer as a parameter (in the same way as described in the previous examples).

We can make a customer drop-down selector appear in the WebCenter user interface. You do this by linking a parameter to a WebCenter element.



1. Create a parameter CustomerId. With this parameter selected, click on ... in the properties of the parameter, and add a property uiTypeID (mind spelling and uppercases) with value CUSTOMER.
2. You will also have to update the WHERE clause with this parameter

```
WHERE Project_StatusName = ${Status} and Project_CreatedDate > ${StartDate} and
Project_CustomerId = ${CustomerId}
```

3. Also make the new parameter and Status parameter required.
4. Save the report (File/Save) and upload it to WebCenter. You'll now get a customer parameter with a drop-down filled with the WebCenter customers visible to the current user (see the picture below).

Report Parameters

Title: Text

Enter the status. Mind the spelling.:* Text

Start Date:

Customer:*

Report Output

Upload Report to Project:

Folder:

- Please Select...
- Please Select...
- 113588
- BIGLOAD.org
- Banzai
- BP
- Company1
- Company2

If it doesn't work, check the spelling again.

In the same way, you can connect other WebCenter elements.

9. Create your Own Custom Report Template

To create a new custom report, you can start from a blank template, use the template provided by Esko, or just modify the built-in reports. For our first exercise, we will create one from a blank template.

9.1. Blank Template

How to create a new custom report from a blank template is described [Create your first Custom Report](#) on page 22. We do only recommend this way for creating simple reports consisting of a single jrxml file.

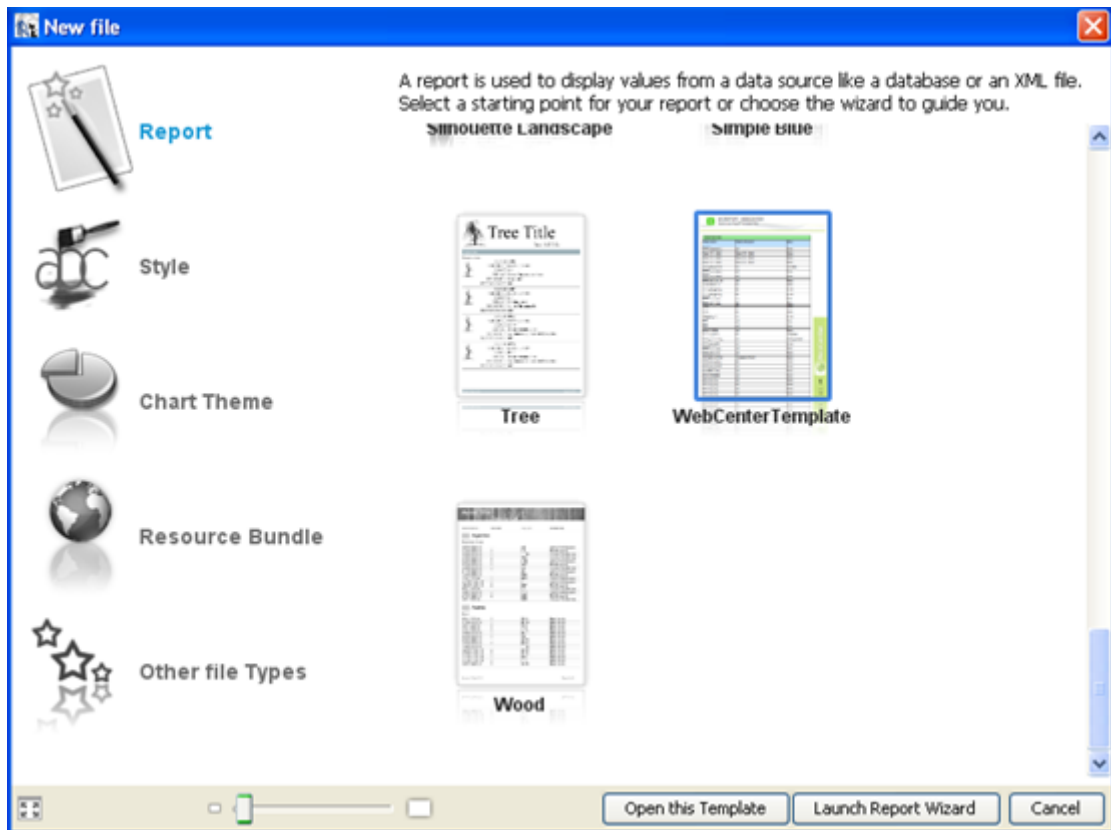
9.2. Using the WebCenter Report Template

For more advanced reports, we have designed a report template which can help you quickly on the road.

- Extract the contents of WebCenterReportTemplate.zip (this is one of the files in the ZIP file you found on help.esko.com) to the "ireport\templates" folder in iReport Installation folder.

If you followed the default installation, this will be C:\Program Files (x86)\Jaspersoft\iReport-4.0.1\ireport\templates

- In iReport, click **File > New**. This opens a dialog, showing various available templates



- Select the **WebCenterTemplate** and click on **Launch Report Wizard** button. Complete the wizard to start with the template. The new report will be created based on the selected template, you may remove stuff that is not needed and start filling up your report contents.
- Once you are done with your report, upload the report (complete zip) to WebCenter.



Note: : If you save your report at a different location, make sure you copy the subfolders (images, styles, subreports) too. It is not done by the wizard.

9.3. Modifying Default Reports

Following is the process to modify the default reports:

- From the WebCenter Admin/Reports overview page, click 'Download Report' next to the report you want to modify
- Save and extract the downloaded zip file on your computer.
- Open the report's .jrxml file with the iReport Designer and customize the report to the desired level. Make sure to use Preview regularly and make intermediate saves. Some of the default reports are rather complex.
- Save from iReport. Go to the location on your hard disk and zip the entire folder (with subfolders). Make sure to exclude the .jasper files though (these are compiled versions you don't need on WebCenter).
- Upload the report (complete zip) to WebCenter. Login as admin user, go to **Admin > Preferences > Reports**
- **In case of updating a report:** Click **Update Report** link on the desired report. The user will be taken to **Update Report** page where user can upload the modified report as a zip. Click **Update**.

- **In case of creating a new report:** Click **Upload Report** button, enter the name and description of the report and then specify the location of the zip and click **Upload**.

On successful creation or update of the report, the corresponding report will be available on the **Reports** overview page.

10. Fine-tuning your Report

Find below some tips that are always good to follow if you create a new custom report.

10.1. Date Formatting

In the simple report, we saw a date formatting like 2/22/13 6:46 PM.

In many cases you want to present the date in a different way. To have consistent date formatting within the report, it is the best to define a private parameter to hold the formatter object and to use this one to format all date fields in the report.

To do so you can use expression like:

```
#{REPORT_FORMAT_FACTORY}.createDateFormat("MMM d, yyyy z", #{REPORT_LOCALE},
    java.util.TimeZone.getTimeZone("GMT"))
```

Date format

The **MMM d, yyyy z** part defines the date/time format. By changing it into other patterns, you'll get different results.

Pattern	Result
'EEE dd-MMM'	Fri 22-Feb
'MMM dd, yyyy kk:mm'	Feb,22 2013 13:45 (this is the standard WebCenter full format with military time)

Locale

The result depends on the chosen locale for the report. That's what **#{REPORT_LOCALE}** does.

When running a report from WebCenter, the locale is passed as a Parameter from WebCenter. You can choose it with the Language setting at the Report Details/Run page.

Run Report

Name: HADW Report 2

Description: created by HADW as part of the Jasper report training prep

Language:

To test in iReport, you can set the locale at Main Menu > Tools > Options > Compilation and Execution > Report Locale

Timezone

The WebCenter server timezone will be used by the reports by default.

The `java.util.TimeZone.getTimeZone("GMT")` part will help you to set a custom (fixed) timezone for the dates in your report.

You can use timezone abbreviations such as "PST", a full name such as "America/Los_Angeles", or a custom ID such as "GMT-8:00". At this moment it is not possible to set the timezone for the report from the WebCenter UI.

Example

In the report template from example 1 we are going to set the format for the project created date.

1. Create a new variable:

- Name: DateFormatter
- Parameter class: `java.text.DateFormat` (this is not available in the dropdown, you have to open the more dialog and type it in manually)
- Default value expression:

```
$P{REPORT_FORMAT_FACTORY}.createDateFormat("MMM d, yyyy z",
$P{REPORT_LOCALE}, java.util.TimeZone.getTimeZone("GMT-2"))
```

2. In the report from the example 1, select the 'Project_CreatedDate' field and click on ... of the text field expression.

3. In the dialog, change the expression to

```
$P{DateFormatter}.format($F{Project_Created Date})
```

4. Make sure the expression class is `java.lang.String`.
5. Preview the report.

This will show the date as Jan 23, 2013 GMT.

You can define more formats for your reports (for example one that will show the date only and another one that will include the hours too).

10.2. Determine What to Display if no date is found

In some cases, a report (or a subreport) will display no data as the (filtered) result is empty. Instead of a white page or section, a text message can be displayed to pass this information to the user.

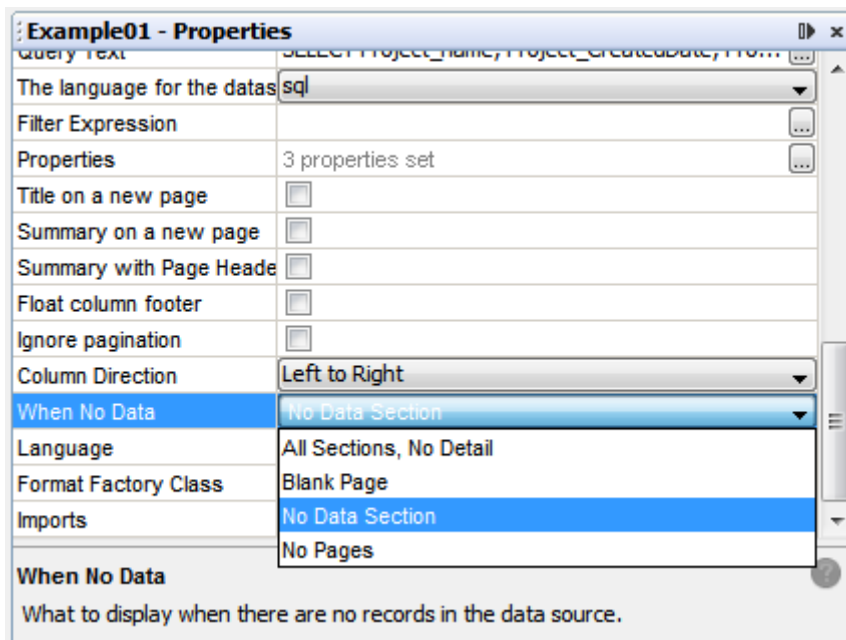
This can be achieved by using the **No Data** band.

Example

1. In the **Report Inspector** right-click on the **No Data** node (it is typically greyed out, but you can still right-click on it). Click **Add Band**. This adds a band to the layout. Whatever you put in that band will be displayed if no data can be displayed for the report. A moving static text can also be added in this section.
2. Put a Text Field in the **No Data** band.
3. In the Text Field Expression type (including the double quotes):

```
"No project with status \" + $P{Status} + "\" found."
```

4. In the report property 'When No Data' select 'No Data Section'.

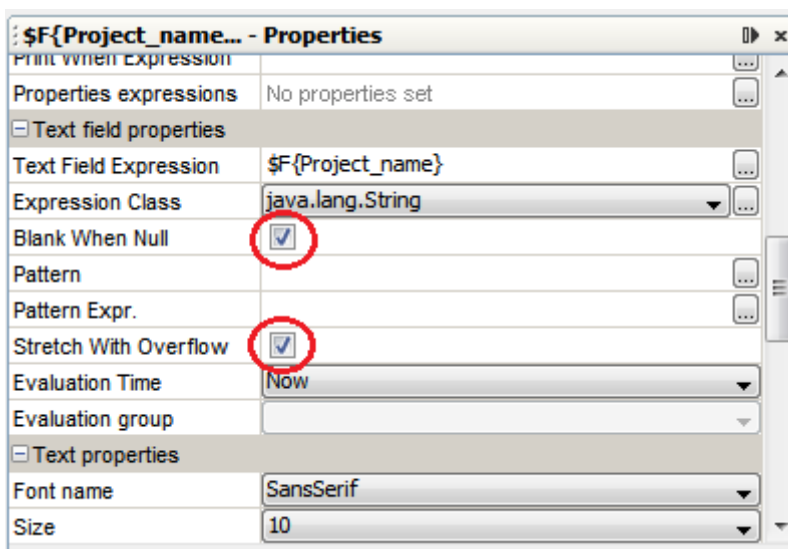


5. Preview the report and enter some non-existing project status for the 'Status' parameter.
6. You will get a report with the message specified in step 3.

10.3. Field Properties

It is always good to check the following field properties:

- Stretch With Overflow – stretch the field vertically if the text does not fit in the element
- Blank When Null – print a blank string instead of null

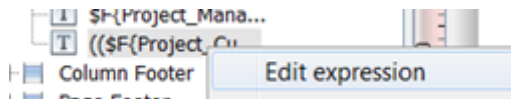


_NULL_values

In the database views, we have used **_NULL_** keyword to indicate empty value (not NULL value). This removes null check dependency for all the Ids.

In this case, Jasper shows "_NULL_" in the report. This is ugly and most of the time you want to replace this with either "None" or just nothing at all (empty space).

The best way to do this is to define the field expression. You can either edit it from the element properties dialog, or by selecting the field in the Report Inspector and right click/Edit expression.



Example 1

In the expression editor, type something like:

```
"" : ${Project_Customer} ((${Project_Customer}.equals("_NULL_")) ? "" :
${Project_Customer})
```

What this expression says is:

- When the field Project_Customer is **_NULL_**, then just put nothing there (what sits between "").
- else, put the Customer's value there

Example 2

The below example will set "No Customer" instead of **_NULL_**:

```
(${Project_Customer}.equals("_NULL_")) ? "No Customer" : ${Project_Customer}
```

10.4. Custom JR Data Source

New data source support is introduced to create custom reports.

- Custom JR DataSource

With the introduction to new data source, we will be able to write our own class that can fetch results from WebCenter backend. To create any custom report that cannot be created from database (eg: Annotation reports), you can write your own Custom Data source class that reads the ROMD files and generate report in the specified Jasper Template.

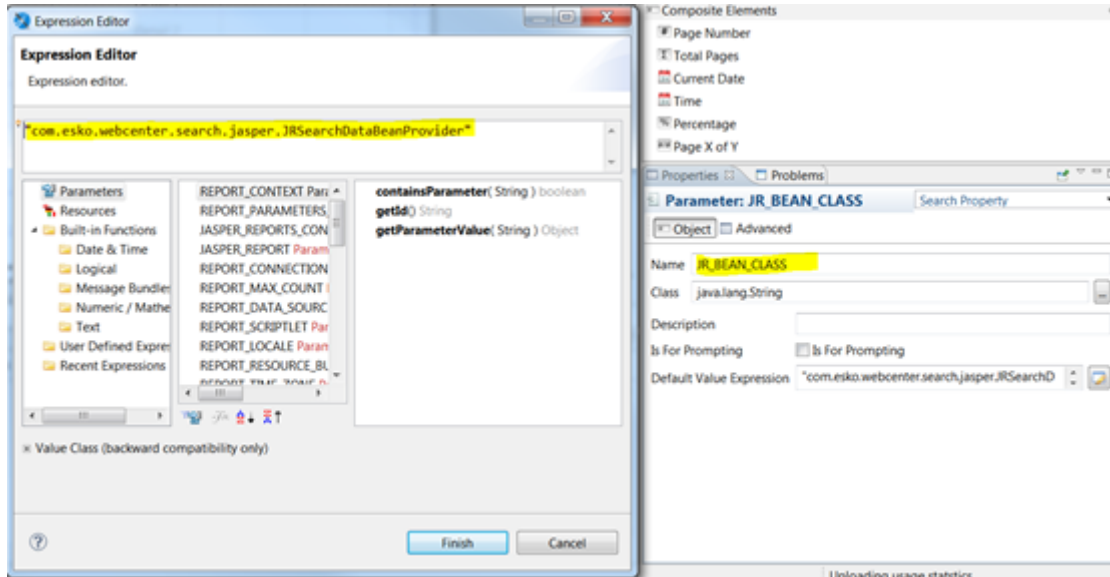
In Jasper template, add the two mandatory parameters to use custom data source.

Parameter Name	Type	Default Value	WCR	Description
JR_BEAN_CLASS	java.lang.String	Jasper data source provider class name	yes	This parameter holds class name of data provider
DATABEAN	java.util.Map	NA	yes	This parameter holds collection report data should be filled in report

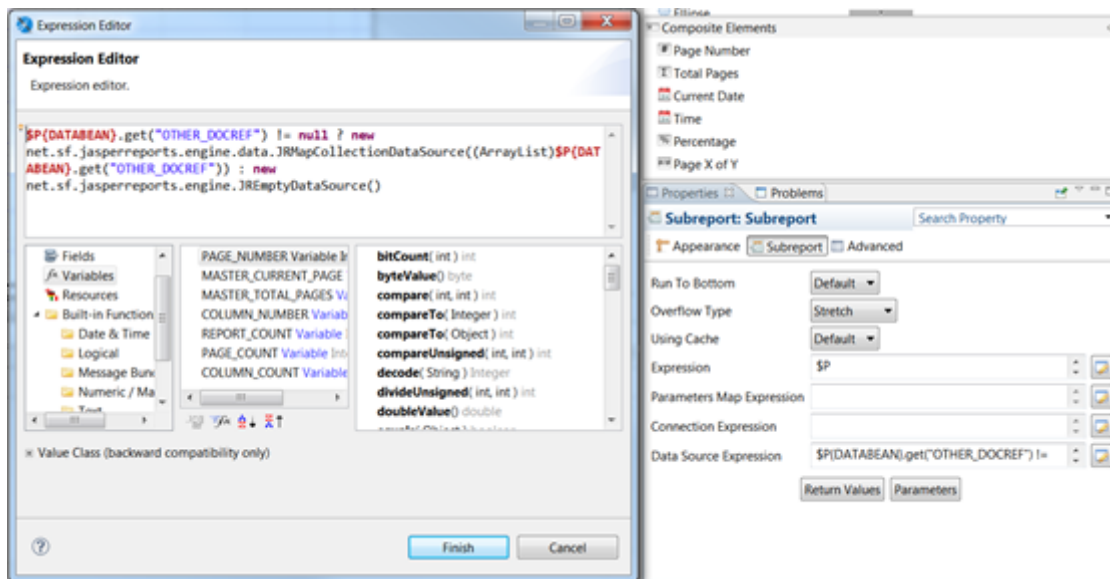
Example: SampleAttributeCategory_PDF

From WebCenter 18 and above, **SampleAttributeCategory_PDF** report is available. This is a sample report to export Attribute Category to PDF. Custom JR Data source class is defined in parameter.

“JR_BEAN_CLASS”.



You can download this template and learn how a template can be created for JR data source. This report has many sub-reports. Each sub-report data source has to be configured according to the key set in “DATABEAN” parameter. Each key in “DATABEAN” will have list of data that will be filled in sub-reports.

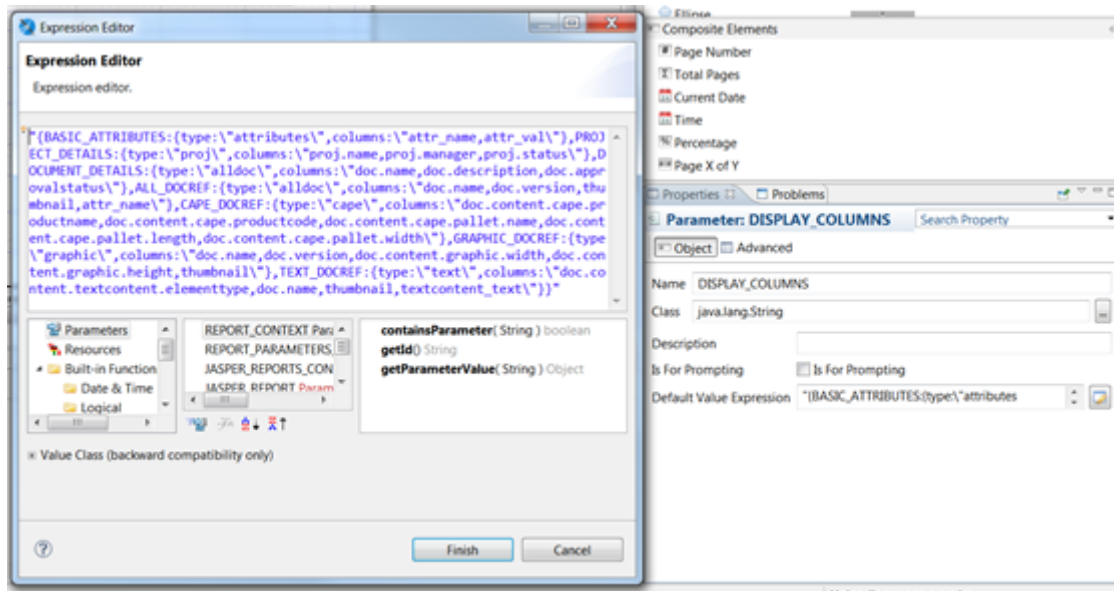


Edit/Update SampleAttributeCategory_PDF Jasper Template

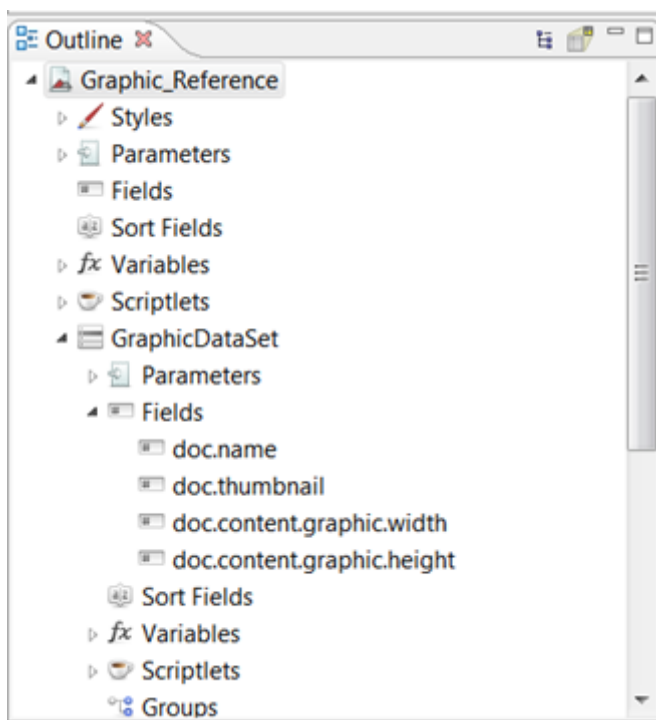
SampleAttributeCategory_PDF is a sample Jasper Template that is installed along with WebCenter 18. If users want to update this Jasper Template to add more columns for different document types, follow the steps below:

- Download the Jasper Template from WebCenter, open it in Jasper Studio.
- Verify each sub-report. If existing sub-report has to be updated for new columns, open the sub-report.


- Refer the section, JRSearchDataBeanProvider which contains all the possible columns that can be used in Jasper Template for different document types.
- Open “DISPLAY_COLUMNS” parameter and update the JSON to add new column name to appropriate “type”.



- Add this field in required sub-report data set.



- Now add this new field to table in report canvas.

Graphic Documents			Title
Thumbnail	Name	Width	Height
	<code>\$F{doc.name}</code>	<code>\$F{doc.content.graphic.width}</code>	<code>\$F{doc.content.graphic.height}</code>

- Save the template, zip it and upload to WebCenter.

11. Advanced Topics

11.1. Dataset

The main report allows only one SQL query to get the report data. You may want the report to show more information in Tables or Charts or List elements. This can be performed by defining the (sub) **dataset** objects.

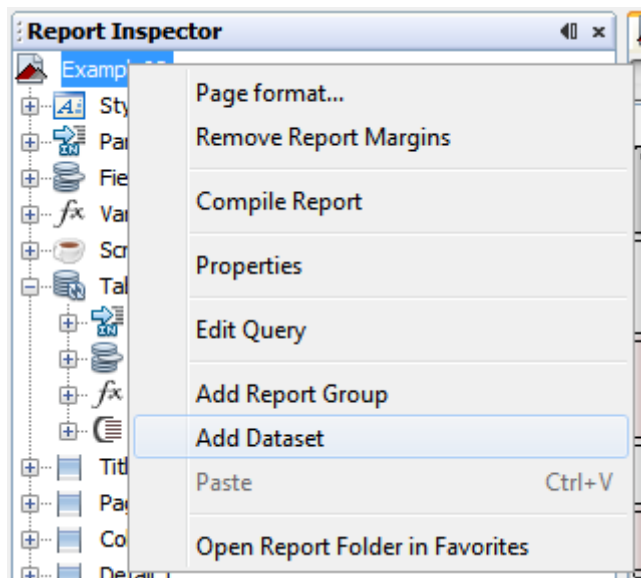
Each dataset has its own query, and the report can have as many datasets as you require. You have to specify a dataset for every table and chart datasource.

Datasets can have parameters, fields, variables and groups just like a report can. The report built-in parameters are available in the dataset by default. You have to explicitly define and pass the custom parameters to the dataset when used as the report element datasource. You have to define the parameters that you want to use in your query only. The same dataset can be used by more report elements, each time passing different parameter values.

Example: Define a dataset

We are going to redo the example 1 using a table for the info in the report detail band.

1. Create a copy of the Example01.jrxml and name it Example02.XML.
2. In the report properties, change the report name to Example 02.
3. Right click at the report root node in the Report Inspector and select **Add Dataset**. The New Dataset wizard will start.

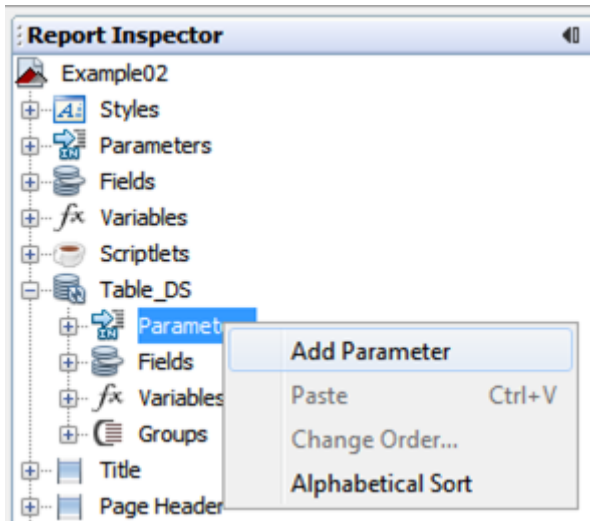


4. Enter **Table_DS** as the dataset name, select create new dataset from a connection or datasource and click **Next**.
5. Type in the below query in the box and click **Next** (we can't use any parameters at this step, so we will have to modify the query later).

```
SELECT Project_name, Project_CreatedDate, Project_Manager, Project_Customer,
Project_Duration_Days
```

```
FROM webcenter.WCReports_ProjectsQ ORDER BY Project_CreatedDate
```

6. Select all fields and click **Next**.
7. Click **Finish**.
8. Select the new dataset parameters node in the Report Inspector, right click and select **Add Parameter**.



9. Define first parameter.
 - Name: Status
 - Parameter Class: java.lang.String
 - Keep default for the rest (not important)
10. Define second parameter.
 - Name: StartDate
 - Parameter Class: java.util.Date
 - Keep default for the rest (not important)
11. You might also need the DateFormatter parameter [Exporting Custom Fonts](#) on page 69.
 - Name: DateFormatter
 - Parameter Class: java.text.DateFormat
12. Select the Table_DS dataset in the Report Inspector, right click and select Edit Query. You should see the two (three) parameters you just defined in the Available parameters list.
13. Modify the query as below, click **OK**.

```
SELECT Project_name, Project_CreatedDate, Project_Manager, Project_Customer,
Project_Duration_Days FROM webcenter.WCReports_ProjectsQ
WHERE Project_StatusName = ${Status} and Project_CreatedDate > ${StartDate} ORDER BY
Project_CreatedDate
```

We will show how to define a table using this data set in the next chapter.

11.2. Using Tables

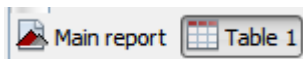
The Table component displays data coming from a secondary dataset. It is an extremely powerful component which can replace the use of subreports.

The Table wizard allows you to create a complex table with few clicks. Each table cell can be a simple text element or it can contain an arbitrary set of report elements including nested tables, creating very sophisticated layouts.

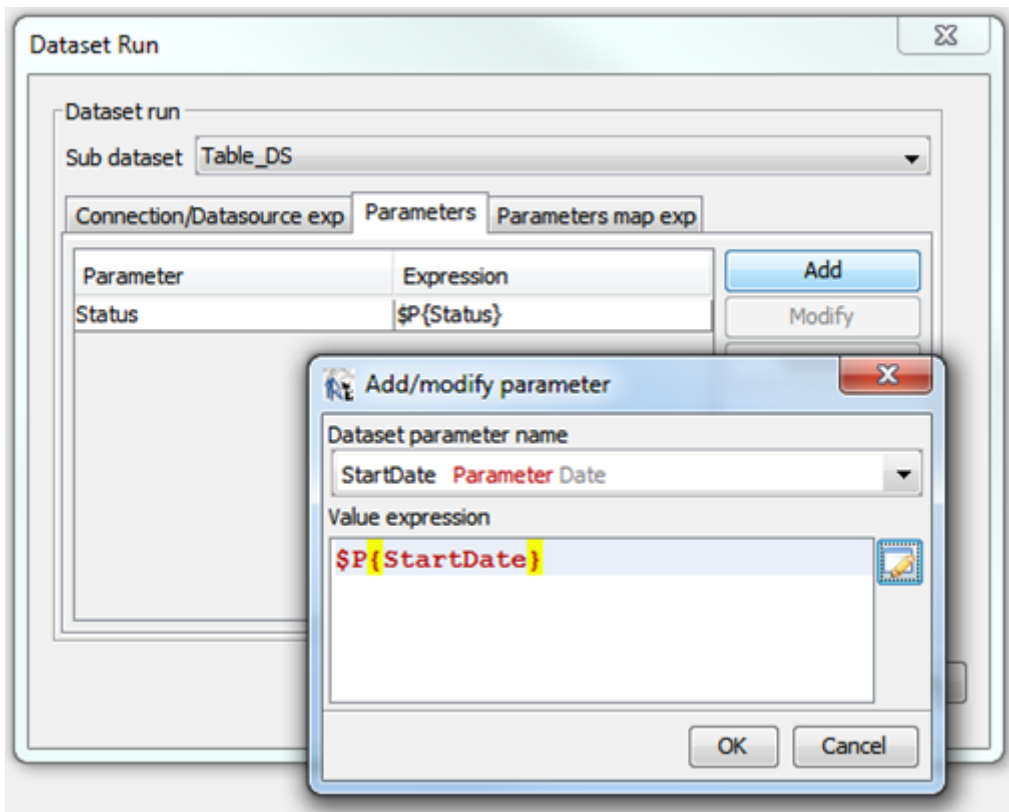
Example: Organize your main report data in a table

In the first example, we have put all the data in fields. We did not have lines around the cells and we are very limited in the formatting of the resulting cells. We can do better by putting our results in a table. Let's continue with the example 2 from the previous chapter.

1. Delete all the fields from the **Column Header** and **Detail 1** bands.
2. Drag and drop a Table element from the Palette to the 'Detail 1' band. Table wizard will start.
3. Select **Create a table from this dataset** and select **Table_DS** (this will be most probably already pre-selected as this is the only dataset in the report). Click **Next**.
4. Select all fields and click **Next**.
5. Select **Use the same connection used to fill the master report**.
6. You can define styles of your table or just uncheck **Create a new set of styles for this table** check box. Click **Finish**.
7. Once the table is created, you can switch between the Main report and your table layout at the bottom of your Designer window.



8. Go to the Main report layout and make sure the table placeholder fits into the band. Otherwise you will get **Warning: Element bottom reaches outside band arealike** error when you try to preview the report.
9. In the Table layout view, you can change the width of columns, change the table headers labels and adapt the field properties as you did in the example 1.
10. Select the Table in the Report Inspector, right click and select **Edit table datasource**.
11. Go the **Parameters** tab and make a link between the two (three) dataset and report parameters.



12. Preview your report.

Example: Add info about the user who ran the report

As of WebCenter 12.1, you can add info about the user who ran the report into your report.

When running the report from WebCenter, Id of the user who ran the report is available via the **REPORT_USER_ID** report parameter.

To add information about the user at the end of your report:

1. Create a new Dataset – e.g. UserDS.

Query:

```
SELECT DISTINCT User_FullName, User_Email, User_MobileNumber FROM WcReports_UsersQ
WHERE User_Id=$P{REPORT_USER_ID}
```

2. Add a new table in the Summary band.
Select the UserDS as the table Sub dataset (don't forget to pass the REPORT_USER_ID parameter).
3. Format the info as you want.

11.3. Using Images

An image is the one of the graphic elements and can be used to insert images (such as GIF, PNG and JPEG images) in the report.

When you drag an image element from the Palette into the Designer, iReport pops up a dialog. iReport will not store the selected image anywhere, it will simply use it from the original location.

Having "absolute path" of the image file will have a problem after deploying report in WebCenter server (or any other machine), since the file may not be found. Therefore we require the two mandatory report parameters – ROOT_DIR and IMAGE_DIR.

The correct image expression property should be set like:

`#{IMAGE_DIR} + "WebCenter_Logo.png"`.

For WebCenter custom reports, you have to place the images, in the 'images' folder [Using Tables](#) on page 59.

Display project thumbnail

As of WebCenter 12.1, you can show a project thumbnail (if available) in your reports too.

To display a project thumbnail in your report, you need to know the project thumbnail file name.

It is stored in the WebCenter database and it is available as '**Project_ThumbnailUrl**' column in the following reporting views:

WcReports_ProjectsQ, WcReports_DocVersionQ, WcReports_TasksQ, WcReports_DocApprovalQ.

So you have to make sure this column is available in your query result and is available as the report field – e.g. `#{Project_ThumbnailUrl}`.

To include a project thumbnail in your report template you have to add an 'Image' element into your report:

1. Don't select any image file (just click 'Cancel' in the browser).
2. Change the image size (WebCenter icons are always squared).
3. Set the following properties:
 - a. Print When Expression: `#{Project_ThumbnailUrl} != null`
 - b. Image expression:


```
#{WCR_THUMBNAIL_DIR} + "/" + #{Project_ThumbnailUrl}
```
 - c. On error type: Blank.

If you want to preview the report in iReport, make sure that you have correctly set the WCR_THUMBNAIL_DIR parameter default value.

Display document thumbnail

As of WebCenter 12.1, you can show a document thumbnail (if available) in your reports too.

To display a project thumbnail in your report, you need to know the project thumbnail file name. It is stored in the WebCenter database and it is available as **Document_ThumbnailUrl** column in the following reporting views: WcReports_DocVersionQ, WcReports_DocApprovalQ.

So you have to make sure this column is available in your query result and is available as the report field.

e.g. `#{Document_ThumbnailUrl}`.

You will need the Document_Id column too.

To include a document thumbnail in your report template you have to add an 'Image' element into your report:

1. Don't select any image file (just click 'Cancel' in the browser).
2. Change the image size (WebCenter icons are always squared).
3. Set the following properties:
 - Print When Expression:


```
#{Document_ThumbnailUrl} != null
```

- Image expression:

```
${WCR_DOCUMENT_DIR} + "/" + ${Document_Id}.substring(6,12) + "/" +
${Document_ThumbnailUrl}
```

- On error type: Blank

If you want to preview the report in iReport, make sure that you have correctly set the WCR_DOCUMENT_DIR parameter default value.



Note: There is only one thumbnail per document in WebCenter which corresponds to the latest version of the document.

11.4. Report Styles

In a report template, you can create a style with a particular font, size, colour, etc. and assign the style to any report element. You can create any number of styles.

It is possible to mark a particular style as **Default**. It makes all the report elements use this style by default if no style assigned to them explicitly.

Conditional Style

You can design your report so that a style changes dynamically. For example, you can set the foreground color of a textfield to black if a particular value is positive and red when it is negative. A style can have an arbitrary number of conditional styles.

In iReport the conditional styles are created by deriving from an existing style, for which we set:

- condition (boolean expression)
- change some properties

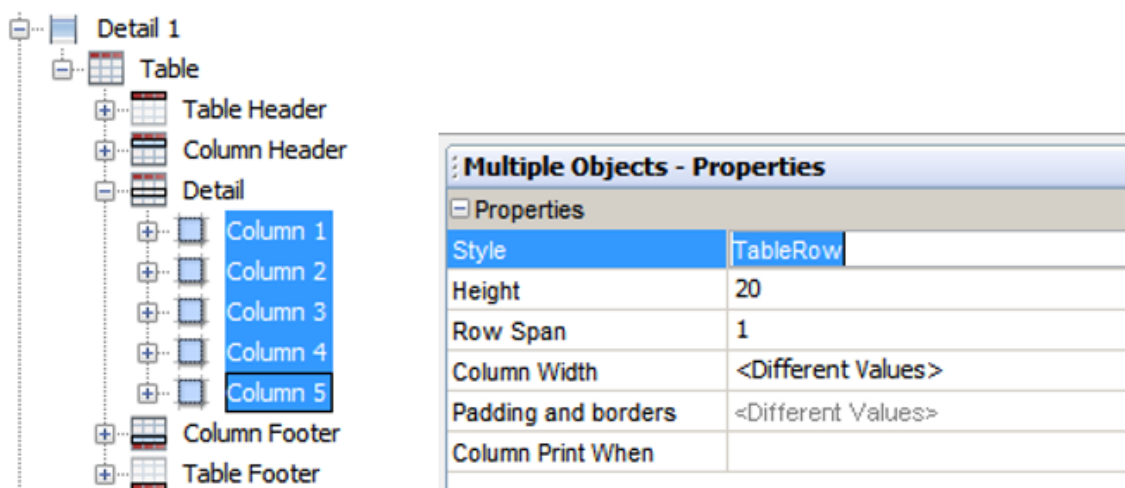
In the condition expression you can use all the properties of the report object.

Example: Conditional style

Let's see an example of using a conditional style to achieve the effect of having an alternating background for odd and even rows. We will continue in extending the example 2.

1. Select the Styles root in the Report Inspector, right click and select **Add > Style**.
2. Give it a name (for example Table Row).
3. Change the Backcolor to white and make the style 'Opaque' (check box).
4. Select the new style in the **Report Inspector**, right click and select **Add conditional style**.
5. In the properties window set the following condition expression (this will be evaluated to true for even rows)


```
new Boolean(${REPORT_COUNT}.intValue()%2==0)
```
6. Change the Backcolor for the style. You can select one from the drop down list or click at the more button to open the color selector. If you have troubles opening the color selector, make sure the property value is empty or white (the first one in the list).
7. You have to check the 'Opaque' checkbox too to make the background color change effective.
8. Now you have to assign the style to all columns in the table detail section. Select the column in the Report Inspector and select the 'TableRow' style in the properties window.



You can preview your report now.

Style templates

Adding styles directly to a report template does have one disadvantage: if we need to add the same style to several reports (incase if you have multiple reports), we need to add the style individually to each and every report. Furthermore, if we need to change a style, then we need to go and modify the style definition in each report template.

Jasper Reports allows us to use **style template**, styles can be defined separately from the report template and reused across several reports.

- A style template is an XML file that contains one or more style definitions.
- Style template files use by convention the *.jrtx extension.
- If same style defined both in report and style template, then local style will be used.

For WebCenter custom reports, you have to place the style templates, in the 'styles' folder.

11.5. Sub Reports

Subreports simplify report design significantly by allowing us to extract complex report sections into a separate report and then incorporating that separate report into a master report.

For WebCenter custom reports, you have to place the sub-reports, in the 'subreports' folder.

By default Report parameters that are defined in the master report are not available to sub-reports, you need to explicitly define and pass the required parameters to the sub-reports.

SumDuration - Properties	
Properties	
Name	SumDuration
Variable Class	java.lang.Double
Calculation	Sum
Reset type	Report
Reset group	
Increment type	None
Increment group	
Incrementer Factory Class	
Variable Expression	#{Project_Duration_Days}
Initial Value Expression	0

11.6. Expressions

Many settings in a report are defined using formulas (such as conditions to hide an element, special calculations, text processing, and so on). An expression is just a formula that operates on some values and returns a result.

Expression language

The default expression language is Java, but you may change it to either JavaScript or Groovy. In all cases, it requires a minimum knowledge of the selected language.

Expression type

When you define an expression, you also need to select the expression type (class) in the properties view. The type of an expression is the nature of the value resulting from it; the type is determined by the context in which the expression is used.

If you are creating an expression that should be displayed in a textfield, it will probably be a String or a number (Integer or Double).

Expression examples

Find below some examples:

```
#{LastName} + ", " + #{FirstName}
#{SpecialOffer}.booleanValue() ? #{SpecialPrice} : #{Price}
#{Customer}.equalsIgnoreCase(" NULL ") ? "All": #{Customer}
#{name}.length() > 50 ? #{name}.substring(0,50) : #{name}
#{MyParam}.equals("true") ? "Yes" : "No"
```

Print when expression

There is an important property called "Print When Expression". This property is useful whenever you want to display something conditionally, that is based on a mathematical or logical condition.

The Print When Expression property can be used to hide or display the band or Report elements under the circumstances described by the expression. The expression must return a Boolean value.

In particular, it must return true to display the band and false to hide it. By default, when no value is defined for the expression, the band is displayed.

For example you may want to show a detail section of a report only if required by the user by setting a report parameter to true.

ShowDetails - Properties	
[-] Properties	
Name	ShowDetails
Parameter Class	java.lang.Boolean
Use as a prompt	<input checked="" type="checkbox"/>
Default Value Expression	true
Description	WCREPORT_SHOW_DETAILS
Properties	One property set

[-] Band properties	
Band height	50
Print When Expression	SP{ShowDetails}
Split Type	<Default>

11.7. How to fill a text field with RTF Content

Sometimes you want to show some already pre-formatted text in your report like HTML or RTF. This can be for example WebCenter RTF attributes, which are stored as BLOBs in the WebCenter database.

11.7.1. Reading BLOB's from the database

BLOB (binary large object) data type is used to store large objects (like images or XML data) in the database.

If you want to work with fields which result from a BLOB type database column, you should make sure that the field (dataset field) type is '`java.io.InputStream`'.

PROJECT_ATTRVALUE_XML - Properties	
[-] Properties	
Name	PROJECT_ATTRVALUE_XML
Field Class	java.io.InputStream
Description	
Properties	No properties set

For **Image** fields, you can directly set the image field expression class to '`java.io.InputStream`' and the image will be read from the BLOB for you.

However for **Text field**, this is not possible. To display data from a BLOB in a text field you have to fill in the following properties:

- Expression class:

```
java.lang.String
```

- Text filed Expression:

```
new java.util.Scanner(${PROJECT_ATTRVALUE_XML}, "UTF-8").useDelimiter("z").next()
```

11.7.2. Markup text field property

By default, the data are displayed as plain text – not interpreted. This might not be very user friendly and wanted to show the xml/html tags. In the report template you can define **Markup** property for a text field.

This attribute allows you to format the text using a specific markup language. The possible values are:

- **None:** No processing on the text is performed (default).
- **Styled:** This markup is capable to format the text using a set of HTML-like tags.
- **HTML:** If you want to print some HTML text into your report – primary use is to format text. The following tags are available: , <u>, <i>, , <sup>, <sub>, ,
. All others are ignored (the tags itself are omitted in the output).
- **RTF:** The content will be interpreted as RTF code.

11.7.3. Reading WebCenter RTF Attributes

As of WebCenter 12.1 it is possible to show RTF attributes in your report too.

WebCenter RTF attributes are stored as BLOBs in the database. There is **Project_AttrValue_UsesXML** ('DocVersion_AttrValue_UsesXML') flag which is saying if the attribute value is stored as BLOB in the **'Project_AttrValue_XML'** ('DocVersion_AttrValue_XML') column, or as String in the **'Project_AttrValue'** ('DocVersion_AttrValue') column.

To read the RTF attributes from a database, you can either use the approach described in [Scriptlets](#) on page 84, or you can use the Project Scriptlet. The **'getProjectAttributeValue'** method will return the attribute value as String regardless its type (so it does the detection of the attribute value type, reading of the BLOB and conversion to String for you).

The best is to use the **html** option for the text field markup property.

You can display the text field based on the attribute value type by setting the 'Print when expression' to:

```
${PROJECT_ATTRVALUE_USESXML} == '1' (or '0')
```

12. Localization Support

12.1. Report Parameters labels in WebCenter UI

You can define a label for the report parameter by specifying the report parameter property called **Description** (see [Report fields, Parameters and Variables](#) on page 36). This string will be localized in the standard WebCenter way.

12.2. Localization Support for the generated Reports

It is assumed that the customers will create reports in the target language already (or create one report per language variant). However, several extensions were made to make it possible to localize the generated reports too. This approach is used for the built-in reports.

Language selector

There is a standard WebCenter language selector available for all reports at the report details page. 'Locale' object is created based on the 'Language' value and passed to the reporting engine. The value can be accessed via **REPORT_LOCALE** parameter in the report itself.

This value is mainly useful for formatting dates (see [Date Formatting](#) on page 51).

Resource Bundle

Jasper engine accepts standard Java property files. Those files should be located in the class path. An extension was made to support UTF-8 encoded property files.

Property files with all WebCenter language strings can be found at the following location:

```
jboss\server\default\deploy\wcr-ear.ear\wcr-ejb-.22.03.jar\languages\.
```



Note: These files will be overwritten when installing a new WebCenter version.

Corresponding property file is loaded based on the 'Language' selector value. The resource bundle base name can be overridden by a value specified in the report template.

In order to localize the report Strings, use the syntax **\$R{Resource Key}**, for example **\$R{WCREPORT_WebCenter}**. At run time, It will look for the key in the resource bundle if it finds the entry use the value specified against the key else it will use the same key (another behavior can be set in the template properties).

13. Exporting Custom Fonts

To use a specific custom font in the custom reports, export the font and bundle it as a **separate jar file**.

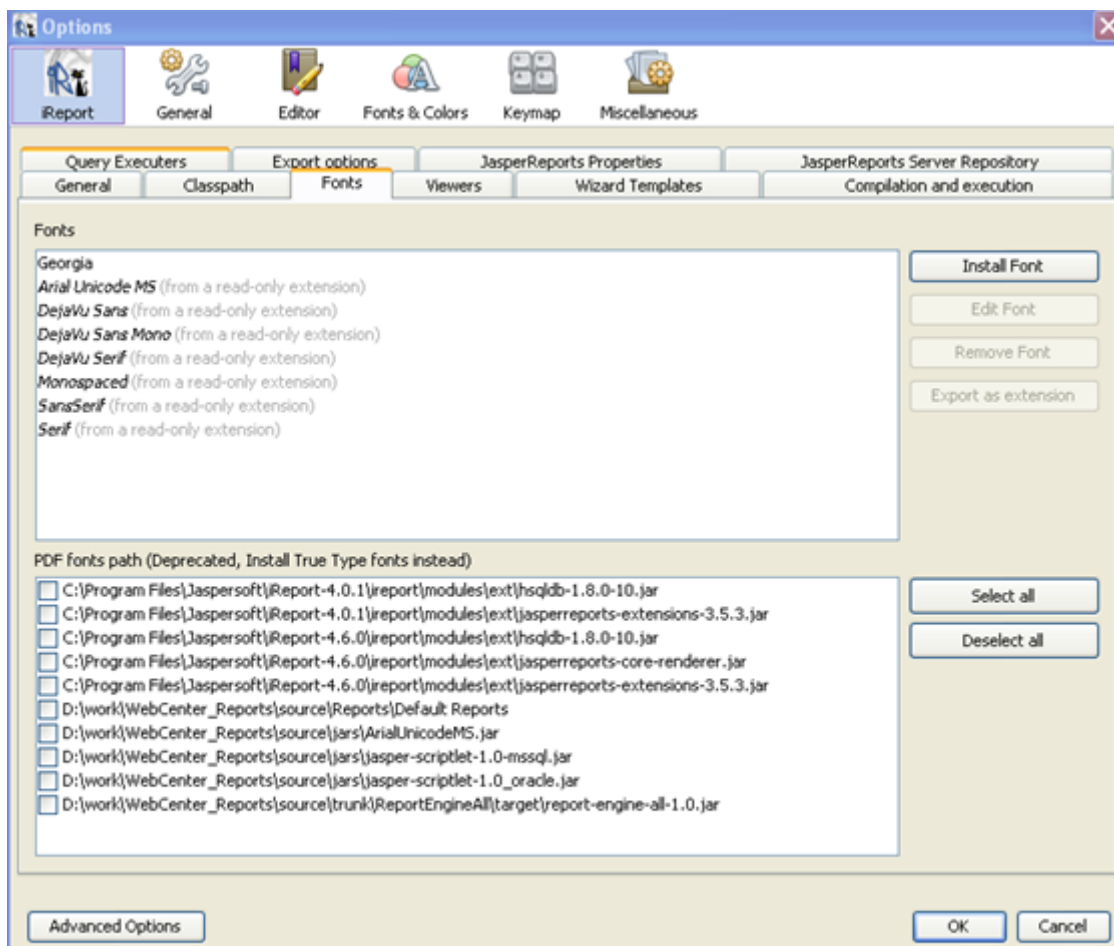
For the WebCenter custom reports, the default font is defined in the **StyleTemplate.jrtx** file. WebCenter overrides the default settings to ignore any missing font.



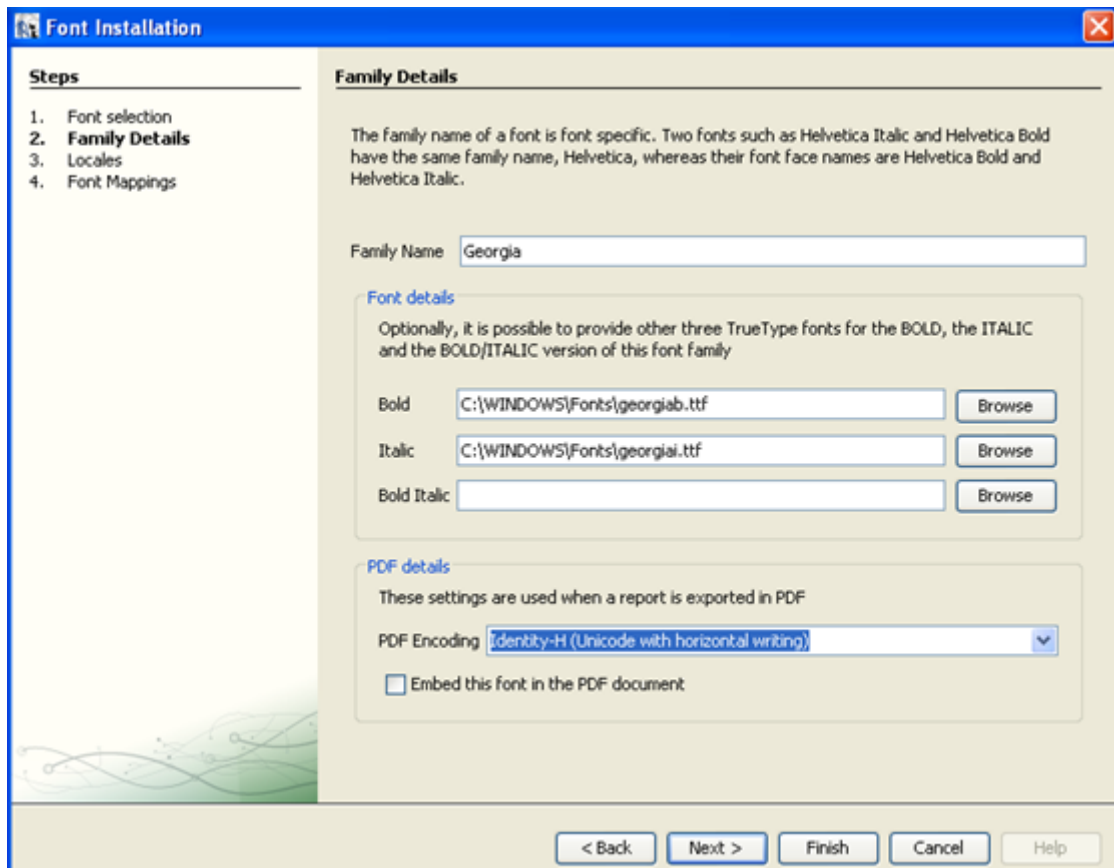
Note: The recommended font for non-Latin-1 languages is 'Arial Unicode MS'. This font should be part of the MS Office distribution.

Steps to export a font into a jar file are as follows:

1. In iReport, go to **Tools -> Options -> Fonts**, First, install fonts in iReport and let it create the required configuration.
2. When you install a font, ensure you install bold and italic version of the font also.



3. Select the PDF Encoding as "**Identity-H (Unicode with horizontal writing)**".



4. After installing the font, the font appears in the fonts section.
5. Choose the location and save the font extension as Jasper_Report_Fonts.jar file.
6. The jar file would contain the selected font and an XML file.
7. Add the exported jar file in the following WebCenter path

ApplicationServer\Jboss\modules\com\esko\webcenter\jasper\main\Jasper_Report_Fonts.jar



Note: In case of upgrading WebCenter into a new version, step 7 should be repeated. So make sure you backup the font jar file.

Please refer the following links for more information about exporting font.

<http://grysz.com/2010/06/01/use-custom-fonts-in-jasperreports-pdf-exporter/>

<http://jasperforge.org/uploads/publish/jasperreportswebsite/trunk/sample.reference/fonts/index.html>

14. Tips and Tricks

What to do with compilation errors

More often you may face Jasper compilation errors in iReport. This may be mostly due to mistakes in report definition which may lead to errors in viewing the preview.

```

iReport console Finished [MyReport2.jrxml]
Compiling to file... C:\Jasper\MyReport2\MyReport2.jasper
Errors compiling C:\Jasper\MyReport2\MyReport2.jasper!
Compilation exceptions: com.jaspersoft.ireport.designer.compiler.ErrorsCollector@1f137c9
net.sf.jasperreports.engine.JRException: Errors were encountered when compiling report express
org.codehaus.groovy.control.MultipleCompilationErrorsException: startup failed:
calculator_MyReport2_1362997955969_652156: 131: unexpected token: Project @ line 131, column 4
1 error
    
```

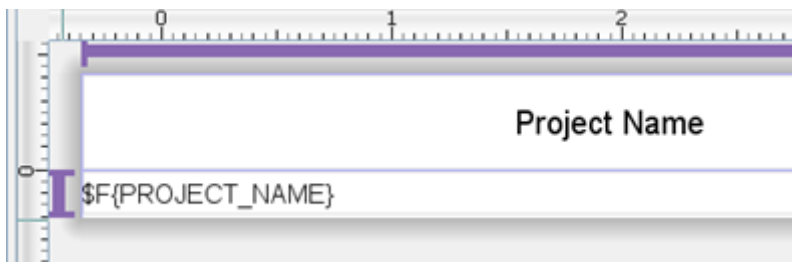


Note: To resolve this

- you can search in @line ... column
- you can revert back to intermediate versions.
- you can always go 1 step back to where it still worked (sometimes it also helps to use Ctrl-Z).

Setting the height of a table row.

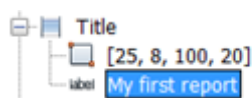
As soon as you have multiple columns, iReport does not allow you to diminish the height lower than the height of other columns. If you want to diminish the row height, you are stuck (since you cannot change the height of all columns together numerically).



Only way found is to drag the height first to as small as possible, then change it numerically in all columns (if you want to have numbers).


How to select an element which sits behind another element

Select it in the report inspector. In the below picture, the label and the rectangle overlap completely. If you want to change the label, you need to select it from the report inspector.



How does the 'Page 1 of 10' trick work?

When using the Page X of Y element, iReport will actually put 2 elements.



The first writes out Page 1 of and the second one puts the total number of pages. However, they both seem to have the same variable `#{PAGE_NUMBER}`. Why does the one print the current page while the other prints the total number of pages?

For this you have to look at the corresponding Text Field properties. The first one has the property **Evaluation Time** set to 'Now', while the second one 'Report'. It means that the expression will be evaluated at the end of the report (so it basically puts the page number of the last page on each page).

Which database view should I use to get approver information?

The `WcReports_DocApprovalQ` view contains all information about the document (version) approval status, but it doesn't include information about the users (groups, roles) who should do/did the approval.

To get information about the submitted approvals you can use the following WebCenter internal view: **Submitted_ApprovalsQ**.

To get information about the pending approvals, you can use **User_ToDo_ListQ** view

What query to use to get the documents included in the project bill of material (BOM)?

Bill of Material concept has been introduced in WebCenter 12.1. To get a list of documents included in the project BOM, you can use the following SQL query:

```
SELECT * FROM WcReports_DocVersionQ WHERE Project_Id = ${ProjectId} AND Project_Bom_Id != '_NULL_'
ORDER BY Project_BOM_Order
```

To get the document version revision (ACAD feature) properly formatted, you can use the following text field expression (the field names are case sensitive, so make sure they match the names defined in your report):

```
#{DocVersion_Revision } > 0 ? #{DocVersion_Counter} +
"(" + new String((char)(65 + #{DocVersion_Revision}) - 1) + ")" :
#{DocVersion_Counter
```

My report shows January 1, 1970 as the project due date in case no due date is set

The data type used for dates stores the value as number of milliseconds since January 1, 1970. Therefore no date (0/null) is interpreted as January 1, 1970.

```
#{ Project_DueDate }.getYear() > 70 ? ${DateFormatter}.format(#{Project_DueDate}) : ""
```



Note: Read more about date formats here [Date Formatting](#) on page 51

15. WebCenter Database Interface

Report Queries should use WebCenter views to get the required data. All the reporting views starts with **WcReports_** prefix. If you need other information from the WebCenter database that is not available via those views, please contact the WebCenter team.

In the database views, we have used **_NULL_** keyword to indicate empty value not NULL value. This removes null check dependency for all the Ids, we replaced null with **_NULL_**.

The following is a list of reporting views available in WebCenter. Columns in **bold** are only available as of WebCenter22.03.

WcReports_UsersQ

There is one row per user/group combination in this view.

Column Name	Type	Description
User_Id	String	User Id
User_Name	String	User Name
User_CustomerId	String	User company Id
User_CustomerLocationId	String	User company location Id
User_CustomerLocation	String	User company location name
User_CustomerLocationShort	String	User company location short name
User_Customer	String	User company name
User_CustomerLegalName	String	User company legal name
User_FullName	String	Full name – Smith, John (JOSM)
User_FirstName	String	First Name
User_LastName	String	Last Name
User_Email	String	Email
User_PhoneNumber	String	Phone Number
User_MobileNumber	String	Mobile Number
User_OwnerId	String	Owner Id: Id of the user who has created this user
User_Owner	String	Owner full name
User_Function	String	Function
User_IsActive	Integer (0/1)	If User is active
User_IsRegistered	Integer (0/1)	If User is a registered WebCenter User
User_GroupName	String	Group Name
User_GroupId	String	Group Id
User_GroupManager	Integer (0/1)	If the user is a group manager of this group

This view is a join of the following tables: Users, Users_In_Group, Groups and Locations.

WcReports_ProjectsQ

There is one row per project in this view.

Column Name	Type	Description
Project_Id	String	Project Id
Project_Name	String	Project Name
Project_Description	String	Description
Project_ThumbnailUrl	String	Project Thumbnail File Name
Project_CreatedDate	Timestamp	Project Created Date
Project_ModifiedDate	Timestamp	Project Modified Date
Project_CreatorId	String	Project Creator Id
Project_Creator	String	Project Creator Name
Project_CreatorDelegatId	String	Project Creator Delegate Id
Project_CreatorDelegate	String	Project Creator Delegate Name
Project_DueDate	Timestamp	Project Due Date
Project_StatusId	String	Project Status Id
Project_StatusName	String	Project Status
Project_IsActive	Integer (0/1)	Is Project Active
Project_ManagerIsGroup	Integer (0/1)	Is Project Manager a group
Project_ManagerId	String	Project Manager Id
Project_Manager	String	Manager Name (group/user)
Project_SalesPersonId	String	Sales Person Id
Project_SalesPerson	String	Sales Person
Project_CustomerId	String	Customer Id
Project_CustomerLocationId	String	Customer Location Id
Project_CustomerLocation	String	Customer Location Name
Project_CustomerLocationShort	String	Customer Location Short Name
Project_Customer	String	Customer Name
Project_CustomerLegalName	String	Customer Legal Name
Project_Duration_Days	Double	Project Duration
Project_SourceProjectId	String	Source Project Id
Project_SourceProjectName	String	Source Project Name
Project_SourceTemplateId	String	Source Template Id
Project_SourceTemplateName	String	Source Template Name
Project_AttrCatId	String	Project Attribute Category Id

This view is a join of the following tables: Projects, Project_Statuses, Companies and Locations.

WcReports_Project_AttrQ

There is one row per project attribute value in this view.

Column Name	Type	Description
Project_Id	String	Project Id
Project_AttrId	String	Project Attribute Id
Project_AttrName	String	Attribute Name
Project_AttrLanguageTag	String	Attribute Language Tag
Project_AttrValue	String	Attribute Value
Project_AttrValue_UsesXML	Integer (0/1)	If the attribute value is stored in the 'Project_AttrValue_XML' column (used for RTF attributes for example)
Project_AttrValue_XML	Blob	Attribute Value Blob
Project_AttrUnitId	String	Attribute Unit Id
Project_AttrUnitName	String	Attribute Unit Name
Project_AttrUnitType	Integer	Attribute Unit Type
Project_AttrCatId	String	Project Attribute Category Id

This view is a join of the following tables: Projects, Attribute_Category_Map, Attributes, Locations and Project_Attributes.

WcReports_Project_CharQ

There is one row per project characteristic in this view.

Column Name	Type	Description
Project_Id	String	Project Id
Project_CharId	String	Characteristic Id
Project_CharName	String	Characteristic Name

This view is a join of the following tables: Projects, Project_Characteristics and Characteristics.

WcReports_DocVersionQ

There is one row per document version/project pairs in this view

Column Name	Type	Description
Document_Id	String	Document Id
Document_Name	String	Document Name
Document_Description	String	Document Description
Document_ThumbnailUrl	String	Document Thumbnail File Name
Document_LastVersionIndex	Integer	Document Last Version Index
Document_LastRevision	Integer	Document Last Revision
DocVersion_Id	String	Document Version Id

Column Name	Type	Description
DocVersion_Counter	Integer	Document Version Number
DocVersion_Revision		Document Version Revision (ACAD)
DocVersion_FileSize	Integer	Document Version Size
DocVersion_CreatedDate	Timestamp	Document Version Created Date
DocVersion_Description	String	Document Version Description
DocVersion_AuthorId	String	Document Version Author Id
DocVersion_Author	String	Document Version Author
DocVersion_AuthorOwnerId	String	Document Version Author Owner Id
DocVersion_DelegateId	String	Document Version Author Delegate Id
DocVersion_Delegate	String	Document Version Author Delegate Full Name
DocVersion_OrigFileName	String	Document Version Original File Name
DocVersion_UploadDocType	String	Document Version Document Type
DocVersion_ApprovalStatusId	String	Doc Version Approval Status Id
DocVersion_AttrCatId	String	Document version Attribute Category Id
Project_Id	String	Project Id
Project_Name	String	Project Name
Project_Description	String	Description
Project_ThumbnailUrl	String	Project Thumbnail File Name
Project_CreatedDate	Timestamp	Project Created Date
Project_ModifiedDate	Timestamp	Project Modified Date
Project_DueDate	Timestamp	Project Due Date
Project_StatusId	String	Project Status Id
Project_StatusName	String	Project Status
Project_IsActive	Integer	Is Project Active
Project_ManagerIsGroup	Integer (0/1)	Is Project Manager a group
Project_ManagerId	String	Project Manager Id
Project_Manager	String	Manager Name
Project_SalesPersonId	String	Sales Person Id
Project_SalesPerson	String	Sales Person
Project_CustomerId	String	Customer Id
Project_CustomerLocationId	String	Customer Location Id
Project_CustomerLocation	String	Customer Location Name
Project_CustomerLocationShort	String	Customer Location Short Name

Column Name	Type	Description
Project_Customer	String	Customer Name
Project_CustomerLegalName	String	Customer Legal Name
Project_BOM_Id	String	Bill of Materials Id (if the document version is part of BOM)
Project_BOM_Order	Integer	Bill of Materials Order
Project_BOM_Count	Float	Bill of Materials Count
Project_BOM_Details	Blob	Bill of Materials Details (ACAD)
Project_FolderId	String	Project Folder Id
Project_FolderName	String	Project Folder Name
Project_CreatorId	String	Project Creator Id
Project_Creator	String	Project Creator Name
Project_CreatorDelegatId	String	Project Creator Delegate Id
Project_CreatorDelegate	String	Project Creator Delegate Name

This view is a join of the following tables: Doc_Version, Documents, Users, Project_Docs, Projects, Project_Statuses, Companies and Locations, Project_Doc_BOM, Project_Folder.

WcReports_DocVersion_AttrQ

There is one row per document (version) attribute value in this view.

Column Name	Type	Description
DocVersion_Id	String	Document Version Id
Document_Id	String	Document Id
Project_Id	String	Project Id
DocVersion_AttrId	String	Attribute Id
DocVersion_AttrName	String	Attribute Name
DocVersion_AttrLanguageTag	String	Attribute Language Tag
DocVersion_AttrValue	String	Attribute Value
DocVersion_AttrValue_UsesXML	Integer (0/1)	If the attribute value is stored in the 'Project_AttrValue_XML' column (used for RTF attributes for example)
DocVersion_AttrValue_XML	Blob	Attribute Value Blob
DocVersion_AttrUnitId	String	Attribute Unit Id
DocVersion_AttrUnitName	String	Attribute Unit Name
DocVersion_AttrCatId	String	Document version Attribute Category Id
DocVersion_AttrUnitType	Integer	Attribute Unit Type
Project_FolderId	String	Project Folder Id

This view is a join of the following tables: Doc_Version, Documents, Project_Docs, Attribute_Category_Map, Attributes, Doc_Attributes and Attrib_units.

WcReports_Document_CharQ

There is one row per document characteristic in this view.

Column Name	Type	Description
Document_Id	String	Document Id
Project_Id	String	Project Id
Document_CharId	String	Characteristic Id
Document_CharName	String	Characteristic Name
Project_FolderId	String	Project Folder ID

This view is a join of the following tables: Documents, Project_Docs, Doc_Characteristics and Characteristics.

WcReports_TasksQ

There is one row per project task in this view.

Column Name	Type	Description
Task_Id	String	Task Id
Task_Name	String	Task Name
Task_Description	String	Task Description
Task_StartedDate	Timestamp	Task Start Date
Task_CompletedDate	Timestamp	Task Completed Date
Task_DueDate	Timestamp	Task Due Date
Task_EstimatedDate	Timestamp	Task Estimated Date
Task_ModifiedDate	Timestamp	Task Modified Date
Task_LeadTime	Integer	Task Lead Time
Task_StatusId	String	Task Status Id
Task_StatusName	String	Task Status
Task_StatusType	Number	Task Status Type 1,2 – Not started/Active 3 – Completed 4 – Failed (WF cancelled, WF error)
Task_AssigneelsGroup	Integer (0/1)	Task Assignee is group
Task_Assigneeld	String	Task Assignee Id
Task_Assignee	String	Task Assignee Full Name
Task_Delegateld	String	Task Assignee Delegate Id
Task_Delegate	String	Task Assignee Delegate Full Name
Task_PreviousAssigneelsGroup	Integer (0/1)	Task Previous Assignee is group

Column Name	Type	Description
Task_PreviousAssigneeId	String	Task Previous Assignee Id
Task_PreviousAssignee	String	Task Previous Assignee Full Name
Task_CreatorId	String	Task Creator Id
Task_Creator	String	Task Creator
Task_CreatorDelegateId	String	Task Creator Delegate Id
Task_CreatorDelegate	String	Task Creator Delegate Full Name
Task_TypeId	String	Task Type Id
Task_TypeName	String	Task Type Name
Task_TypeDescription	String	Task Type Description
Task_Duration_Days	Double	Task Duration Days
Task_MasterId	String	Id of the Master Task (for subtasks)
Task_HasWorkflow	Integer (0/1)	If the tasks has assigned a workflow
Task_IsSigned	Integer (0/1)	If the task was completed with a signature
Task_CreatedDate	Timestamp	Task Created Date
Project_Id	String	Project Id
Project_Name	String	Project Name
Project_Description	String	Description
Project_ThumbnailUrl	String	Project Thumbnail File Name
Project_CreatedDate	Timestamp	Project Created Date
Project_ModifiedDate	Timestamp	Project Modified Date
Project_DueDate	Timestamp	Project Due Date
Project_StatusId	String	Project Status Id
Project_StatusName	String	Project Status
Project_IsActive	Integer (0/1)	Is Project Active
Project_ManagerIsGroup	Integer (0/1)	Is Project Manager a group
Project_ManagerId	String	Project Manager Id
Project_Manager	String	Manager Name
Project_SalesPersonId	String	Sales Person Id
Project_SalesPerson	String	Sales Person
Project_CustomerId	String	Customer Id
Project_CustomerLocationId	String	Customer Location Id
Project_CustomerLocation	String	Customer Location Name
Project_CustomerLocationShort	String	Customer Location Short Name
Project_Customer	String	Customer Name

Column Name	Type	Description
Project_CustomerLegalName	String	Customer Legal Name
Project_CreatorId	String	Project Creator Id
Project_Creator	String	Project Creator Name
Project_CreatorDelegatId	String	Project Creator Delegate Id
Project_CreatorDelegate	String	Project Creator Delegate Name

This view is a join of the following tables: Project_WcTasks, Projects, WC_States, WcTask_Type, Project_Statuses, Companies and Locations.

WcReports_Task_SpecificationQ

There is one row per task specification in this view.

Column Name	Type	Description
Task_Id	String	Task Id
Project_Id	String	Project Id
Task_SpecId	String	Task Specification Id
Task_SpecName	String	Specification Name
Task_SpecLanguageTag	Short	Specification Language Tag
Task_SpecValue	String	Specification Value
Task_SpecUnitId	String	Specification Unit Id
Task_SpecUnitName	String	Specification Unit Name
Task_SpecUnitType	Integer	Specification Unit Type

This view is a join of the following tables: Project_WcTasks, Projects, Project_WcTask_Specifications, Attributes and Attrib_units.

WcReports_DocApprovalQ

Column Name	Type	Description
Document_Id	String	Document Id
Document_Name	String	Document Name
Document_Description	String	Document Description
Document_ThumbnailUrl	String	Document Thumbnail File Name
Document_LastVersionIndex	Integer	Document Last Version Index
Document_LastRevision	Integer	Document Last Revision
DocVersion_Id	String	Document Version Id
DocVersion_Counter	Integer	Document Version Number
DocVersion_FileSize	Integer	Document Version Size
DocVersion_CreatedDate	Timestamp	Document Version Created Date
DocVersion_Description	String	Document Version Description

Column Name	Type	Description
DocVersion_AuthorId	String	Document Version Author Id
DocVersion_Author	String	Document Version Author
DocVersion_DelegateId	String	Document Version Author Delegate Id
DocVersion_Delegate	String	Document Version Author Delegate Full Name
DocVersion_OrigFileName	String	Document Version Original File Name
DocVersion_UploadDocType	String	Document Version Document Type
DocVersion_ApprovalStatusId	String	Document Version Approval Status Id
DocVersion_ApprovalStatus	String	Doc Version Approval Status
DocVersion_AttrCatId	String	Document version Attribute Category Id
Project_Id	String	Project Id
Project_Name	String	Project Name
Project_Description	String	Description
Project_ThumbnailUrl	String	Project Thumbnail File Name
Project_CreatedDate	Timestamp	Project Created Date
Project_ModifiedDate	Timestamp	Project Modified Date
Project_DueDate	Timestamp	Project Due Date
Project_StatusId	String	Project Status Id
Project_StatusName	String	Project Status
Project_IsActive	Integer (0/1)	Is Project Active
Project_CustomerId	String	Customer Id
Project_CustomerLocationId	String	Customer Location Id
Project_CustomerLocation	String	Customer Location Name
Project_CustomerLocationShort	String	Customer Location Short Name
Project_Customer	String	Customer Name
Project_CustomerLegalName	String	Customer Legal Name
Project_FolderId	String	Project Folder Id
Project_FolderName	String	Project Folder Name
ApprovalInfo_StartedDate	Timestamp	Approval Started Date
ApprovalInfo_EndDate	Timestamp	Approval End Date
ApprovalInfo_DueDate	Timestamp	Approval Due Date
ApprovalInfo_ApprovalState	String	Approval State
ApprovalInfo_Duration_Days	Double	Approval Duration
ApprovalStage_Id	String	Approval Stage Id
ApprovalStage_ApprovalRuleId	String	Approval Rule Id

Column Name	Type	Description
ApprovalStage_StageName	String	Approval Stage Name
ApprovalStage_StageCounter	Integer	Approval Stage Counter
ApprovalStage_DueDate	Timestamp	Approval Stage Due Date
ApprovalStage_LeadTime	Integer	Approval Stage Lead Time
DocVersion_IsApproved	Integer (0/1)	Is document approved
Project_CreatorId	String	Project Creator Id
Project_Creator	String	Project Creator Name
Project_CreatorDelegatId	String	Project Creator Delegate Id
Project_CreatorDelegate	String	Project Creator Delegate Name

This view is a join of the following tables: Doc_Version, Documents, Projects, Project_Docs, Project_Statuses, Project_Folder, Companies, DocVersion_Approval_Info, Approval_Stage, Approval_Type and Locations.

WcReports_UserHistoryQ

Function Name	Input Parameters	Description
UserHistory_Id	String	User History record Id
UserHistory_EventId	String	Event Id
UserHistory_EventUserId	String	User ID
UserHistory_EventUser	String	User Name
UserHistory_OperatorId	String	Operator ID
UserHistory_Operator	String	Operator Name
UserHistory_DelegatorId	String	Delegator ID
UserHistory_Delegator	String	Delegator Name
UserHistory_EventDate	Time Stamp	Event Date
UserHistory_NewValue	String	Event Detailed Message
UserHistory_EventName	String	Event Name
UserHistory_EventLangStr	String	Event Language String
UserHistory_EventDisplayStr	String	Event Display String

This view is a join of the following tables: User_Management_History, WC_Events, Project_Folder.

Database Functions

The following are the list of database functions used in default reports:

Function Name	Input Parameters	Description
get_user_fullname	User_id (String)	It returns user's full name.
is_approval_status_approved	Status_id (String)	It returns either 0 or 1

Function Name	Input Parameters	Description
get_customer_fullname	Company_id (String) Location_id (String)	It returns customer's full name
get_business_days	StartTime (DateTime) EndTime (DateTime)	It returns number of business between the given dates (integer)
get_WCTASK_TYPE_name	Task_type_id (String)	It returns Task Type Name
get_group_name	Group_id (String)	It returns Group Name

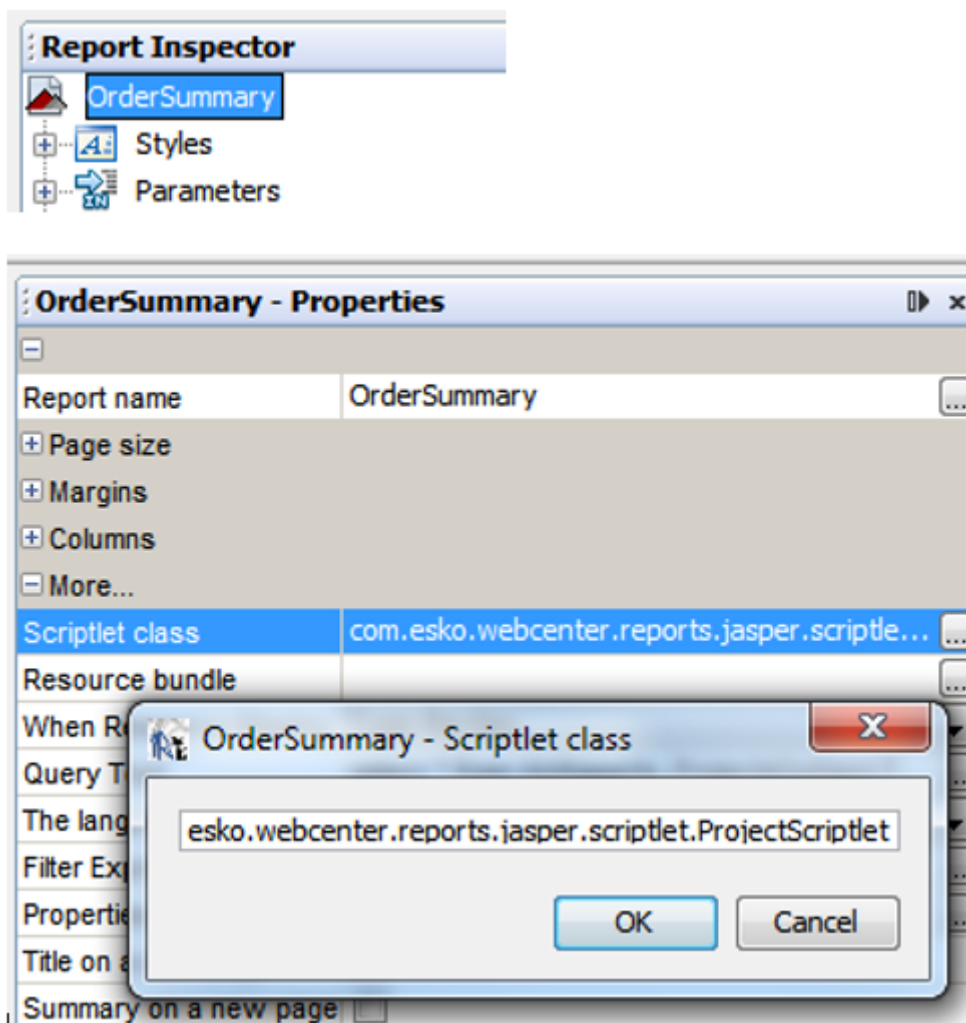
16. Scriptlets

JasperReports allows us to execute snippets of Java code at certain points during the report filling process.

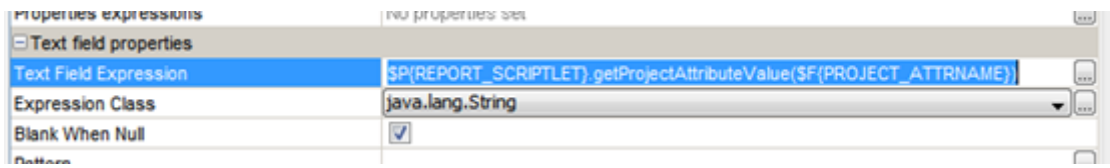
Scriptlets are sequences of Java code that are executed every time a report event occurs. Through scriptlets, users have the possibility to affect the values stored by the report variables.

All the WebCenter custom scriptlets are located in a single jar file (wcr-ireport-22.03.jar), so it is easy to use them within iReport too.

If you want to use a scriptlet in your report, you have to define it (class name) in the Report properties (More section). If you want to add more scriptlets, use the Report Inspector view, in which you can find a Scriptlets node.



REPORT_SCRIPTLET is the built-in parameter that references the scriptlet in the report. The other scriptlets can be referenced by the name_SCRIPTLET (for example scriptlet1_SCRIPTLET).



16.1. Project Scriptlet

Project scriptlet is available in WebCenter as of version 12.0.2 (build R158).

The project scriptlet is a helper class to read and access the **project attributes** from a custom report.

- The project attributes are read and stored by the scriptlet after the report is initialized (based on the 'ProjectId' report parameter).
- The project attributes can be easily accessed by calling the project scriptlet methods from the custom **report** then.

Scriptlet class name: *com.esko.webcenter.reports.jasper.scriptlet.ProjectScriptlet*

The project scriptlet works with the following custom report parameters:

- ProjectId, String (required)
- DateFormat, String, default value "MMM dd, yyyy HH:mm"

Method Summary	
boolean	hasProjectAttribute (String AttributeName) Returns true if there is a value in the database for the project attribute with the given name.
String	getProjectAttributeValue (String AttributeName) Returns the value (or null) of the project attribute with the given name as String. Date attributes are formatted according to the pattern specified in the DateFormat report parameter.

As of WebCenter 12.1, the `getProjectAttributeValue` method supports reading of the XML attribute values (BLOB) too.

They are not cached in the scriptlet, but loaded from the database at the time the attribute value is requested by the report via this method. Read [this](#) chapter for details of how to display RTF attributes in your report.

16.2. Database Scriptlet

Out of the box, **DatabaseScriptlet** is provided as part of the Default Reports.

This scriptlet has various methods that enable us to write report query independent of the database (MS SQLServer, Oracle).

Use this scriptlet if you want to make a report query database independent, otherwise you can directly specify db specific constructs.

You may refer some 'funcWhereclauseXXXXXX' parameters defined in any of the default reports to understand how DatabaseScriptlet helps to write database independent query.

In order to properly initialize the scriptlet both for using within WebCenter and iReport, it is necessary to define **DbType** variable and call the scriptlet initialization method **initDatabaseType** in the main report jrxml file. The **DbType** variable value will be set to the correct database type when called from WebCenter. iReport will use the default value.

```
<parameter name="DbType" class="java.lang.String" isForPrompting="false">
<defaultValueExpression><![CDATA["SQLSERVER"]]></defaultValueExpression>
</parameter><parameter name="funcDbInit" class="java.lang.String"
isForPrompting="false">
<defaultValueExpression><![
CDATA[com.esko.webcenter.reports.jasper.scriptlet.DatabaseScriptlet.
initDatabaseType($P{DbType})]]></defaultValueExpression></parameter>
```

DatabaseScriptlet Provides various database related utility methods that can make reports run independent of the database. Those utility methods are explained in the below table:

Method Summary	
void	<p>initDatabaseType(String dbType)</p> <p>Initializes the scriptlet. This method must be called before any other call. The dbType can be either "SQLSERVER" or "ORACLE".</p>
String	<p>getFuncDateDiff (String datepart, String column1, String column2)</p> <p>If there are 2 date columns on which we need to do a diff, then this method should be used. This method generates a date difference string that is specific to the db.</p> <p>For e.g., StartDate & EndDate are the two date columns on which we want to do diff and we want the result in days.</p> <p>Invocation: getFuncDateDiff("day", "StartDate", "EndDate")</p> <p>Result: MSSQL : DATEDIFF(day, StartDate, EndDate)</p> <p>Oracle : (EndDate – StartDate)</p> <p>Allowed values for datepart: second, minute, hour, day, month</p>
String	<p>getTopNRecordQuery(String column, int numOfRec, String orderType)</p> <p>Generates a string, which when prefixed to a query fetches the top n records of a query result.</p> <p>For e.g., If we want to fetch top 6 rows with query sorted on "Task_Due_Date" by asc, then</p> <p>For Oracle, we just add the (rownum AS RowNum)column to the select list and then later filter the query by using a where clause(RowNum >= 6) and we should explicitly add the order by clause to the query(even though the same is passed as a parameter here)</p> <p>For MSSQL, we add the column {TOP 6 ROW_NUMBER() OVER(ORDER BY Task_Due_Date ASC) AS RowNum} to the select list.</p>
String	<p>getFuncWhereClause(String column, String operator, String value)</p> <p>When non-mandatory parameters are not set, their value is 'null'. It should be interpreted like 'all' in the database query. This method constructs the where clause for the given input. It just returns "1=1" is the value is null or "_NULL_"</p>
String	<p>getFuncAttributeWhereClause(String attrNameColumn, String attrNameValue, String attrValueColumn, String attrValue, boolean skipWhenNull)</p>

Method Summary	
	Constructs the whereClause for WebCenter attributes. It takes in attribute name/value column names and their corresponding values and builds the whereClause.
String	getFuncConcatenation() Concatenation symbol for oracle and MSSQL Server is different. Example: It returns '+' for MSSQL Server and ' ' for Oracle
String	getFuncContainsAll() It returns database specific like operator. Example: It returns '*' for MSSQL Server and '%' for Oracle
String	getFuncLength() It returns database specific length function keyword. Example: It returns 'LEN' for MSSQL Server and 'LENGTH' for Oracle
String	getDummyQuery() Dummy query means that do not need data from any table. This query is database specific. Example: It return 'SELECT 1 value' for MSSQL Server and 'SELECT 1 value FROM dual' for Oracle
String	getFuncSubstring() It returns database specific Sub String function keyword. Example: It return 'SUBSTRING' for MSSQL Server and 'SUBSTR' for Oracle
String	getFuncConvertToChar It returns database specific Convert To Character function keyword. Example: It return 'CONVERT(varchar(255),' for MSSQL Server and 'TO_CHAR()' for Oracle
String	getFuncConvertToInt It returns database specific Convert To Integer function keyword. Example: It return 'CONVERT(int,' for MSSQL Server and 'TO_NUMBER()' for Oracle
String	getFuncConvertToFloat It returns database specific Convert To Float function keyword. Example: It return 'CONVERT(float,' for MSSQL Server and 'TO_NUMBER()' for Oracle
String	getFuncConvertToDate It returns database specific Convert To Date function keyword. Example: It return 'CONVERT(datetime,' for MSSQL Server and 'TO_TIMESTAMP()' for Oracle
String	getFuncNotNull It returns database specific Null Check function keyword.

Method Summary	
	Example: It return 'ISNULL' for MSSQL Server and 'NVL' for Oracle
String	<p>getFuncChar</p> <p>It returns database specific Char function keyword that returns character based on the number code.</p> <p>Example: It return 'CHAR' for MSSQL Server and 'CHR' for Oracle</p>
String	<p>getDateFormat</p> <p>It returns database specific date format construct.</p> <p>Example: It return '121' for MSSQL Server and 'yyyy-mm-dd hh24:mi:ss.ff' for Oracle</p>
String	<p>getFuncDualQuery</p> <p>To perform any calculation or write a dummy query, it is mandatory to specify a table name in oracle query but not in SQL Server. This function can be used to append 'From Dual' keyword if you use oracle database, otherwise it returns empty string.</p> <p>Example: It return '' for MSSQL Server and 'FROM Dual' for Oracle</p>

16.3. ImageGenerator Scriptlet

Provides methods to generate horizontal bar images. Refer "TaskManagement" report template for usage.

Method Summary	
BufferedImage	<p>generateImage_DualBar(int canvasWidth, int canvasHeight, int z, int x, int y)</p> <p>Constructs a rectangle image with given width & height. Within the rectangle constructs two regions with red & green color. The region are constructed based on the x, y & z values. x indicates the amount of green region and y indicates the amount of red region. z indicates the whole rectangular area.</p>
BufferedImage	<p>generateImage_SingleBar(int canvasWidth, int canvasHeight, int z, int x, int threshHold)</p> <p>Constructs a rectangle image with given width & height. Within the rectangle constructs a region with the value x. The region color can be red or green based on the threshold.</p>

17. Chart Customizer

JasperReports use jFreeChart as a charting library. However many features in jFreeChart are not yet supported in JasperReports. With the help of customizer, you can fully control jFreeChart.

Like **Scriptlets**, **Customizer** is also automatically invoked during the filling process. Your customizer class must implement **JRChartCustomizer** interface. This interface has only one method **customize**.

The default WebCenter reports uses a chart customizer **BarChartCustomizer** to control the bar charts' bar colors, the public method **customize** checks bar value against a property value called **Tolerance** and if the bar value is greater than the tolerance value, it shows the bar in red color.

The property **Tolerance** is defined in the bar chart with the value as sum of two parameters values.

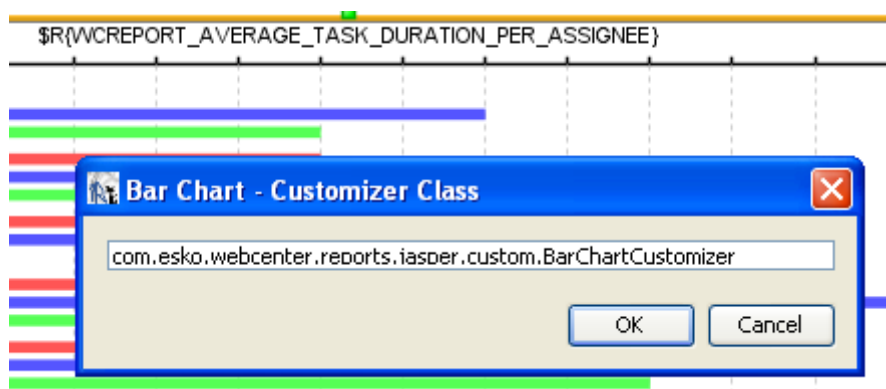
Tolerance = Target Average Completion Time (days) + Target Average Completion Tolerance (days)

The values of the above parameters are passed from WebCenter Report Details page and the default values for the parameters are follows:

Target Average Completion Time (days):*

Target Average Completion Tolerance (days):*

The chart customizer is assigned to the bar chart in the iReport using the property **Customizer Class**.



Note: Please ensure the class is available i Report's class path.

18. Custom JR Data Source Providers

JRSearchDataBeanProvider

These are custom fields for Project and Documents that are available when using the 'JRSearchDataBeanProvider' custom JR data source.

Basic Attributes (type = "attributes")

Field Name	Field Id
Attribute Name	attr.name
Attribute Value	attr.val

Projects Details (type = "proj")

Field Name	Field Id
Project Name	proj.name
Description	proj.description
Customer/Location	proj.customer
Project Manager	proj.manager
Modified	proj.datemodified
Created	proj.datecreated
Due	proj.datedue
Characteristics	proj.characteristics
Salesperson	proj.salesperson
Manager's Location	proj.manager.location
Project Template	proj.template
Parent Projects	proj.parentprojects
Child Projects	proj.childprojects
Project Thumbnail	proj.thumbnail
Project Duration	proj.duration
Project Time Left	proj.timeleft
Project Attribute Category	proj.attributecategory
Any Project Attribute	proj.attributes[<i>attribute name</i>]
Role Members	proj.roles[<i>role name</i>]
Any Project Status Duration	proj.status.duration[<i>any project status name</i>]
Any Project Status Started	proj.status.started[<i>any project status name</i>]
Any Project Status Completed	proj.status.completed[<i>any project status name</i>]

Field Name	Field Id
Any Project Status Due	proj.status.due[any project status name]
Any Project Status Time Left	proj.status.timeleft[any project status name]

ALL Type Documents (type = "alldoc" or "text" or "des" or "mfg" or "canvas" or "graphic" or "graphicimage")

Field Name	Field Id
Document Name	doc.name
Version Author	doc.author
Author's Location	doc.author.location
Description	doc.description
Created	doc.datecreated
Modified	doc.datemodified
Approval Status	doc.approvalstatus
Current Approval Stage	doc.approvalstage
Approval Started	doc.dateapprovalstarted
Approval Due	doc.dateapprovaldue
Approval Status Completed	doc.dateapprovalcompleted
Approval Completed	doc.datecompleted
Revision	doc.revision
Version Count	doc.version
Last Version Comment	doc.versioncomment
Characteristics	doc.characteristics
Document Thumbnail	doc.thumbnail
Approval Duration	doc.duration
Approval Time Left	doc.timeleft
Pending Approvers	doc.pendingapprovers
Reason(s) for Rejection	doc.reasonsforrejection.currentversion
Reason(s) for Rejection - Any Version	doc.reasonsforrejection.allversions
Version Used in	doc.versionusedin
Approval Version Count	doc.versionswithapproval
Attribute Category	doc.attributecategory
Characteristics	doc.characteristics
Version Size	doc.versionsize
Folder	doc.folders
Any Attribute	doc.attributes[attribute name]

Field Name	Field Id
Any Reference Property	doc.referenceproperty[<i>property name</i>]
Project Name	proj.name
Project Description	proj.description
Customer/Location	proj.customer
Project Manager	proj.manager
Project Created	proj.datecreated
Project Started	proj.datestarted
Project Modified	proj.datemodified
Project Due	proj.datedue
Project Completed	proj.datecompleted
Project Characteristics	proj.characteristics
Salesperson	proj.salesperson
Project Creator	proj.projectcreator
Manager's Location	proj.manager.location
Project Template	proj.template
Parent Projects	proj.parentprojects
Child Projects	proj.childprojects
Project Duration	proj.duration
Project Time Left	proj.timeleft
Project Attribute Category	proj.attributecategory
Any Project Attribute	proj.attributes[<i>attribute name</i>]
Role Members	proj.roles[<i>role name</i>]
Any Project Status Duration	proj.status.duration[<i>any project status name</i>]
Any Project Status Started	proj.status.started[<i>any project status name</i>]
Any Project Status Completed	proj.status.completed[<i>any project status name</i>]
Any Project Status Due	proj.status.due[<i>any project status name</i>]
Any Project Status Time Left	proj.status.timeleft[<i>any project status name</i>]

ACAD Design Documents (type = "des")

Field Name	Field Id
Length	doc.content.acad.length
Width	doc.content.acad.width
Depth	doc.content.acad.depth
Board Name	doc.content.acad.boardname

Field Name	Field Id
Board Flute	doc.content.acad.boardflute
Grain Direction	doc.content.acad.graindirection
Blank Length	doc.content.acad.blanklength
Blank Height	doc.content.acad.blankheight
Area	doc.content.acad.area
Rule Length	doc.content.acad.rulelength
Caliper	doc.content.acad.caliper

ACAD Manufacturing Documents (type = "mfg")

Field Name	Field Id
Sheet Width	doc.content.acad.sheetwidth
Sheet Length	doc.content.acad.sheetlength
Run Length	doc.content.acad.runlength
Board Name	doc.content.acad.boardname
Print Items	doc.content.acad.printitems
Machines	doc.content.acad.machines
Print Ite Doc Id	doc.content.acad.printitemdocid
Board Flute	doc.content.acad.boardflute
Grain Direction	doc.content.acad.graindirection
Caliper	doc.content.acad.caliper

Image Graphic Documents (type = "graphicimage")

Field Name	Field Id
Width	doc.content.graphic.width
Height	doc.content.graphic.height
Vertical Resolution	doc.content.graphic.vres
Horizontal Resolution	doc.content.graphic.hres
Inks	doc.content.graphic.inks

Cape Documents (type = "cape")

Field Name	Field Id
Product Name	doc.content.cape.productname
Product Code	doc.content.cape.productcode
Pallet Name	doc.content.cape.pallet.name
User Text	doc.content.cape.usertext

Field Name	Field Id
Secondary Pack (OD) Length	doc.content.cape.sp_od_length
Secondary Pack (OD) Width	doc.content.cape.sp_od_width
Secondary Pack (OD) Height	doc.content.cape.sp_od_height
Pallet Length	doc.content.cape.pallet.length
Pallet Width	doc.content.cape.pallet.width
Area efficiency (Pallet)	doc.content.cape.pallet.area_efficiency
Cube efficiency (Pallet)	doc.content.cape.pallet.cube_efficiency

Canvas Documents (type = "canvas")

Field Name	Field Id
Length	doc.content.acad.length
Width	doc.content.acad.width
Depth	doc.content.acad.depth
Board Name	doc.content.acad.boardname
Part Name	doc.content.acad.canvas.partname
Board Flute	doc.content.acad.boardflute
Grain Direction	doc.content.acad.graindirection
Blank Length	doc.content.acad.blanklength
Blank Height	doc.content.acad.blankheight
Area	doc.content.acad.area
Rule Length	doc.content.acad.rulelength
Caliper	doc.content.acad.caliper

Graphic Documents (type = "graphic")

Field Name	Field Id
LPI	doc.content.graphic.lpi
Dotshape	doc.content.graphic.dotshape
Fonts	doc.content.graphic.fonts
Width	doc.content.graphic.width
Height	doc.content.graphic.height
Graphics Name	doc.content.graphic.graphicsname
Graphics Project	doc.content.graphic.graphicsproject
Barcodes	doc.content.graphic.barcodes
External References	doc.content.graphic.externalrefs
Design Files	doc.content.graphic.designfiles

Field Name	Field Id
Inks	doc.content.graphic.inks

Text Content Documents (type = "text")

Field Name	Field Id	Description
Element Type	doc.content.textcontent.elementtype	To display text content element type
Document Status	doc.content.textcontent.status	To display text content document status
Statement of Text Content	doc.content.textcontent.text	To display statement of text content element
Language	doc.content.textcontent.language	To display text content language
Modified	doc.content.textcontent.modified	To display language modified date