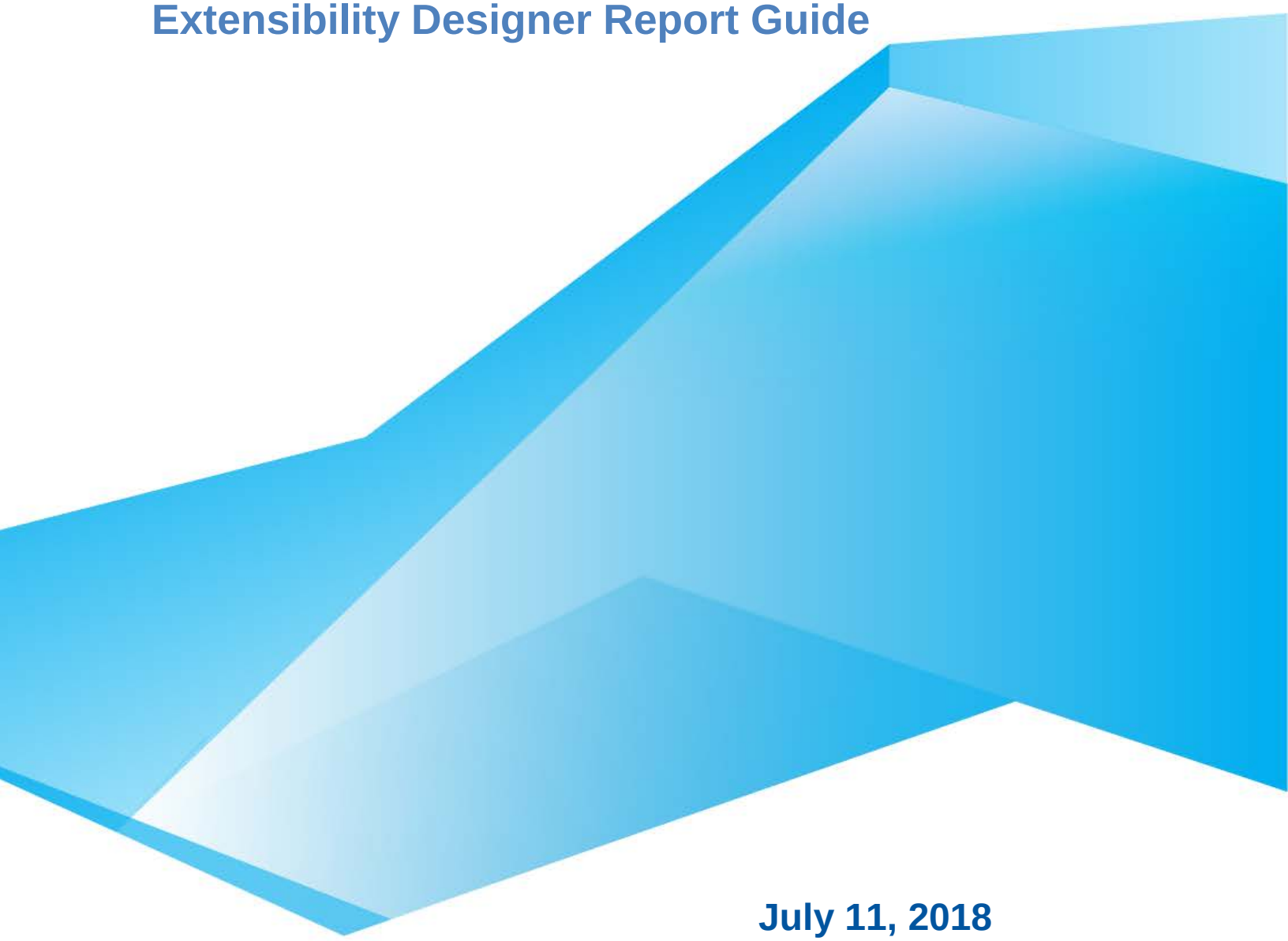
Deltek Know more. Do more.™

Deltek Costpoint® 7.1.1

Extensibility Designer Report Guide



July 11, 2018

While Deltek has attempted to verify that the information in this document is accurate and complete, some typographical or technical errors may exist. The recipient of this document is solely responsible for all decisions relating to or use of the information provided herein.

The information contained in this publication is effective as of the publication date below and is subject to change without notice.

This publication contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, or translated into another language, without the prior written consent of Deltek, Inc.

This edition published July 2018.

© Deltek, Inc.

Deltek's software is also protected by copyright law and constitutes valuable confidential and proprietary information of Deltek, Inc. and its licensors. The Deltek software, and all related documentation, is provided for use only in accordance with the terms of the license agreement. Unauthorized reproduction or distribution of the program or any portion thereof could result in severe civil or criminal penalties.

All trademarks are the property of their respective owners.

Contents

Introduction 1
Standard Components of a Costpoint Report 2
Report Components 2
Report Application 3
Report Parameter Result Set 3
Report Definition 4
Data Source Result Set 6
Report Template 7
Extend a Report in the Extensibility Designer 9
Extend a Report Definition 9
Extend a Data Source Result Set 10
Extend a Report Template 13
Eclipse IDE for BIRT 14
Eclipse IDE Installation 14
Start Eclipse 15
Create a Project 15
Set Up Report Design Preferences 15
Test Costpoint BIRT Plugins 15
Customize a Report Template in Eclipse 17
Report Template Outline 17
Data Source 18
Data Set 18
Body 19
MasterPages 27
Deploy the Extension 28
Guidelines and References 29
Customizing Controls 29
Scripting and Report Event Handlers 29
Writing Expressions 29
Naming Conventions 30
Hiding Element 31
Report Page Settings 32
Cover Page 33
Page Header/Footer 34
Access DataSet Variables in Page Header 36

Lines 39
Column Headers 39
Page Break (Conditional Page Break) 40
Suppress Repeated Values 43
Data Suppression (Labor, Cost, Price) 44
Page Numbers (Restart Page #) 46
Fonts 48
Group Sections and Aggregates 50
Handling "No Data Found" 51
Multiple Report Sections in One Template (Posting Applications) 52
Presenting Text 56
Localization and I18n 59
Image Controls 64
Report Table of Contents 64
Utility functions 65

Introduction

This guide explains the steps on how to extend a report. Before reading this guide, refer to the *Deltek Costpoint Extensibility Designer User Guide* on how to navigate the Designer and basic steps in extending a Costpoint component. The extensible components include Result Set, Action, Report, Messages, Constants, and Application Title. Extending a report involves several interconnected components.

At a high level, extending a report involves the following steps.

Step	Description
1	Create a report extension using the Extensibility Designer
2	Extend the reporting data result set (if additional data is to be shown in the report)
3	Extend the java classes that populate the reporting data (if data was constructed with the help of java code)
4	Modify the report template used for such report
5	Deploy the extensibility unit to the system the same way as done for other components

Some of the steps are optional depending on how the report is implemented by Deltek and the extent of the change.

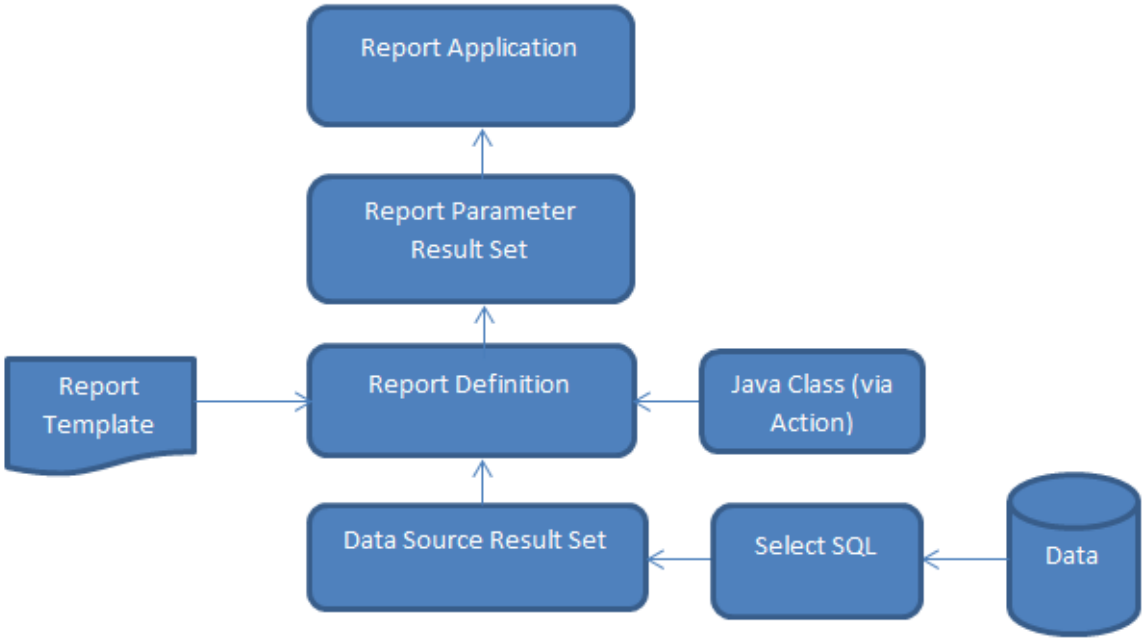
Standard Components of a Costpoint Report

Before you extend or customize a Costpoint report, you must first understand the standard components of a Costpoint report and how they are linked together. This will help you identify which components you need to extend for a report.

Report Components

The following are the standard components of a report.

- Report application
- Report parameter result set
- Report definition (and Java classes that populate the data for the report)
- Data source result set (with Select SQL)
- Report template



Report Application

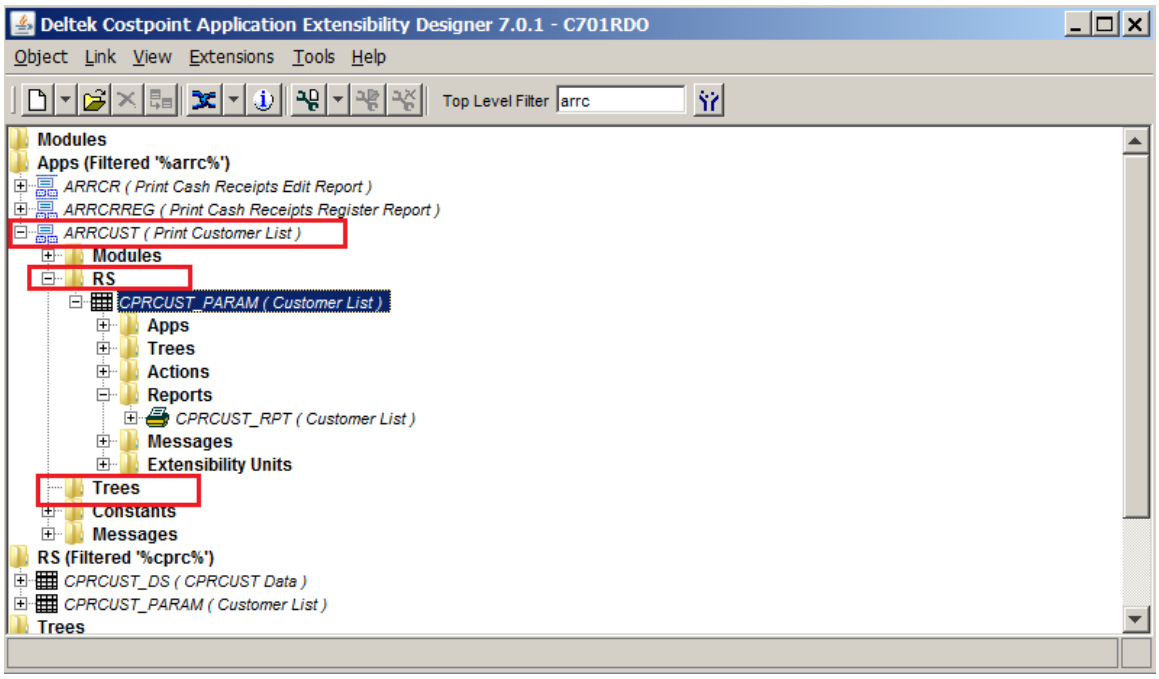
An application is the top level component that acts as the container to the screens inside the application. It does not contain any logic. UI and business logics are done entirely in the screens (result sets) and the java classes that supports them. You can only extend an application by changing its application name.

To view a list of all available applications in Costpoint, refer to the Apps folder of the Extensibility Designer's Object Explorer. These applications are sorted by application ID.

If you know the name of the application but not the application ID, open the application in Costpoint and then go to the View Help About screen (**Administration » System Administration » System Administration Reports/Inquiries Help » About Costpoint**). The application is listed on the Open Applications table window with its application ID.

Report Parameter Result Set

A report parameter result set is the screen where the user selects various options to run the report. It is assigned underneath an application. In the example below, CPRCUST_PARAM is the parameter result set assigned to the Print Customer List application.

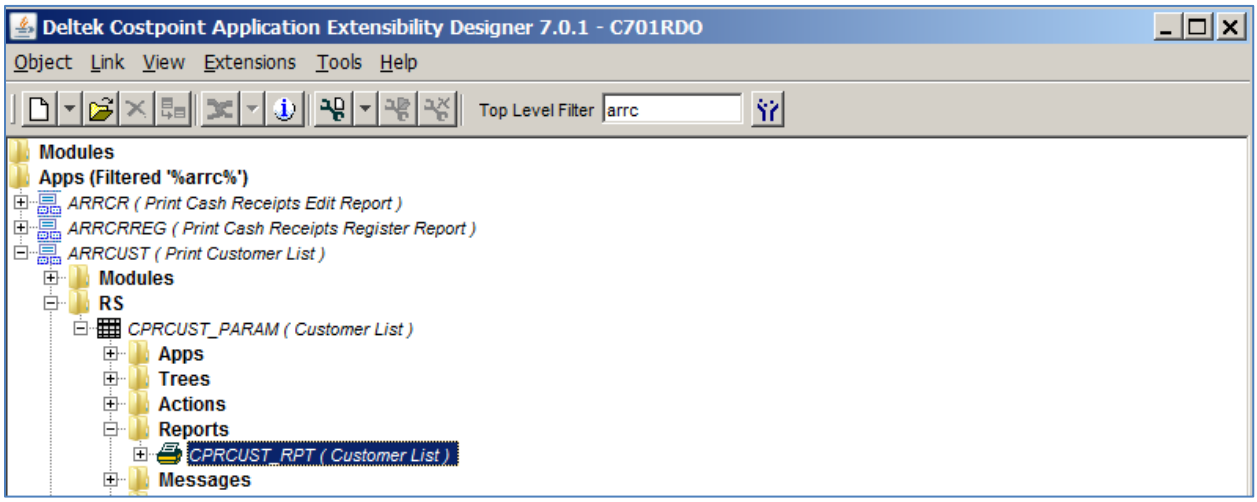


If a report application contains more than one subtask, you can locate its parameter result set under the Trees folder, instead of the RS folder for that application in the Object Explorer.

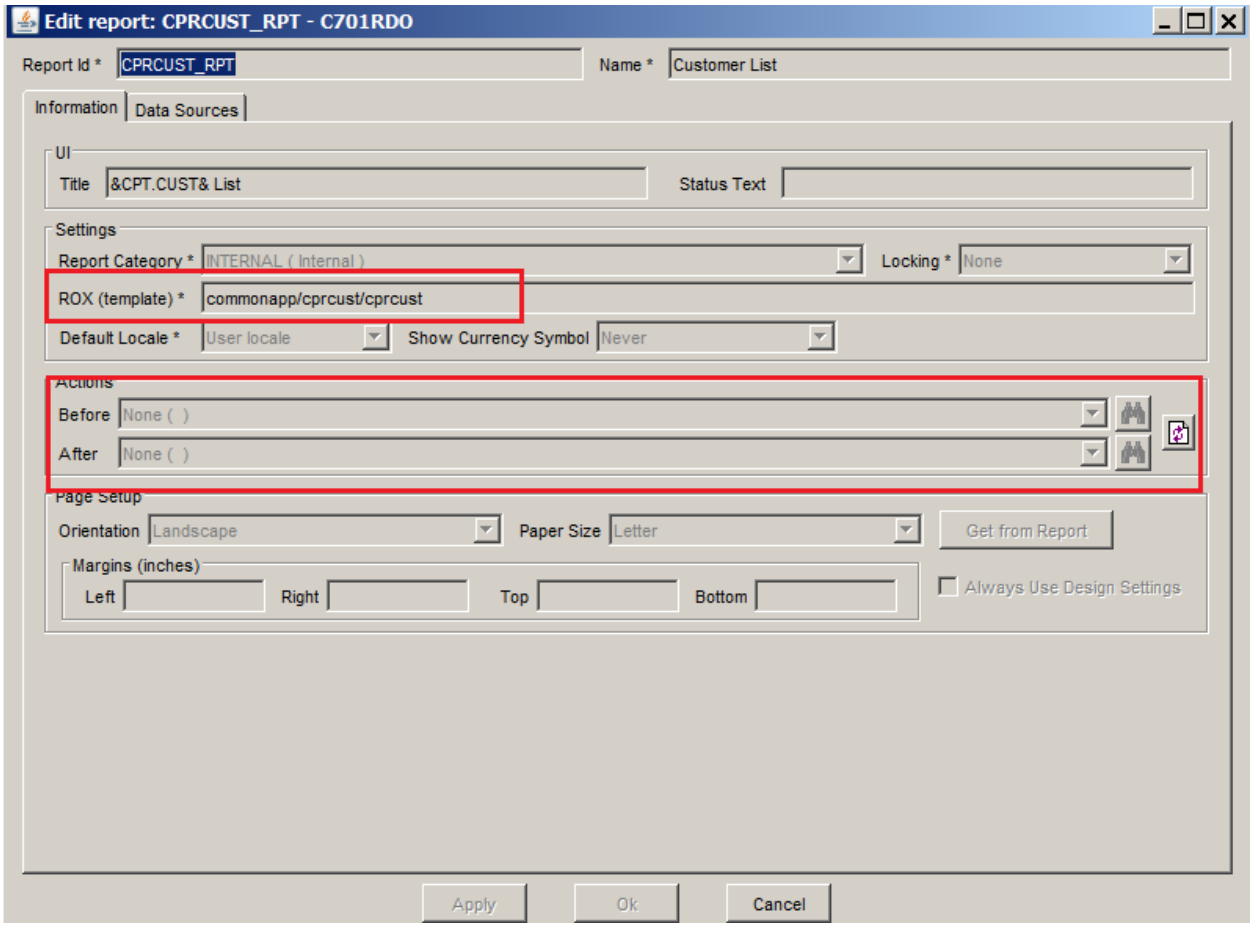
A parameter result set does not contain logic for the report. It only drives what data should be gathered for the report. Generally, you do not extend a parameter result set except for cosmetic changes such as changing the labels on the screen. Options such as adding an additional selection, filter, or sort option cannot be added since the existing Java class logic for populating the data source are not extensible (see Data Source Result Set).

Report Definition

A report definition (or report list) includes the metadata that describes a report such as title, status text, Java class action, and the data fields that feed into the report. In the Object explorer, you can find the report listed directly underneath the parameter result set.



In the example above shows the report definition for CPRCUST_PARAM is CPRCUST_RPT.



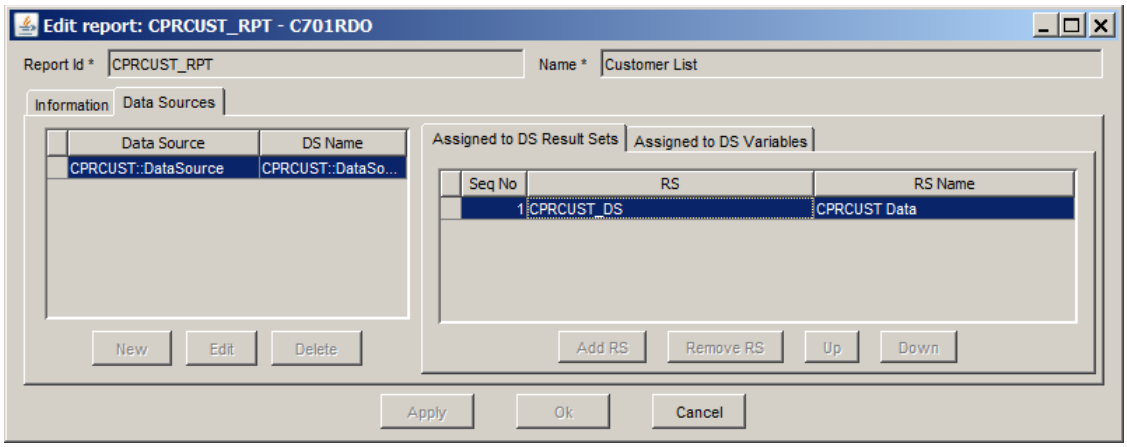
The name and location of the original report template is specified in the **ROX (template)** field on the Information tab. This path is a sub path underneath the root path where report templates for all applications reside in the Costpoint server. The root path for all report templates is C:\deltek\costpoint\70\applications\birt\report\bin. The example above displays the report template for Customer List at:

C:\deltek\costpoint\70\applications\birt\report\bin\commonapp\cprcust\cprcust.rptdesign

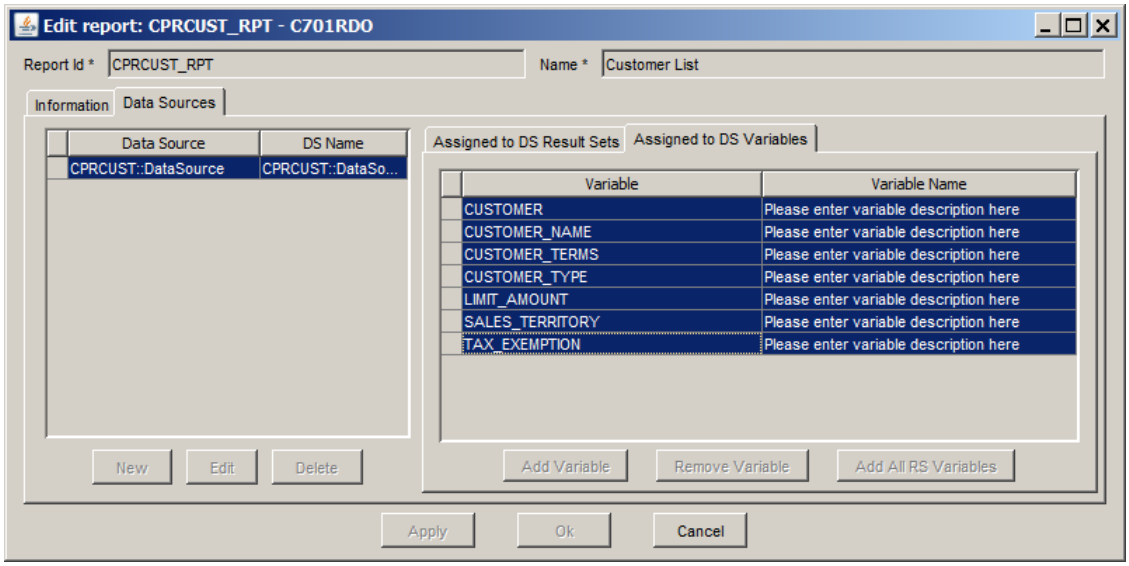
The .**rptdesign** extension name is omitted in the **ROX** field since it is implied. If there is any standard Costpoint action to be run before or after the report is run, they are displayed here.

For more information about an action, see the *Deltek Costpoint Extensibility Designer Guide*.

On the Data Sources tab, the left pane displays the data source ID and name used inside the report template. The right pane displays the ID of the data source result set which supplies the data for the report. In the following example, there is only data source, and the ID is CPRCUST_DS.



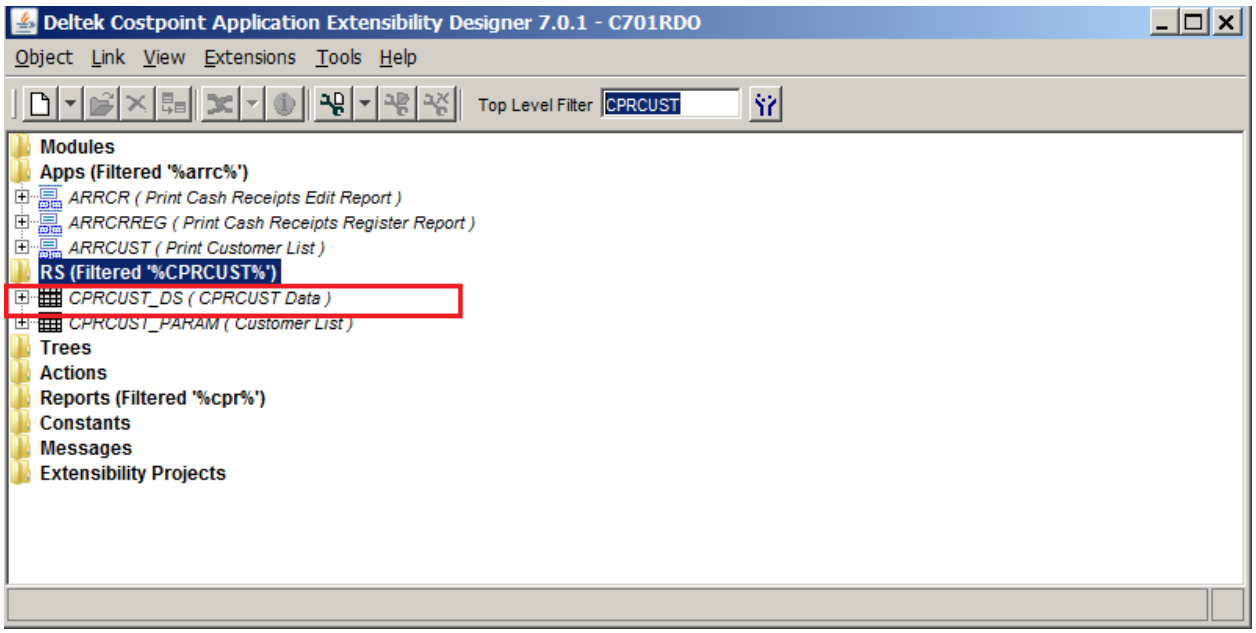
Click the Assign to DS Variables tab to see all the fields from the data source result set that have been selected to feed into the report. In the following example, seven columns from the data source result set are selected to be used on the report.



You can extend a report definition by changing the report title, status text, the report template, action to be run before or after the report, and the data source columns.

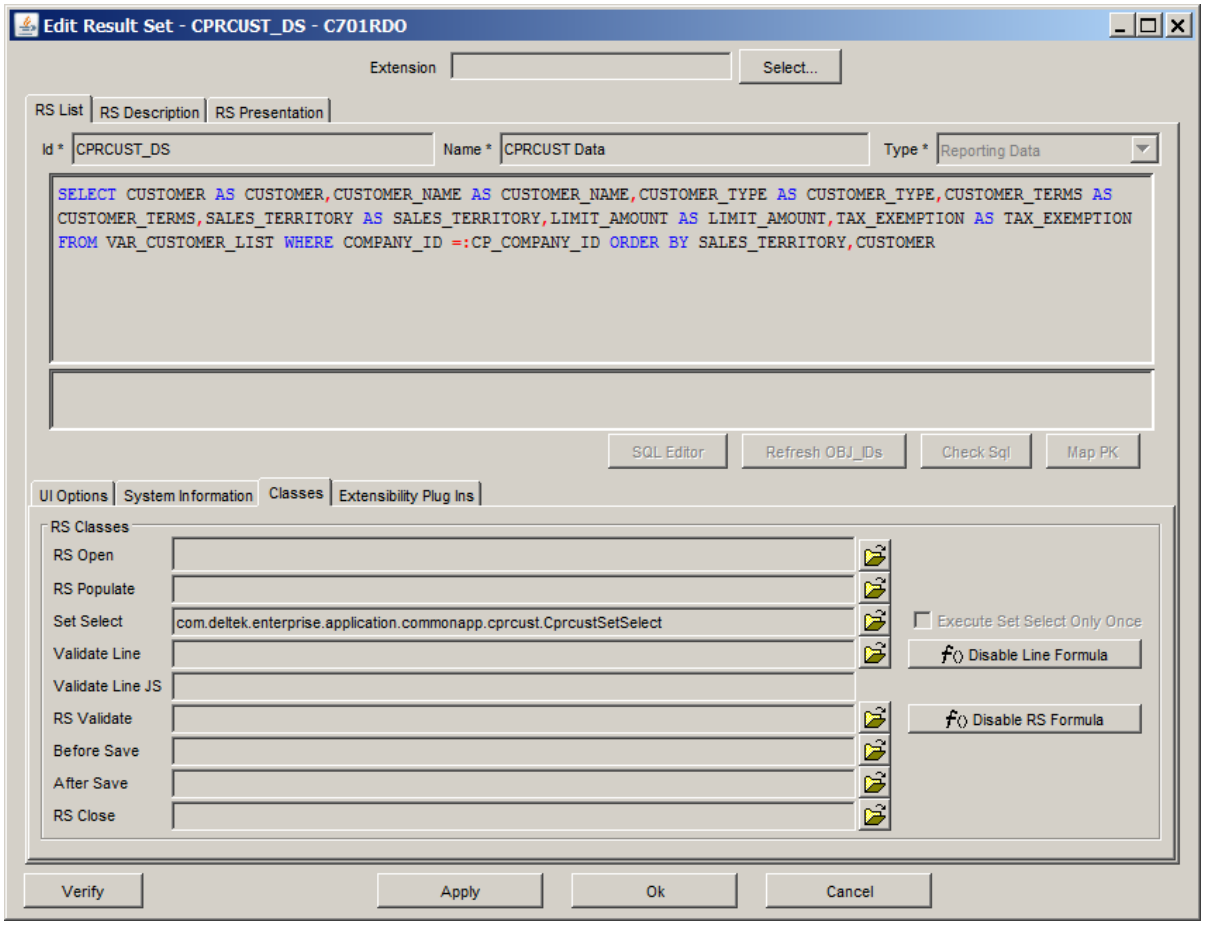
Data Source Result Set

The data source result set acts as a data source for a report. It is like any other result set except there is no need to design the UI layout. The data source result set for a report is not displayed directly in the Object Explorer under the respective application branch. You must take note of the ID of the data source result set in the Report Definition and find it using the **Top Level Filter** field, as shown in the following example.



At the top of the RS list tab, there is usually a SQL Select statement defining where the data for the report will come from. In the lower half of the Classes tab, there is usually a Java class in the **Set Select** field. This Java class defines or refines the SQL statement at run time to filter and sort the data according to what the user specifies on the parameter result set.

You cannot extend this Java class as it is not an event that is extensible. For this reason, you generally do not extend the parameter result set to add additional options for filtering as you cannot extend the set select Java class to cover the additional options. You can, however, add options such show or hide since this can be done in the report template itself.



You can extend the data source result set by adding additional columns to the SQL select statement in the lower half of the SQL select panel.

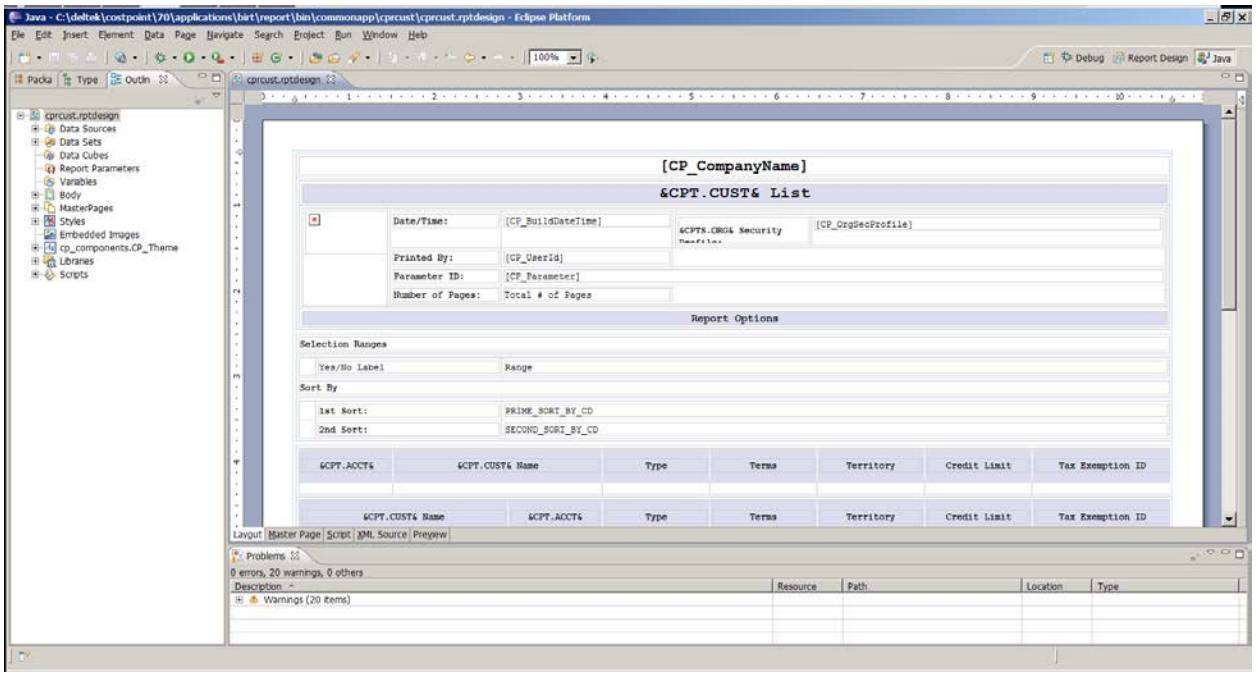
Report Template

Costpoint uses Eclipse open source BIRT (Business Intelligence and Report Tools) platform for reporting. A report template is a layout/design of the report created with Eclipse BIRT IDE.

Refer to the Eclipse IDE Installation section for more information.

All report template files have the .rptdesign extension name. For example, the Customer List report template file name is cprcust.rptdesign.

You can extend a report template by copying an existing Costpoint template, saving it with a new name, and then modifying it using the Eclipse IDE. The new template is then placed in the custom folder (specified in the extended report definition) under the BIRT report main folder on the server, as shown in the following example.


[CP_CompanyName]
&CPT.CUST& List
Date/Time:
[CP_BuildDateTime]
Printed By:
[CP_UserId]
Parameter ID:
[CP_Parameter]
Number of Pages:
Total # of Pages
[CP_OrgSecProfile]
Report Options
Selection Ranges
Yes/No Label
Range
Sort By
1st Sort:
PRIME_SORT_BY_CD
2nd Sort:
SECOND_SORT_BY_CD
&CPT.ACCT&
&CPT.CUST& Name
Type
Terms
Territory
Credit Limit
Tax Exemption ID

Extend a Report in the Extensibility Designer

You can extend a report with simple to complex changes. The level of effort depends on how complex the report is in Costpoint and how many changes are desired for the extended report. Below are some examples of simple and complex changes:

- **Change the report title or report status text on the screen (not the report)** — This change only involves extending the report definition in the Extensibility Designer and setting the new report title or status text to be shown on the screen (not the report).
- **Change the report title on the report** — This change involves extending the report definition in the Extensibility Designer and a simple change in the report template for the report title in Eclipse IDE for BIRT report.
- **Change the labels, column headings, or the layout of the fields on the report** — This change involves extending the report definition and more in-depth changes to the report template in Eclipse IDE. You need proficient knowledge of how to use Costpoint plug-in classes and methods for BIRT reports for this change.
- **Add additional columns from a related table to the main Costpoint report table to the report** — This change involves extending the report definition, making changes to the data source result set, and making changes to the report template.

Extend a Report Definition

To extend a report definition:

1. Select the Report Definition for the report in the Extensibility Designer.

   The Edit Report screen displays. The Information tab contains Deltek's original settings. The template location and name in the **ROX (template)** field is the subfolder under the root path for all report templates on the server (C:\deltek\costpoint\70\applications\birt\report\bin). The report template file (rptdesign) is actually a text file, so you can copy and rename it when you intend to customize it.

   The .rptdesign extension name is omitted in the **ROX (template)** field since it is implied.

2. Click the Extensibility Information tab. You can make the following modifications:
   - Chang the **Title** and **Status Text** fields.
   - If you want to change the content of the report (report template), specify a new template name. The path must start with custom/<Unit ID>/<new report template name without extension>. At run time, Costpoint expects this file to be at the report root path and sub folder specified here.
   - Change the **Show Currency Symbol** option. The valid options are:
     - **Never** — A currency field in the report template will not print with the currency symbol at run time.
     - **Always** — A currency field in the report template will always print with the currency symbol at run time.
     - **Depends** — Printing will depend on the settings in the System Settings application.
   - Add before and/or after actions (**Additional Extensibility Actions** group box) if you need to run some special process before or after generating the report. These are available for

new actions (not extended actions) you have created for this extensibility unit. These actions are executed in addition to the standard actions that are set in the Information tab for this report.

- Change the values in the **Page Setup** group box, such as **Orientation**, **Paper Size** or **Margins**. These are defaults only and will display as defaults in the Page Setup option dialog box when a user prints the report. These settings should match how the report template is designed. At run time, a user can override these defaults unless you select the **Always Use Design Settings** check box.

3. Click the Data Sources tab.

   The left pane displays the data source ID and name used inside the report template. The right pane displays the ID of the data source result set that supplied the data for the report.

4. Click the Assigned to DS Variables tab.

   This tab displays the data fields from the data source result set that were selected for the report.

5. Click Add Variable to add additional fields from the data source result set.

You must first add new fields to the data result set, by extending the data source result set, before you can select and then add them to the Assigned to DS Variables tab.

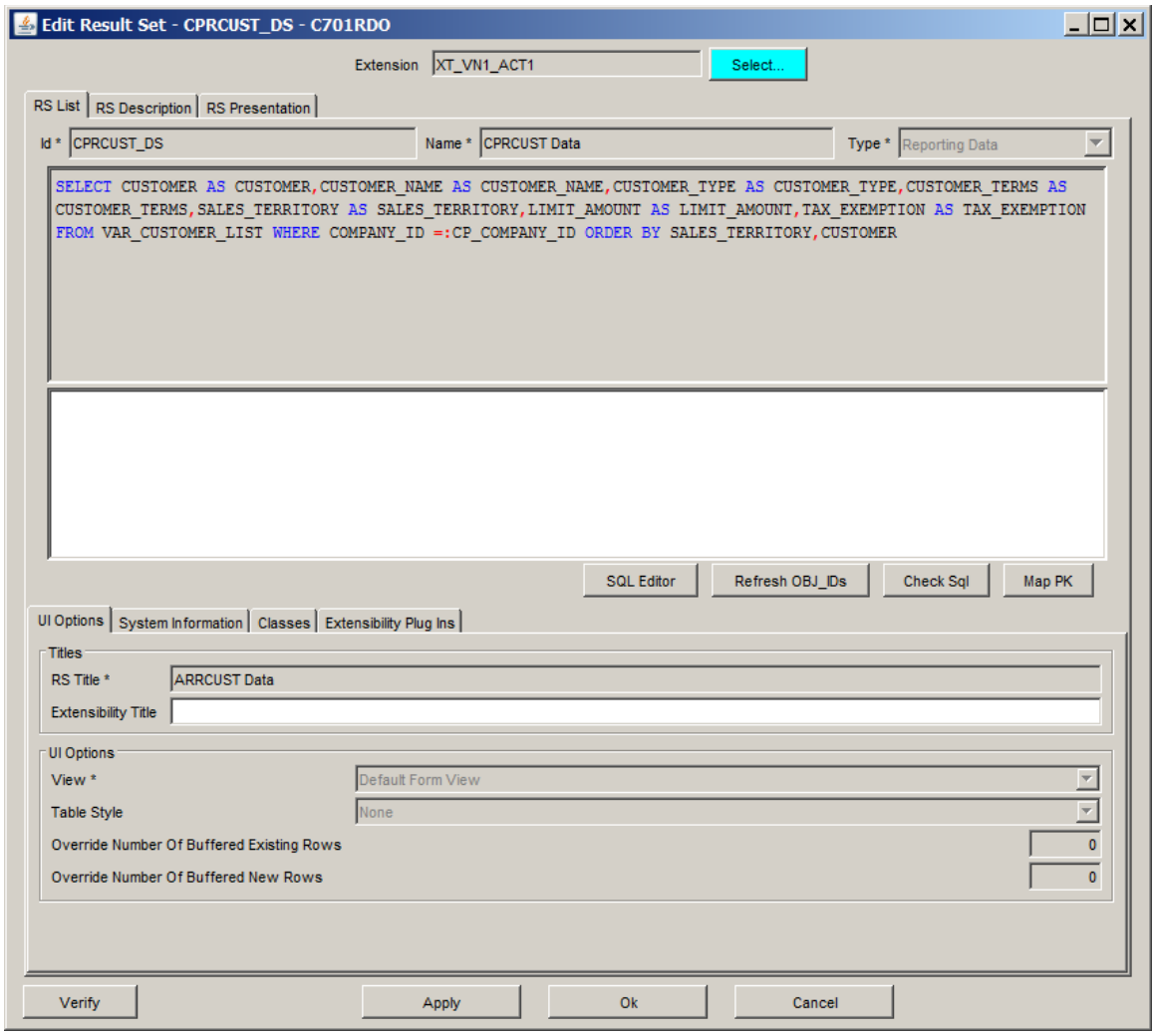
Extend a Data Source Result Set

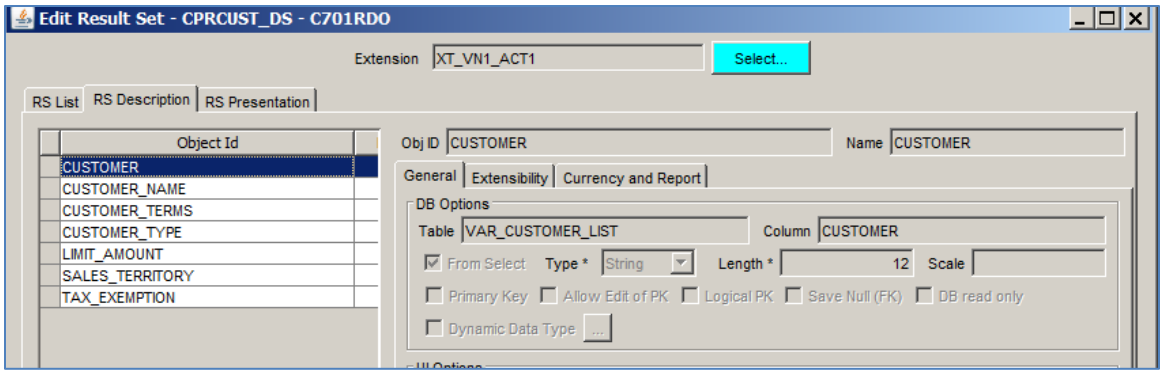
From the report definition, you can get the ID of the data source result set. Then, use the Object Explorer to find this result set and extend this result set to add more data to the report.

To extend a data source result set:

1. Display the Edit Result Set screen.

   The upper text field displays the SQL SELECT statement that selects data for the report. In the following example, the SELECT statement selects data from a table (or view in this case) named AR_CUSTOMER_LIST. It selects seven columns which you actually see in the report definition earlier. To verify, you can select the RS Description tab and those seven columns should display there.

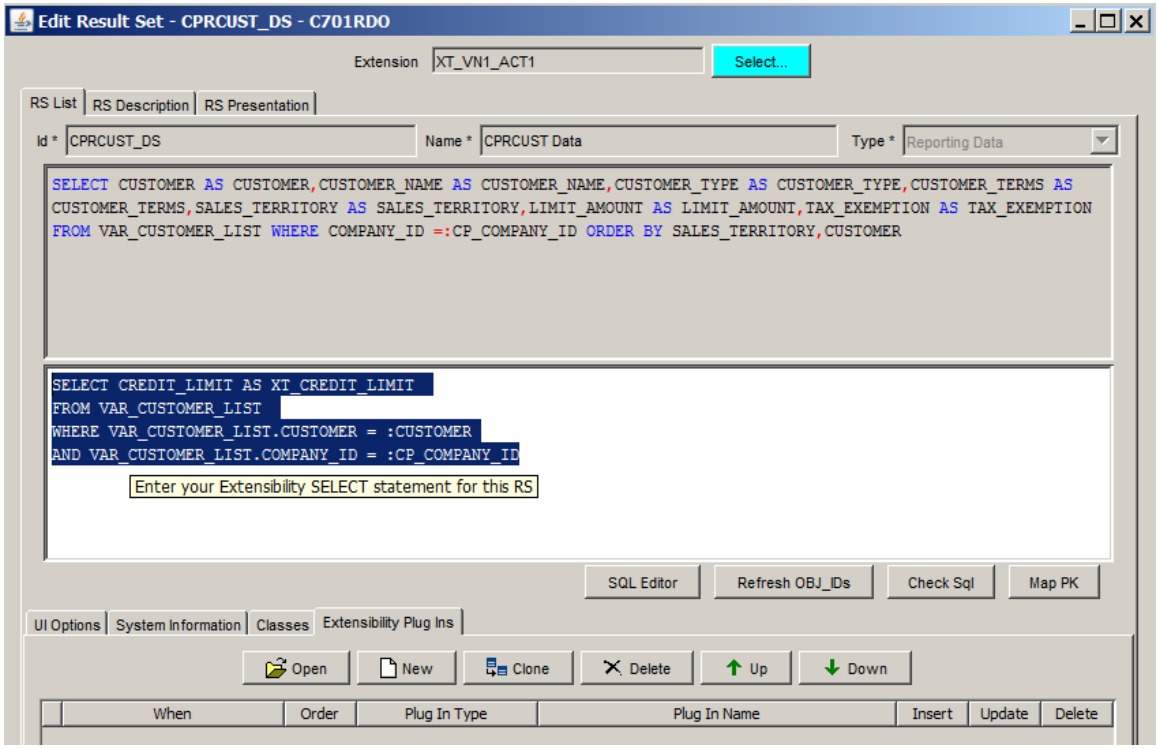




2. To add additional data for the report, click the RS List tab and use the lower text field to enter a SELECT statement. Then select other columns from the same table (or from any other table) where the WHERE clause can be formed with columns selected in the original SELECT statement. Treat the value of the columns in the original SELECT statement as a bind variable (represented by the colon followed by the object ID/alias).

   In the following example, the key fields to table/view VAR_CUSTOMER_LIST are CUSTOMER and CP_COMPANY_ID. So the WHERE clause of the extended SELECT is "VAR_CUSTOMER_LIST.CUSTOMER =: CUSTOMER AND

VAR_CUSTOMER_LIST.COMPANY_ID = :CP_COMPANY_ID". The bind variable :CUSTOMER is the alias for the CUSTOMER column in the original select. It is also the ID of the resulted object seen in the RS Description tab. :CP_COMPANY_ID is the other bind variables for the where clause.



At run time, when each row from the original SELECT statement is being retrieved, the system will execute the extended SELECT statement to get the additional data to add on to that row before sending it to the report.

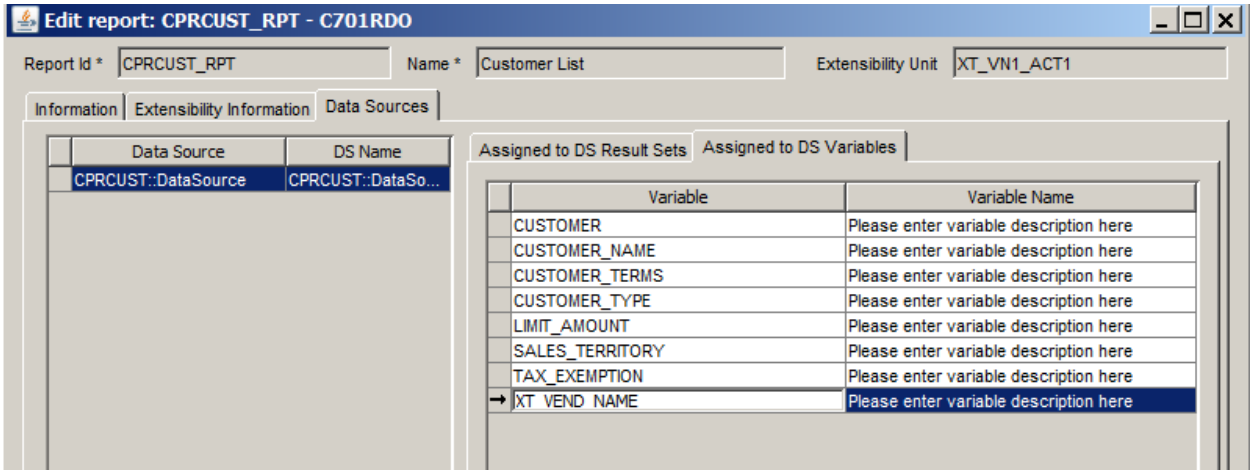
You can select additional data from any table as long as the WHERE clause can be constructed with these bind variables.

For example, instead of adding CREDIT_LIMIT from the VAR_CUSTOMER_LIST, you want to add the VEND_NAME from the VEND table for the VEND_ID contain in this VAR_CUSTOMER_LIST. You must join the VEND table with the CUST table. Your extended SELECT statement would be like this:

```
SELECT V.VEND_NAME AS XT_VEND_NAME FROM VEND V, CUST C WHERE V.VEND_ID =
C.VEND_ID AND C.CUST_ID = :CUSTOMER AND C.COMPANY_ID = :CP_COMPANY_ID AND
V.COMPANY_ID = :CP_COMPANY_ID
```

3. To test if the SQL is valid, click **Check Sql**, and click **Refresh OBJ_IDs**. An object with ID = XT_VEND_NAME displays on the RS Description tab.
4. Click **Apply**.

   This column is now available in the data source result set. You must then make it available for the report
5. Click the RS Description tab, and click the Currency and Report tab.
6. Enter the same ID in the **Report Variable** field, and click **Save**.
7. Go back to the Report Definition screen, and add this new column to the list of variables for the report.



Extend a Report Template

The report template is not extended in the Extensibility Designer. You must specify the new template name and location in the Report Definition extension and then use the Eclipse IDE to design the changes.

To extend a report template:

1. Find the location of the original template.

   For example, if the original path is commonapp/cprcust/cprcust, then the file is located at C:\deltek\costpoint\70\applications\birt\report\bin\commonapp\cprcust\cprcust.rptdesign.

2. Copy the report template file to the new location, and customize it using Eclipse IDE for BIRT.

   f the new path is custom/XT_VN1_ACT1/cprcust_xt, then the new file must be copied, renamed, and placed at C:\deltek\costpoint\70\applications\birt\report\bin\custom\XT_VN1_ACT1\cpr_cust_xt.rptdesign.

See the next section for information on how to install Eclipse IDE for BIRT and how to customize the report template inside Eclipse.

After a report template is customized, any hot fixes to the Deltek report template will not automatically carry over to the custom version. Unlike all other objects in the Designer where you extend it, the extension inherits the changes in the original object. Customizing the report template is actually an override, not an extension. So when the standard Report template is changed by Costpoint development, you will need to take new standard report template and redo your customizations in that file.

Eclipse IDE for BIRT

Report templates are designed using Eclipse IDE for BIRT. You must install the same version of Eclipse IDE for BIRT that Costpoint framework uses (currently, version 4.6.0). Contact Deltek if you cannot find this exact version to download from Eclipse.

Eclipse IDE Installation

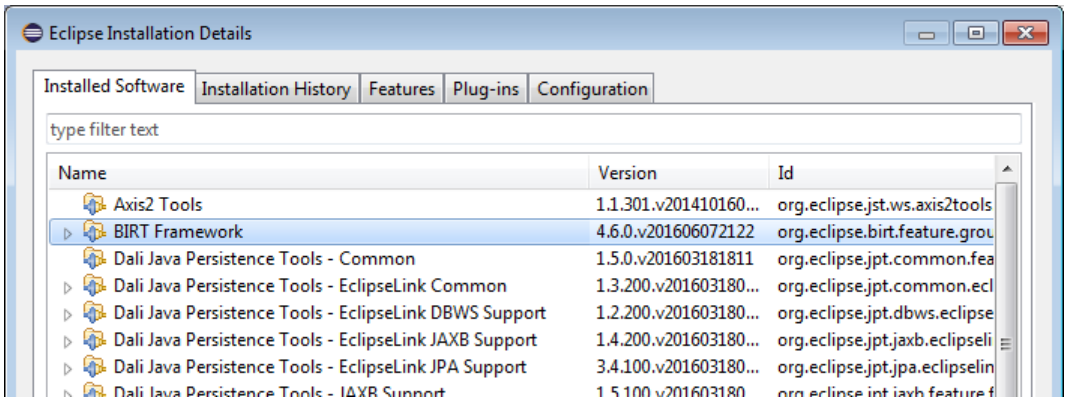
You can download the Eclipse IDE for BIRT 4.6.0 using the following link:

http://www.eclipse.org/downloads/download.php?file=/birt/downloads/drops/R-R1-4.6.0-201606072112/birt-report-designer-all-in-one-4.6.0-20160607-win32.win32.x86_64.zip

You can use the same Java 8 runtime environment that is installed with Costpoint or download latest Java 8 runtime from Oracle download site.

Deltek recommends that you install Eclipse on the Costpoint server machine so that you can access the database and deploy the customized template easily. If you would like to install Eclipse IDE on a machine other than the Costpoint server machine, that machine must be able to connect to the database/system where the extensibility is being done (Weblogic client, JDBC drivers, and so on) as well as the toolconnections.properties file which contains the connection information.

After installation, start Eclipse and verify that the BIRT Framework component under **Help » About Eclipse Platform » Installation Details » Installed Software** displays version **4.6.0**.



Under the installation folder for Eclipse (for example, C:\Eclipse), you will see a plugins subfolder. Go to DSM and download DeltekCostpoint711BIRTInstaller.exe, which will copy two Eclipse plugins (jar files) to your hard drive. You must add these two jar files to the Eclipse\plugins subfolder:

- com.deltek.enterprise.system.reporting.client.birt.oda.designer_1.0.0.jar
- com.deltek.enterprise.system.reporting.client.birt.oda.runtime_1.0.0.jar

These Deltek-supplied plugins allow Eclipse IDE to interact and design the report according to the Costpoint framework.

Start Eclipse

After you install Eclipse, start it by running the eclipse.exe file located under C:\Eclipse. Or, you can create a batch file under the same folder with the following command line in it:

```
c:\eclipse\eclipse.exe -vm "C:\oracle\jdk1.8.0_121\bin\javaw.exe"
```

The –vm parameter allows you to specify any version of the JRE (Java Runtime Environment) to run Eclipse (described in previous section). Eclipse will prompt you for the location and name of the workspace. You can give your workspace any name and any location. The location of the workspace does not need to be the same location as the report template. It's just a place for Eclipse to store the workspace settings. If there is a welcome screen, close the screen and proceed to create a report project in the next step.

Create a Project

To create a project:

1. Click **File » New and select Project » Business Intelligence and Reporting Tools » Report Project**.
2. On the Report Project page of the New Report Project wizard, enter a name for the project, and clear the **Use default location** check box.
3. Click **Browse** to locate the C:\deltek\costpoint\70\applications\birt\report\bin\custom folder, and click **Finish**.
4. If the Open Associated Perspective message box displays, select the **Remember my decision** check box, and click **Yes**. The Report Design perspective displays.
5. If the Navigator view is not displayed, click **Window » Show View » Navigator**.

   You can then expand the Navigator view to locate the report template that you have placed in the custom folder.

Set Up Report Design Preferences

To set up report design preferences:

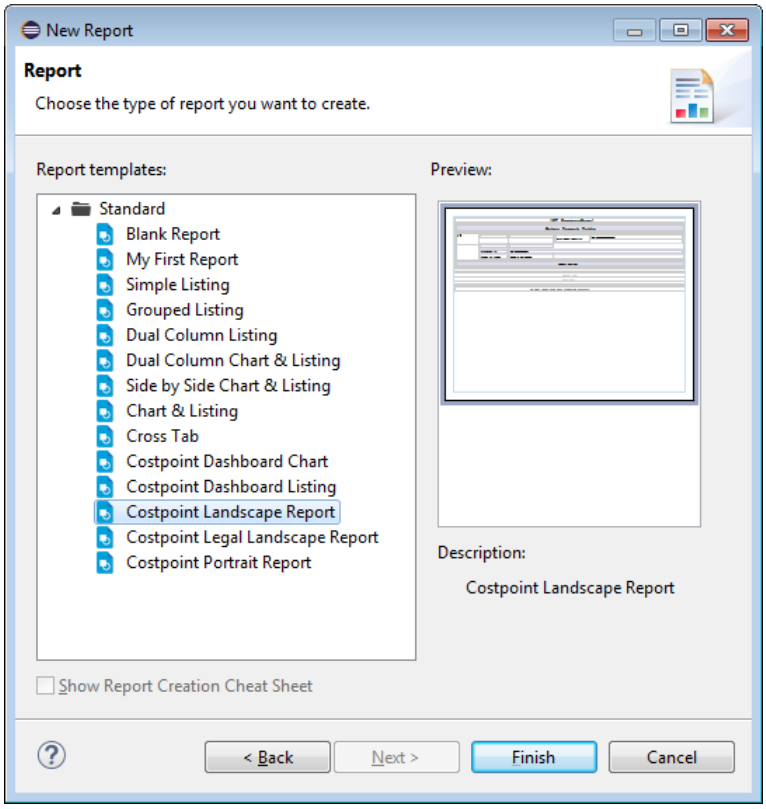
1. Click **Window » Preferences » Report Design**. The Preferences dialog box appears.
2. Select **Window » Preferences » Report Design » Resource** from the hierarchy tree on the left, and enter the report folder of the Costpoint server (for example, C:/Deltek/costpoint/70/applications/birt/lib/report) in the **Resource folder** field.
3. Select **Window » Preferences » Report Design »Template** from the hierarchy tree, and enter the report template folder of the Costpoint server (for example, C:/Deltek/costpoint/70/applications/birt/lib/report/template) in the **Template folder** field.

Test Costpoint BIRT Plugins

To test if the settings to Costpoint were correctly configured:

1. In Eclipse, right-click the extension unit in the Navigator view, and click **New » Report** the shortcut menu. The New Report wizard appears.
2. On the New Report wizard, leave all as defaults, and click **Next**.

You should see multiple choices for Costpoint template available.



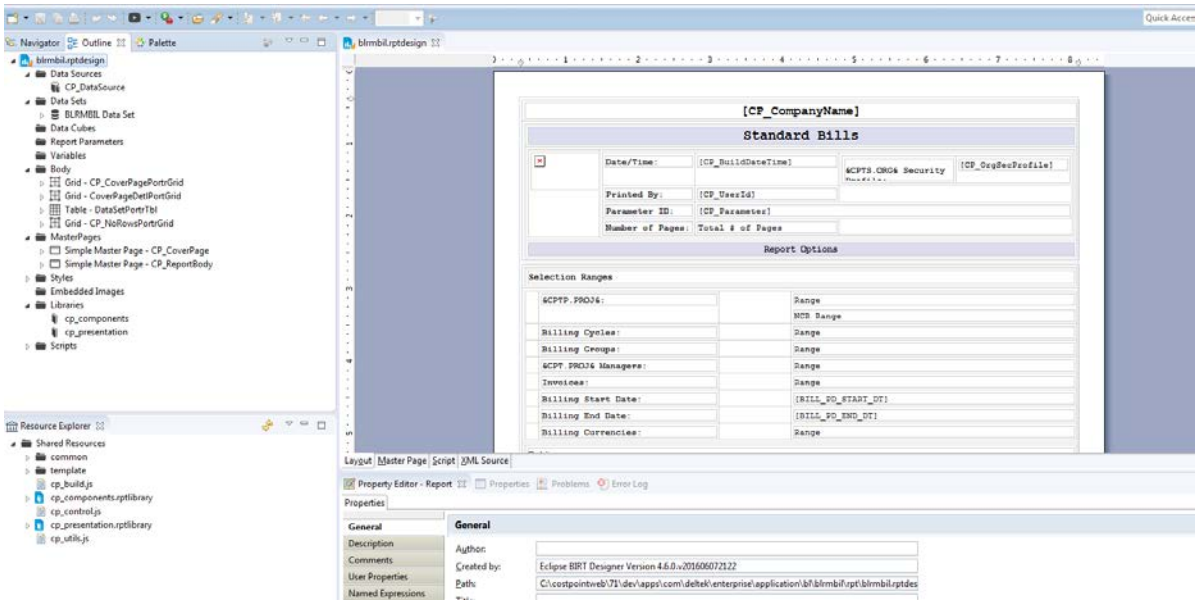
3. Click **Cancel**.

   You can now open and customize the report template.

Customize a Report Template in Eclipse

Report Template Outline

To view the outline of a report template in Eclipse, browse for the report template in the Navigator view, and open the Outline view.

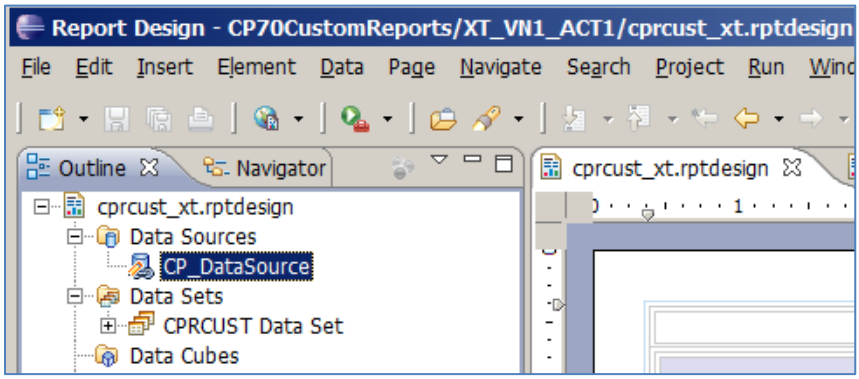


As shown in the example above, the following are the main components of a Costpoint report:

- **Data Sources**: Costpoint report uses its own data source type. The report does not connect and receive data directly from the database. Rather, it receives data from the Costpoint application server. This is one reason why you need to have the Costpoint plugins for Eclipse.
- **Data Sets**: Refers to the data setup in the report definition. The dataset defines which columns (fields) are being fed into the report from the data source result set.
- **Body**: This is the content of the report. Usually, it has two grids: cover page grid and report content grid. If a report contains multiple reporting sections, then you may see more than one table object in the Table section.
- **MasterPages**: Defines report cover page and main report data that repeats on each page, such as page headers, footers, and page numbers.

Data Source

On the Outline view, select and expand the **Data Sources** section.

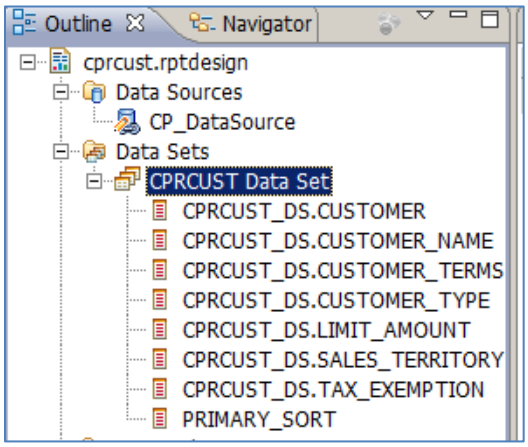


To change Costpoint data source settings:

1. Right-click **CP_DataSource** in the Outline view, and click **Edit** on the shortcut menu. The Edit Data Source dialog box displays. The data source may register the configuration folder location that does not exist in your local environment.
2. Change the value in the **Configuration Folder Location** field to C:\deltek\costpoint\71\applications\enterprise\properties, where the server's enterprise.properties file is located. This allows Eclipse to get the additional data fields added to the report (see next step).

Data Set

To view data sets, expand the Data Set folder in the Outline view.



The Data Set folder displays existing registered data fields that come from the data source result set. This was done by Deltek when the report was created. You may see additional columns in the folder. These are additional data fields computed in this report from other columns but not directly from the data source.

In the example, the extra field for XT_VEND_NAME to the data source and report definition in the Extensibility Designer is not listed. You must register it here before we can include it in the customized layout.

To register an extra field in the Data Sets folder:

1. Right-click the CPRCUST Data Set, and click **Edit** on the shortcut menu.
2. On the Edit Data Set dialog box, select **Query** from the hierarchy tree on the left, and complete the following actions in the Select System, Extension, Report, Data Source page.
   - From the **System Name** drop-down list, select the system in which you extended the report.
   - From the **Extension** drop-down list, select the unit in which the unit was extended.
   - From the **Report** drop-down list, select the report you extended.
   - From the **Data Source** drop-down list, select the data source that you extended.
3. Select **Output Columns** from the hierarchy tree on the left side.

   The additional column that you have added to the data set in the report definition is listed in the Output Columns page.
4. Click **Save**.

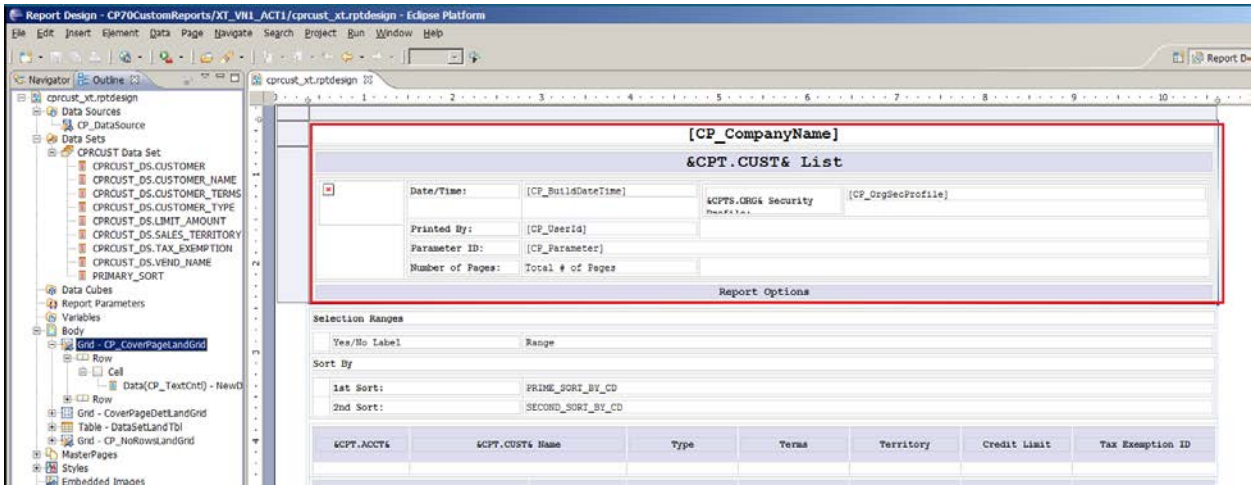
   The additional field is now available for use in the report.

Body

The body contains all the information for the report.

Grid CP_CoverPage

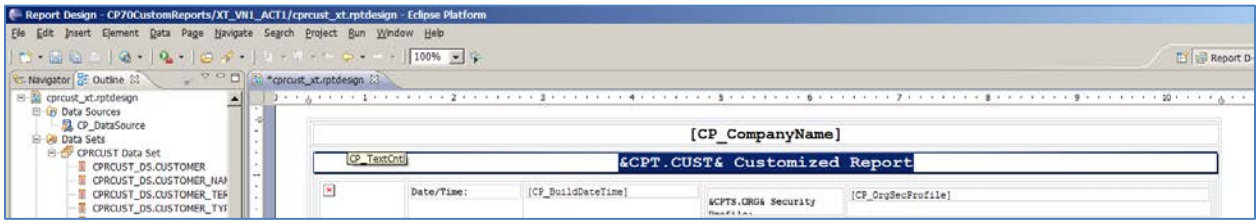
To view the top section of the cover page where the company name, report title, printed by information, and so on are printed, click **Body » CP_CoverPageLandGrid** from the Outline view.



The naming convention for CP_CoverPageLandGrid refers to the Costpoint cover page grid in landscape mode. If the report is in portrait mode, the name would be CP_CovePagePortrGrid. These are objects provided by Deltek in the library. You can browse for all supplied Costpoint BIRT objects in the Resource Explorer view.

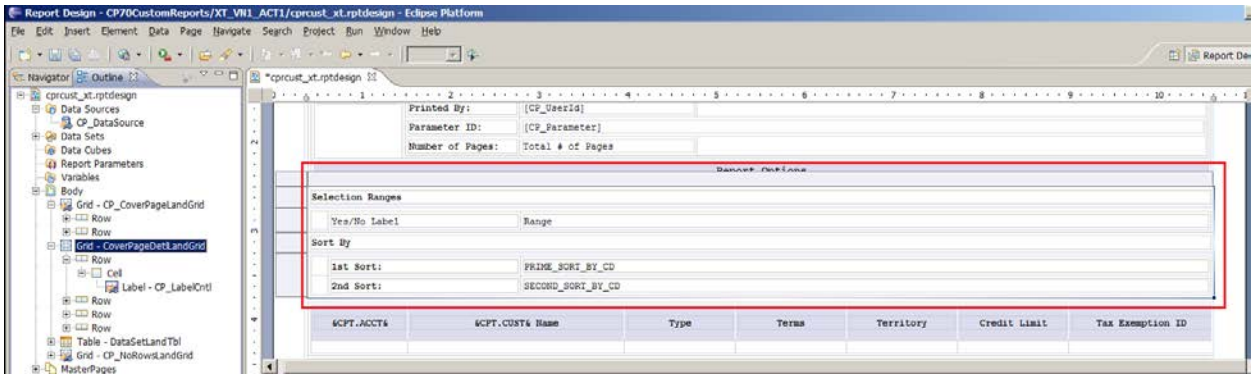
You can add additional info or rearrange the layout of the top section of the cover page. The title of “Customer List” report is set as &**CPT.CUST& List**. At run time, the &**CPT.CUST&** variable is replaced with the customized term set from the **Manage Standard Label** application in Costpoint. There are seven system-wide terms that you can customize. Use these variables instead of the hard coded English words for them if they are one of these.

To change the title, click the title on the report, and enter the new title.



Grid CP_CoverPageDetlLandGrid

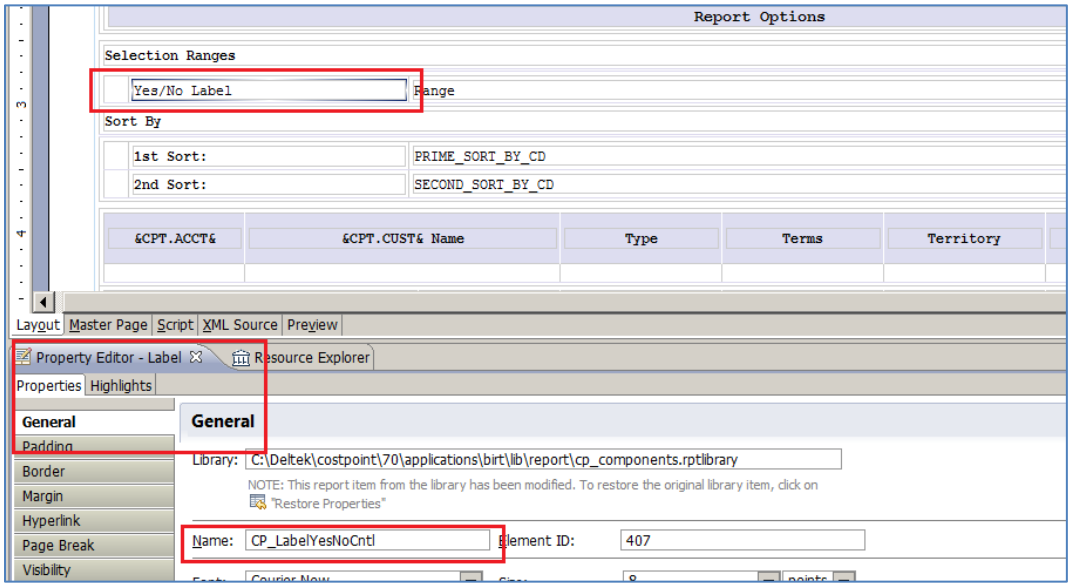
To view the selection a user has selected on the parameter page before the report (for example, selection range, sort, and so on), click **Body » Grid – CP_CoverPageDetlLandGrid** in the Outline view.



To edit the grid of the report:

1. Click any of the fields on the Layout view (for example, **Yes/No label** field), and click **Property Editor view » Properties**.

   This uses the CP_LabelYesNoCntl Costpoint class.

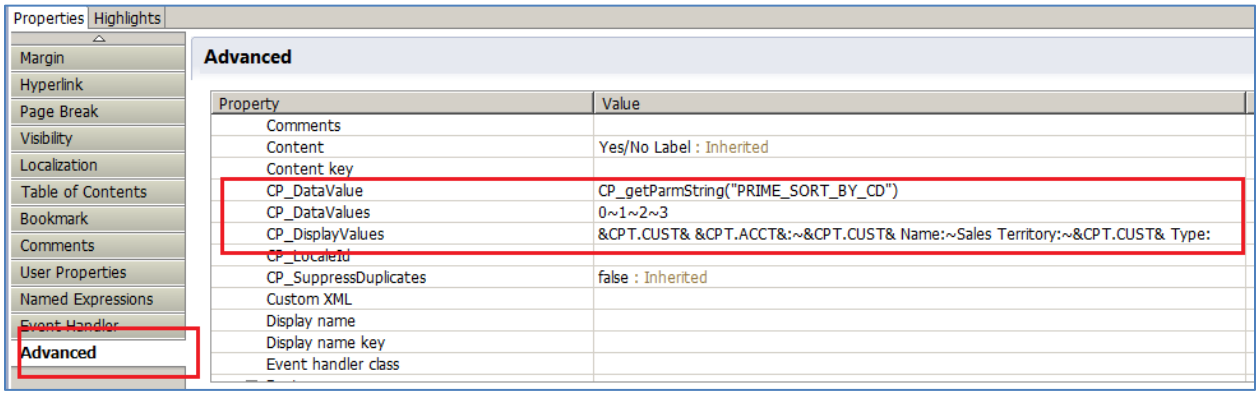


2. Click the Advanced tab in the Properties tab in the Property Editor view to view settings for this object.

For this CP_LabelYesNoCntl class, the CP_DataValue points to the data entered in the field PRIME_SORT_BY_CD (which is the object ID of the field on the parameter Result Set). CP_DataValues are the possible values for this field. There are four possible values, separated by a tilde (~) symbol.

CP_DisplayValues are the corresponding text to be displayed on the report according to the value returned in CP_DataValues. There are four possible text separated by a tilde (~) symbol. So if the return value is **0**, it will display **&CPT.CUST& &CPT.ACCT&**. If the return value is **1**, it will display **&CPT.CUST& Name**.

Since you are not likely to customize/extend the parameter result set, there is probably not much to change in this section.



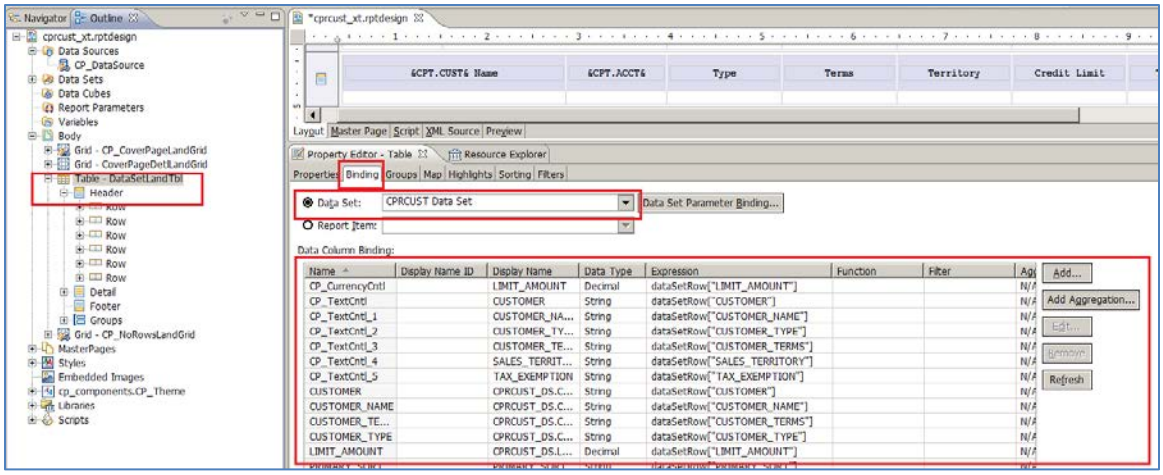
Table

The table is the main component of a report. It is the representation of the data source in visual format. It contains the detail content of the report. Generally, each table corresponds to a data set that supplies the data to the table. If you offer users multiple formats for a report (for example, choice of portrait or landscape), you can add an additional table format, but choose the same data set.

To add a table format in the report:

1. Select a table in the Outline view.
2. Click the Binding tab in the Property Editor view.

   The data set assigned to this table displays. All the fields from the data set are listed in the **Data Column Binding** grid. Each time you use these fields and place a control on the report for it, an additional row is listed here for the control. For example, on the first line of the grid, LIMIT_AMOUNT was shown on the report via the CP_CurrencyCtnl object. LIMIT_AMOUNT is also shown as an item coming from the data set.



3. Click the Properties tab, and then **Table of Contents**.

   On the right side is the text that will display at run time when the user selects the Table of Content on the report preview. CP_TranslateText is used to translate text (to another language for a Costpoint system that supports multiple languages – coming in version 7.1). Usually, when the text is set for a CP_Label control, it is automatically translated. When text is imbedded in another control such as this one, you must indicate that this text needs to be translated at run time.

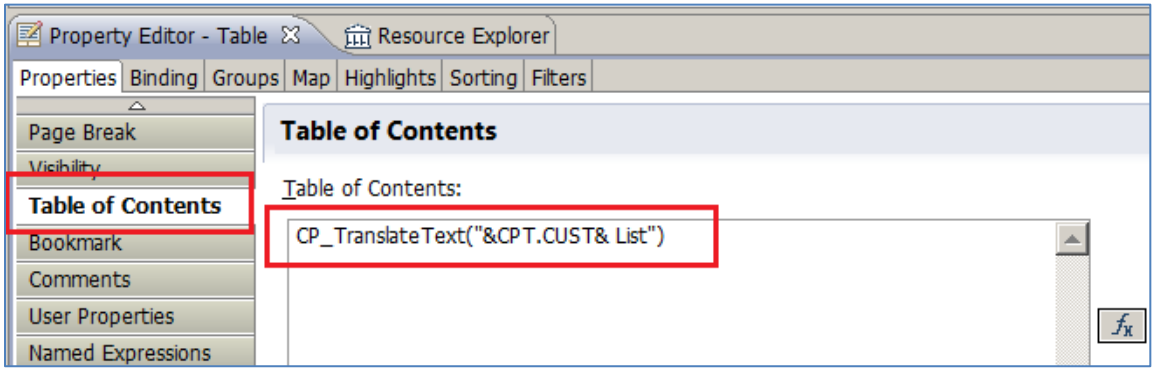
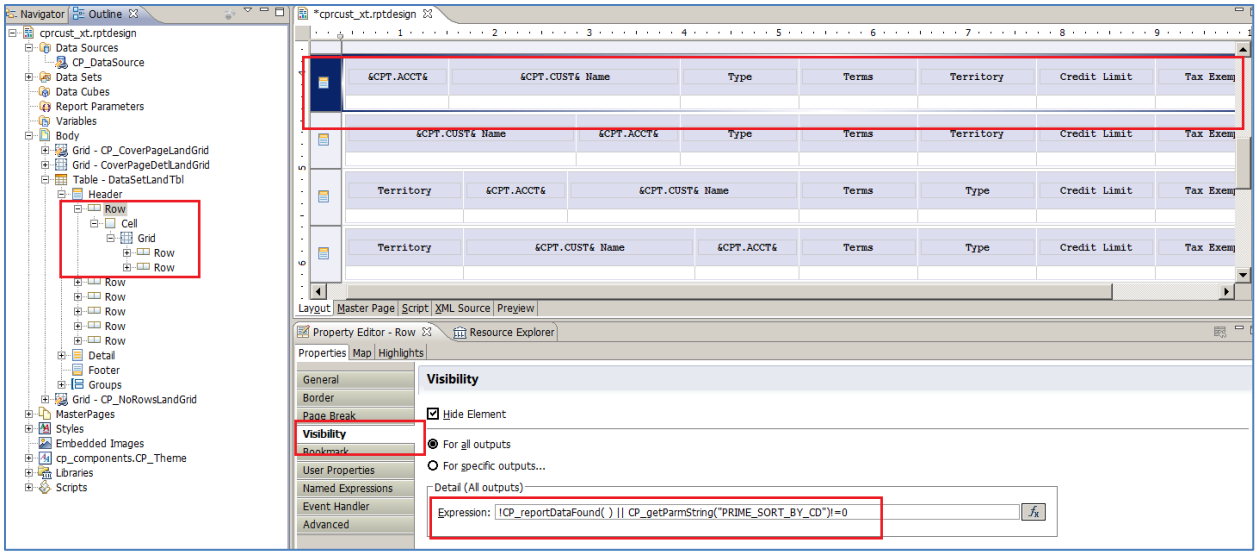


Table Header

A table header prints on each page of a report. It is generally used for static text such as column headings. It does not change from page to page. If there is data in it, it displays the data for the first row and remains the same for all pages. Only use table headers for static values.



To view visibility formulas, click **Visibility** on the Properties tab of the Property Editor view. A visibility formula controls when the object (in this case, the table row) should be hidden. If the formula returns true, it is hidden. Otherwise, it is visible.

In the following example, CP_reportDataFound is a Costpoint method that returns true if there is any row returned to this table. CP_getParmString gets the value of the object on the parameter result set. If there is no data found or the sort code is not 0, it is hidden. Otherwise, it is visible.

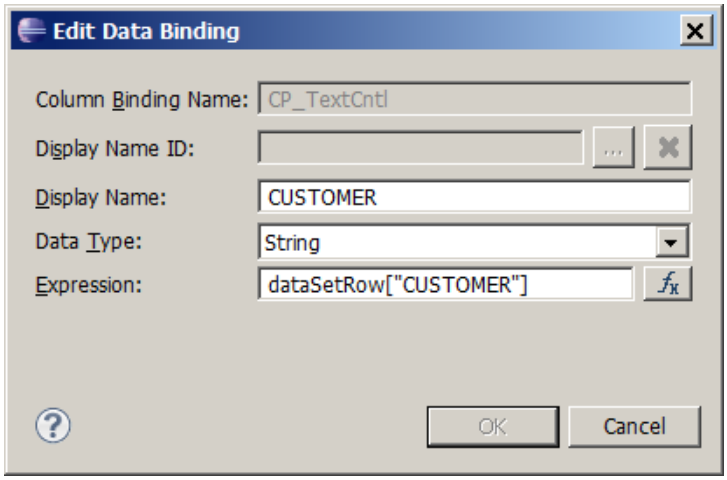
```
!CP_reportDataFound( ) || CP_getParmString("PRIME_SORT_BY_CD")!=0
```

Table Detail

Table detail prints for each row returned from the data set. You can have multiple rows for the detail if there are many columns. You generally use a grid to do indentation and spacing within a row. You can add additional rows to a grid if the additional row shares the same indentation and spacing. Otherwise, you add a new row and a new grid within it for different indentation.

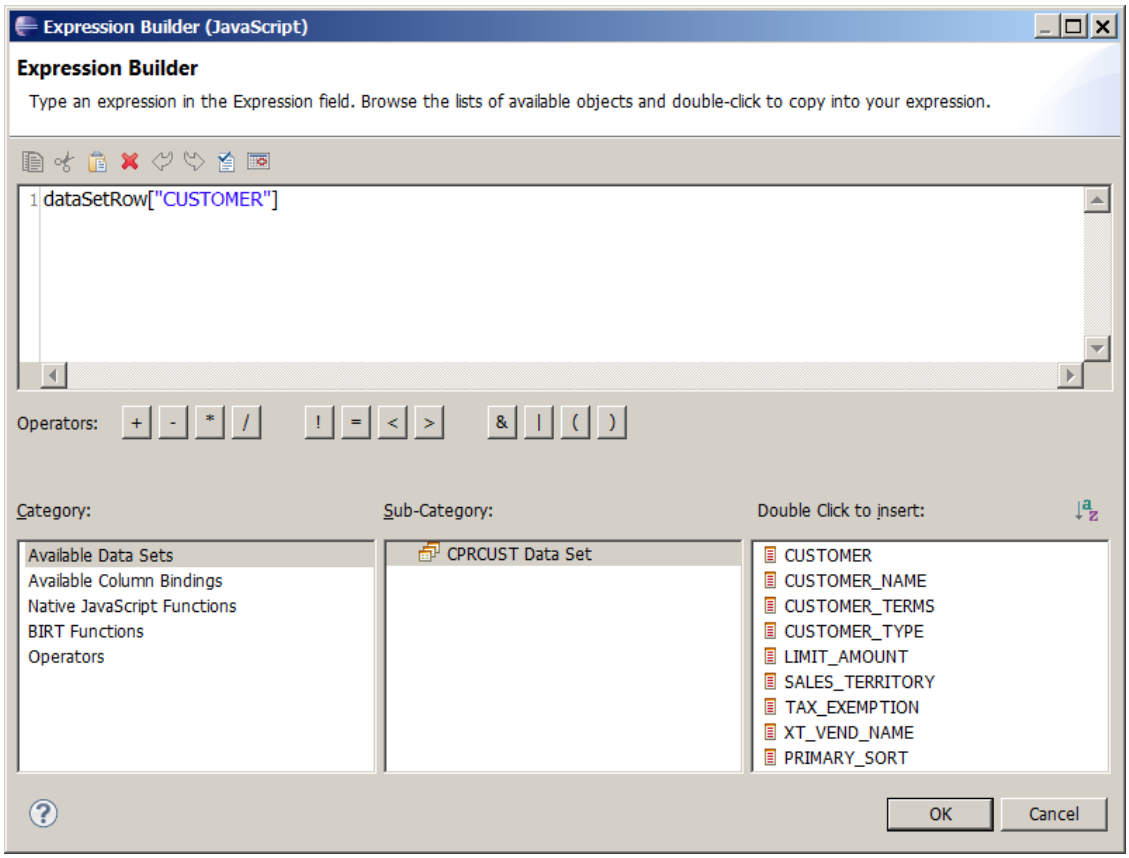
To extend a table detail:

1. Double-click a control on the Table Detail row (for example, CUSTOMER to display the Edit Data Binding dialog box.



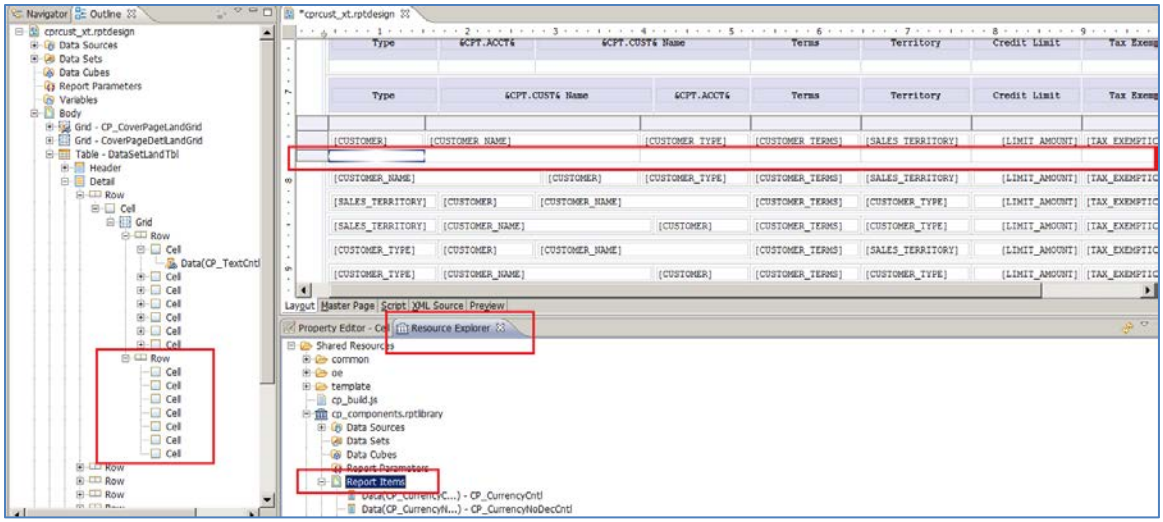
2. Click the formula (**fx**) button.

The Expression Builder dialog box displays. The pane at the top allows you to build the value of this field using the fields available in the lower panes. For example, if you want to display CUSTOMER, then select **Available Data Sets** in the **Category** pane, select **CPRCUST Data Set** in the **Sub-Category** pane, and double-click **CUSTOMER** in the **Double Click to insert** pane to set the value of in the top pane.

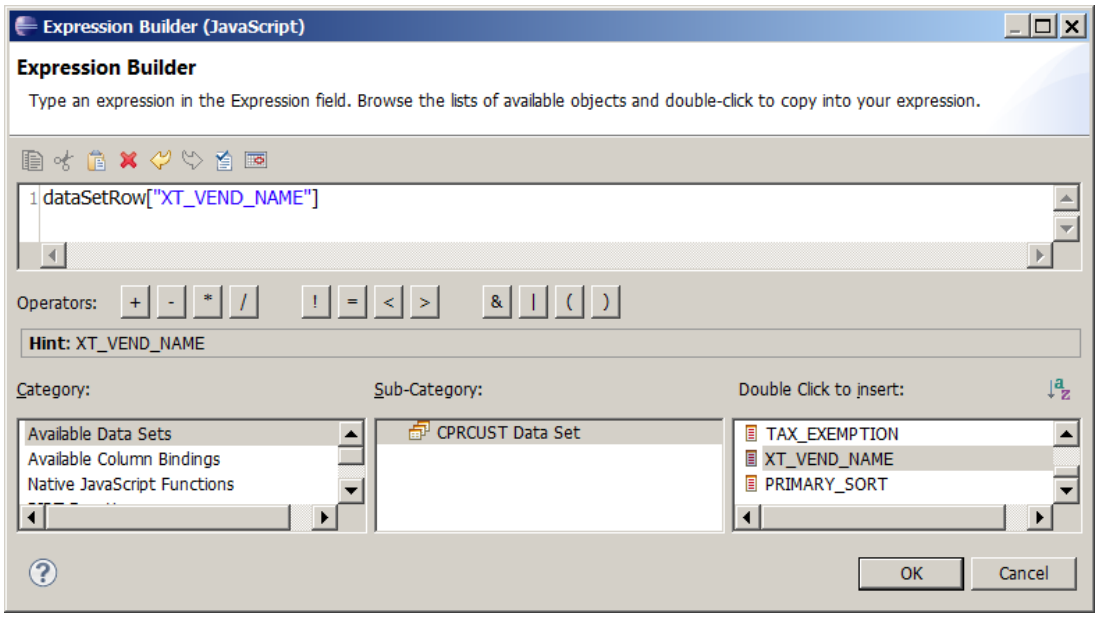


To add a row for the XT_VEND_NAME (from the data source example):

1. Right-click a grid in the Outline view, and click **Insert Row** on the shortcut menu. A new row is added on the Layout view.
2. Select the first cell of the new row, and open the Resource Explorer view. The list of Costpoint classes available for use as report controls display.



3. Drag and drop **CP_LabelCntl** to the first cell of the new row, and rename the text of the label to **&CPT.VEND& Name**.
4. For the second cell, drag and drop **CP_TextCntl** from the Resource Explorer tab to the second cell.
5. On the Edit Data Binding dialog box, click the formula (**fx**) button.
6. On the Expression Builder dialog box, double-click **EX_VEND_NAME** in the **Double Click to insert** pane, and click **OK**.



7. On the Edit Data Binding dialog box, change the **Display Name** to **XT_VEND_NAME**.

   This display name is not important, but setting it to the same name in the data set helps to quickly identify the object.
8. Click **OK**. The new field is added to the report by extending the data source result set, the report definition, and the report template.

To set or change the format of the object, click **Property Editor – Data » Properties » General**. You can change the font, point size, color, indentation, and so.

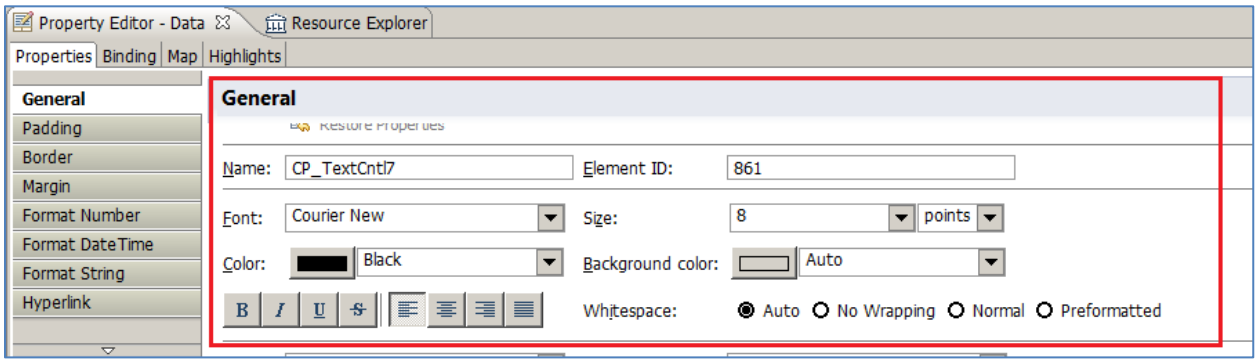


Table Footer

Similar to a table header, a table footer is used to print static data, such as some disclaimer text, and so on. You can also set the static text on the Master Page.

Table Groups

A table group is used to control and display information for data when there is a break in the group. A group is a key column used for sorting. For example, assume you have data coming in as in the following table.

ID	Line No
A	1
A	2
B	1
B	2

If an ID is chosen as the group key, when the value changes from A to B, there is a break. You can display the ID once and omit it from the table detail section which will only print Line No.

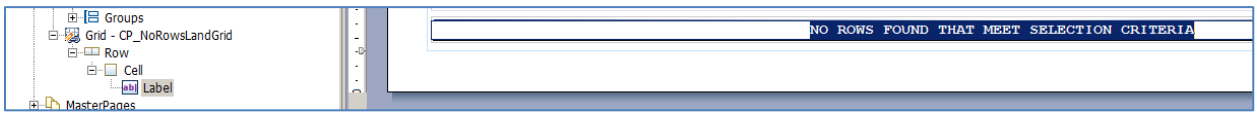
To add a group:

1. Right-click **Groups**, and click **Insert Group** on the shortcut menu.
2. On the New Group dialog box, enter a descriptive name for this group in the **Name** field.
3. In the **Group On** drop-down list, select a column from the data set as key for this group.
4. In the **Page break** group box, set the page breaks.
5. Enter the values to display (usually the same data set column) in the **Item Expression** field. This will display as an index on the table of contents for quick jump/navigation.

To edit an existing group, right-click the group, and click **Edit Group** on the shortcut menu.

Grid CP_NoRowsLandGrid

This control is used to print the text when there is no row found on the report. Click the field to customize this standard text.



MasterPages

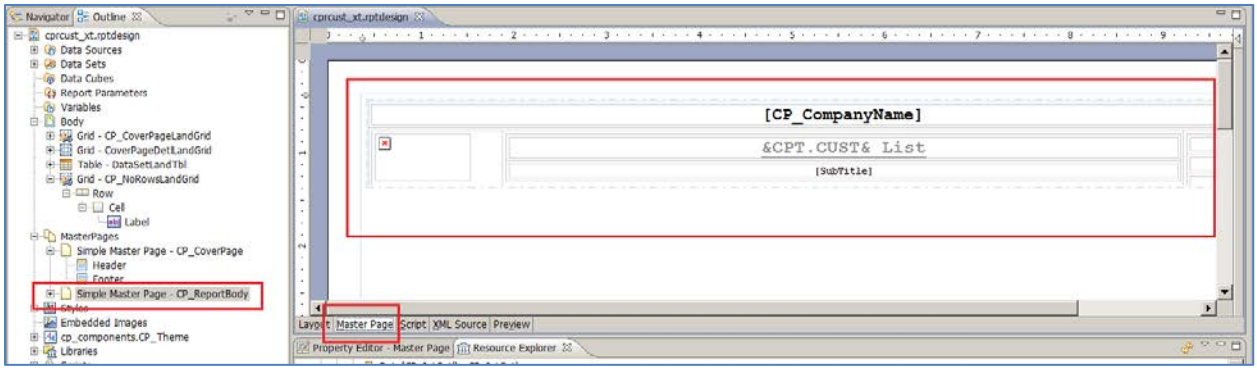
A MasterPage is a frame where the body information is inserted.

CP_CoverPage

This is the mat for the cover page. A standard Costpoint report does not have any text for the cover page MasterPage. You can add additional text or add the text directly on the cover page.

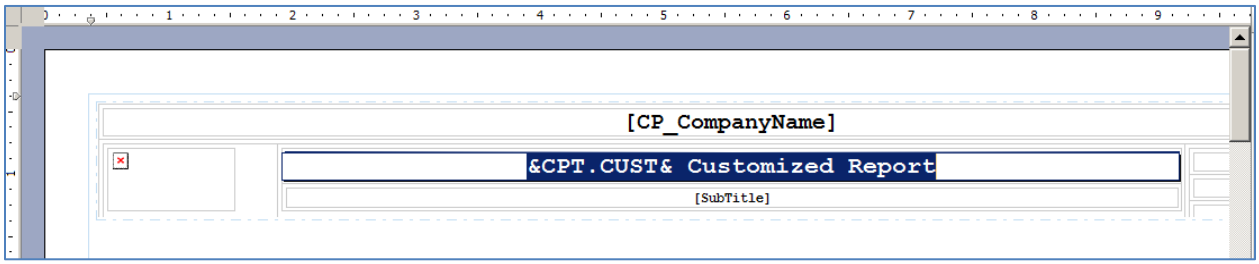
CP_ReportBody

This is the mat for the report body. To view the layout of the Master Page in the Layout view, open the Master Page view.



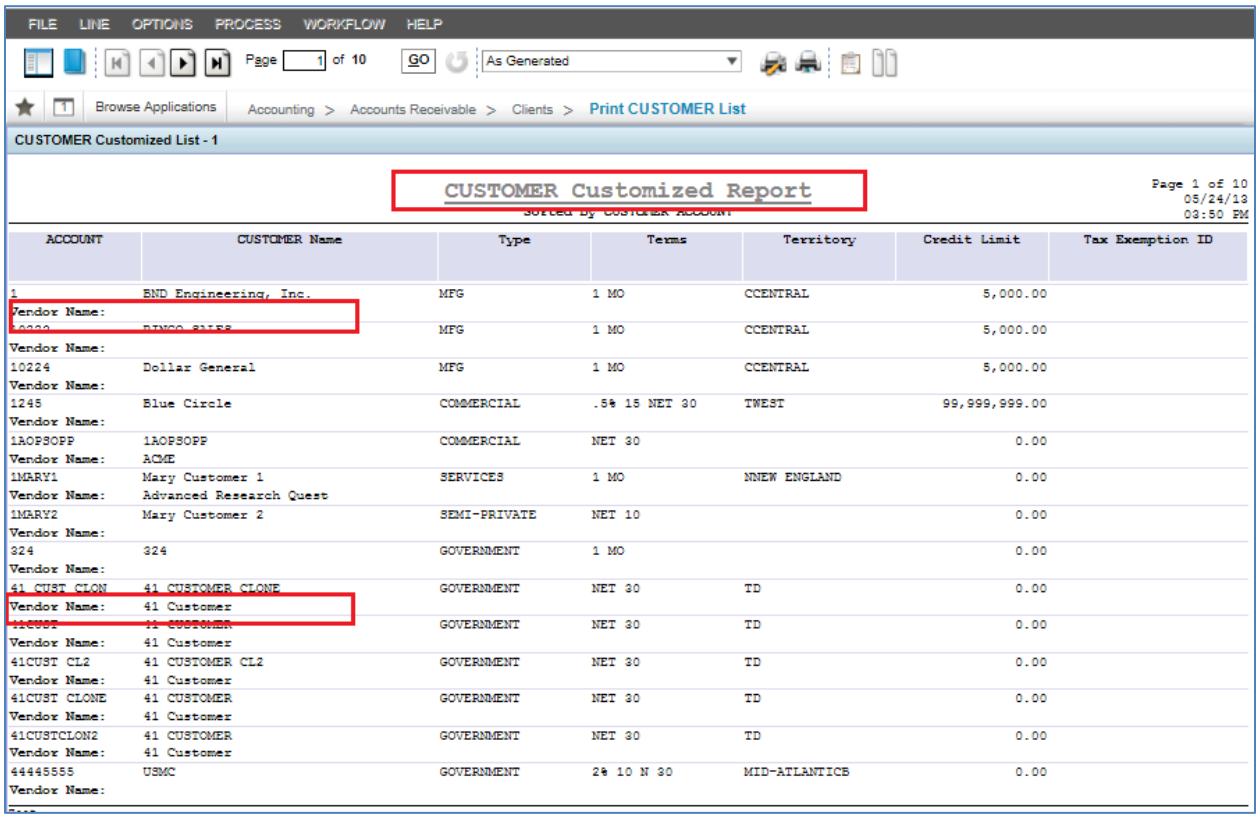
Similar to the standard information on the cover page, the placement for company logo, the company name, the report name, and any extra information is displayed in the **SubTitle** row.

To change the title of the report, you must make the change in the Master Page view.



Deploy the Extension

You need to make sure the template is copied or located at the correct location at the server. When you deploy the extension unit to the server, you will be able to see the result of the customization when you run the report in Costpoint.



ACCOUNT	CUSTOMER Name	Type	Terms	Territory	Credit Limit	Tax Exemption ID
1	BND Engineering, Inc.	MFG	1 MO	CCENTRAL	5,000.00	
Vendor Name:						
[illegible]	[illegible]	MFG	1 MO	CCENTRAL	5,000.00	
Vendor Name:						
10224	Dollar General	MFG	1 MO	CCENTRAL	5,000.00	
Vendor Name:						
1245	Blue Circle	COMMERCIAL	.5% 15 NET 30	TWEST	99,999,999.00	
Vendor Name:						
1AOPSOPP	1AOPSOPP	COMMERCIAL	NET 30		0.00	
Vendor Name:	ACME					
1MARY1	Mary Customer 1	SERVICES	1 MO	NNEW ENGLAND	0.00	
Vendor Name:	Advanced Research Quest					
1MARY2	Mary Customer 2	SEMI-PRIVATE	NET 10		0.00	
Vendor Name:						
324	324	GOVERNMENT	1 MO		0.00	
Vendor Name:						
41_CUST_CLON	41 CUSTOMER CLONE	GOVERNMENT	NET 30	TD	0.00	
Vendor Name:	41 Customer					
[illegible]	[illegible]	GOVERNMENT	NET 30	TD	0.00	
Vendor Name:	41 Customer					
41CUST CL2	41 CUSTOMER CL2	GOVERNMENT	NET 30	TD	0.00	
Vendor Name:	41 Customer					
41CUST CLONE	41 CUSTOMER	GOVERNMENT	NET 30	TD	0.00	
Vendor Name:	41 Customer					
41CUSTCLON2	41 CUSTOMER	GOVERNMENT	NET 30	TD	0.00	
Vendor Name:	41 Customer					
44445555	USMC	GOVERNMENT	2% 10 N 30	MID-ATLANTICB	0.00	
Vendor Name:						

Refer to the *Deltek Costpoint Extensibility Designer Guide* for more information about deploying your extension unit.

Guidelines and References

Customizing Controls

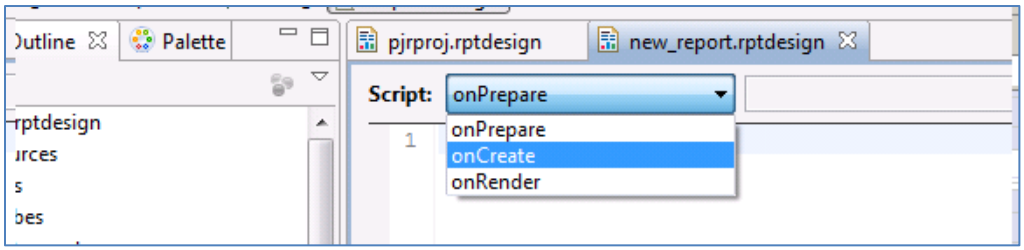
Only controls from Costpoint BIRT libraries should be used. Controls can be customized in different ways:

- Value expression
- BIRT control properties
- Costpoint (CP) control properties
- Visibility (show/hide)
- Scripting (use scripting only when other means for control customization cannot provide the desired result)

Scripting and Report Event Handlers

BIRT provides a powerful scripting capability that allows a report developer to create custom code to control various aspects of report creation. Each control or component in BIRT provides scripting capabilities. When a report runs, it first goes through a generation phase and then goes through a rendering phase. Therefore, there are event handlers (control scripting methods) that are called at generation time only and event handlers that are called at rendering time only.

- If you need to customize a control at generation phase, use the **onCreate** event.
- If you need to customize a control at the presentation phase, use the **onRender** event.



Writing Expressions

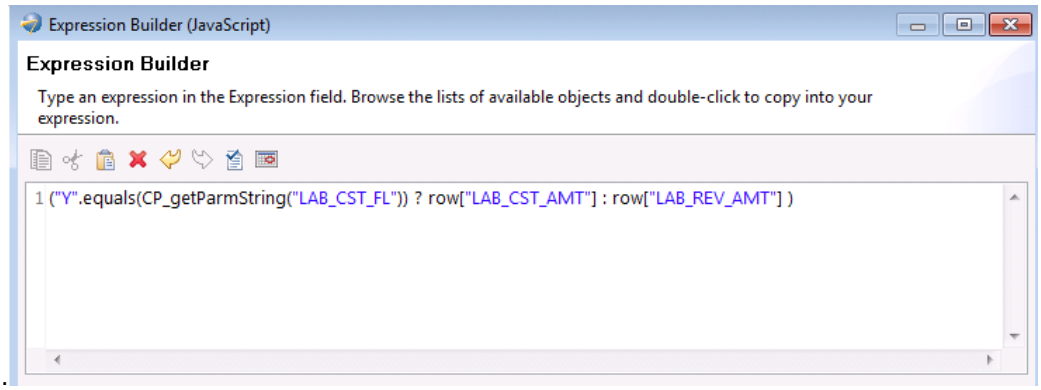
The BIRT Scripting language derives from Java/JavaScript. A BIRT expression is a statement that produces a value. An expression can be a numeric, date or literal value, such as:

- 3.14
- "abc" + var1

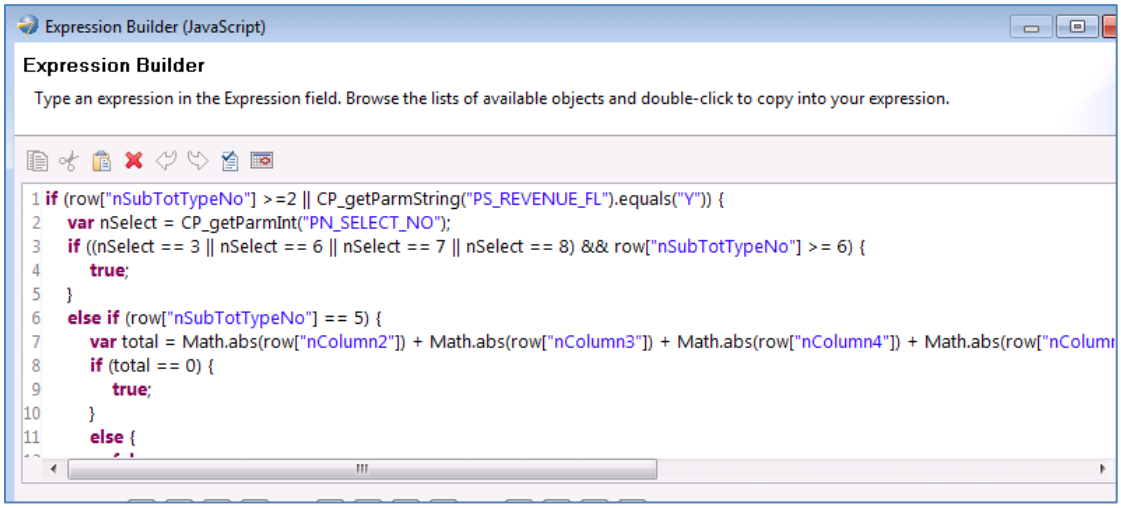
When you drag a field into the report, BIRT Report Designer creates a column binding with the correct expression for you. An expression can contain any combination of literal values, fields, operators, variables, and functions as long as it evaluates to a single value:

- "Order Total: " + row["orderTotal"]
- row["orderDate"].getYear()
- row["itemQuantity"] * row["itemPrice"]

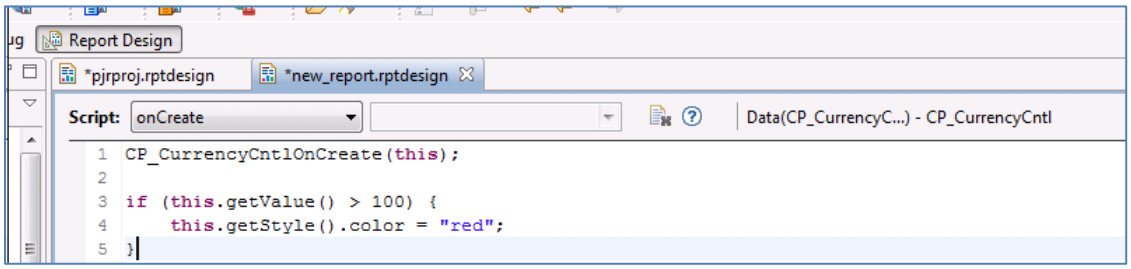
You use expressions when you need to change the value dynamically. See the following example.



Or, you use expressions when you need to hide or show the control dynamically as shown in the following example.



Or, you use expressions to customize the appearance dynamically as shown in the following example.

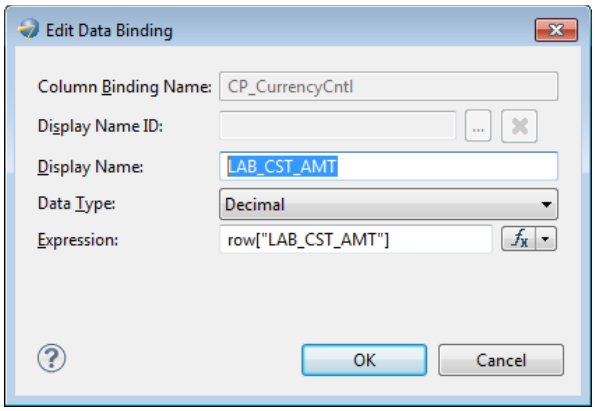


Use scripting only when other means for control customization cannot provide the desired result.

All standard Costpoint controls have CP_ControlNameEventHandler(this) methods. If you override a control event handler, keep CP_ControlNameEventHandler(this) as the very first line of custom code. Never remove the CP_ControlNameEventHandler(this) call.

Naming Conventions

Use a descriptive name for the control display name (for example, PROJ_ID, ACCT_ID, ORG_ID, PageHeaderProj).

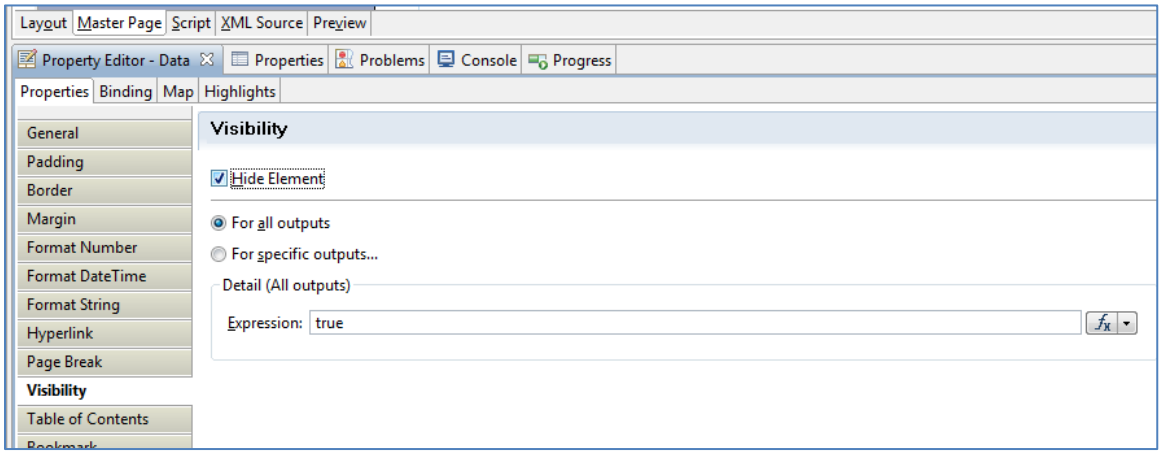


Hiding Element

A visibility formula is available for all data controls, grids, tables, grid rows, and table rows.

To engage conditional show/hide:

1. Click **Properties** » **Visibility** tab
2. Select the **Hide Element** check box.
3. Display the Expression Builder, and enter an expression to hide an element.



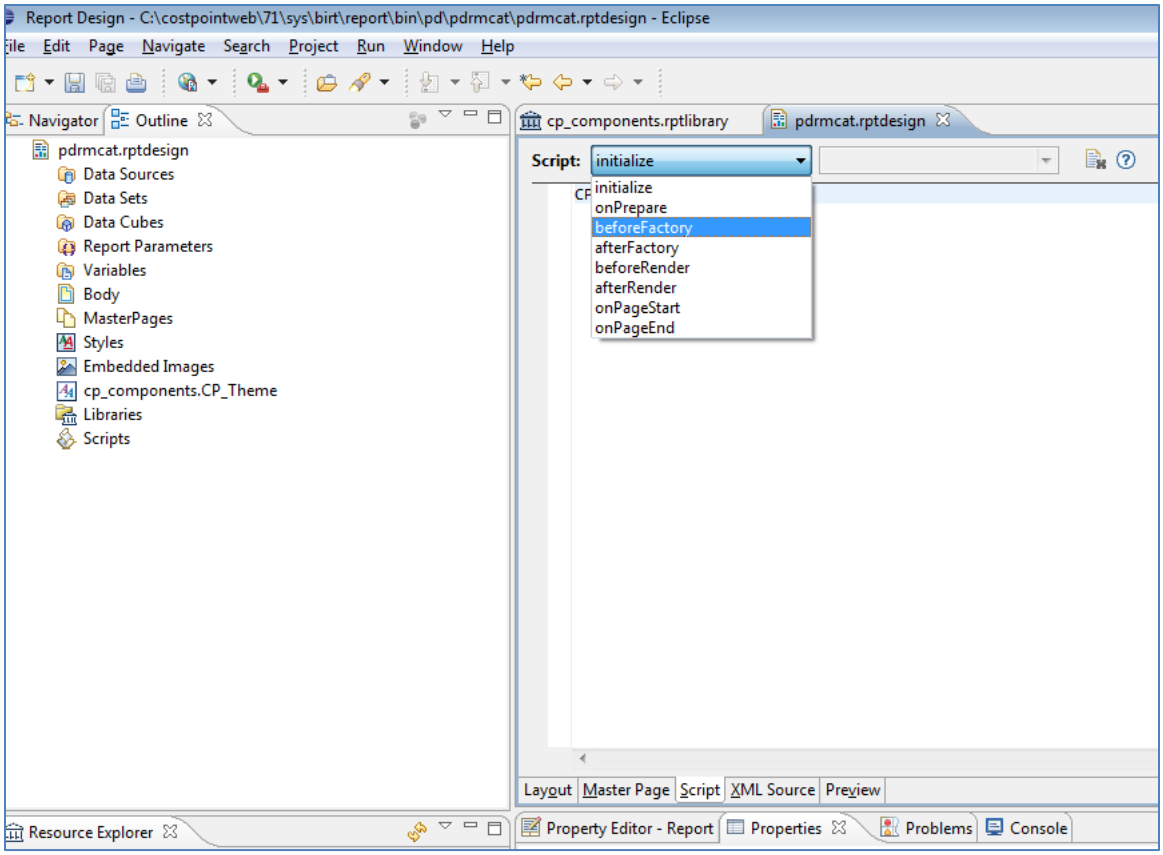
Important Note about Hiding a Table Object

If you need to hide a table object, then do not use the table visibility formula. While using the table visibility formula will hide the table when the report is rendered, the table itself is still executed at runtime. This means that the table processes all assigned data sources/sets and all grids and controls that the table contains. Report performance will be impacted badly because of this.

Instead of using a table visibility formula, you need to suppress the table from being executed in the first place.

To suppress a table from being executed:

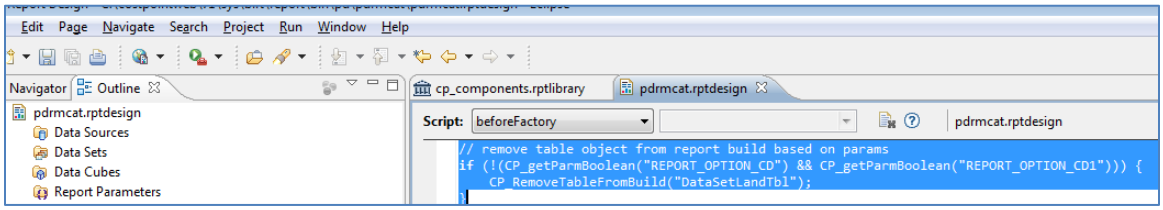
1. Display the Report Outline, and locate the top most report element (for example, pdrmcat.rptdesign).
2. On the Script tab, override the **beforeFactory** method.



You may reference only report parameters and/or report constants within the expression that drives whether or not to remove the table from report execution.

3. Enter the following script:

```
// remove table object from report build based on params
if (!CP_getParmBoolean("REPORT_OPTION_CD")) {
	CP_RemoveTableFromBuild("DataSetLandTbl");
}
```



Report Page Settings

Unless it is an external report with a special page size and margins requirements, the guideline is to keep all margins equal to 0.38”. This margin size is the minimum standard that must be followed. Less than 0.38" will cause the report not to fit properly and look undesirable when switching to A4 paper.

If your report margins are less than 0.38", notify the Design and Architecture teams so the report margins can be reviewed for the possible need of redesign or rework.

Possible options:

- If it is a Landscape report and C/S left/right margins are less than 0.38", then keeping the C/S margins on the Web (using less than 0.38") is fine. An alternative would be to decrease the font size from 8 to 7 in some sections of a report (for example, the detail section) and keep the standard margins. This alternative is acceptable, but less preferable, compared to decreasing the margins and keeping the font size standard.
- If it is a Portrait report and C/S left/right margins are less than 0.38", then, on the web, margins must be 0.38" and the font size may be decreased from 8 to 7 in some sections of a report (for example, the detail section) to fit the data.

The Design team must be informed before choosing either approach.

Margins cannot be smaller than 0.25” for Landscape reports and 0.38” for Portrait reports. Having appropriate margins ensures that a report can be printed without data truncation. Generally, printers have their own small margins as well.

If you are working on an external form that has specific page size and margins requirements, then follow these requirements precisely since the only rule is to replicate a special form as close as possible.

Margins and paper size/orientation in the design tool MUST match margins and paper size in the *.rptdesign file. The system applies the following defaults if margins and/or paper size/orientation are not specified in the design tool:

- paper size code=Letter
- orientation=landscape
- all margins=0.38

So if a report template falls in this default category, then it is not required to provide page size/orientation and margins in the design tool. In all other cases, developers are required to enter page settings in the tool.

If a report is designed for Legal or A4 paper or if a report is a government form, then you should select the **Always Use Design Settings** check box on the Enter General Report Info screen in the design tool. Selecting this check box instructs the system to ignore the system/user page settings and use the report design time page settings.

Cover Page

All Costpoint reports must have a cover page. A cover page must be one page only. If there are many options, they should be arranged in columns across the page in order to use up as much space as possible. If the report has non-contiguous selections, the printing of selections should be limited so that they do not cause a page break. If necessary, some non-contiguous selections should not be printed if there is no room for them on the page.

- The cover page is provided by the Costpoint templates as the starting point when creating a new report.
- The cover page object can be re-inserted from cp_presenation.rptlibrary intothe report design.

To add a cover page:

1. In the Resource Explorer, locate cp_presenation.rptlibrary.
2. Drag the cover page object from the library into the report design.

Page Header/Footer

All internal reports must have a standard page header and footer. The page header/footer is provided by the Costpoint templates as the starting point when creating a new report. These objects can be re-inserted from cp_presentation.rptlibrary into the report design.

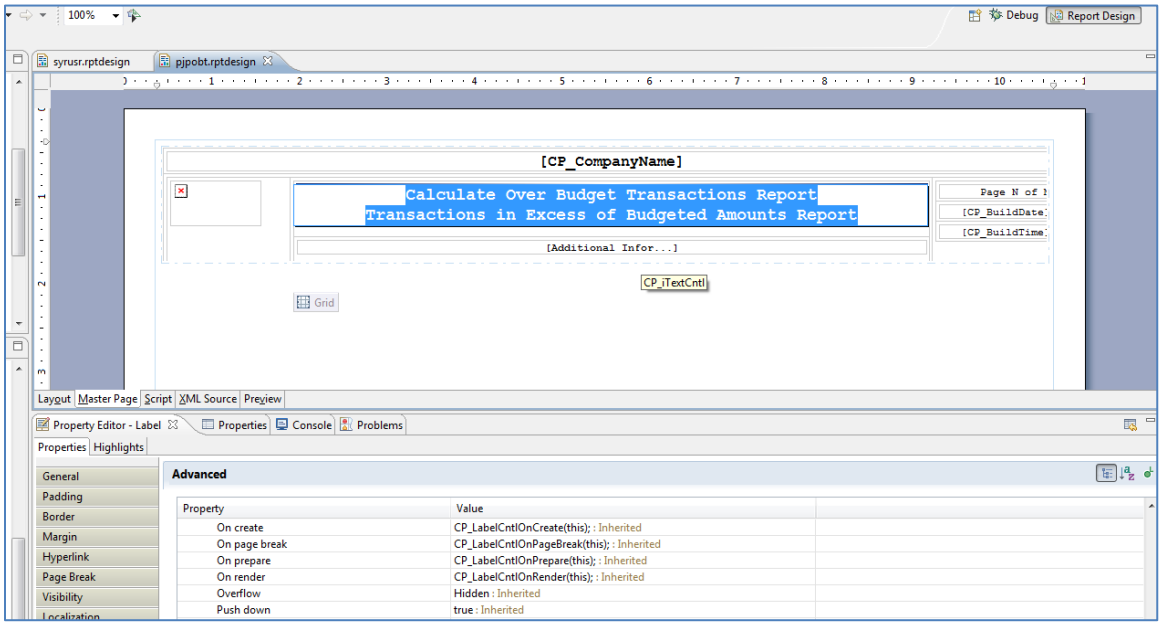
To create a page header/footer:

1. On the Master Page tab in the Report Design window, use Outline view to locate the CP_ReportBody page under the MasterPages component.
2. In Resource Explorer, locate cp_presenation.rptlibrary and drag the page header/footer objects from the library to the **CP_ReportBody » Header/Footer** folder in the Outline view.

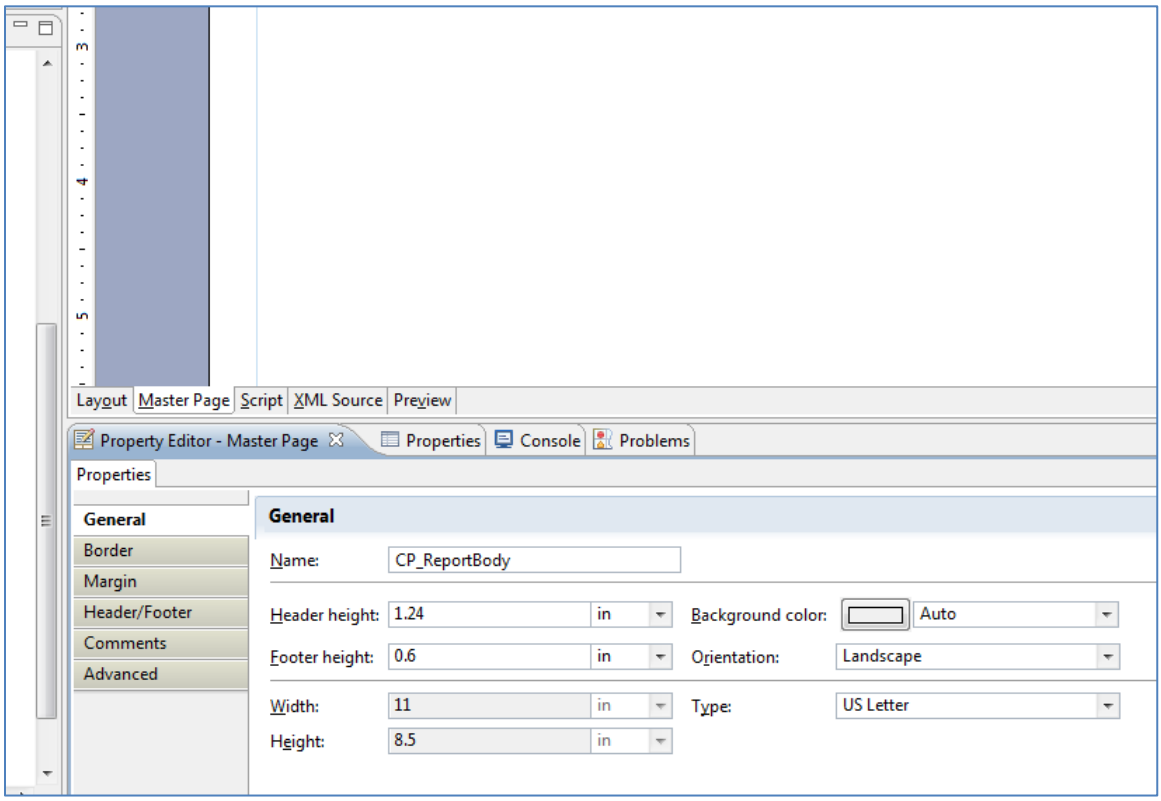
In case of multiple report sections and different page settings, you may need to create additional page objects and place standard Costpoint page headers/footers onto new pages.

To create the new page object, right-click **MasterPages** in the Outline view, and click **Insert Element** on the shortcut menu. Rename the new object (via the right-click menu), and change the page settings (page size, margins, and so on), if needed.

The default height of the standard Costpoint header is 1 inch. You should change the Page Header default height only when you need to present more data that may not fit vertically. In the following example, the report title has two lines of text.



You can change the **Master Page » CP_ReportBody (page) » Properties » General » Header Height** from 1 inch to 1.24 inches.

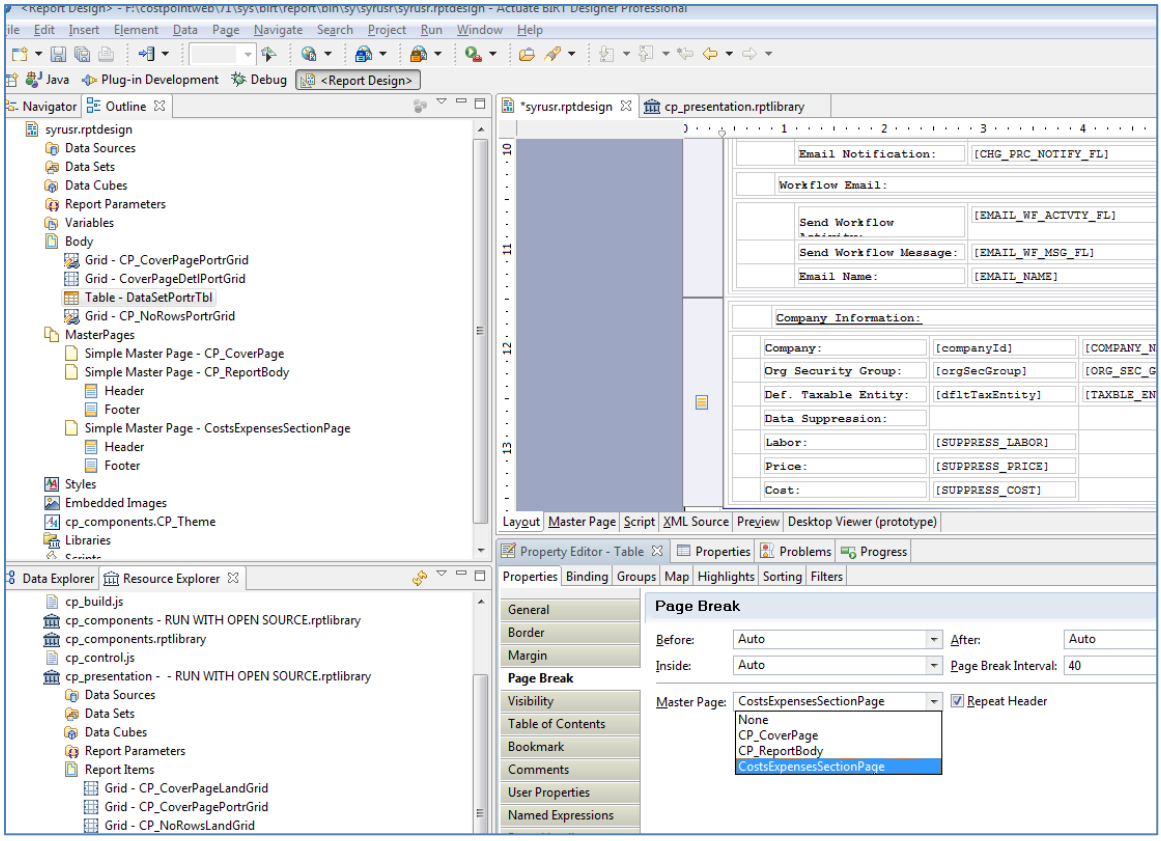


- If your report title has 1 line, then the standard page header height should be **1 in** - default.
- If your report title has 2 lines, then the standard page header height should be **1.24 in**.
- If your report title has 3 lines, then the standard page header height should be **1.48 in**.

The default height of the standard Costpoint footer is **0.6** in. You should not change the footer height.

You can insert standard Costpoint page header/footer objects from cp_presenation.rptlibrary into the new page by dragging page header/footer objects from the library directly into **CP_ReportBody page » Header/Footer** in the Outline view.

The new page object should be correctly mapped to your table object in the report design. Switch to Layout view and in the **Body** folder of the Outline view, locate the table object that should display data for different report section. Click **Table » Property Editor » Page Break**, and change the master page to the page that you crated. As the result, all data that goes into the above table will be displayed on a page with possibly different layout margins, headers, and footers.



To add report abbreviation data in the page footer:

1. On the the Master Page tab in the Report Design window, use the Report Outline view on the left to locate CP_ReportBody page under the Master Pages component.
2. Scroll to the bottom of the page, and select the page footer grid.
3. Select the grid row that contains the **Enter Report Abbreviation Data Here** text.
4. In the **Row » Property Editor » Visibility** section, and click the formula (fx) button beside the **Expression** field. The Expr Builder dialog box displays.
5. Set the visibility expression to **false**, and click **Ok**.
6. Click the **Enter Report Abbreviation Data Here** text, and change its value to a report specific abbreviation text (for example, ACCT: Account; PROJ: Project).

Access DataSet Variables in Page Header

By default, BIRT does not allow access to DataSet variables on the page header. This is the limitation of BIRT where, for performance reasons, page header/footer objects are static and created only once for whole report/table section/object.

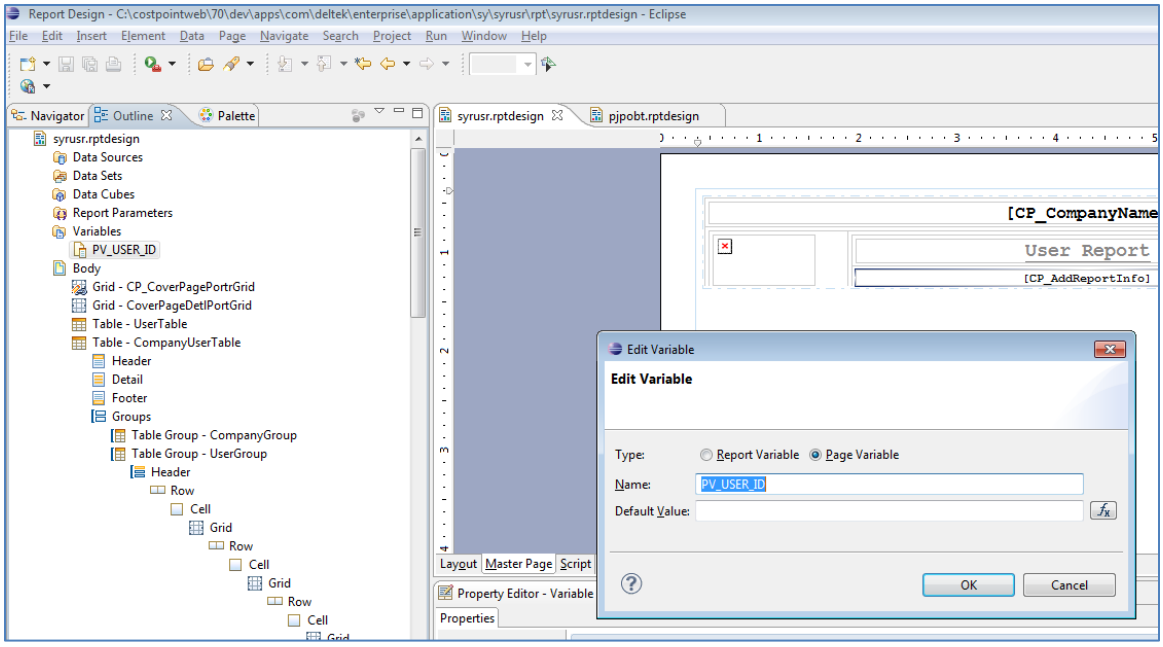
However, you can display DataSet variables in **Page Header » CP_AddReportInfo** (CP_iTextCntl) control by using the BIRT Page Variables.

- The BIRT page variable are only to be used in the page header and only within the CP_iTextCntl control.
- The BIRT Page Variable will hold the value of the last DataSet row that falls on that page.

- The variable must be a break group control

To use the BIRT page variables:

1. Add a new page variable in **Variables » New Variable » Page Variable**.

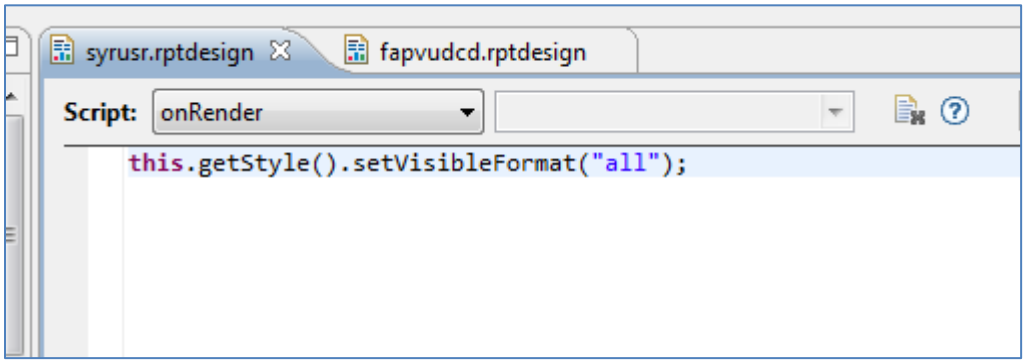


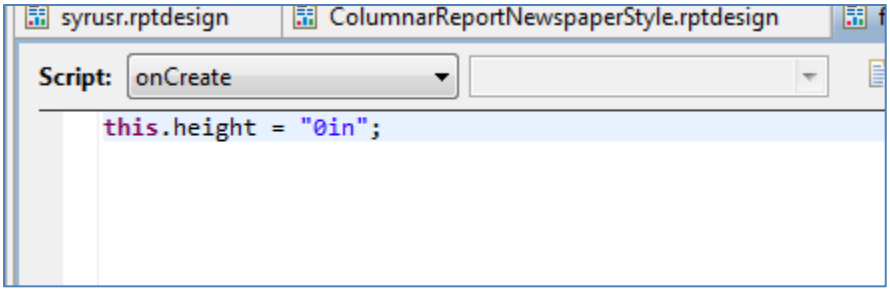
2. Name the variable using the following naming convention:
   - The variable name should be uppercase and should start with PV_ prefix (for example, PV_USER_ID).
3. Set the value of the page variable by overriding the **onPageBreak()** method for the control that displays the DataSet data that you want to use in the page header.

If such a control is not there or should not be in the report by design, you need to create a dummy table/grid row, put a dummy control on that row, set control value, and make sure that this dummy control's parent row is hidden through:

```
overridden Row>Scrip>OnRender()
method>this.getStyle().setVisibleFormat("all");
overridden Row>Scrip>OnCreate() method>this.height = "0in";
```

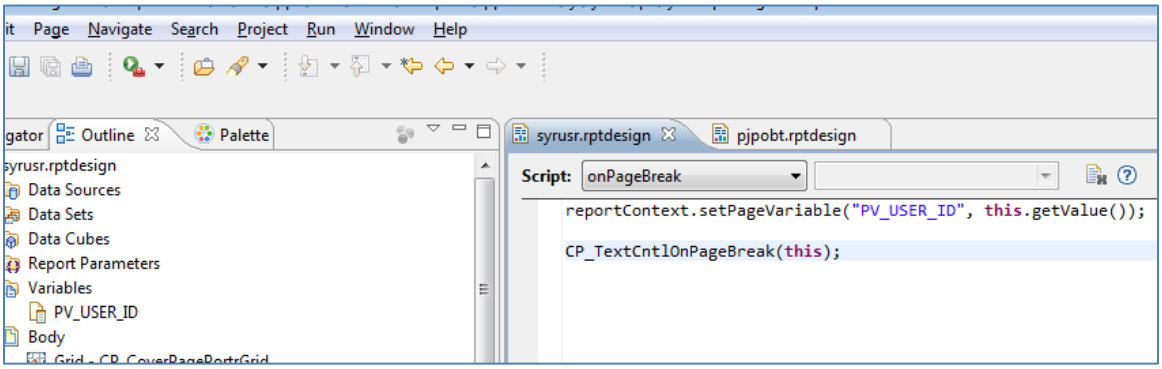
Do not hide the row or control using the **Hide Element** check box or set row height to **0**. Use the above **setVisibleFormat("all")** approach only.





The control's **onPageBreak()** method should be overridden to set the page variable value. For example:

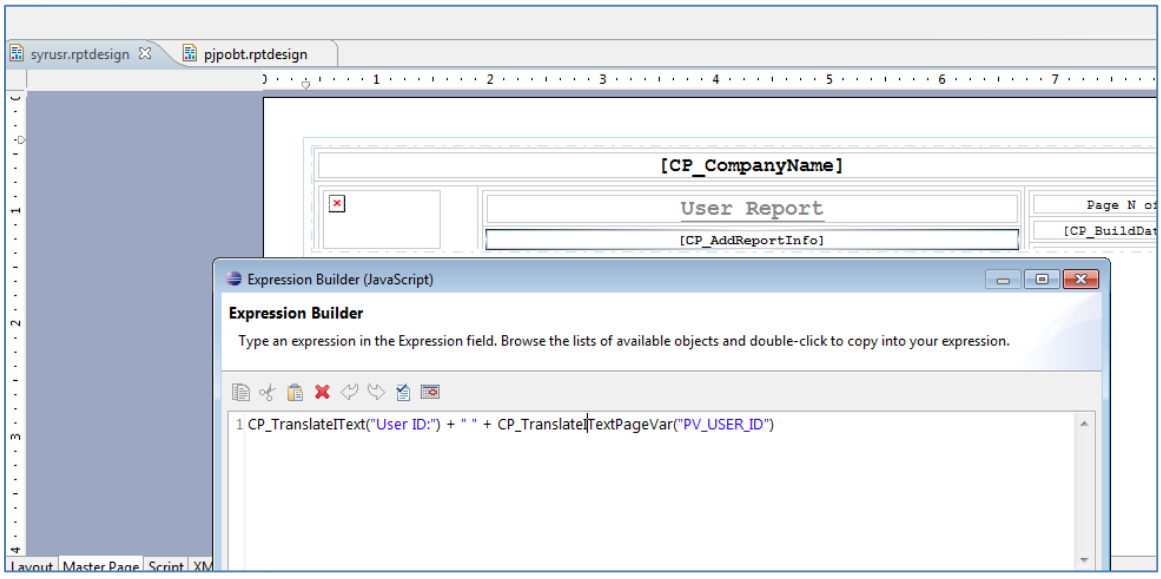
```
reportContext.setPageVariable("PV_USER_ID", this.getValue()
```



4. Go to **Master Page » CP_ReportBodyPage » CP_AddReportInfo » Expression**, and reference the page variable value using the **CP_TranslateITextPageVar(pvName)** function

   For example:

   ```
   CP_TranslateIText("User ID:") + " " + CP_TranslateITextPageVar("PV_USER_ID").
   ```



You can also use the **CP_TranslateITextPageVar(pvName, format)** function in case you need to format data. For example:

```
CP_TranslateITextPageVar("PV_PSWD_CH_DT", "CP_DateCntl")
```

Be sure to add this variable as a break group and set the page break properties so that it will break on this break group change. Otherwise, you may have a page where the variable has changed and it will take the value of the last one.

Lines

You must use CP styles in order to provide separation lines for the cover page, report detail, and page header/footer sections.

Select the row(s) of a grid or a table for which you want to draw a line under. Right-click and click **Styles » Apply Styles**. Then choose a line as one of the following styles:

- **CP_2ContentLineStyle (double content line)** — Use on the cover page.
- **CP_ContentLineStyle (single content line and bottom line)** — Use on the cover page, table detail section, and at the bottom of column headers.
- **CP_HeaderLineStyle (header line for Page and Group Headers)** — Use on page and group headers.

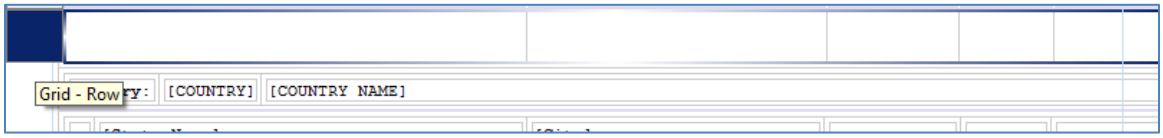
The line style provides required line width, style, and color.

Column Headers

The column headers must be placed in the table header or group header sections. Use the CP_ColHdrStyle style for cells where you place column header labels. This style provides the required background color and minimum white space in-between the headers.

To create column headers:

1. Use table or grid row for placing column headers. Initially, the row should have empty cells.



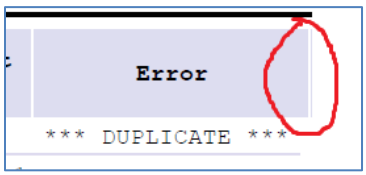
2. Hold SHIFT, and select all the cells in the row
3. Apply **CP_ColHdrStyle** to the cells.

Do not apply the style to the row or to the column header labels. For the right-most column header cell, you may need to apply CP_ColHdrNoRightBorderStyle.

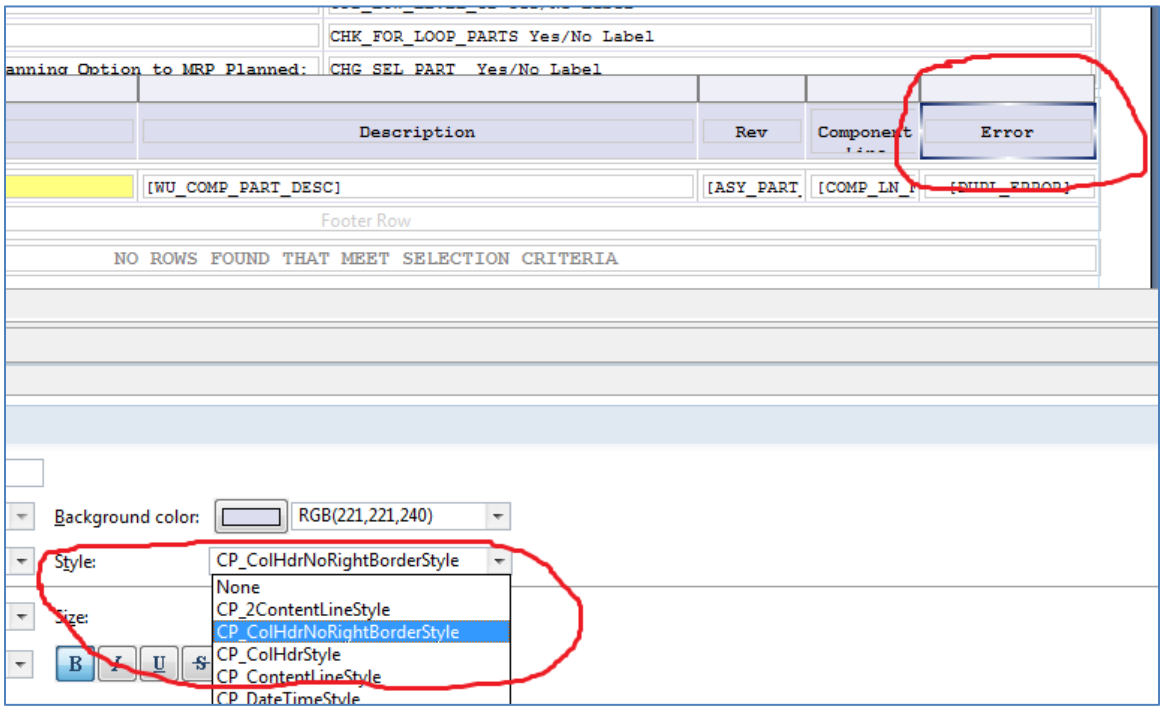
4. Drag the CP_LabelCntl controls (labels) from the library to the column headers row/cells, and adjust text alignment if necessary.
5. Adjust the row height setting in **Row » Property Editor - Row » General » Height** if necessary.

For the Right-most Column Header Cell Only

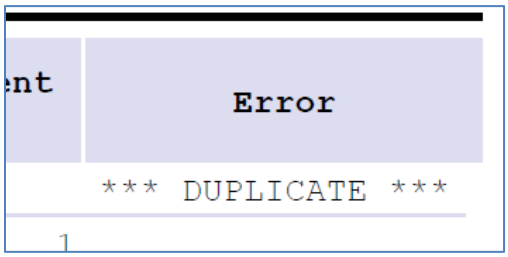
To avoid displaying a tiny vertical line just to the right of the last column header, you may need to apply **CP_ColHdrNoRightBorderStyle** .



Select the right-most column header cell, and apply **CP_ColHdrNoRightBorderStyle** to remove the right vertical border line.



The report out will look like this then:



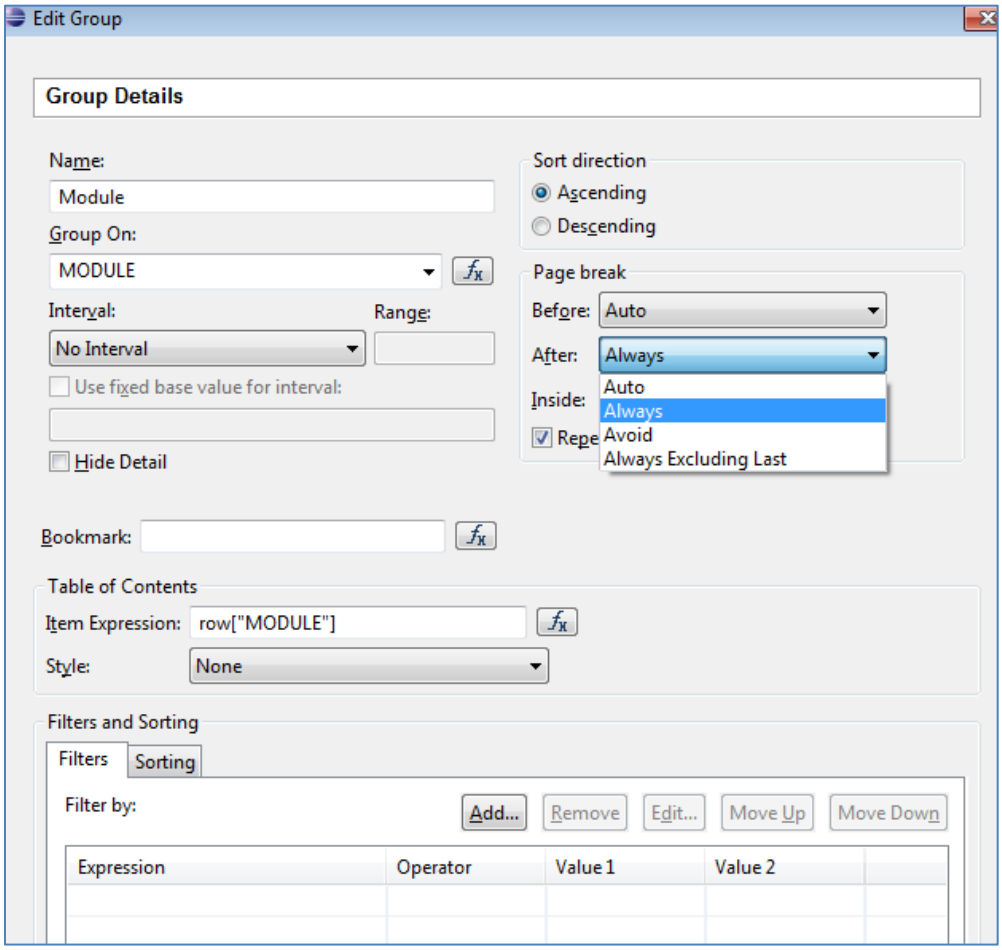
Page Break (Conditional Page Break)

Page Break for a Group

In order to add a page break after the group, click **Group properties**, and then click **Page Break » After » Always Excluding Last (Always)**.

- For the top most Group, select **Always Excluding Last**.

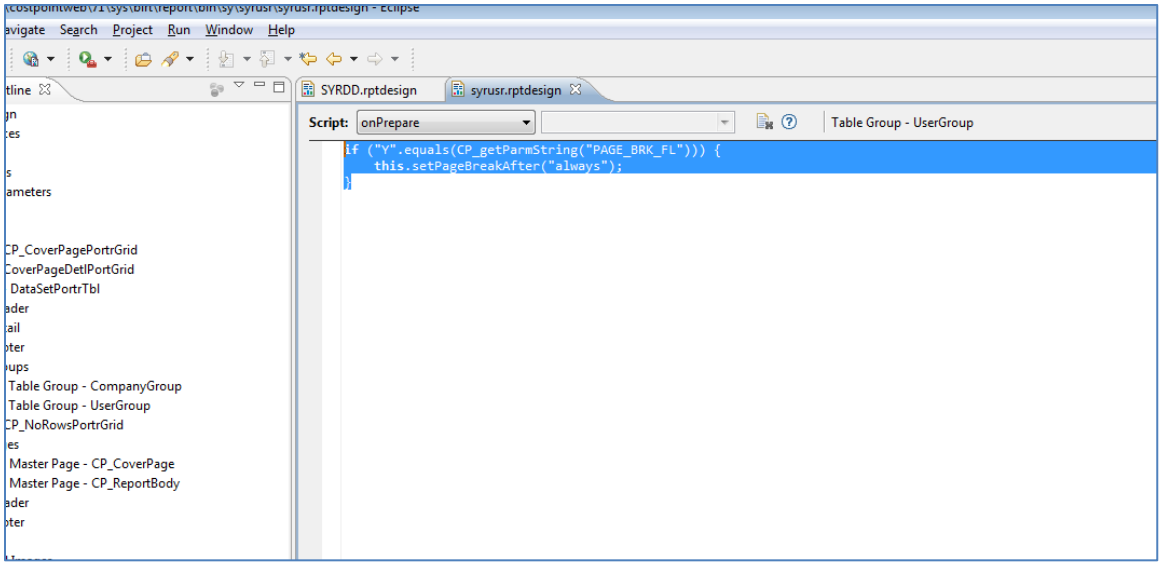
- For the Nested (Inner) Groups, select **Always**.



Conditional Page Break

A conditional page break can be done through scripting. For example, you have to provide a page break after a group driven by a report parameter:

1. Go to **Outline » Body » Table » Group object**
2. Click on script tab, and override the **onPrepare** method to set a page break driven by a report parameter.
   - For the top most group, use the **always-excluding-last** value.
   - For the nested (inner) groups, use the **always** value.



For the Top most Group (onPrepare method):

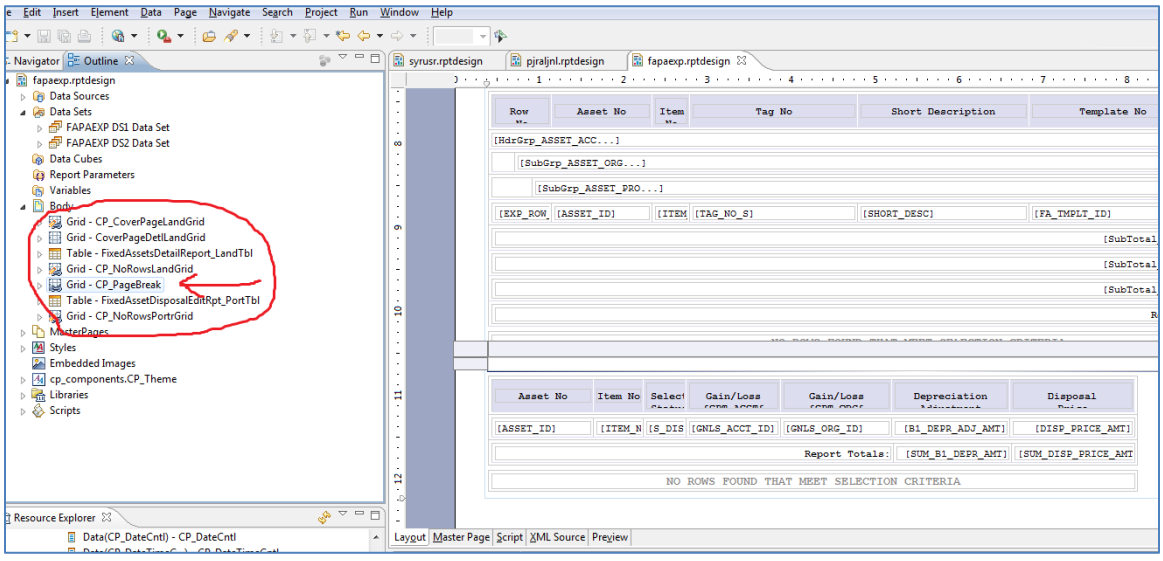
```
if ("Y".equals(CP_getParmString("PAGE_BRK_FL"))) {
        this.setPageBreakAfter("always-excluding-last");
}
```

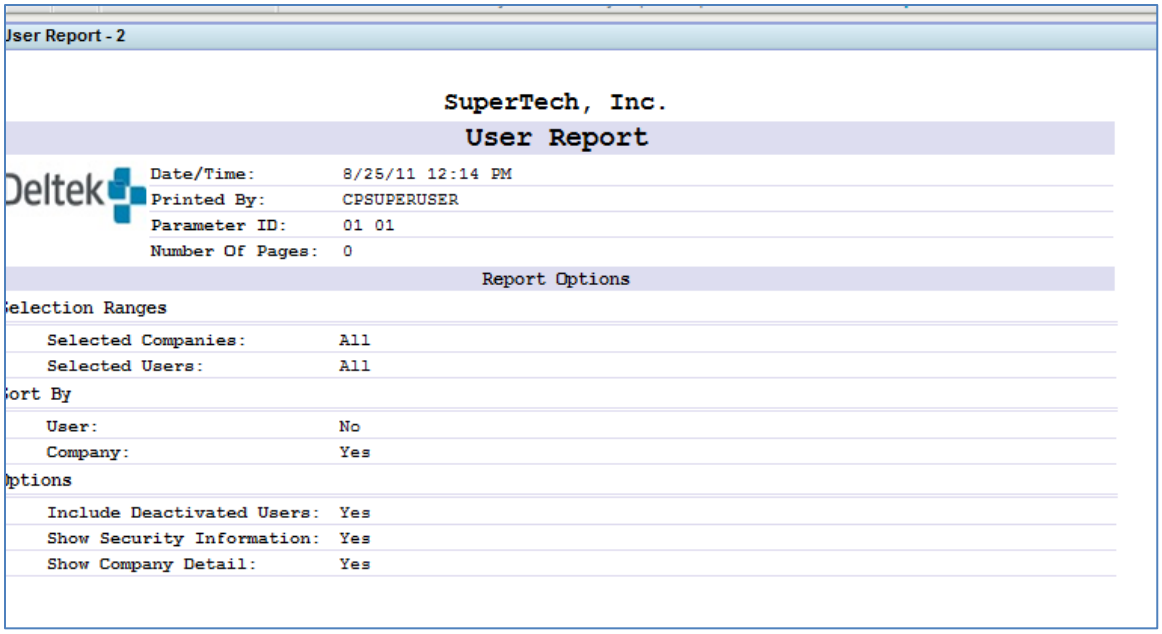
For the Inner Groups (onPrepare method):

```
if ("Y".equals(CP_getParmString("PAGE_BRK_FL"))) {
        this.setPageBreakAfter("always");
}
```

Page Break between Report Sections (Tables)

In you need to have a page break between report sections (table objects), do not set the page break at the table level. Instead, drag the CP_PageBreak component from cp_components.rptlibrary and insert it right after CP_NoRows Grid for each table in report design.



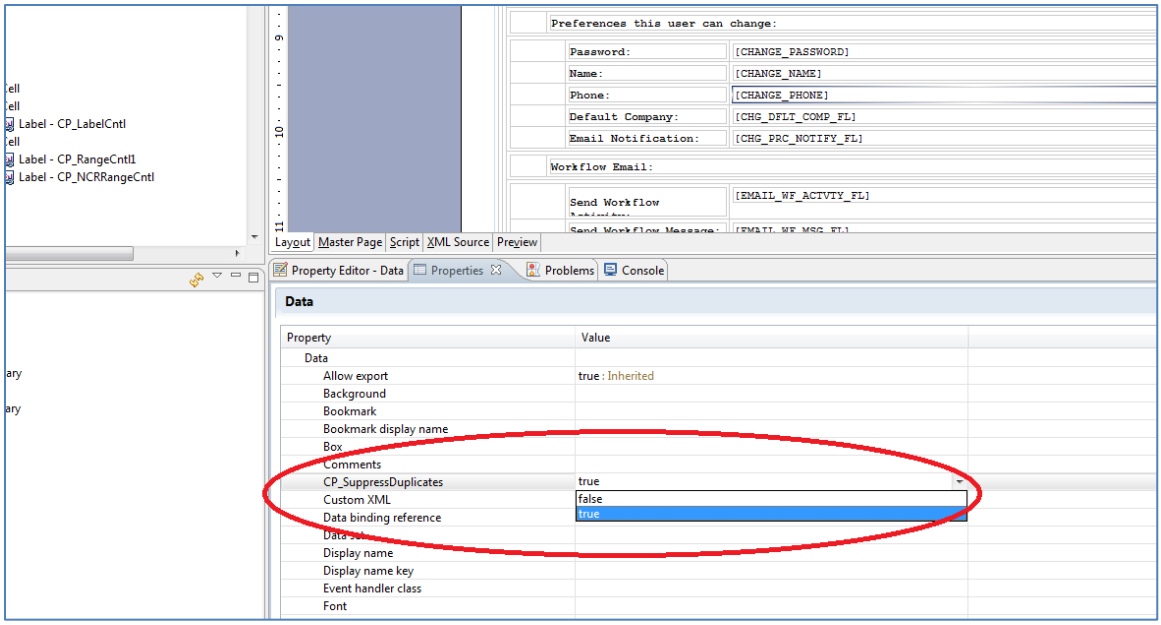


Suppress Repeated Values

Suppress repeated values is supported for the following controls:

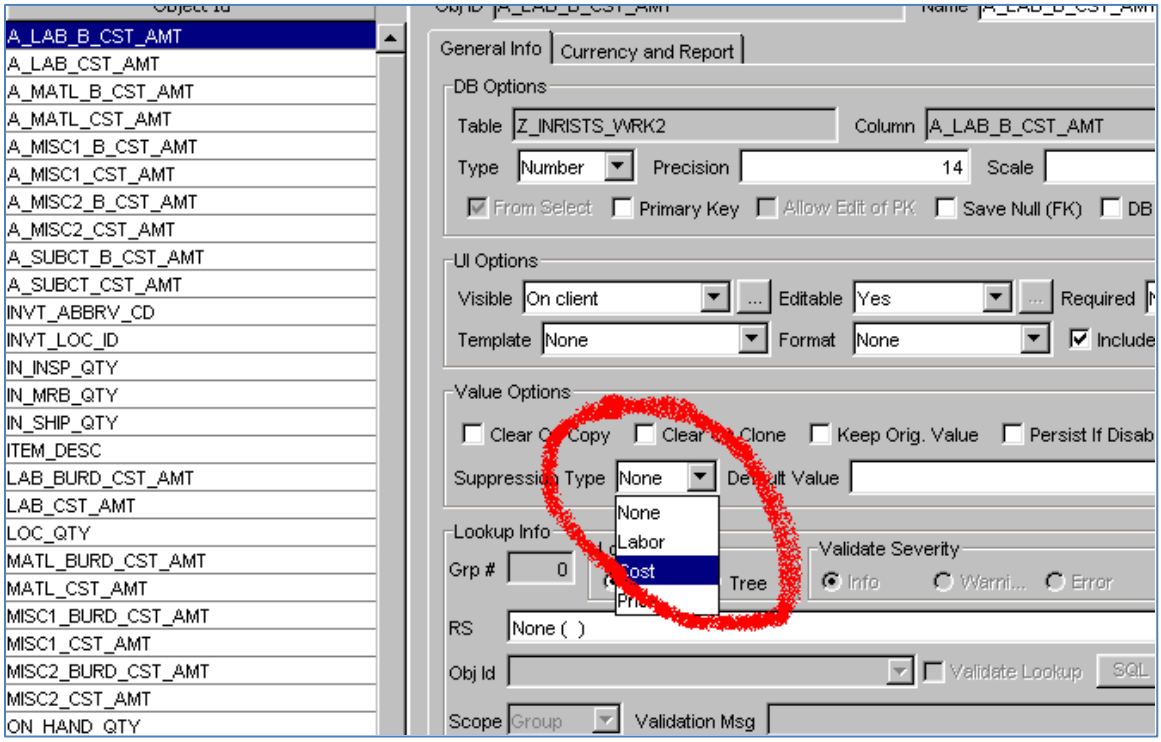
- CP_TextCntl
- CP_LabelCntl
- CP_LabelYesNoCntl
- CP_DateTimeCntl
- CP_DateCntl
- CP_TimeCntl

Use this feature only for controls that are being placed into the table/table group detail section. In order to enable **Suppress repeated values**, go to **Control » Properties**, and set **CP_SuppressDuplicates** to **True**.



Data Suppression (Labor, Cost, Price)

The correct approach to implement data suppression is to identify the suppressed column in the Extensibility Designer for report data source RS. The framework relies on correct RS metadata when implementing security filtering for archived reports.



In addition, you can hide these columns and column headers in *.rptdesign based on the suppression constants passed to a report at runtime by changing control Visibility formula:

```
// Hide if user suppress cost flag is Y
```

```
"Y".equals(CP_getConstantString( "CP_WUSERCOMPANY_SUPPRESSCSTFL" ) )
// Hide if user suppress labor flag is Y
"Y".equals(CP_getConstantString( "CP_WUSERCOMPANY_ SUPPRESSLABFL " ) )
// Hide if user suppress price flag is Y
"Y".equals(CP_getConstantString( "CP_WUSERCOMPANY_ SUPPRESSPRCFL" ) )
Dynamic Page Layout
```

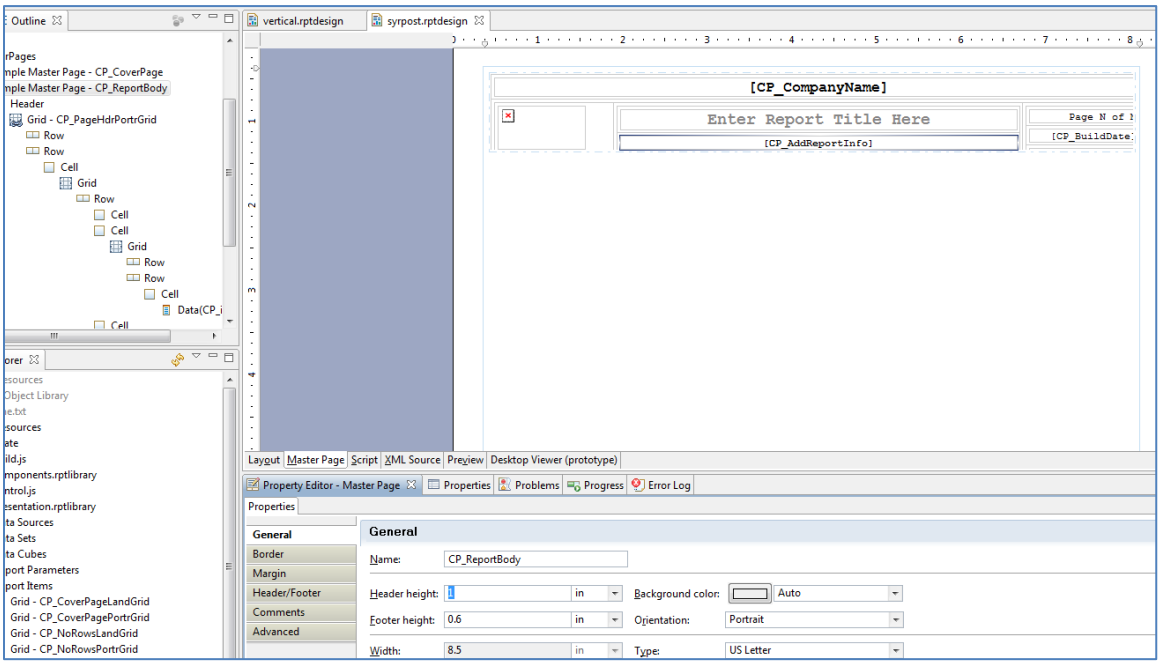
You can create page objects of different size, margins, and page layout (lanscape or portrait) and then assign this page to a table object in **Outline » Body » Table » Page Break » Master Page**.

For multiple report sections and different page settings, you may need to create additional page objects and place standard Costpoint page headers/footers onto new pages.

To create a new page object:

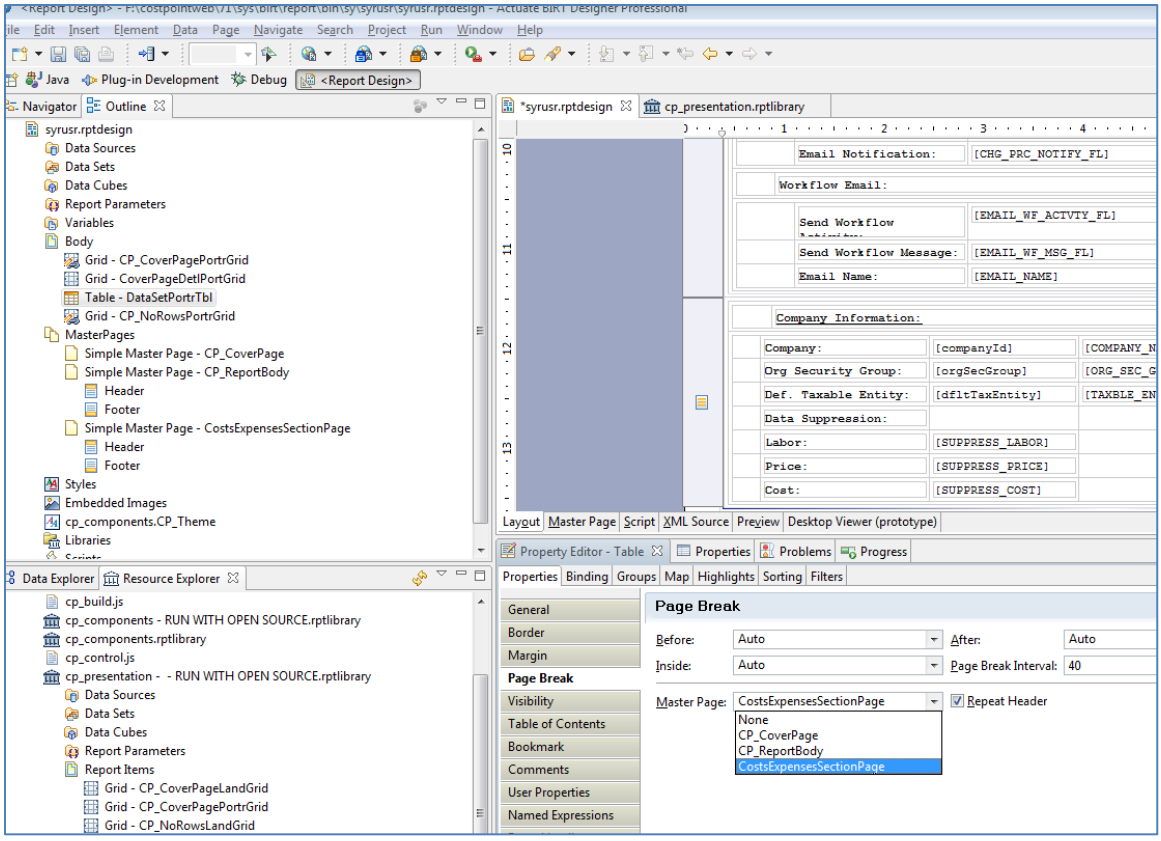
1. Right-click **Master Pages** in the Outline view, and click **Insert Element** on the shortcut menu.
2. Rename the page object, and change the page settings if needed.

   The default height of the standard Costpoint Header is 1 inch. The default height of the standard Costpoint Footer is 0.6 inch.



3. Insert standard Costpoint page header/footer objects from cp_presenation.rptlibrary into the new page by dragging page header/footer objects from the library directly into **Outline » CP_ReportBody page » Header/Footer**.

The new page object should be correctly mapped to your table object in the report design. Switch to Layout view, and in **Outline » Body,** locate the table object that should display data for the different report section. Go to **Table » Property Editor » Page Break**, and change the master page to the newly created page. The result is that all data that goes into the above table will be displayed on a page with possibly different layout margins, headers, and footers.



Page Numbers (Restart Page #)

The page number control should be displayed at the top right corner of the page. The page number is a part of the standard Page Header that is provide by the libraries.

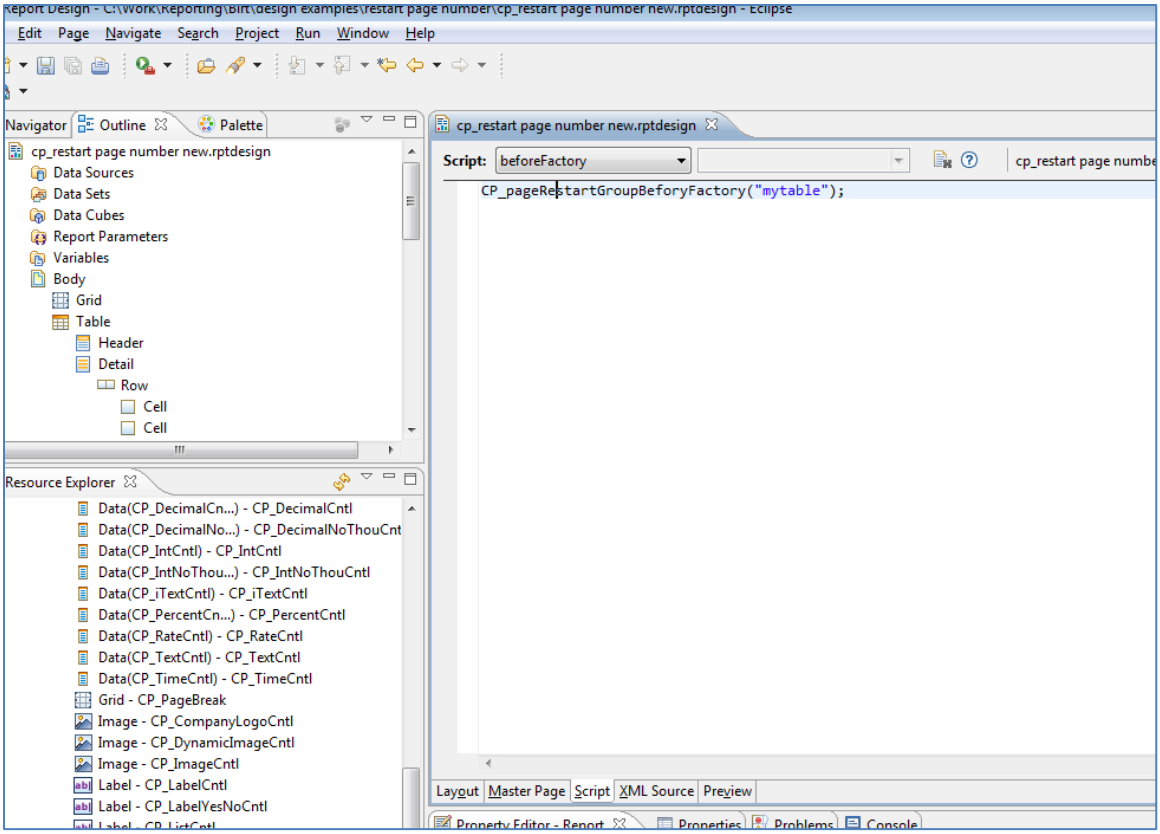
- Use CP_PageNofMCntl control to display standard Page N of M text.
- Use CP_TotalPageCntl control to display report total page count.

Restart Page Number

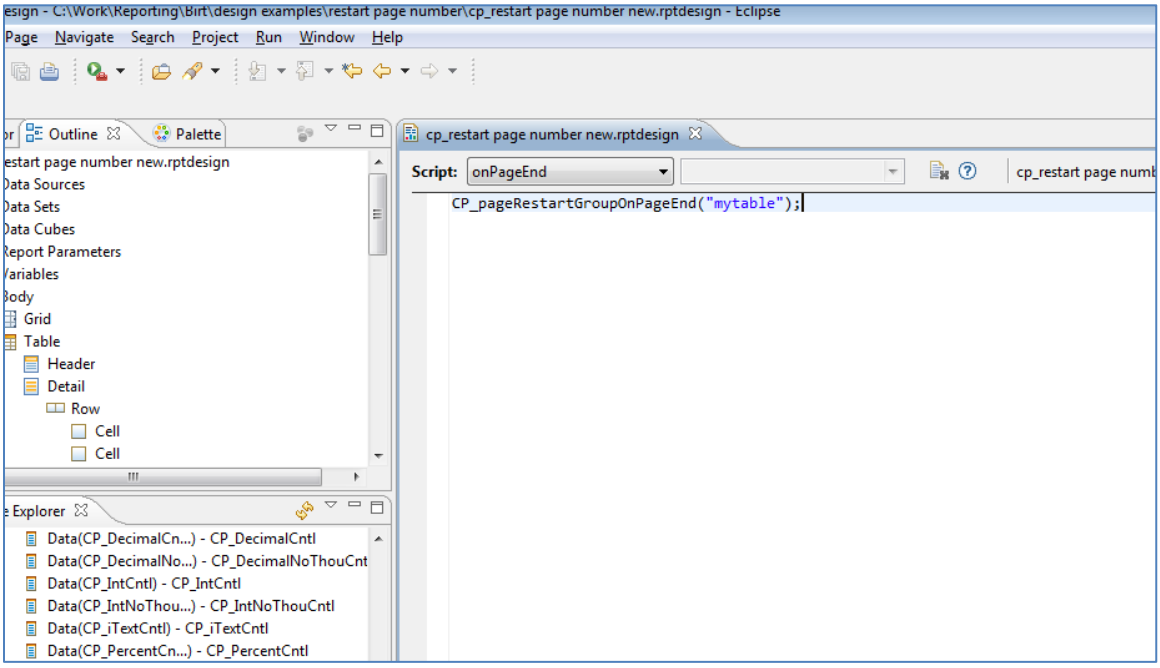
There are cases when you need to restart page numbers within a group (for example, display new pagination for each voucher or invoice).

Use the following technique/rules to achieve restartable page numbers per break group:

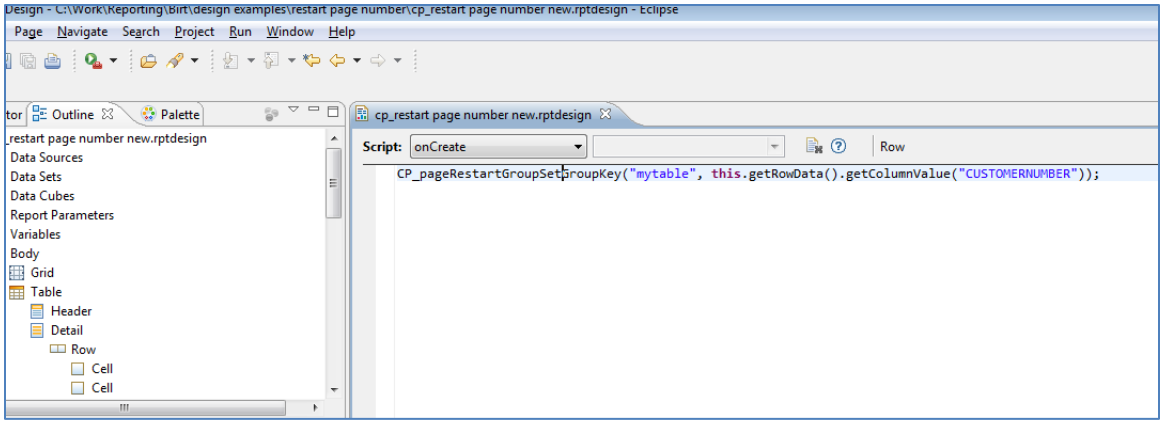
- The break group must have a page break on changing of a group key.
- Call the CP_pageRestartGroupBeforyFactory(tableName) function in report » beforeFactory script.



- Call the CP_pageRestartGroupOnPageEnd(tableName) function in report » onPageEnd script.

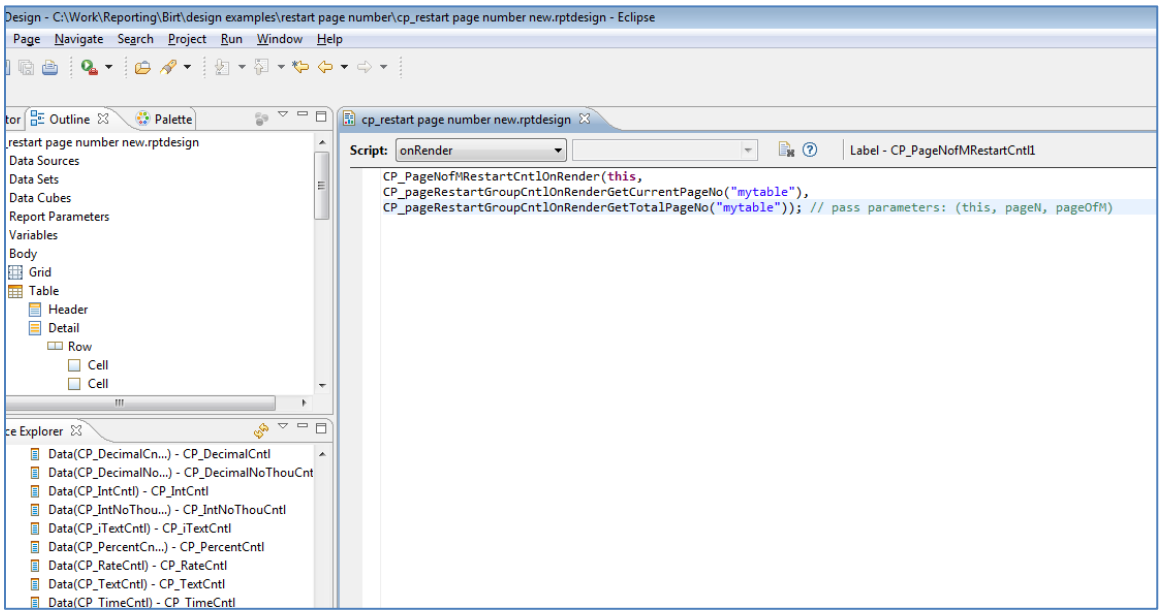


- Call CP_pageRestartGroupSetGroupKey(tableName, groupKey) function in Table » Detail » Row » onCreate function to set the current group key.



- Drag the CP_PageNofMRestartCntl control from the libraries. This control can be used on both Page Header/Footer or on Report/Group Header/Footer. In CP_PageNofMRestartCntl » onRender(), override CP_PageNofMRestartCntlOnRender(this, -1,-1); // pass parameters: (this, pageN, pageOfM). For example:

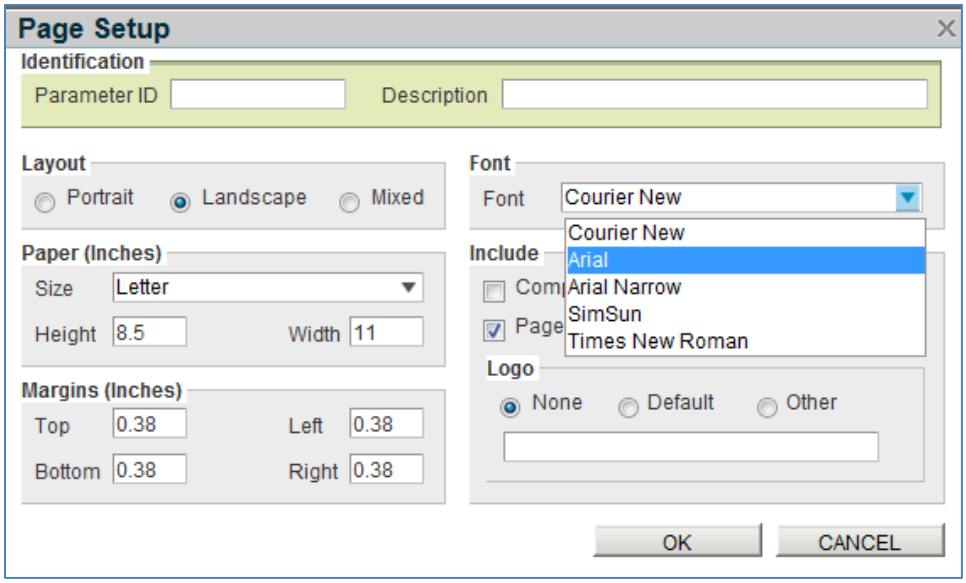
```
CP_PageNofMRestartCntlOnRender(this,
CP_pageRestartGroupCntlOnRenderGetCurrentPageNo("mytable"),
CP_pageRestartGroupCntlOnRenderGetTotalPageNo("mytable")
);
```



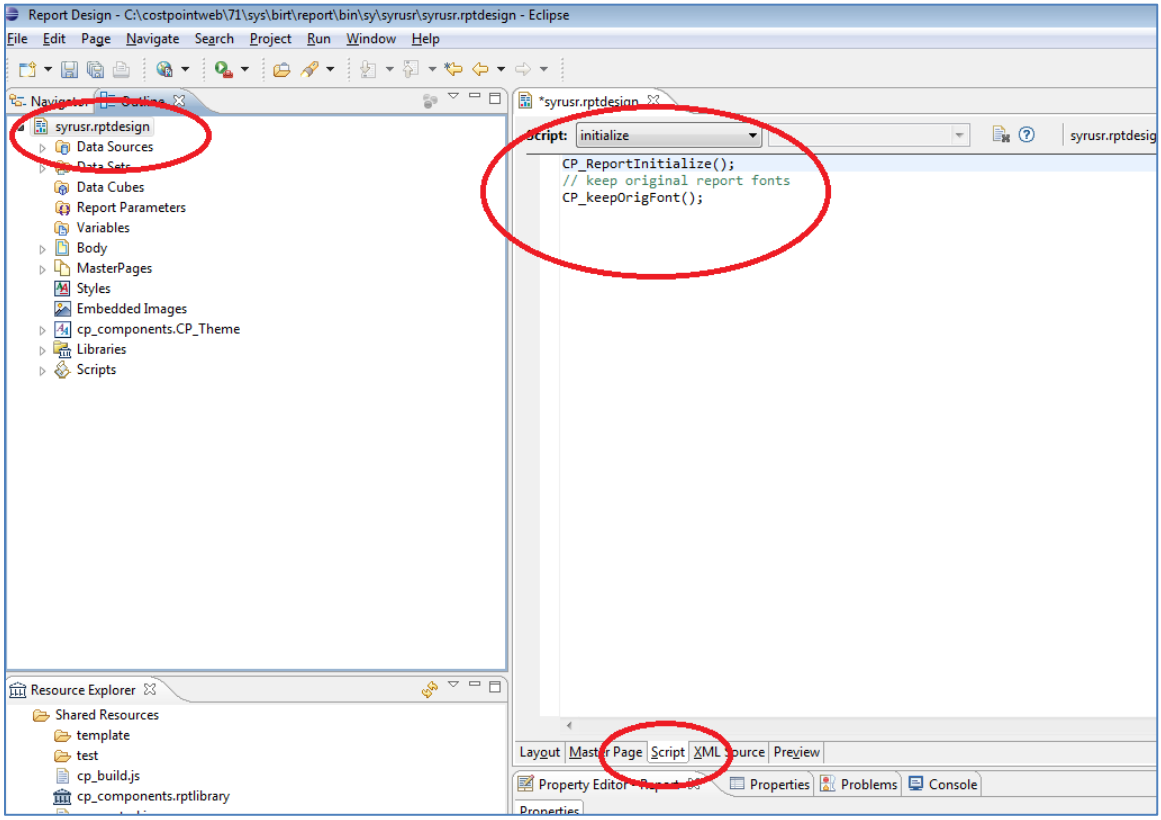
Fonts

The default font is Courier New and the default font size is 8. When you create new controls derived from the library controls, do not change the default font (and/or size) unless the Design suggests changes. The smallest allowable font size for Costpoint reports is 7.

At runtime, the system may automatically substitute design time font (Courier New) by a font selected on the Page Setup dialog box.



For some reports, due to application business logic and report presentation rules, font substitution should not be allowed. In order to ensure that font substitution does not occur, override the **initialize()** function for the top most report component and call **CP_keepOrigFont()** to prevent the system from substituting fonts at runtime.



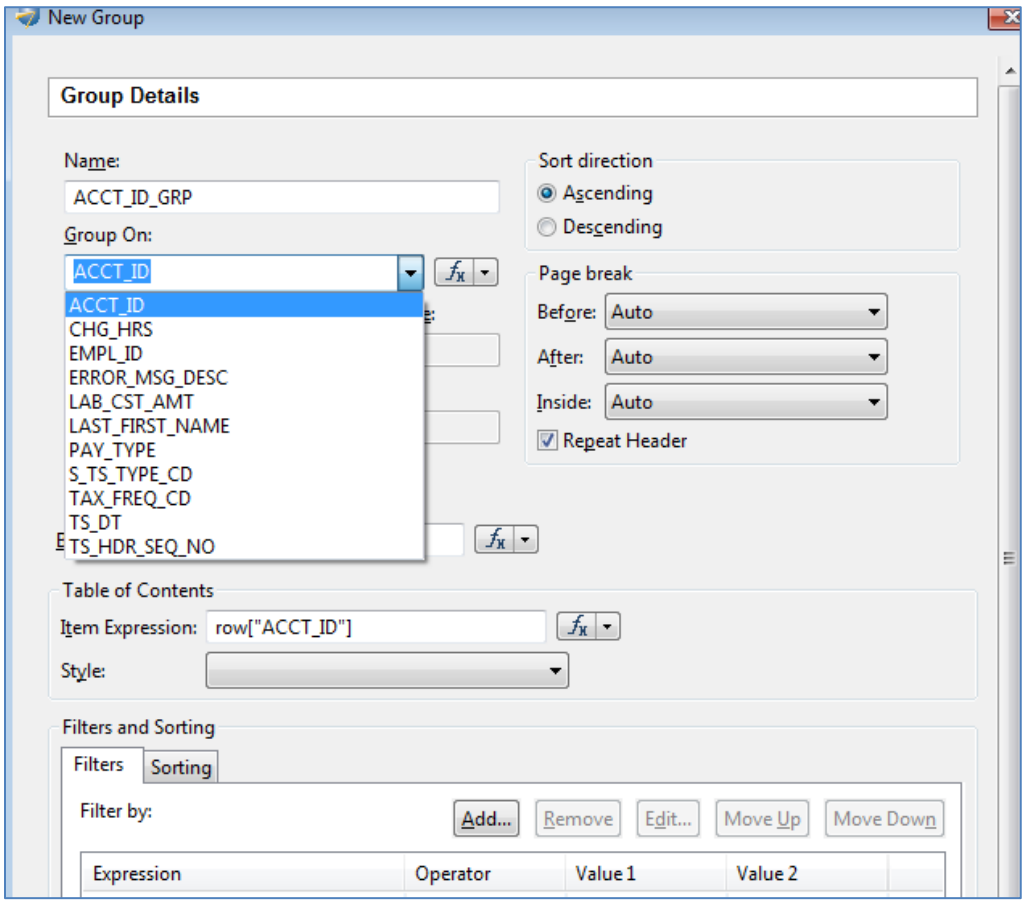
Group Sections and Aggregates

One of the key features of any report is the ability to display summary, or aggregate, information. For example, a sales report can show the overall sales total; sales subtotals by product type, region, or sales representatives; average sales figures; or the highest and lowest sales figures.

The aggregates are usually created on a group.

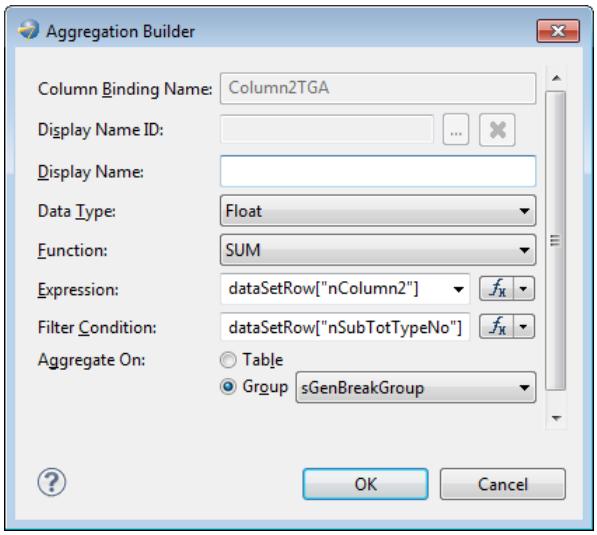
To create a group:

1. Right-click the table, and click **Insert Group** on the shortcut menu.
2. In a the New Group dialog box, enter a group name, and select a key from the **Group On** drop-down list as a simple column, or use the Expression Builder.
3. If it is the top most group, enter a Table of Contents (TOC) expression (generally, the same as group key). If it is the inner group clear, the TOC value as the TOC should be provided at the top most group level.



To create an aggregate:

1. Click the Table Binding tab, and click **Add Aggregate**.
2. Enter a name in the **Column Binding Name** field, and select an aggregate function from the **Function** drop-down list.



Examples of aggregate expressions:

- dataSetRow["nColumn2"]
- dataSetRow["nColumn2"] / dataSetRow["nColumn3"]
- dataSetRow["nColumn2"] * (CP_getParmInt("PN_SELECT_NO") == 0 ? 1 : 0) - dataSetRow["nColumn3"]

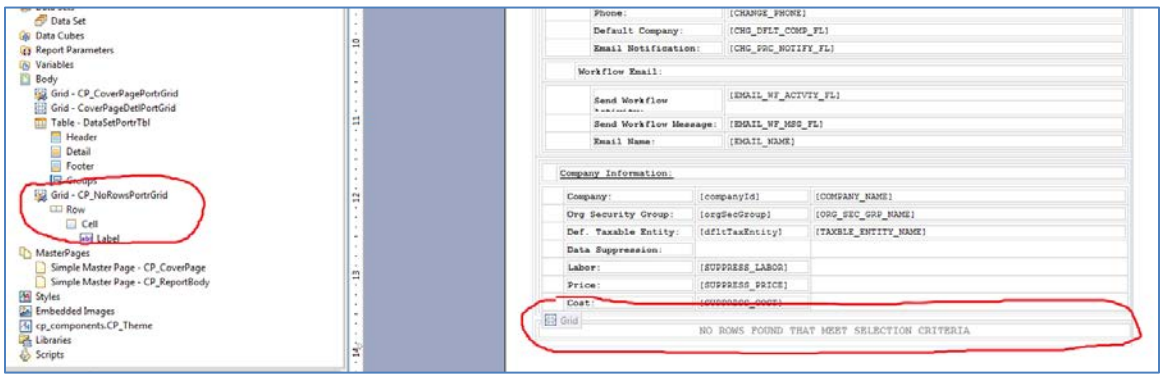
Examples of filter conditions:

- dataSetRow["nSubTotTypeNo"] == 10
- dataSetRow["nSubTotTypeNo"] > 1
- dataSetRow["nSubTotTypeNo"] >=2 && dataSetRow["nSubTotTypeNo"] <=5

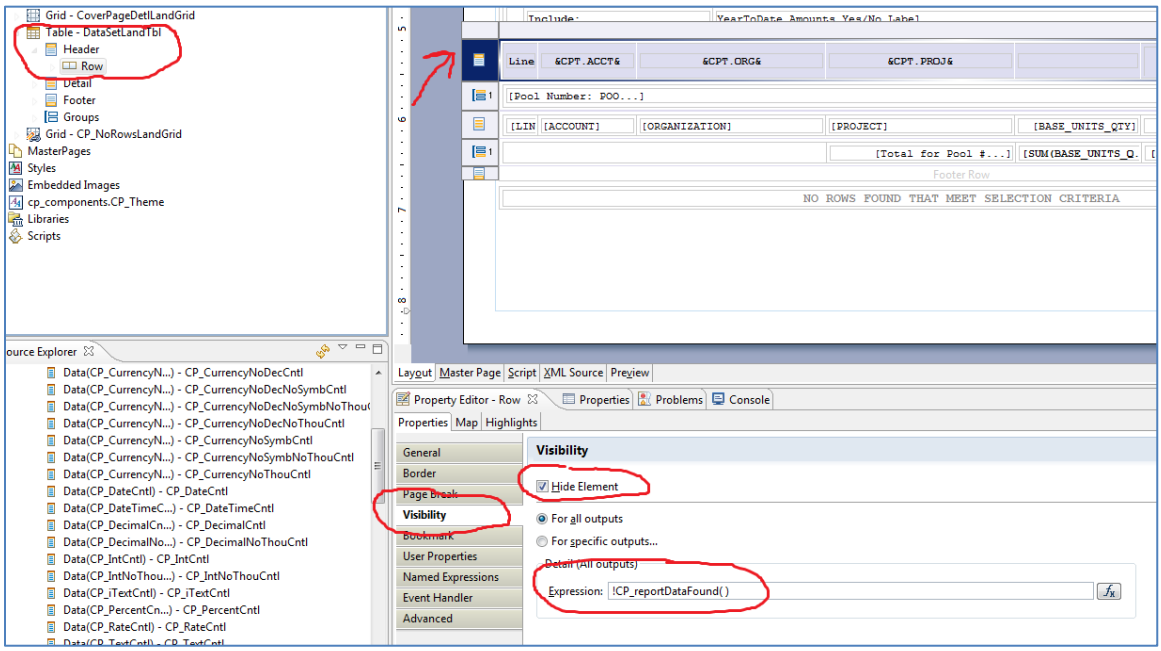
When you divide aggregates (or any numbers), Deltek recommends that you use **CP_safeDivide(n1, n2, nIfn2IsZero)**.

Handling "No Data Found"

When a report has no data to process, the generic message "NO ROWS FOUND THAT MEET SELECTION CRITERIA" should be printed on a report page. This is handled by the system libraries through cp_presentation.rptlibrary » CP_NoRowsPortrGrid and CP_NoRowsLandGrid. The appropriate (landscape or portrait) CP_NoRows Grid should be used after each report section (table object) in the report design.



When the report layout requires you to have a table header (for example, table wide column headers) and/or table footer (for example, report totals) row(s), these rows should have a **Table » Header (Footer) » Row » Visibility** formula set to **!CP_reportDataFound( )**. Failure to add the formula will result in Table Header/Footer Row(s) showing when report has no data.



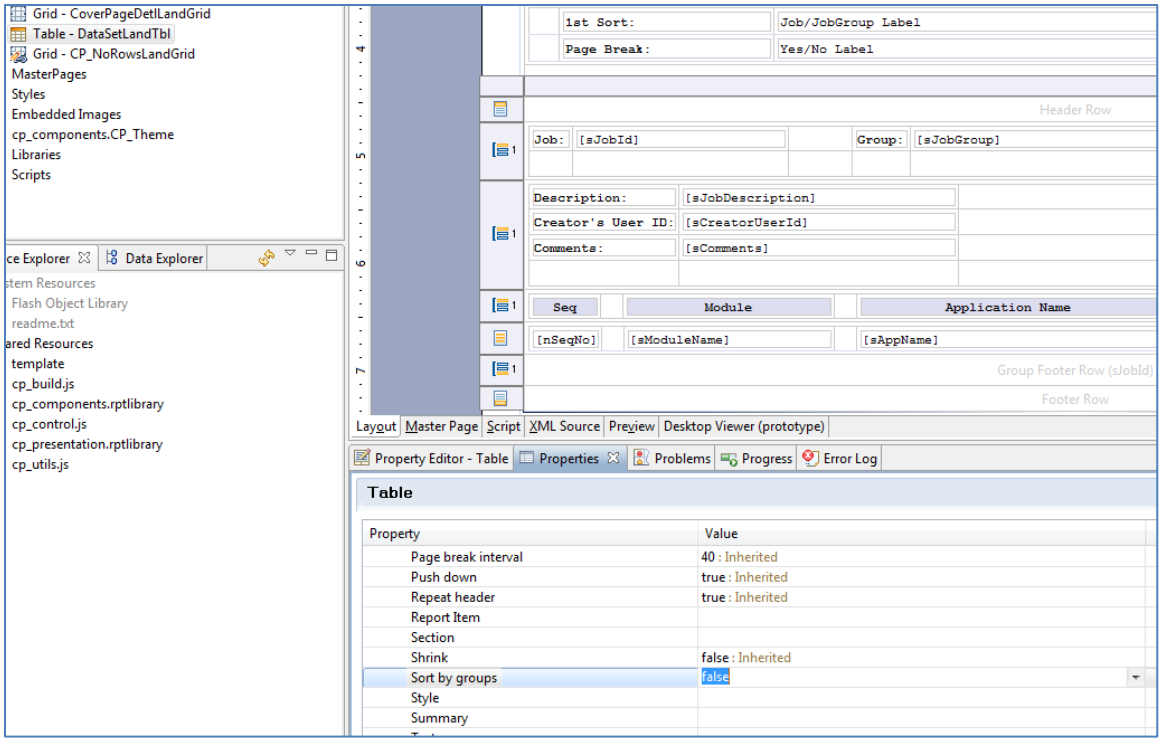
You add this formula only for Table Header and/or Table Footer rows. If you do not use the Table Header or Footer rows and, for instance, you have Column Headers placed in a Group Header, then you don't have to add the formula.

Multiple Report Sections in One Template (Posting Applications)

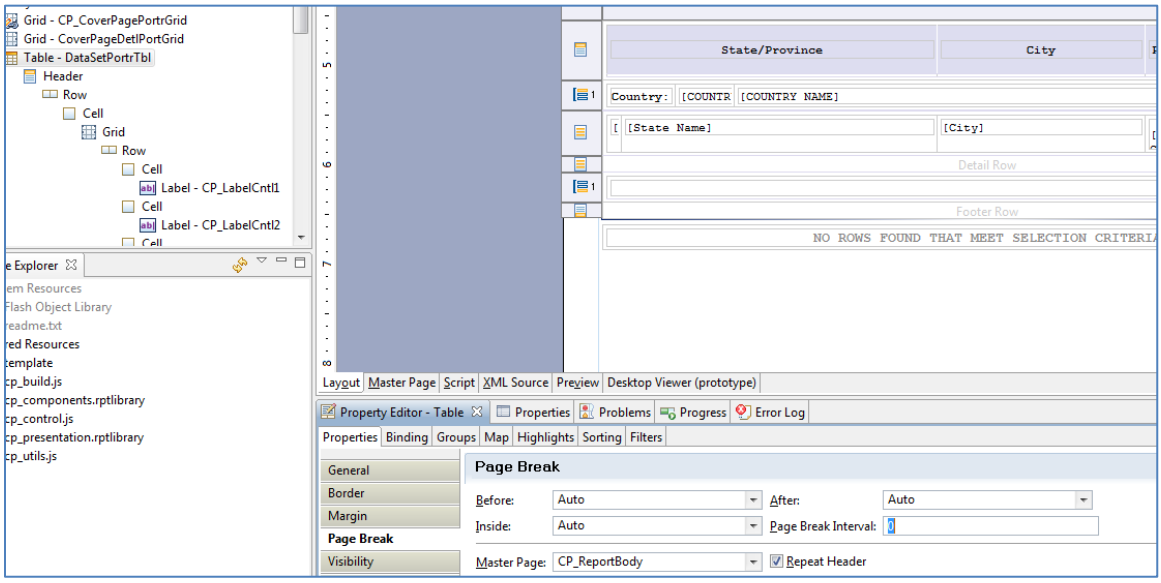
You can print several tables in a sequence. The processing for a subsequent table starts after the previous one finishes, and so on. You can build a sequential report by placing multiple tables one after another.

Table Object

When you insert a new table object in the report, always set the **Table » Properties (Advanced) » Sort By Groups** to **false**. This will disable re-sorting of the data (BIRT default behavior). The Costpoint data is already pre-sorted through SQL.

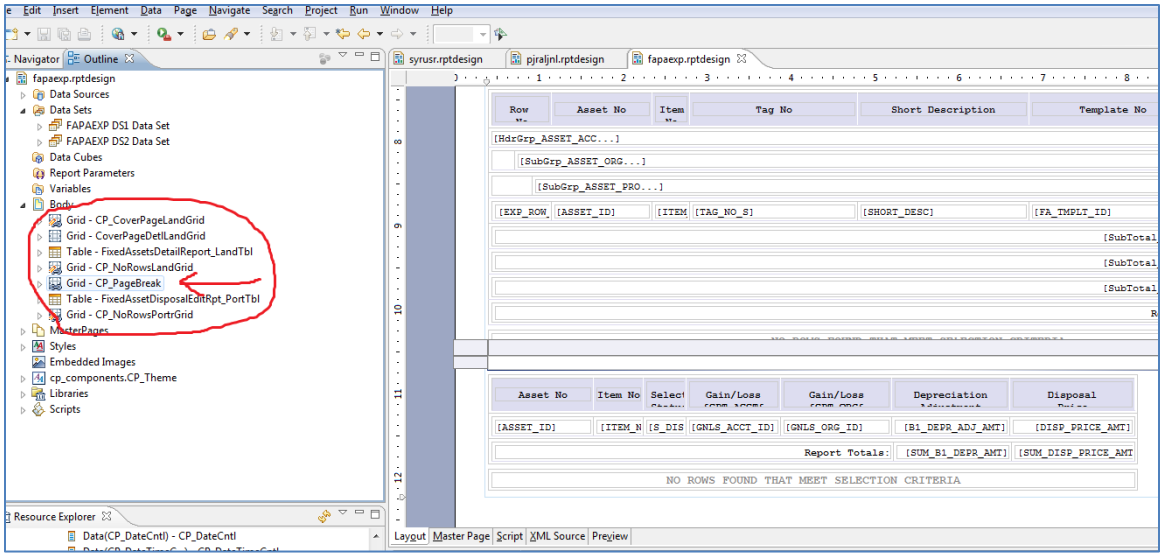


In **Table » Property Editor - Table » Properties » Page Break**, set **Page Break Interval** to **0**.

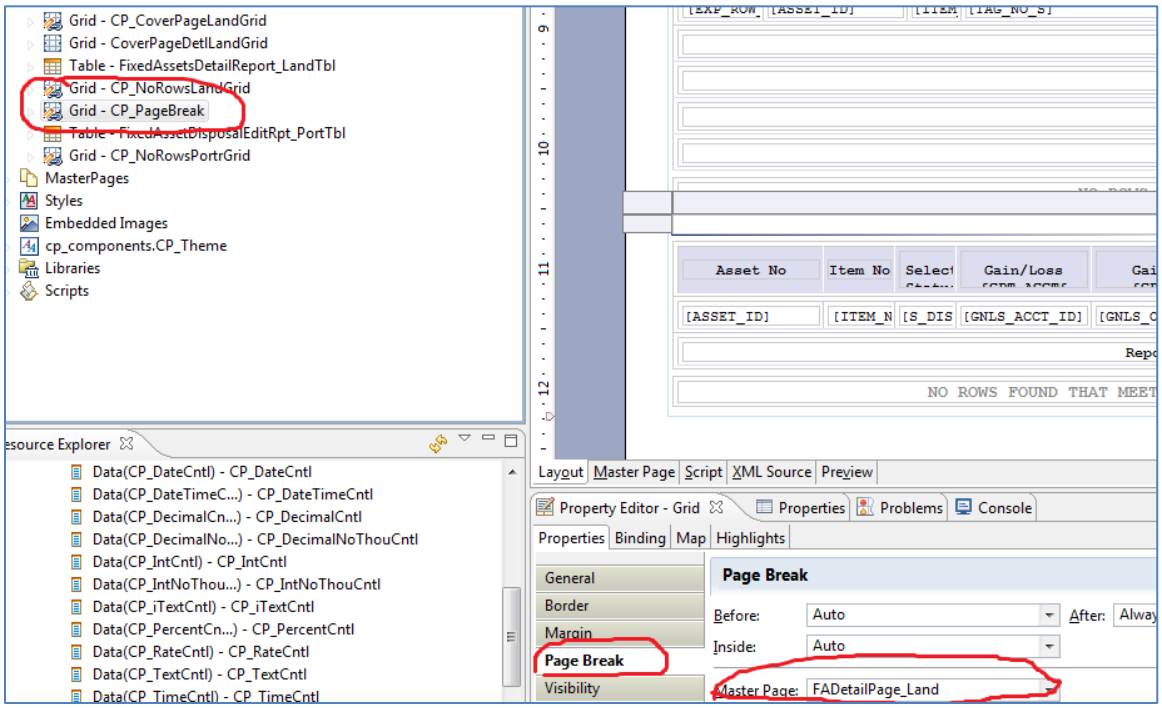


The appropriate (landscape or portrait) CP_NoRows Grid should be used after each report section (table object) in the report design. For example, if you have two report sections in the report, then you must have two table objects and two CP_NoRows Grids following each table object.

When you need to have a page break in between report sections (table objects), do not set the page break at the table level. Instead drag CP_PageBreak component from cp_components.rptlibrary and insert it right after CP_NoRows Grid for each table in report design.



Go to **CP_PageBreak Grid » Properties » Page Break**, and assign a master page property to the same page that is used by the table object just before the CP_PageBreak component.



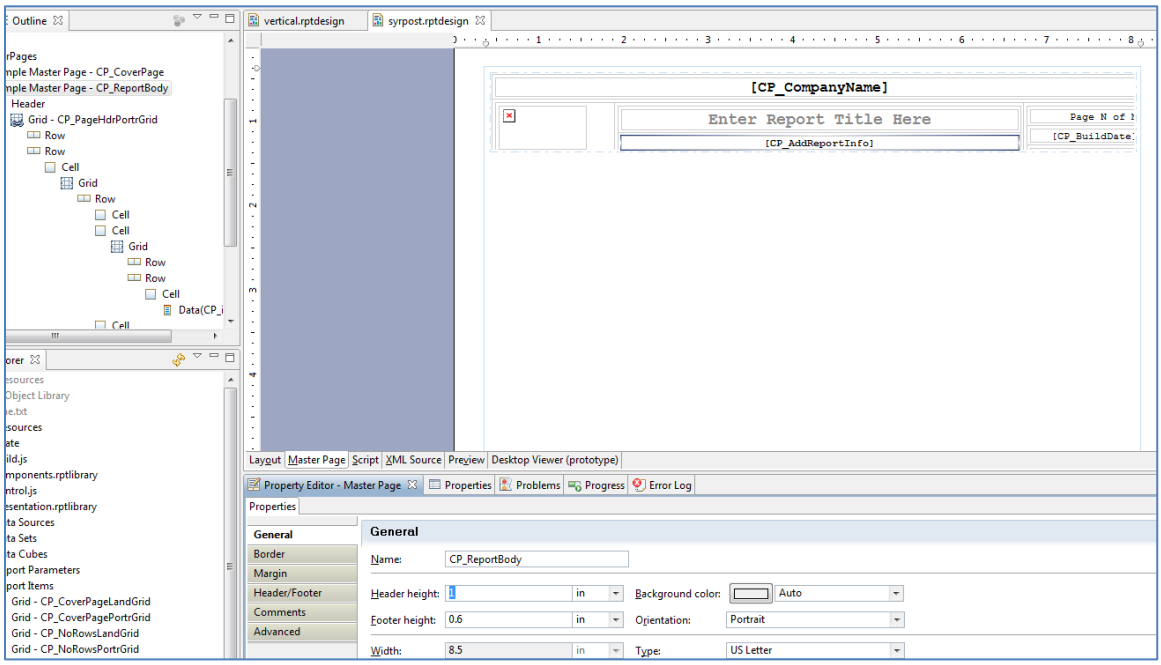
Page Layout

You can create a page object of different size, margins, and page layout (landscape or portrait) and then assign this page to a table object in table properties/page break/Master Page.

In case of multiple report sections and different page settings, you may need to create additional page objects and place standard Costpoint page headers/footers onto new pages.

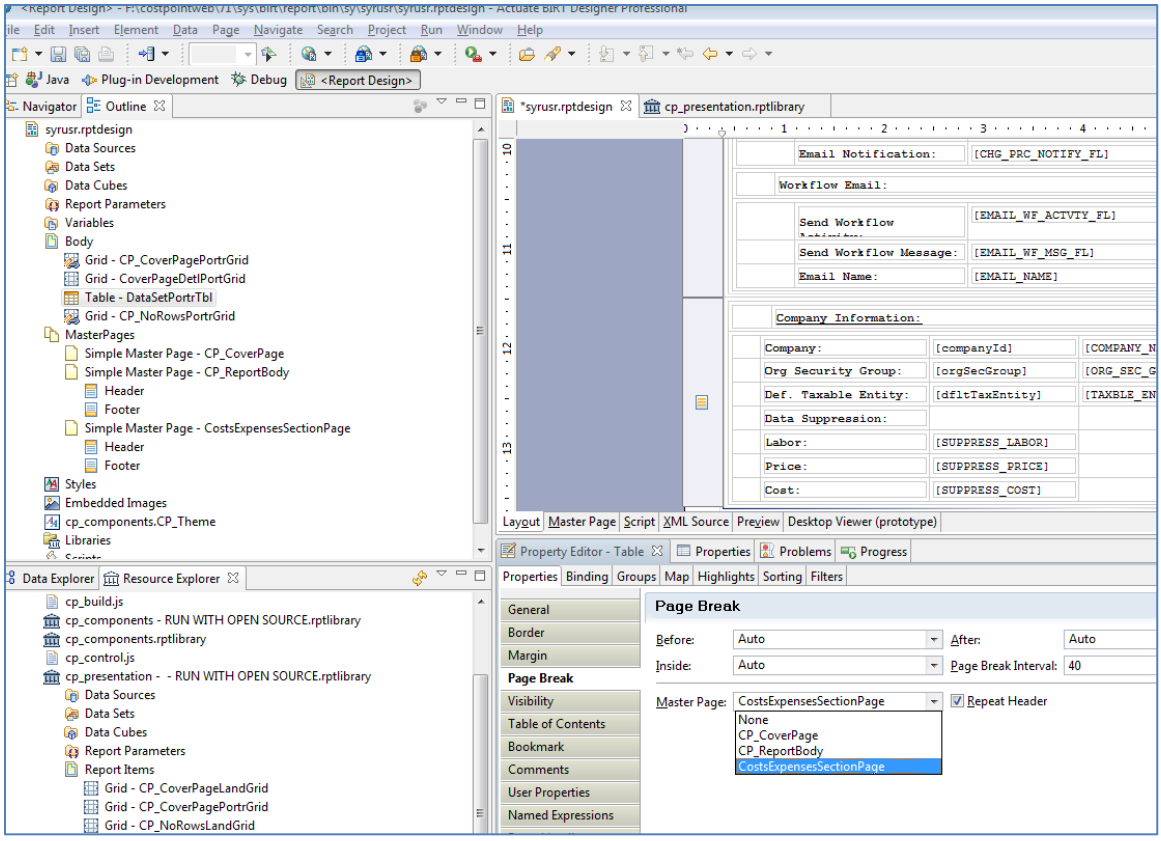
To create a new object, right-click **Master Pages** in the Outline view, and click **Insert Element** on the shortcut menu. Rename the object, and change the page settings if needed.

The default height of the standard Costpoint Header is 1 inch. The default height of the standard Costpoint footer is 0.6 inch.



Insert standard Costpoint page header/footer objects from cp_presenation.rptlibrary into the new page by dragging page header/footer objects from the library directly into **Outline » CP_ReportBody page » Header/Footer**.

The new page object should be correctly mapped to your table object in the report design. Switch to Layout view, and in **Outline » Body**, locate the table object that should display data for different report sections. Go to **Table » Property Editor » Page Break**, and change the master page to the newly created page. The result is that all data that goes into the above table will be displayed on a page with possibly different layout margins, headers and footers.



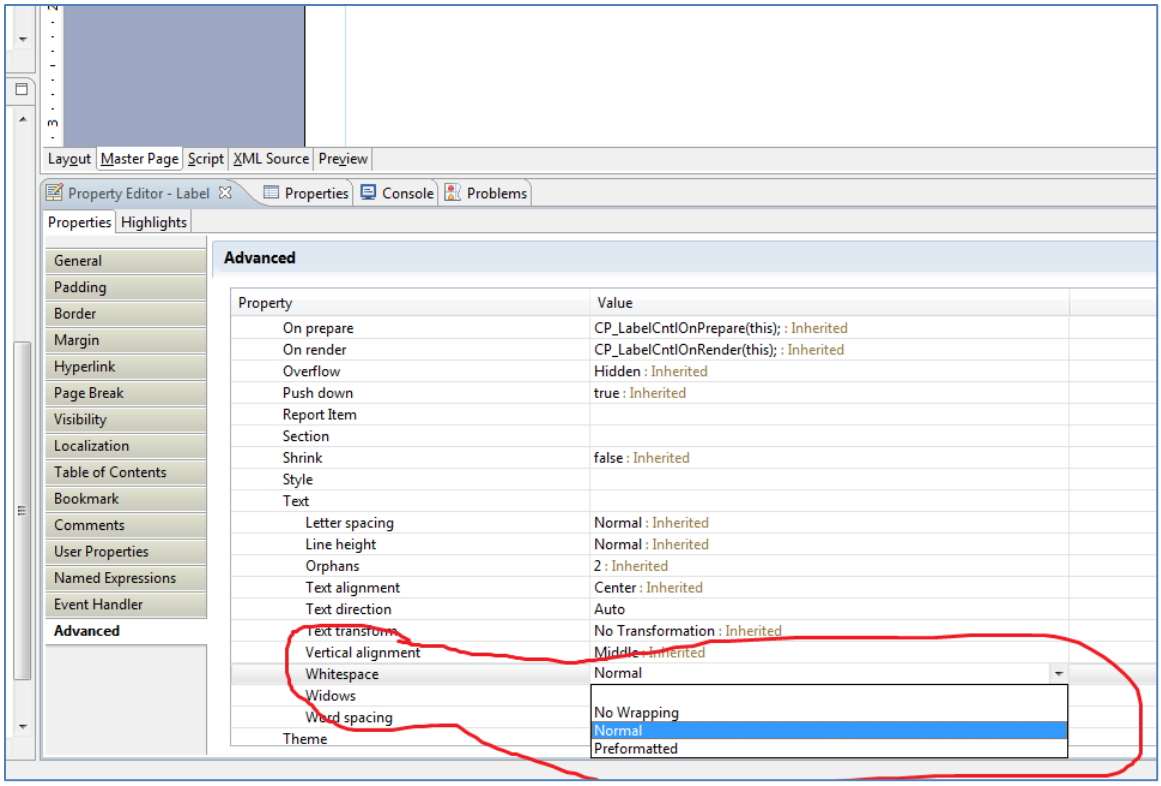
Presenting Text

A report typically presents most of its information in a textual format. Textual information can be any of the following:

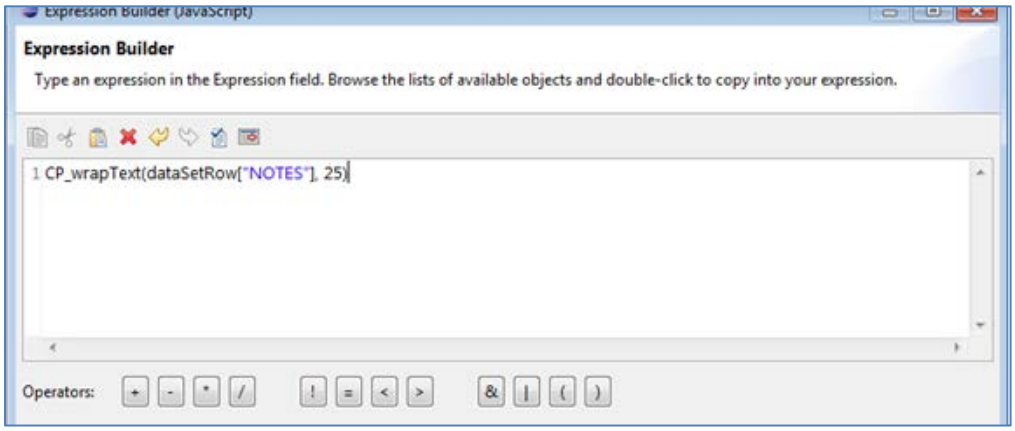
- Static text — Labels, column headers, report title. This data is subject to localization, language translation, and CP terms translation. Use only CP_LabelCntl and CP_LabelYesNoCntl.
- Mixture of static text and data values supported by the system — Range text, NCR range text, text list, page N of M, and total Page Count…. Use specially provided controls such as CP_RangeCntl, CP_NCRRangeCntl, CP_ListCntl, CP_PageNofMCntl, CP_PageNofMRestartCntl, and CP_TotalPage.
- Dynamic text data to present Data Set fields: dataSetRow["COMPANY_ID"], dataSetRow["USER_ID"]…. Use CP_TextCntl control.
- Mixture of dynamic data that is subject to localization, language translation and CP terms translation — [static text] + [text] + [date/time] + [number] + …. For any of these, use CP_iTextCntl control and CP_...IText functions to present data within the control (for example, CP_TranslateIText("Project End Date:") + CP_TranslateITextDateTime(dataSetRow["PROJ_END_DT"], "CP_DateCnlt").

Text wrapping:

By default, all text data is wrapped onto multiple lines in case it does not fit within the control or grid/table boundaries. The text line breaks by the whitespace and is controlled by the **Control » Properties » Advanced » Text » Whitespace** setting. The default value is **Normal**. You can turn off text wrapping by changing the setting to **No Wrapping**.



In addition to whitespace wrapping of text, you can use the CP_WrapText(text, nchars) function to wrap the text based on a number of characters as well as the whitespaces.

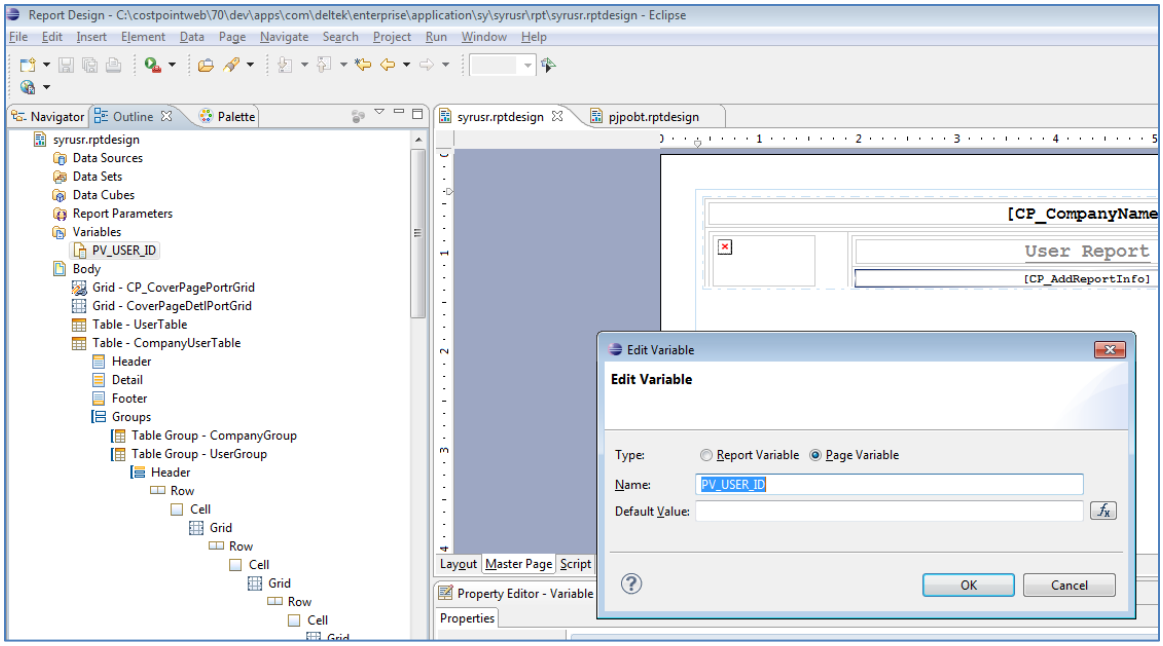


Display Page Header » Report Title Based on Current DataSet Row Variable

There are a few cases (limited to some reports) when you need to display the report title on the page header where the text of the report title depends on the value of the current DataSet row variable.

In order to do that, you have to use a BIRT page variable and override the onRender() method for the **Page Header » Report Title** label control.

1. Add a new page variable in **Outline » Variables » New Variable » Page Variable**.



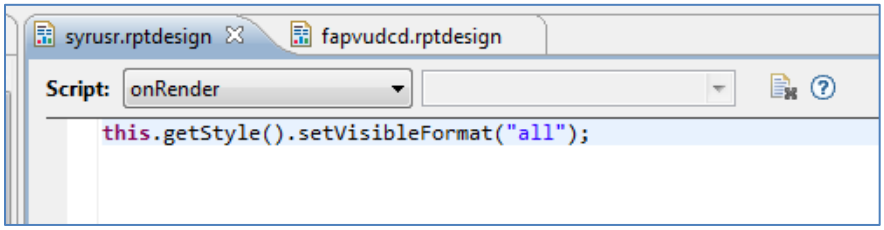
2. Name the variable using the following naming convention: the variable name should be uppercase and should start with PV_ prefix (for example, PV_USER_ID).

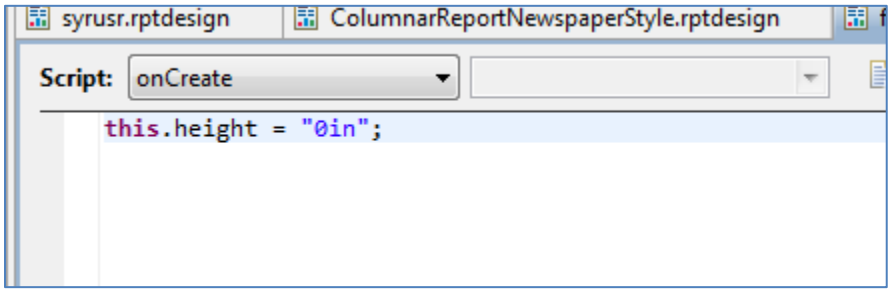
3. Set the value of the page variable by overriding the **onPageBreak()** method for the control that displays the DataSet data that you want to use in the page header.

If such a control is not there or should not be in the report by design, you need to create a dummy table/grid row, put a dummy control on that row, set control value, and make sure that this dummy control's parent row is hidden through:

- overridden Row>Scrip>OnRender() method>this.getStyle().setVisibleFormat("all");
- overridden Row>Scrip>OnCreate() method>this.height = "0in";

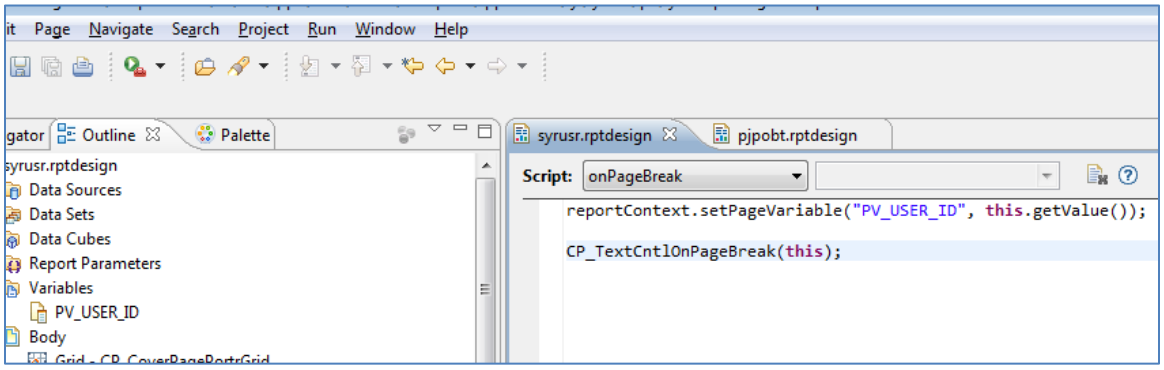
Do not hide the row or control using the **Hide Element** check box or set the row height to **0**'. Use the setVisibleFormat("all") approach only.





The control's **onPageBreak()** method should be overridden to set the page variable value. For example:

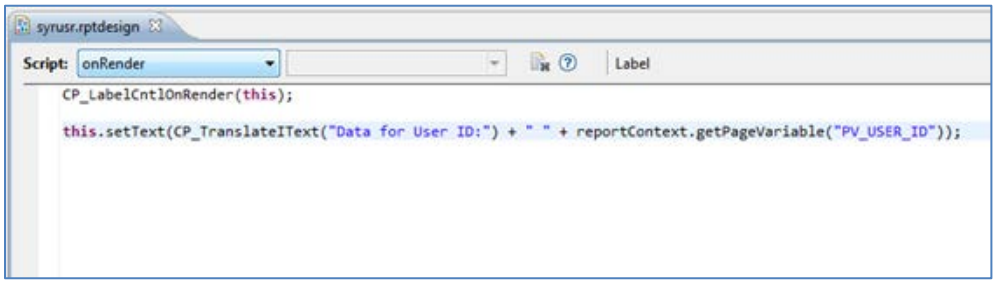
```
reportContext.setPageVariable("PV_USER_ID", this.getValue());
```



4. Go to **Master Page » CP_ReportBodyPage » Report Title Label**, and override the **onRender()** method based on the value of the page variable created in the earlier steps

   Inside the onRender() script, access the Page Variable value through the reportContext.getPageVariable() function. For example:

   ```
   CP_LabelCntlOnRender(this);
   this.setText(CP_TranslateIText("Data for User ID:") + " " +
   reportContext.getPageVariable("PV_USER_ID"));
   ```



Always use the **CP_TranslateIText()** method for static text and labels. You can use other CP_TranslateIText functions to format:

- Date/time — (CP_TranslateITextDateTime())
- Numbers — (CP_TranslateITextNumber())
- Currencies — (CP_TranslateITextCurrency).

Localization and I18n

Costpoint reports can be printed in different locales. All localization and i18n processing and formatting occurs at report rendering (view) time. This allows you to generate the report once, but render it differently based on different locale rules.

What Is Subject to Localization in Costpoint Reports

- Labels (text, text ranges)
- Date/Time controls
- Numeric controls (Integer, Decimal)

- Currency Controls
- TOC labels

Displaying Text

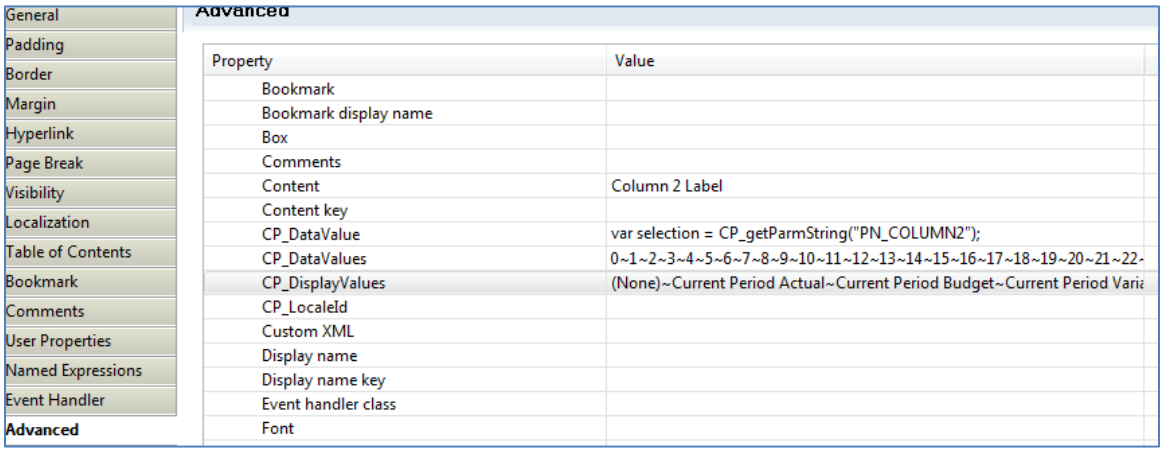
All static text and labels in a report have to be localized and translated. You must use the CP_LabelCntl control that provides multi-language translation. You always use the standard CP_TextCntl controls for data that does not require multi-language translation.

Labels can have a single display value or multiple display values; one of which will be picked based on a condition.

To create a single display value label, drag CP_LabelCntl from the library, and rename the label text.

To create a multiple display values label, drag CP_LabelCntl from the library, and in the Advanced Properties, set the values for:

- CP_DataValue (expression)
- CP_DataValues (~ separated list of data values)
- CP_DisplayValues (~ separated list of display values)



For reports where locale is data-driven (coming from RS data), set the CP_LocaleId expression (for example, **row["LOCALE_ID"]**).

Dictionary Terms

There are common terms that users would like to customize according to their practice. For example, a customer would like to use the term **Department** instead of **Organization**, **Client** instead of **Customer**, and so on. These terms are scattered throughout the product. They appear on the screen in labels, menu items, status help text, messages, and so on as well as on reports as titles, headings, and so on.

There are seven terms—Project, Organization, Account, Vendor, Customer, Employee, and Warehouse—that are defined with the following term IDs: PROJ, ORG, ACCT, VEND, CUST, EMPL, and WHSE.

Each term has Singular, Plural, Singular Abbreviation, and Plural Abbreviation forms. For each occurrence of the term in metadata (display title, status text, messages, and so on) or report labels, the following syntax is used:

- **Singular** — &CPT.TERM_ID&
- **Plural** — &CPTP.TERM_ID&

- **Singular Abbreviation** — &CPTS.TERM_ID&
- **Plural Abbreviation** — &CPTSP.TERM_ID&

For example:

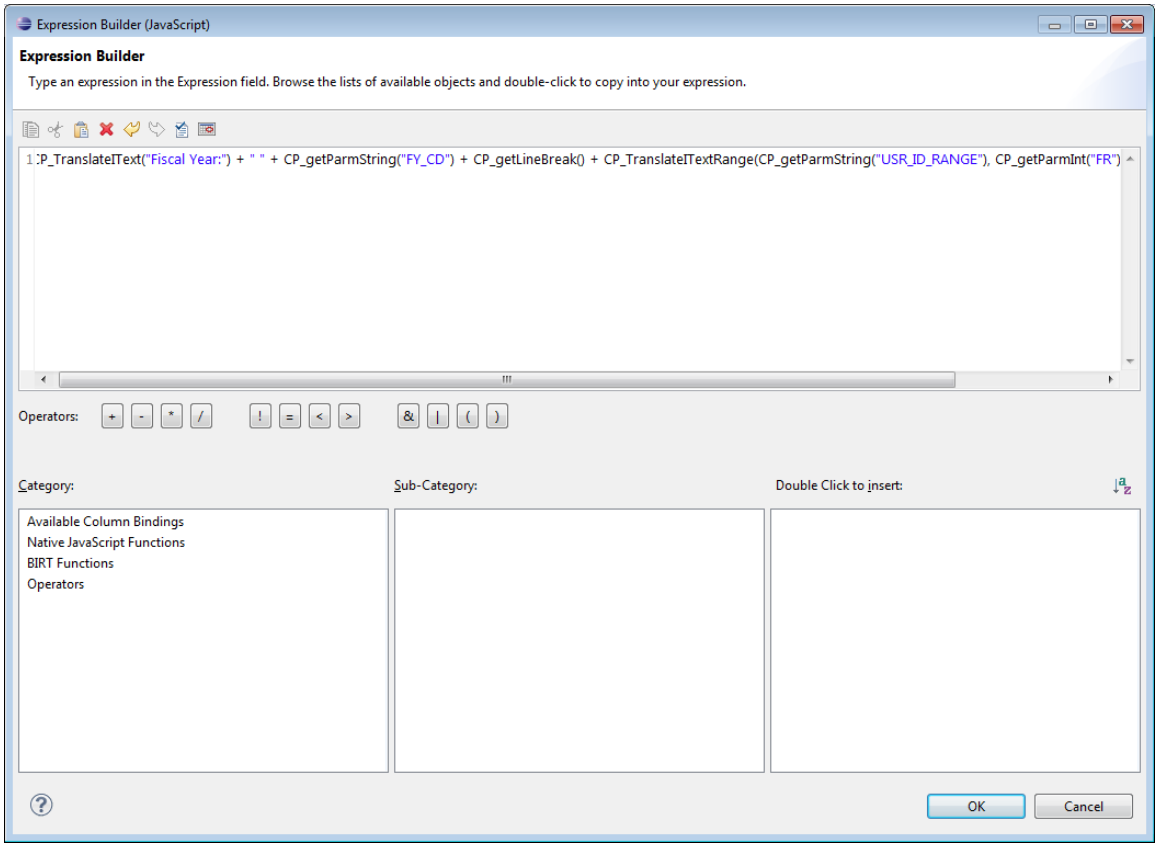
- Old label: “Organization:” — New syntax: “&CPT.ORG&:”
- Old label: "Projects:" — New syntax: "&CPTP.PROJ&:”

Since the very same report can be viewed and printed in different locales and languages, you should take caution when concatenating and presenting data of different types in one control.

For example, assume you want to show a static text “For Subperiod Ending” concatenated with the date value 01/01/2000. If you place both items in one control and concatenate them, then at view time, the text will not be translated properly and the date value will not be formatted according to the runtime locale.

Use the CP_iTextCntl control to address the above case. The CP_iTextCntl control can combine static text data with dynamic data. You must use special functions in controls value Expression:

- CP_TranslateIText(text)
- CP_TranslateITextDateTime(value, format)
- CP_TranslateITextCurrency(value, format, currencyCd)
- CP_TranslateITextNumber(value, format)
- CP_TranslateITextRange(rangeCd, from, to)
- CP_TranslateITextRange(rangeCd, from, to)
- CP_TranslateITextCustomRange(rangeCd, from, to, displayValues, dataValues)



Use the CP_getLineBreak() function to break the new line within a CP_iTextCntl control.

Example:

```
CP_TranslateIText("Fiscal Year:") + " " + CP_getParmString("FY_CD") +
CP_getLineBreak() + CP_TranslateITextRange(CP_getParmString("USR_ID_RANGE"),
CP_getParmString("FR"), CP_getParmString("TO"))
```

Text Ranges

Use CP_RangeCntl and CP_NCRRangeCntl controls to display text ranges. CP_RangeCntl - set CP_DataValue Expression for Range Code, CP_RangeFrom and CP_RangeTo Expressions to set Range From and To values. CP_NCRRangeCntl - set CP_NCRResultSetId value to identify NCR Result Set

For reports where locale is data-driven (coming from RS data), set the CP_LocaleId expression (for example, **row["LOCALE_ID"]**).

Displaying Date/Time

The format for dates and times is locale specific. The following controls provide this functionality:

- CP_DateCntl
- CP_DateTimeCntl
- CP_TimeCntl

In addition, you can use CP_DateCntl, CP_DateTimeCntl, and CP_TimeCntl constants as format masks in conversion functions. For example:

```
CP_translateITextDateTime(row["MY_DATE"], "CP_DateTimeCntl")
```

For reports where locale is data-driven (coming from RS data), set the CP_LocaleId expression (for example, **row["LOCALE_ID"]**).

Displaying Numbers

The format for numbers is locale specific. The following controls provide this functionality:

- CP_IntCntl
- CP_IntNoThouCntl
- CP_DecimalCntl
- CP_DecimalNoThouCntl
- CP_PercentCntl
- CP_RateCntl

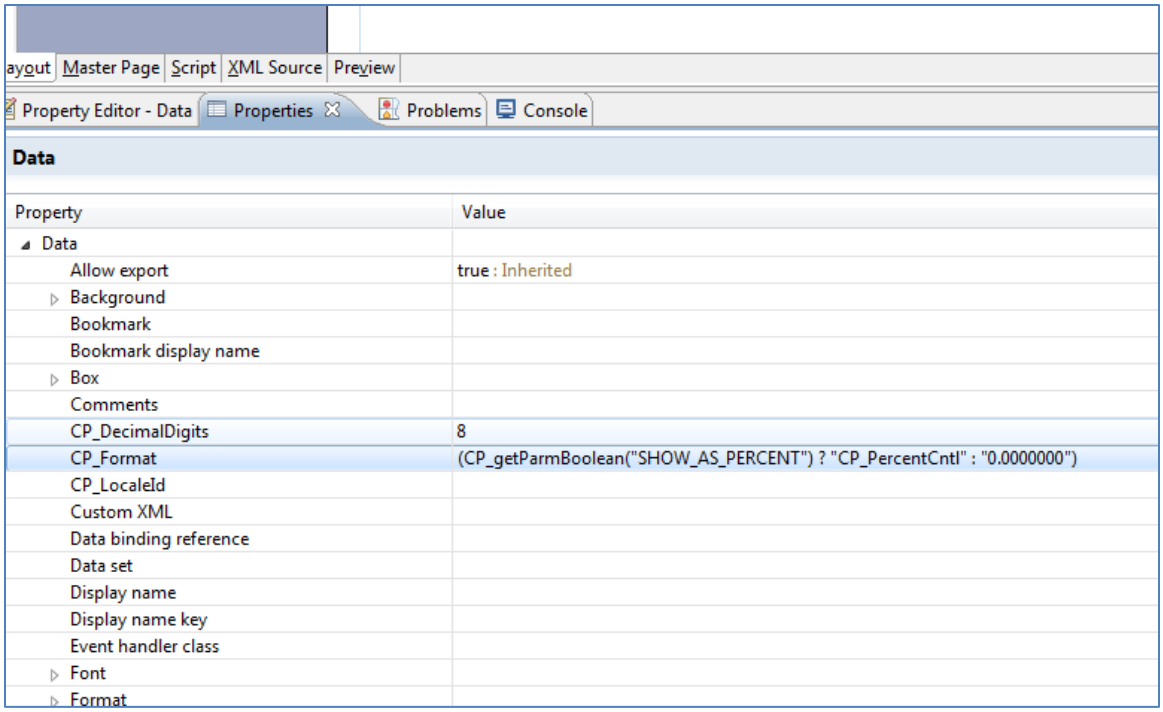
In addition you cany use CP_IntCntl, CP_IntNoThouCntl, CP_DecimalCntl, CP_DecimalNoThouCntl, and CP_PercentCntl, CP_RateCntl constants as format masks in conversion functions. Fopr example:

```
CP_translateITextNumber(row["MY_AMT"], "CP_DecimalCntl")
```

For reports where locale is data-driven (coming from RS data), set CP_LocaleId expression (for example, **row["LOCALE_ID"]**.

Within the CP_Format property, you can use custom format masks to override default formatting (for example, "0.000000", "#,##0.0000000").

Change the CP_DecimalDigits property to override the default formatting in regards to the number of decimal digits.



Displaying Currency

The format for currency data is locale specific. The following controls provide this functionality:

- CP_CurrencyCntl
- CP_CurrencyNoDecCntl
- CP_CurrencyNoDecNoSymbCntl
- CP_CurrencyNoDecNoSymbNoThouCntl
- CP_CurrencyNoDecNoThouCntl
- CP_CurrencyNoSymbCntl
- CP_CurrencyNoSymbNoThouCntl
- CP_CurrencyNoThouCntl

In addition you can use CP_CurrencyCntl, CP_CurrencyNoDecCntl, CP_CurrencyNoDecNoSymbCntl, CP_CurrencyNoDecNoSymbNoThouCntl, CP_CurrencyNoDecNoThouCntl, CP_CurrencyNoSymbCntl, CP_CurrencyNoSymbNoThouCntl, and CP_CurrencyNoThouCntl constants as format masks in conversion functions. For example:

```
CP_translateITextCurrency(row["MY_AMT"], "CP_CurrencyCntl")
CP_translateITextCurrency(row["MY_AMT"], "CP_CurrencyCntl", row["CURRN_CD"])
```

For reports where locale is data-driven (coming from RS data), set the CP_LocaleId expression (for example, **row["LOCALE_ID"]**).

For reports where currency code is data-driven (coming from RS data), set the CP_CurrencyCd expression (for example, **row["CURRN_CD"]**).

Change the CP_DecimalDigits property to override the default formatting in regards to the number of decimal digits.

Displaying TOC

For TOC text, use the CP_translateText(text) function. For example:

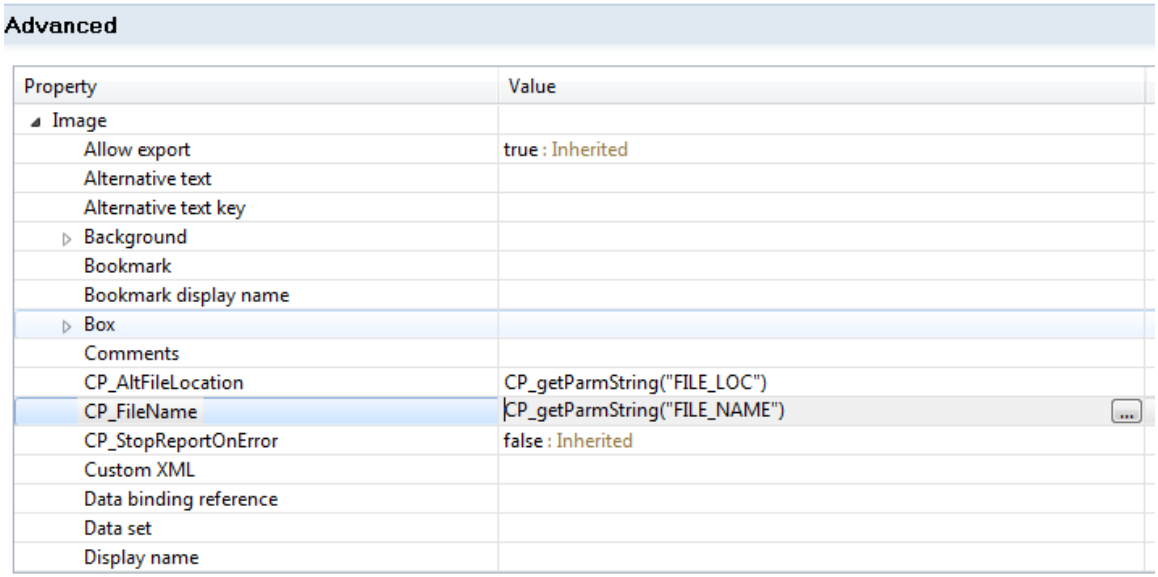
```
CP_translateText("Project Status Report")
```

Image Controls

The following controls are provided to support images:

- CP_ImageCntl — Basic image control
- CP_DynamicImageCntl — Image control that allows you to load image files from Costpoint File Locations.
- CP_CompanyLogoCntl — Image control for displaying a company logo image

The CP_DynamicImageCntl control allows you to load image files stored in Costpoint File Locations. Set the CP_AltFileLocation and CP_FileName Expression values.



Report Table of Contents

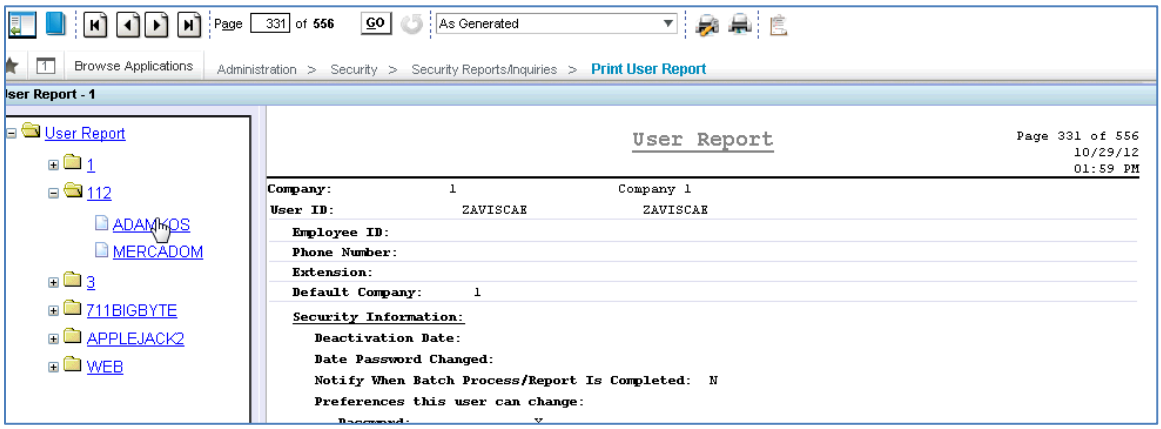
The Table of Contents (TOC) allows users to proceed directly to a specific report item in Preview mode.

Follow these rules to code the TOC:

- The Cover Page is not part of the TOC
- The Report Section title is part of the TOC. If there are multiple report sections (for example, posting reports), then there will be multiple top-level report titles in the TOC.
- Under the Report Section, a developer should look for primary group by/sort by logical keys and not go for inner groups down the tree requested by Design. A label should not be used at this level. For example, if the break group key is PROJ_ID, then a developer should use just the key value in the TOC, but not the key value pre-pended by a label (for example, **1003** instead of **Project 1003**).
    - For hard coded (static) text or CP terms, use the CP_TranslateText(value) function. For example:

- CP_TranslateText("Project Status Report")
- CP_TranslateText("&CPT.EMPL& Salary Report")

- For formatted date/time and numbers, use the CP_toString(value, format) function. For example:
  - CP_toString(row["MY_DT"], "CP_DateCntl"),
  - CP_toString(row["VOUCHER_NO"], "CP_IntCntl")
- For skipping a TOC level in your report, leave the TOC expression blank for the report section/group that you need to skip.



Utility functions

The cp_utils.js file contains the list of public functions that can be used by report developers. Check the cp_utils.js file for complete list of functions and descriptions of each.

Here is a list of the most useful functions:

- **CP_getParm... (name)** — Returns a parameter value specified by name
- **CP_getConstant...(name)** — Returns a constant value specified by name
- **CP_TranslateIText...(value[, format, currencyCd])** — Concatenates and translates text, number, and currency date/time values for the CP_iTextCntl control
- **CP_RemoveTableFromBuild(tableName)** — Removes a Table Object froma report build. Override *.rptdesign **beforyFactory** to call this function.
- **CP_removeCellStyle(objName, row, column)** — Removes cell style dynamically. Override *.rptdesign **beforyFactory** to call this function.
- **CP_TranslateText(value)** — Used for TOC text translation.

  Do not use inside CP_iTextCntl!

- **CP_toString(value, format)** — Used to format TOC date/time/number at report execution time.

Deltek is the leading global provider of enterprise software and information solutions for government contractors, professional services firms and other project- and people-based businesses. For decades, we have delivered actionable insight that empowers our customers to unlock their business potential. 20,000 organizations and millions of users in over 80 countries around the world rely on Deltek to research and identify opportunities, win new business, recruit and develop talent, optimize resources, streamline operations and deliver more profitable projects. Deltek – Know more. Do more.®

deltek.com