TEAMCENTER

Report Management Using Report Builder

Teamcenter 2412

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow’s idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: **support.sw.siemens.com**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

Configuring reports for Active Workspace

Defining reports in Reporting and Analytics

Generating reports

1. Introduction to Report Builder

What are reports in Teamcenter?

In Teamcenter, each *report* mainly consists of a report definition and an associated XSL style sheet:

- *Report definition* specifies the data you want to display in the report. This includes the data source, additional properties to be extracted, and the rules for importing the data.

- *XSL style sheets* allow you to control the properties you want to display in the report and the display order. They also help you define the look and format of the report.

What is Report Builder?

As a Reports administrator (with or without DBA privileges), you use Report Builder to create and manage report definitions. End users use these report definitions to generate various types of reports in the rich client or Active Workspace. There are three main types of reports:

- **Summary**

  Reports that summarize similar information, for example, reports that show all the employees, the items belonging to a user, or the release status of items.

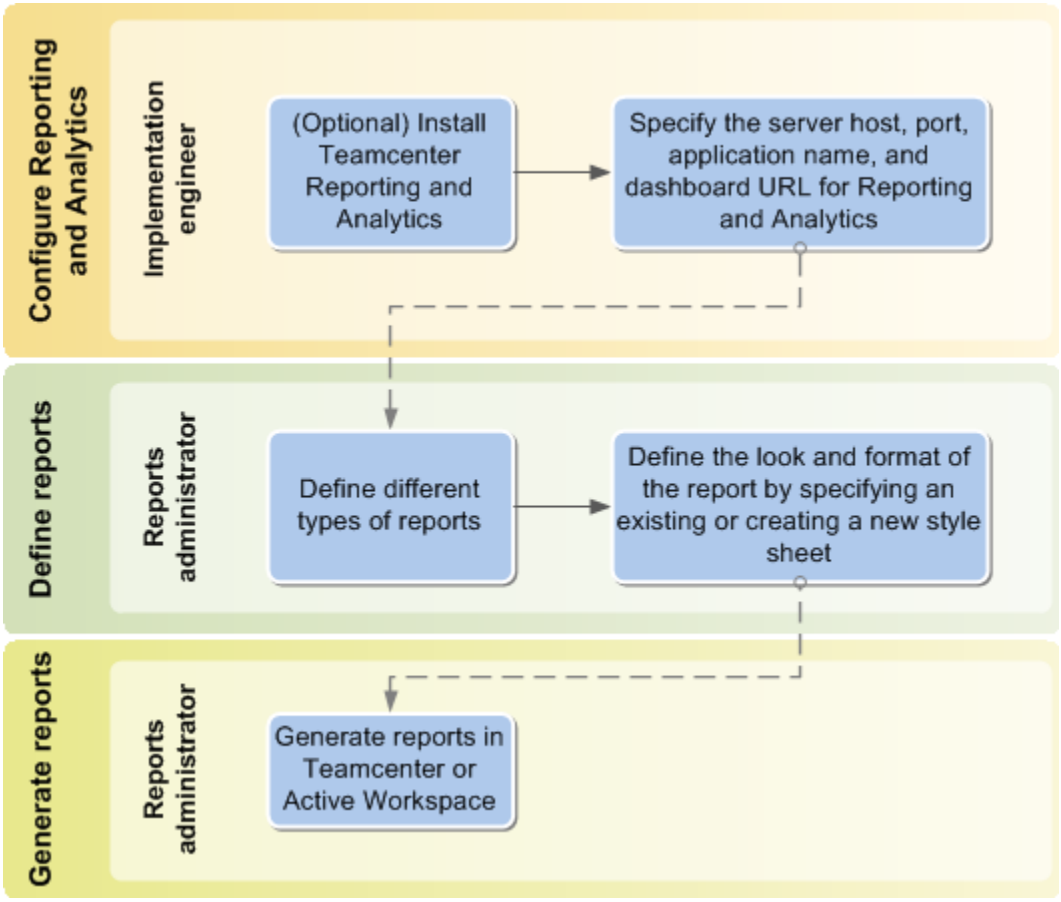
- **Item**

  Reports that can be run on a particular item or on multiple items, for example, reports that show the BOM list for an item or the workflow signoff for an item.

- **Custom**

  Reports that address special cases such as complex processing or calculations done through custom code, or those done with data obtained from external sources.

Report Builder processes and user roles



About configuring Report Builder and creating report definitions

The following are the high-level tasks to configure Report Builder and create report definitions:

Role	Task	Description
Reports administrator	Specify site preferences	Specify site preferences for the source or reports and server parameters for Reporting and Analytics.
Reports administrator	Import and export report definitions	Use the **import_export_reports** utility to export report definitions, their dependent data (for example, saved query definitions and property

Role	Task	Description
		set definitions), and their associated style sheets from one Teamcenter server and import them to another server.
Query administrator, Data administrator, and Reports administrator	**Define a summary report**	Use Query Builder to create saved queries and PLM XML/TC XML Export Import Administration to specify additional properties to extract and the rules for importing the data.
Reports administrator	**Create a summary report definition**	Create a summary report definition to define the content for the report and select style sheets to define the style and display the report.
Query administrator, Data administrator, and Reports administrator	**Define an item report**	Use Query Builder to create saved queries and PLM XML/TC XML Export Import Administration to specify additional properties to extract and the rules for importing the data.
Reports administrator	**Create an item report definition**	Create an item report definition to define the content for the report and select style sheets to define the style and display for the report.
Reports administrator	**Define a custom report**	Create custom reports for specific cases, such as complex processing or calculations performed through custom code. These may also be used for data obtained from external sources.
Reports administrator	**Create a custom report definition**	Create a custom report definition to define the content for the report and select style sheets to define the style and display for the report.
Reports administrator	**Create style sheets to define the look and the format of reports**	Use XSL style sheets to provide more control over the properties you want to display in the report and the display

Role	Task	Description
		order. They also help define the look and format of the report.
Reports administrator	**Define reports with a watermark**	(Optional) Define a watermark for reports to identify copyright information and the status of the generated report. For example, add a watermark to specific types of reports to display your company name and the report status, such as confidential.
Reports administrator	**Enable preview for Excel reports**	Modify a preference to enable preview for Excel reports.
Reports administrator	**Define Reporting and Analytics report definitions**	Use Reporting and Analytics Mapper to set user rights and to create the cubes (similar to property sets) that are displayed in reports and BuildNPlay to create and manage the report definitions specific to Reporting and Analytics.

2. Install and configure Report Builder and Reporting and Analytics

Before installing Teamcenter Reporting and Analytics

Open the Teamcenter downloads page on Support Center, click the **Additional Downloads** tab, and find **Reporting and Analytics 2412**. Download the following packages:

- Software: **TcRA2412.zip**
- Documentation: **TcRA2412_Deployment_Guide.zip**

Expand both packages to your local machine.

In the **Deployment Guide** directory, find the *Teamcenter Reporting and Analytics Deployment Guide* for your language.

Install Reporting and Analytics as described in *Install Reporting and Reporting and Analytics applications using TEM* or *Install Reporting and Reporting and Analytics applications using Deployment Center*.

After you complete these steps and install Reporting and Analytics, you can begin using the Reporting and Analytics integration in Report Builder.

Install Reporting and Reporting and Analytics applications using TEM

Add **Reporting** and **Reporting and Analytics** applications to your existing Teamcenter environment.

Procedure

1. Stop all Teamcenter services *except* FSC services before beginning installation.
2. Launch Teamcenter Environment Manager (TEM) for the corporate server to which Server Extensions features are to be added.
3. In the **Maintenance** panel, select **Configuration Manager**.
4. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**.
5. In the **Old Configuration** panel, select the corporate server configuration.

6. In the **Feature Maintenance** panel, select **Add/Remove Features**.

7. Select the following features:

Feature name	Description
Reporting	Adds the ability to view report templates, generate reports based on selected criteria, style sheets, or both, and view them in HTML, Excel, or raw XML formats in the client.
Teamcenter for Reporting and Analytics	Installs the Teamcenter Reporting and Analytics (TcRA) integration. TcRA is a standalone reporting application that introduces a new folder in Report Builder called **TcRA Reports**, which contains reports created with TcRA.

8. Continue the installation as described in *Install patches on a Teamcenter server* in *Teamcenter Upgrade Using TEM* in the Teamcenter documentation.

Postrequisites

If you are installing the **Reporting** feature in the Client, you must install the **Reporting** Server Extensions feature in the corporate server.

Additionally, to allow for asynchronous report generation, install the **AsyncService** translator in the Dispatcher Server.

Install Reporting and Reporting and Analytics applications using Deployment Center

Add **Reporting** and **Reporting and Analytics** applications to your existing Teamcenter environment.

Procedure

1. Log on to Deployment Center and select the environment to which you want to add **Reporting** and **Reporting and Analytics** applications.

2. Go to the **Applications** task. Click **Add or Remove Selected Applications**.

3. In the **Available Applications** panel, use the web browser search to find the following applications.

Feature name	Description
Reporting	Adds the ability to view report templates, generate reports based on selected criteria,

Feature name	Description
	style sheets, or both, and view them in HTML, Excel, or raw XML formats in the client.
Reporting and Analytics	Installs the Teamcenter Reporting and Analytics (TcRA) integration. TcRA is a standalone reporting application that introduces a new folder in Report Builder called **TcRA Reports**, which contains reports created with TcRA.

4. Select the applications, and then click **Update Selected Applications**.

   Deployment Center automatically selects any additional dependent applications.

5. In the **Selected Components** list, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

   When all components are fully configured, the **Deploy** task is enabled.

6. Go to the **Deploy** task. Click **Generate Install Scripts** to generate deployment scripts you will use to update affected machines.

   When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

7. Locate deployment scripts, copy each script to its target machine, and then run each script on its target machine.

   For more information about running deployment scripts, see *Deploy task* in *Deployment Center — Usage*.

Postrequisites

If you are installing the **Reporting** feature in the Client, you must install the **Reporting** Server Extensions feature in the corporate server.

Additionally, to allow for asynchronous report generation, install the **AsyncService** translator in the Dispatcher Server.

Specify the source for reports and the server parameters for Reporting and Analytics

Site preference	Description
Default_Crf_Office-Template_relation	Specifies the relation between the report definition and the associated Microsoft Office template.
Enable_Report-Builder_Reports_By_Source	Specifies the reports you can display in the **Tools→Reports→Report Builder Reports** menu.
ReportDefinition_Column-Preferences	Determines which columns are displayed in Report Builder.
ReportDefinition_Column-WidthPreferences	Specifies the column width for each column defined in the **ReportDefinition_Column-Preferences** preference.
ReportGenerationDlg_Column-Preferences	Determines which columns are displayed in the Report Generation wizard.
ReportGenerationDlg_Column-WidthPreferences	Specifies the column width for each column defined in the **ReportGenerationDlg_Column-Preferences** preference.
TC_RA_server_parameters	Specify the server host, port, and application name of Reporting and Analytics.
Valid_ReportBuilder_Sources	Specify all the valid sources available for the common reporting framework.

Importing and exporting report definitions

Arguments for importing and exporting report definitions

You use the **import_export_reports** utility to export report definitions, their dependent data (for example, saved query definitions and property set definitions), and their associated style sheets from one Teamcenter server and import them into another server.

```
import_export_reports {-import | -export | -execute}
[-u=user-id -p=password -g=group]
-stageDir=directory -reportId=report-identifier -f=output-filename.xml [-
h]
```

Argument	Definition
import	Imports the report definition.
export	Exports the report definition.
stageDir	Is the fully qualified name of the directory that contains all the report definitions and its associated data in a predefined format. This directory must exist prior to an import and an export.
reportId	Is the ID of the report definition. On export, this is the name of the directory that is created where the report definition and style sheets are written. On import, this is the name of the directory where the report definition and style sheets are located.

Import or export report definitions by using a sample XML file

You can import or export report definitions by using a sample XML file. You can specify the revision rule name, variant rule name, and other details in this XML file.

For more information about sample XML files, see *TC_DATA***crf****crf_item_report_offline_execution_sample.xml** for an item report and *TC_DATA***crf****crf_summary_report_offline_execution_sample.xml** for a summary report.

The following procedure is based on an item report.

1. Create an XML file as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<CrfReport>
  <ItemReport command="execute" id="TC_2007_00_ITM_RPT_0003"
                stylesheet_name="ps_bom_report_excel.xsl" >

    <context_item  name="000004"  rev="A" />

    <report_option name="report_locale" value="en_US" />
    <report_option name="rev_rule" value="Revision_Rule_Name" />
    <report_option name="variant_rule" value="Variant_Rule_Name" />


  </ItemReport>
</CrfReport>
```

2. Edit this XML file to specify the following parameters:

Description	Example
Report definition ID.	**id=***"TC_2007_00_ITM_RPT_0003"*
(Optional) Specify the style sheet name. If **stylesheet_name=** is not specified in the input XML file, then system generates the default report with the **.xml** extension.	**stylesheet_name=***"ps_bom_report_excel.xsl"*
(Optional) Specify the revision rule name. If the revision rule is not specified, the default is **Latest Working**.	**<report_option name=***"rev_rule"* **value=***"Revision_Rule_Name"*
(Optional) Specify the variant rule name.	**<report_option name=***"variant_rule"* **value=***"Variant_Rule_Name"*

3. Run the following command from the Teamcenter command prompt:

   import_export_reports -execute -u=*user_name* **-p=***password* **-g=***dba* **-f=***C:\Temp\sample.xml*

   Where **-f=** is the location of the input XML file, including the complete directory path and filename.

Export a report definition

1. Ensure that you are in the proper environment for running a command utility.

   For example, on a Windows system where the corporate server is installed, click the **Start** button and choose **Programs→Teamcenter→Command Prompt**.

2. Type the **import_export_reports** utility commands:

   a. Enter the utility name: **import_export_reports**

   b. Specify the action argument: **-export**

   c. (Optional) Specify an administrator user: **-u=***user-name* **-p=***password* **-g=***group*

   d. Specify an existing staging directory using the **-stageDir** option, for example, **d:\tc_data**.

   e. Specify an existing report ID in Teamcenter using the **-reportId** option, for example, **TC_2007_00_SUM_RPT_0001**.

   > Note:
   >
   > To find the ID for a report, in the Report Builder application, select the report in the **Reports Home** folder and view the ID on the **Details** tab.

The syntax is as follows:

```
import_export_reports -export -u=user-name -p=password -g=group
-stageDir=staging-directory -reportId=ID
```

> Note:
>
> Data definitions that are dependent on report definitions, for example, saved query definitions and property set definitions, are included with the exported report definition.

Example

1. Consider a situation where you want to export an existing report definition **Admin - Item Ownership** whose report ID is **TC_2007_00_SUM_RPT_0001**.

   Create a staging directory named **Reports** under **C:\Temp** (Windows) or **/tmp** (Linux).

   > Note:
   >
   > Ensure that the user executing the **import_export_reports** utility has *write access* permission to this directory.

2. Start the Teamcenter command prompt and specify the following command:

   ```
   import_export_reports -export -u=username -p=password -g=dba
   -stageDir="C:\Temp\Reports" -reportId="TC_2007_00_SUM_RPT_0001"
   ```

   After the export, Teamcenter creates a new folder in the staging directory. The folder name is the same as the report ID: **TC_2007_00_SUM_RPT_0001**.

   Teamcenter creates another folder named **Resources** in the report ID folder. The associated style sheets are copied to this folder.

   After a successful export, you view the following directories and files:

   C:\Temp\Reports\TC_2007_00_SUM_RPT_0001

   C:\Temp\Reports\TC_2007_00_SUM_RPT_0001\Resources

   C:\Temp\Reports\TC_2007_00_SUM_RPT_0001\TC_2007_00_SUM_RPT_0001.xml

   C:\Temp\Reports\TC_2007_00_SUM_RPT_0001\Resources\admin_ownership_html.xsl

   C:\Temp\Reports\TC_2007_00_SUM_RPT_0001\Resources\l10n_admin_ownership_html.xsl

Import a report definition

1. Ensure that you are in the proper environment for running a command utility.

   For example, on a Windows system where the corporate server is installed, click the **Start** button and choose **Programs→Teamcenter→Command Prompt**.

2. Type the **import_export_reports** utility commands:

   a. Enter the utility name: **import_export_reports**

   b. Specify the action argument: **-import**

   c. (Optional) Specify an administrator user: **-u=***user-name* **-p=***password* **-g=***group*

   d. Specify the staging directory where the reports reside, using the **-stageDir** option, for example, **d:\tc_data**.

   e. Specify an existing report ID from the staging directory, using the **-reportId** option, for example, **TC_2007_00_SUM_RPT_0001**.

   The syntax is as follows:

   ```
   import_export_reports -import -u=user-name -p=password -g=dba
   -stageDir=staging-directory -reportId=ID
   ```

Example

1. Consider that you must import a (previously) **exported report definition**.

   > Note:
   >
   > Ensure that the staging directory (**C:\Temp\Reports**) contains the report ID folder (the folder whose name is the same as the report ID). This folder must contain the report definition template and other style sheet files.
   >
   > The directory structure looks similar to this:
   >
   > **C:\Temp\Reports\TC_2007_00_SUM_RPT_0001**
   >
   > **C:\Temp\Reports\TC_2007_00_SUM_RPT_0001\Resources**
   >
   > **C:\Temp\Reports\TC_2007_00_SUM_RPT_0001\TC_2007_00_SUM_RPT_0001.xml**
   >
   > **C:\Temp\Reports\TC_2007_00_SUM_RPT_0001\-Resources\admin_ownership_html.xsl**

C:\Temp\Reports\TC_2007_00_SUM_RPT_0001\-Resources\l10n_admin_ownership_html.xsl

2. Start the Teamcenter command prompt and specify the following command:

```
import_export_reports -import -u=username -p=password -g=dba
-stageDir="C:\Temp\Reports” -reportId=”TC_2007_00_SUM_RPT_0001"
```

If the report definition does not exist in Teamcenter, the import is successful. Else the import is skipped.

Upgrading standard Report Builder reports

Due to possible customization loss, the upgrade process does not automatically upgrade standard Report Builder reports.

1. Back up report definitions using the **import_export_reports** utility with the **-export** argument. This exports one report definition at a time:

```
import_export_reports -export
-u=user-name -p=password -g=group
-stageDir=${TC_DATA}/crf -reportId=REPORTID
```

If you have style sheet customizations, you can save them using Report Builder by opening the style sheet and saving to a local hard drive.

2. Update all standard reports using the following command:

```
import_export_reports -import -overwrite
-u=user-name -p=password -g=group
-stageDir=${TC_DATA}/crf -reportFile=${TC_DATA}/crf/CrfReports.xml
```

3. Defining summary, item, and custom reports in Report Builder

Summary, item, and custom reports

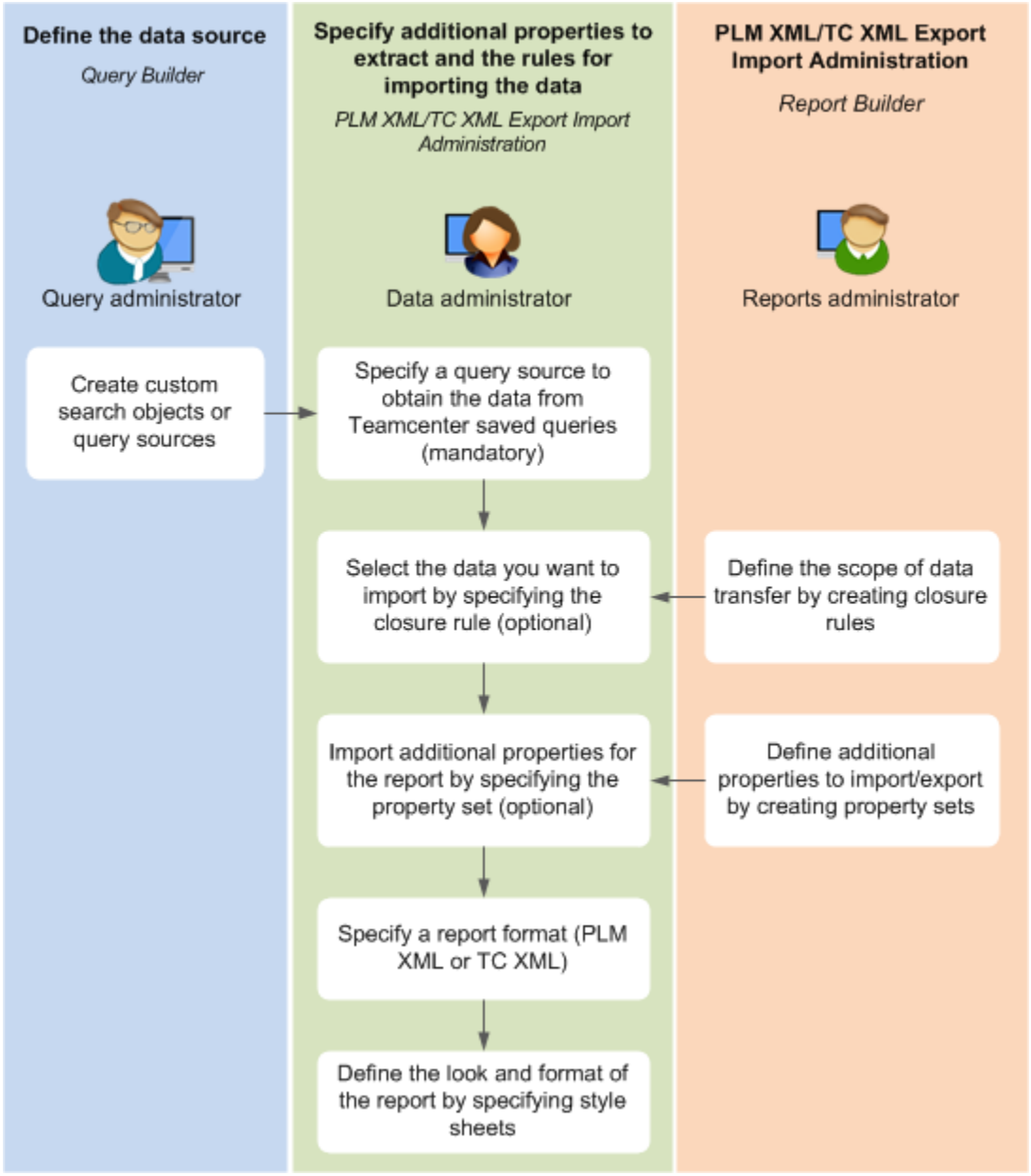
The types of reports in Report Builder

There are three types of reports in Report Builder: *summary*, *item*, and *custom*. Summary and item reports are static reports. They are generated from the persistent properties in a database. Custom reports are dynamic reports and are generated from the runtime properties in a database. For example, you can create a custom report that generates and displays the BOM line attributes of a product structure, and the BOM line information may vary depending on the revision rule you apply. If you apply the **Released revisions only** revision rule, Teamcenter applies only the latest released revisions, and does not select any working revisions.

Defining summary reports

Define a summary report

Summary reports present information based on a Teamcenter saved query definition. You use Query Builder to create saved queries. For example, you can create a saved query to find, in the database, all items that have been shipped, the items belonging to a particular user, or the release status of all items.



Query sources:

Role	Task	Description
Query administrator	Creates query sources in Teamcenter Query Builder.	Teamcenter provides standard saved queries to search for objects. The query administrator creates custom search objects in Teamcenter Query Builder.

Role	Task	Description
		A saved query points to the search class (database table) and contains the search criteria for the attributes of the class.
Reports administrator	**Specifies a query source for the summary report in Report Builder**.	If the standard saved queries do not meet your requirements, you can create your own saved queries for summary reports.

Closure rules:

Role	Task	Description
Data administrator	Creates closure rules in PLM XML/TC XML Export Import Administration.	Teamcenter provides standard closure rules to define the scope of data transfer. The rule defines the traversal from a target object to its related object and whether to process or skip the object.
Reports administrator	**Specifies a closure rule for the summary report in Report Builder**	If the standard closure rules do not meet your requirements, you can create your own closure rules for summary reports.

Property sets:

Role	Task	Description
Data administrator	Creates property sets in PLM XML/TC XML Export Import Administration.	Property sets are collections of Teamcenter data, such as class attributes. They provide a nonprogrammatic way to control attributes or properties you want to store in a container before importing or exporting them.
Reports administrator	**Specifies a property set for the summary report in Report Builder**.	If the standard property sets do not meet your requirements, you can create your own property sets for summary and item reports.

Report format and style sheets:

Role	Task	Description
Reports administrator	**Specifies a report format: PLM XML or TC XML.**	Report Builder uses two schemas: PLM XML and TC XML. PLM XML is an open schema based on standard W3C XML schemas. TC XML schema is a Siemens Digital Industries Software format that uses Teamcenter XML. PLM XML supports the export and import of Teamcenter objects, such as items, datasets, BOMs, forms, and folders, as well as system data, such as business rules and organization data. TC XML is a schema specifically designed to work with Teamcenter data. You use TC XML and transfer option sets with to move data between Teamcenter and Teamcenter Enterprise (Teamcenter Enterprise) sites.
Reports administrator	**Creates new style sheets or specifies existing style sheets.**	The XSL style sheets provide more control over the properties you want to display in the report and the display order. They also help define the look and format of the report. You can create new style sheets or use existing style sheets provided with Teamcenter and customize them for different types of reports. You can add more than one style sheet to a report. The end user can then select different style sheets while generating the report. For example, an end user can generate a summary report in HTML or spreadsheet format if you define both style sheets for the summary report.

Create a summary report definition

Summary reports list information about Teamcenter data. For example, you can use a summary report to list users in a group or to list the **Item** objects released on a particular day.

Saved queries are the data source for summary report definitions. You can use the supplied saved queries, or you can create your own query using Query Builder.

1. Choose **File→Create Report**.

   The system displays the Create Report Definition Template wizard.

2. Select **Summary Report** and click **Next**.

3. Perform the following steps in the **Report Information** pane:

   a. In the **Report ID** box, type an ID or click **Assign** to automatically assign an ID.

   b. In the **Name** box, type a name for the report definition.

   c. In the **Description** box, type a short description of the report definition.

   d. From the **Source** list, select the source of the report.

      The **Source** value is defined in the **Valid_ReportBuilder_Sources** preference.

   e. From the **Query Source** list, select a query definition.

      > Note:
      >
      > To see the values of query definitions, run the Query Builder application, open the **Saved Queries** folder in the left pane, and select the query. The values of the query are displayed in the right pane. You can also create your own query using the Query Builder application.

   f. From the **Closure Rule** list, select a closure rule to process the query definition. Closure rules specify how the data structure is traversed by specifying which relationships are of interest.

      > Note:
      >
      > To see the values of closure rules, run the PLM XML/TC XML Export Import Administration application and select the closure rule in the left pane. The values of the selected closure rule are displayed in the right pane. You can also create your own closure rule using the PLM XML/TC XML Export Import Administration application.

   g. From the **Property Set** list, select a set of properties to use for the report.

      > Note:
      >
      > To see the values of property sets, run the PLM XML/TC XML Export Import Administration application and select the name of the property set in the left pane. The values of the selected property set appear in the right pane. You can also create your own property set using the PLM XML/TC XML Export Import Administration application.

   h. From the **Report Format** list, select the format for the report, **PLMXML** or **TCXML**.

      Select **TCXML** to view all the Teamcenter attributes, and select **PLMXML** to view attributes displayed in PLM XML industry standard names.

TCXML supports transfer option sets. Transfer option sets are created in Tc XML and PLM XML Configuration for Data Import and Export and are used to move data between Teamcenter and Teamcenter Enterprise sites. You can set transfer options for a report by creating a *report_ID*_**transferoptions** user preference. The parameter *report_ID* specifies the ID of the report and is available in the **Details** tab. You can specify transfer options as name-value pairs in the **Values** box of the preference dialog box.

An example of transfer options specified in the **Values** box is as follows:

```
opt_rev_select=allItemRevisionsopt_entire_bom=True
```

i. (For Active Workspace only) To allow users to generate the report asynchronously, select the **Run in Async** check box.

The **AsyncService** translator uses the Dispatcher framework to independently process asynchronous requests from the Teamcenter server using the Teamcenter SOA native C++ framework in the background mode.

j. Click **Next**.

4. Perform the following steps in the **Reports Stylesheets** pane:

a. Select the style sheet you wish to use for the report in the **Defined Stylesheets** pane and click the **+** button to move it to the **Selected Stylesheets** pane.

> Note:
>
> To import a style sheet, click the **Import stylesheets** button, located under the **+** and **–** buttons in the **Report Stylesheets** pane.
>
> In the **Import Stylesheet** dialog box, in the **Dataset Type** box, select a type for the new style sheet: **HTML**, **MSExcel**, **MSWord**, **Text**, or **Xml**. This option is required if the end user wants to save the report as a dataset at the time the report is run.

b. Click **Finish**.

The new report is added to the **Teamcenter Reports** folder.

c. Click **Close**.

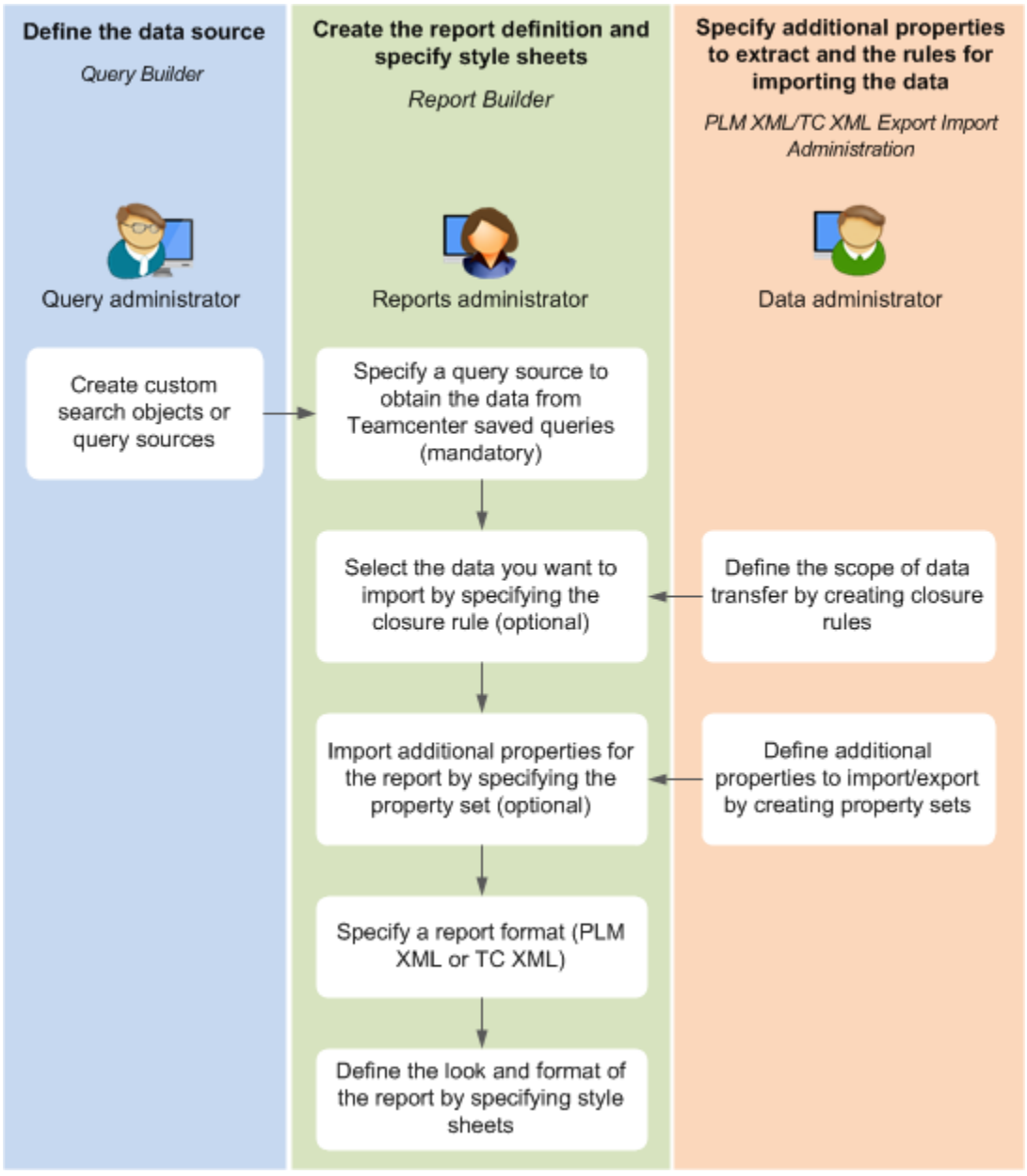
5. To verify that the report definition works as desired, generate a report.

For example, in My Teamcenter, in the rich client, choose **Tools→Reports→Summary Reports** and select your new summary report.

Defining item reports

Define an item report

Reports that can be run on a particular item or on multiple items, for example, reports that show the BOM list for an item or the workflow signoff for an item.



Classes:

Role	Task	Description
Data model administrator	Defines data model objects, including classes, in Business Modeler IDE.	Teamcenter manages data by grouping similar types of data into classes. You create and edit classes in Business Modeler IDE.
Reports administrator	**Specifies a class for the item report in Report Builder**.	A class is a table in a database, and it corresponds to a business object on the user interface. Examples include items, item revisions, datasets, forms, and folders.

Transfer modes:

Role	Task	Description
Data administrator	Creates transfer modes in PLM XML/TC XML Export Import Administration.	Teamcenter provides standard transfer modes that allow end users to import or export data between applications. A transfer mode is a logical grouping of closure rule clauses. Transfer modes allow end users to export or import data by knowing only the transfer mode name that they must use, for example, **WSOTransferModeDefault** for exporting or importing workspace objects.
Reports administrator	**Specifies a transfer mode for the item report in Report Builder**.	If the standard transfer modes do not meet your requirements, you (as a reports administrator) can create your own transfer modes for item reports.

Property sets:

Role	Task	Description
Data administrator	Creates property sets in PLM XML/TC XML Export Import Administration.	Property sets are collections of Teamcenter data, such as class attributes. They provide a nonprogrammatic way to control attributes or properties you want to store in a container before importing or exporting them.
Reports administrator	**Specifies a property set for the summary report in Report Builder**.	If the standard property sets do not meet your requirements, you can create your own property sets for summary and item reports.

Report format and style sheets:

Role	Task	Description
Reports administrator	**Specifies a report format: PLM XML or TC XML.**	Report Builder uses two schemas: PLM XML and TC XML. PLM XML is an open schema based on standard W3C XML schemas. TC XML schema is a Siemens Digital Industries Software format that uses Teamcenter XML. PLM XML supports the export and import of Teamcenter objects, such as items, datasets, BOMs, forms, and folders as well as system data, such as business rules and organization data. TC XML is a schema specifically designed to work with Teamcenter data. You use TC XML and transfer option sets to move data between Teamcenter and Teamcenter Enterprise (Teamcenter Enterprise) sites.
Reports administrator	**Creates or specifies an existing style sheet to define the look and format of the report.**	The XSL style sheets provide more control over the properties you want to display in the report and the display order. They also help define the look and format of the report. You can create new style sheets or use existing style sheets provided with Teamcenter and customize them for different types of reports. You can add more than one style sheet to a report. The end user can then select different style sheets while generating the report. For example, an end user can generate a summary report in HTML or spreadsheet format if you define both style sheets for the summary report.

Create an item report definition

Item reports are run on an individual item, such as an item revision. Each item report object is associated with a transfer mode, which is created in PLM XML/TC XML Export Import Administration.

1. Choose **File→Create Report**.

The system displays the Create Report Definition Template wizard.

2. Select **Item Report** and click **Next**.

3. Perform the following steps in the **Report Information** pane:

   a. In the **Report ID** box, type an ID or click **Assign** to automatically assign an ID.

   b. In the **Name** box, type a short description of the report definition.

   c. In the **Description** box, type a short description of the report definition.

   d. From the **Source** list, select the source of the report.

      The **Source** value is defined in the **Valid_ReportBuilder_Sources** preference.

   e. From the **Class** list, select a class, such as an item revision.

      If you selected **ItemRevision** as the class, **BOM Report** is enabled. Select the **BOM Report** flag if you want to generate a bill of materials report.

      > Note:
      >
      > Teamcenter manages data by grouping similar types of data into classes. Classes are created and edited in the Business Modeler IDE.

   f. From the **Transfer Mode** list, select a transfer mode for the report definition.

      A transfer mode object combines rules and property sets to define the context of the PLM XML import or export operation. It stores the translation intent in the PLM XML product. These objects are presented as context options when you import or export objects or system data. If the transfer mode object does not exist, you can create it with the PLM XML/TC XML Export Import Administration application.

   g. From the **Property Set** list to select a property set.

      > Note:
      >
      > To see the values of property sets, run the PLM XML/TC XML Export Import Administration application and select the name of the property set in the left pane. The values of the selected property set appear in the right pane.
      >
      > You can also create your own property set using the PLM XML/TC XML Export Import Administration application.

   h. From the **Report Format** list, select the format for the report, **PLMXML** or **TCXML**.

Choose **TCXML** to see all the Teamcenter attributes. Choose **PLMXML** to see attributes displayed in PLM XML industry standard names.

i. (For Active Workspace only) To allow users to generate the report asynchronously, select the **Run in Async** check box.

The **AsyncService** translator uses the Dispatcher framework to independently process asynchronous requests from the Teamcenter server using the Teamcenter SOA native C++ framework in the background mode.

j. (For Active Workspace only) To make this report template available in Active Workspace, select the **Use for Print** check box.

k. Click **Next**.

4. Perform the following steps in the **Reports Stylesheets** pane:

a. Select the style sheet you wish to use in the **Defined Stylesheets** pane, and click the **+** button to move it to the **Selected Stylesheets** pane.

> Note:
>
> To import a style sheet, click the **Import stylesheets** button, located under the **+** and **–** buttons in the **Report Stylesheets** pane.
>
> In the **Import Stylesheet** dialog box in the **Dataset Type** box, select a type for the new style sheet: **HTML**, **MSExcel**, **MSWord**, **Text**, or **Xml**. This option is required if the end user wants to save the report as a dataset at the time the report is run.

b. Click **Finish**.

The new report is added to the **Teamcenter Reports** folder.

c. Click **Close**.

5. To verify that the report definition works as desired, generate a report.

For example, in My Teamcenter in the rich client, choose one or more Teamcenter objects, choose **Generate Report** from the shortcut menu, and select your new item report.

Defining custom reports

Define a custom report

Custom reports are drawn up in specific cases, such as complex processing or calculations performed through custom code. These may also be used for data obtained from external sources. You can use custom reports especially when you want to use their output an input for these reports.

Role	Task	Description
Reports administrator	**Selects a process or method**.	You select a **Process** or **Method**.
Reports administrator	**Specifies parameters for the report**.	Select **Process** and specify the full path to the command line application that generates the output file. This is used as the source for the custom report. Select **Method** and **AWCustomAsItem** for custom ITKs. The **BOMLine** and **AuditDefinition** types are reserved methods to implement custom reports for compiling existing code from Teamcenter. You *cannot* use them to add new custom reports. The **AWCustomAsItem** type is also a reserved method, but you can use it to call a custom library that must be run with the Teamcenter server. You can write your own code and build it with the Teamcenter release.
Reports administrator	**Creates or specifies an existing style sheet to define the look and format of the report**.	The XSL style sheets provide more control over the properties you want to display in the report and the display order. They also help define the look and format of the report. You can create new style sheets or use existing style sheets provided with Teamcenter and customize them for different types of reports. You can add more than one style sheet to a report. The end user can then select different style sheets while generating the report. For example, an end user can generate a summary report in HTML or spreadsheet format

Role	Task	Description
		if you define both style sheets for the summary report.

Create a custom report definition

Custom reports address special cases such as complex processing or calculations done through custom code, or when data comes from external sources. Each custom report object is associated with a custom program. When a custom report is selected from a list, the server launches the program and the custom process.

1. Choose **File→Create Report**.

   The system displays the Create Report Definition Template wizard.

2. Select **Custom Report** and click **Next**.

3. Perform the following steps in the **Report Information** pane:

   a. In the **Report ID** box, type an ID or click **Assign** to automatically assign an ID.

   b. In the **Name** box, type a short description of the report definition.

   c. In the **Description** box, type a short description of the report definition.

   d. From the **Source** list, select the source of the report.

      The **Source** value is defined in the **Valid_ReportBuilder_Sources** preference.

      > Note:
      >
      > If you select **Visual Report** as your source, the report information pane changes.

   e. Select **Process** if you are using a process to run the custom report.

      A. In the **Process** box, type the full path to any command line application that generates the output file.

         This path is used to identify the operating system name for a process that generates an XML data file output. This is used as the source dataset for the custom report.

      B. In the **Output** box, type the output file name.

C. Click the **+** button to the right of the **Parameters** pane to add a new line, and type the parameter name and value.

f. Select **Method** to run a custom report against a type.

Select **Method** and **AWCustomAsItem** for custom ITKs.

The **BOMLine** and **AuditDefinition** types are reserved methods to implement custom reports for compiling existing code from Teamcenter. You *cannot* use them to create new custom reports.

The **AWCustomAsItem** type is also a reserved method, but you can use it to call a custom library that must be run with the Teamcenter server. You can write your own code and build it with the Teamcenter release.

A. From the **Type Name** list, select the type that is applicable to the custom report.

B. Click the **+** button on the right of the **Parameters** pane to add a new line and type the parameter name and value.

The parameters you specify here must match the parameters specified in the custom code you want to use.

4. (For Active Workspace only) To allow users to generate the report asynchronously, select the **Run in Async** check box.

The **AsyncService** translator uses the Dispatcher framework to independently process asynchronous requests from the Teamcenter server using the Teamcenter SOA native C++ framework in the background mode.

5. Click **Next**.

6. Perform the following steps in the **Reports Stylesheets** pane:

a. Select the style sheet you wish to use for the report by selecting it in the **Defined Stylesheets** pane and clicking the **+** button to move it to the **Selected Stylesheets** pane.

> Note:
>
> To import a style sheet, click the **Import stylesheets** button, located under the **+** and **–** buttons in the **Report Stylesheets** pane.
>
> In the **Import Stylesheet** dialog box, in the **Dataset Type** box, select a type for the new style sheet: **HTML**, **MSExcel**, **MSWord**, **Text**, or **Xml**. This option is required if the end user wants to save the report as a dataset at the time the report is run.

b. Click **Finish**.

The new report is added to the **Teamcenter Reports** folder.

c. Click **Close**.

7. To verify that the report definition works as desired, generate a report.

   For example, in My Teamcenter, in the rich client, choose **Tools→Reports→Custom Reports**, and select your new custom report.

Create sample custom reports

Create a sample custom report

The following is an example of how to create a custom report.

1. Extract the **wntx64\tc\sample.zip** file from the customer **patches** directory (for example, *Customer_Patches***Tc11.6.0\tc11.6.0_0910\wntx64\tc\sample.zip**), and copy the following files to a *custom* directory.

   - **build_custom_report_ITK.docx**
   - **compile.bat**
   - **link_custom_exits.bat**
   - **sample_custom_as_item_report_itk.cxx**

2. Set the **TC_ROOT**, **MSDEV_HOME**, **TC_INCLUDE**, **USER_INCLUDE**, and **TC_LIBRARY** environment variables. Refer the **build_custom_report_ITK.docx** file in the **sample.zip** you extracted.

3. Open the **sample_custom_as_item_report_itk.cxx** file in Microsoft Visual Studio.

   You can optionally change the library name, but not any other parameters. If you change the library name, you must recompile the files you have changed.

```
/*----------------------------------------------------------------------*/
//main entry point here

extern DLLAPI int libcusir_itk_main_fun( CUMITKINPUTBASE* p_inputs,int*
     n_columns,char***col_names, QRY_user_query_row_t**
     rows,char**report_path_name )
{
    CUMITKINPUT_B* p_rel_input = dynamic_cast<CUMITKINPUT_B*>( p_inputs );
    //this sample send back QRY_user_query_row_t and ask report framework
            to export and apply stylesheet
    if( strcmp( p_rel_input->method_name,"whereRefered_a") == 0 )
```

```
    {
        if(  p_rel_input->objs_num > 0 )
        {
            tag_t theobj = (p_rel_input->n_objs)[0];
            p_rel_input->dataset_created = false;//doesn't created
            dataset yet return generate_referenceBy_report
            ( theobj,n_columns,col_names,rows );
        }
        else
        {
            return ITK_ok;
        }
    }
    //this sample send back the html report directly
    else if( strcmp( p_rel_input->method_name,"whereRefered_b") == 0 )
    {
       if(  p_rel_input->objs_num > 0 )
        {
            tag_t theobj = (p_rel_input->n_objs)[0];
            p_rel_input->applyst = false;
            p_rel_input->dataset_created = false;//doesn't created
            dataset yet return generate_referenceBy_html_report
            ( theobj,report_path_name );
        }
    }
    else if( strcmp( p_rel_input->method_name,"customFun") == 0 )
    {// customer to impement the code here

    }

   return ITK_ok;
}
```

Where the *libname* is the library name that you compile and define in the **TC_custom_item_report_itk_library** Teamcenter preference.

For example, if you compile the **libcusir** sample library, then the main function name is **libcusir_itk_main_fun**.

The **whereRefered_a** and **whereRefered_b** methods have been already implemented in the **generate_referencedBy_report**.

If you have a custom function or method, you can implement it in the **customFun** method provided above.

4. From a command prompt, change to the *custom* directory where you have copied the **sample_custom_as_item_report_itk.cxx** file and compile the file.

Windows example:

```
.\compile -DIPLIB=none sample_custom_as_item_report_itk.cxx
```

When you compile the file, the system creates the **sample_custom_as_item_report_itk.obj** object file.

5. Link the **sample_custom_as_item_report_itk.obj** object file to the library in the *custom* directory.

   Windows example:

   ```
   .\link_custom_exits libcusir
   ```

   The system creates the **libcusir.dll** file.

6. Copy the **libcusir.dll** file to the **tc\bin** folder.

7. To allow Teamcenter to load the library, add the library name to the **TC_custom_item_report_itk_library** preference.

   Edit the preference to add the **libcusir:{method:whereRefered_b}** value as is used in the example.

8. To load the new library, restart Teamcenter.

Create the test_where_ref_a sample custom report

You can use the **whereRefered_a** method defined in the **sample_custom_as_item_report_itk.cxx** file to generate the **test_where_ref_a** sample custom report.

1. Create the **test_where_ref_a** custom report.

   a. In Report Builder, choose **File→Create Report**.

   b. Select **Custom Report** and click **Next**.

   c. Click **Assign** to automatically specify a report ID.

   d. Specify **test_where_ref_a** as the name for the report.

   e. In the **Class** drop down, choose **WorkspaceObject**.

   f. Select **Method** and choose **AWCustomAsItem** as the type name.

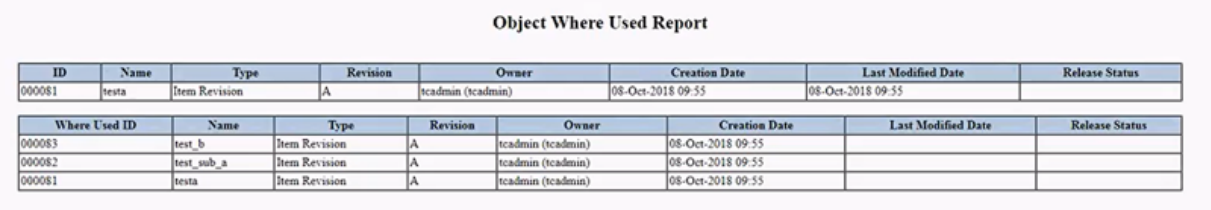
   g. In **Parameters**, specify the method name as **Method_Name** and the value as **whereRefered_a** as in the example used in Create a sample custom report, and click **Finish**.

2. Generate the **test_where_ref_a** sample custom report.

   a. Create a **test** folder in My Teamcenter and create an item.

   b. Right-click an item and choose **Generate Report**.

   c. Select the **test_where_ref_a** report you created to generate the following report:



Object Where Used Report

ID	Name	Type	Revision	Owner	Creation Date	Last Modified Date	Release Status
000081	testa	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55	08-Oct-2018 09:55	

Where Used ID	Name	Type	Revision	Owner	Creation Date	Last Modified Date	Release Status
000083	test_b	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55		
000082	test_sub_a	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55		
000081	testa	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55		

Create the test_where_ref_b sample custom report

You can use the **whereRefered_b** method defined in the **sample_custom_as_item_report_itk.cxx** file to generate the **test_where_ref_b** sample custom report.

1. Create the **test_where_ref_b** sample custom report.

   a. In Report Builder, choose **File→Create Report**.

   b. Select **Custom Report** and click **Next**.

   c. Click **Assign** to automatically specify a report ID.

   d. Specify **test_where_ref_b** as the name for the report.

   e. In the **Class** drop down, choose **ItemRevision**.

   f. Select **Method** and choose **AWCustomAsItem** as the type name.

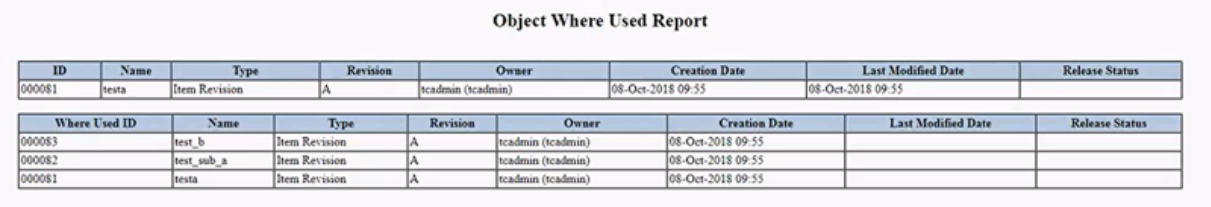
   g. In **Parameters**, specify the method name as **Method_Name** and the value as **whereRefered_b** as in the example used in **Create a sample custom report**, and click **Finish**.

2. Generate the **test_where_ref_b** sample custom report.

   a. Create a **test** folder in My Teamcenter and create an item.

   b. Right-click an item and choose **Generate Report**.

   c. Select the **test_where_ref_b** report you created to generate the following report:



Object Where Used Report

ID	Name	Type	Revision	Owner	Creation Date	Last Modified Date	Release Status
000081	testa	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55	08-Oct-2018 09:55	

Where Used ID	Name	Type	Revision	Owner	Creation Date	Last Modified Date	Release Status
000083	test_b	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55		
000082	test_sub_a	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55		
000081	testa	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55		

Create a report for sample custom code

You can use the **customFun method defined in the sample_custom_as_item_report_itk.cxx file** to generate the **test_my_custom** sample custom report.

> Tip:
>
> The following procedure is only an example. You can use your own custom code to generate a custom report.

1. Create an item in My Teamcenter.

2. Select the item, choose **Window→Show View→Print Object** and copy the UID.

3. Add the following custom code to the **sample_custom_as_item_report_itk.cxx** file.

   The following code is to generate a reference report for the object ID you copied. The object ID is hardcoded in this sample code.

```
{// customer to impement the code here

    tag_t sp_tag = NULLTAG;
    POM_string_to_tag( "BUWt7K4yLQtgjO",&sp_tag);

    p_rel_input->applyst = false;
    p_rel_input->dataset_created = false;//doesn't created dataset yet
    return generate_referenceBy_html_report( sp_tag,report_path_name );

}
```

   Where **BUWt7K4yLQtgjO** is the unique ID (UID) that you have copied.

4. Set the **TC_ROOT**, **MSDEV_HOME**, **TC_INCLUDE**, **USER_INCLUDE**, and **TC_LIBRARY** environment variables. Refer the **build_custom_report_ITK.docx** file in the **sample.zip** you extracted.

5. From a command prompt, change to the *custom* directory where you have copied the **sample_custom_as_item_report_itk.cxx** file and compile the file.

   Windows example:

```
.\compile -DIPLIB=none sample_custom_as_item_report_itk.cxx
```

When you compile the file, the system creates the **sample_custom_as_item_report_itk.obj** object file.

6. Link the **sample_custom_as_item_report_itk.obj** object file to the library in the *custom* directory.

   Windows example:

   ```
   .\link_custom_exits libcusir
   ```

   The system creates the **libcusir.dll** file.

7. Copy the **libcusir.dll** file to the **tc\bin** folder.

8. To allow Teamcenter to load the library, add the library name to the **TC_custom_item_report_itk_library** preference.

   Edit the preference to add the **libcusir:{method:whereRefered_b}** value as is used in the example.

9. To load the new library, restart Teamcenter.

Create the test_my_custom custom report

1. In Report Builder, choose **File→Create Report**.

2. Select **Custom Report** and click **Next**.

3. Click **Assign** to automatically specify a report ID.

4. Specify **test_my_custom** as the name for the report.

5. In the **Class** drop down, choose **WorkspaceObject**.

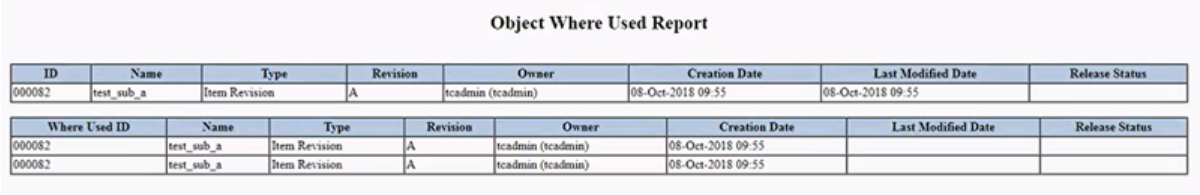
6. Select **Method** and choose **AWCustomAsItem** as the type name.

7. In **Parameters**, specify the method name as **Method_Name** and the value as **customFun** (**as specified in the sample_custom_as_item_report_itk.cxx file)**, and click **Finish**.

Generate the test_my_custom report

1. Right-click any object in My Teamcenter, choose **Generate Report**.

2. Select the **test_my_custom** report you created to generate the following report.



Object Where Used Report

ID	Name	Type	Revision	Owner	Creation Date	Last Modified Date	Release Status
000082	test_sub_a	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55	08-Oct-2018 09:55	

Where Used ID	Name	Type	Revision	Owner	Creation Date	Last Modified Date	Release Status
000082	test_sub_a	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55		
000082	test_sub_a	Item Revision	A	tcadmin (tcadmin)	08-Oct-2018 09:55		

Sample custom report definition

When you create a custom report definition, you associate the custom report object with a custom executable program. When you run the report in Teamcenter, Teamcenter runs the executable program from the full path specified in the report's process path and waits until the process is complete. The full file path specified as the output for the process is a *temporary file* used to share the output XML file between **query_xml.exe** and the report framework. The final report file is *not* specified as the **Output** in the **Report Generation Wizard**. The Report Builder framework takes the XML, applies the style sheet, and creates the dataset in the **New Stuff** folder in the Teamcenter rich client or in the context of the item it is run against.

Perform the following steps to run a sample custom report definition. This report invokes the **%TC_BIN%\query_xml.exe** file that takes an input file named **c6.xml**.

1. Save the following custom report definition as **TC_CUSTOM_RPT.xml** in a temporary directory named **TC_CUSTOM_RPT**. The directory name must be the same as the report name. For example, save the **TC_CUSTOM_RPT.xml** file in **D:\Temp\ TC_CUSTOM_RPT**

```
<?xml version="1.0" encoding="Windows-1252"?>
<ReportDefinition>
    <Id value="TC_CUSTOM_RPT"/>
    <name value="Query Report"/>
    <Description value=""/>
    <Class value=""/>
    <Type value="2"/>
    <QuerySource/>
    <PropertySet value=""/>
    <Stylesheets>
        <Stylesheet>
            <StylesheetType value="CrfHtmlStylesheet"/>
            <StylesheetName value="default_xml_template.xsl"/>
        </Stylesheet>
    </Stylesheets>
    <NoOfParameters value="7"/>
    <Params>
        <param_0 value="process"/>
        <param_1 value="output"/>
        <param_2 value="-f"/>
        <param_3 value="-u"/>
        <param_4 value="-p"/>
        <param_5 value="-g"/>
```

```
            <param_6 value="-o"/>
        </Params>
        <Values>
            <param_0 value="%TC_BIN%\query_xml.exe"/>
            <param_1 value="D:\workdir\test\c6_output.xml"/>
            <param_2 value="D:\workdir\test\c6.xml"/>
            <param_3 value="username"/>
            <param_4 value="username"/>
            <param_5 value="dba"/>
            <param_6 value="D:\workdir\test\c6_output.xml"/>
        </Values>
    </ReportDefinition>
```

2. Save the following process file as **c6.xml** and place it in the **D:\workdir\test** directory:

```
<?xml version="1.0" encoding="UTF-8"?>
<ImanQueryCommandFile site_name="arh" site_id="id">
    <ImanQueryCommand command="execute_tuples">
        <name value="General..."/>
        <query_input_parameter name="Name" value="*"/>
        <query_input_parameter name="Type" value="Item"/>
        <query_pff_post pffName="Admin - Objects By Status" />
    </ImanQueryCommand>
 </ImanQueryCommandFile>
```

3. Import the **TC_CUSTOM_RPT** custom report definition to Teamcenter from the Teamcenter command prompt, using a command similar to the following:

```
import_export_reports -import -u=user-name -p=password
-g=dba -stageDir=temp-directory -reportId=TC_CUSTOM_RPT
```

Use the following command:

```
import_export_reports -import -u=username -p=password
-g=dba -stageDir=”D:\Temp\” -reportId=TC_CUSTOM_RPT
```

4. Verify that the **Query Report** is displayed in the Report Builder application in the **Teamcenter Reports** folder.

5. In the Report Builder application, choose the **Query Report** and import the **default_xml_template.xsl** style sheet.

   The style sheet is located in the *TC_ROOT***tcdata\report_writer** directory.

6. In My Teamcenter in, choose **Tools→Report Builder Reports** and run the **Query Report** report without choosing a style sheet.

This report runs the **%TC_BIN%\query_xml.exe** file and generates the **C6_output.xml** report file.

7. In the rich client, in My Teamcenter, choose **Tools→Report Builder Reports** and run the **Query Report** report and choose a style sheet.

   This report runs the *TC_BIN***query_xml.exe** file and generates the **C6_output.xml** report file and applies the style sheet to generate the final HTML report.

Creating custom visual reports

What are visual reports?

You use visual reports to quickly display and analyze parts of an assembly in meaningful and well-organized ways by coloring parts based on the values of a property. You can create simple visual reports based on existing properties from a visual reporting client such as NX.

Using Report Builder, you can define complex visual reports by creating custom methods and defining a custom report.

You can execute these reports and view their results in a visual reporting client like NX.

The following visual reports are available in Report Builder:

> Note:
>
> You can only view the definitions of these reports in Report Builder. To view the results of these reports, use a visual reporting client like NX.
>
> For more information, see the *Visual Reporting* guide in the NX help.

- **Overall Status in Change Process**

  Shows status of parts currently in a change process. Colors all parts according to their overall status in a change process.

- **Overall Status in Workflow**

  Shows status of parts currently in a workflow process. Colors all parts according to their overall status in workflow.

- **Affected by Problem Report**

  Shows parts that are affected by a specified problem report. Colors all parts that are affected by the specified problem report by its relationship to the specified problem report.

- **Affected by Change Request**

Shows parts that are affected by a specified change request. Colors all parts that are affected by the specified change request by its relationship to the specified change request.

- **Affected by Change Notice**

  Shows parts that are affected by a specified change notice. Colors all parts that are affected by the specified change notice by its relationship to the specified change notice.

- **Has Higher Revision**

  Shows parts that have higher revisions. Colors all parts that have higher revisions than the one that is currently configured in the visual reporting client.

- **Rule Evaluation Result**

  Shows parts that satisfy a specified validation rule. Colors all parts whose validation results do not satisfy the specified validation rule.

- **Check-Mate Result**

  Shows parts that have Check-Mate results. Colors all parts according to their Check-Mate overall result.

  This report is not available in Report Builder. To use this report, import the **TC_9_0_00_VR_RPT_VAL_RESULT.xml** file into Teamcenter using the **import_export_reports** utility.

- **Requirements Validation Status**

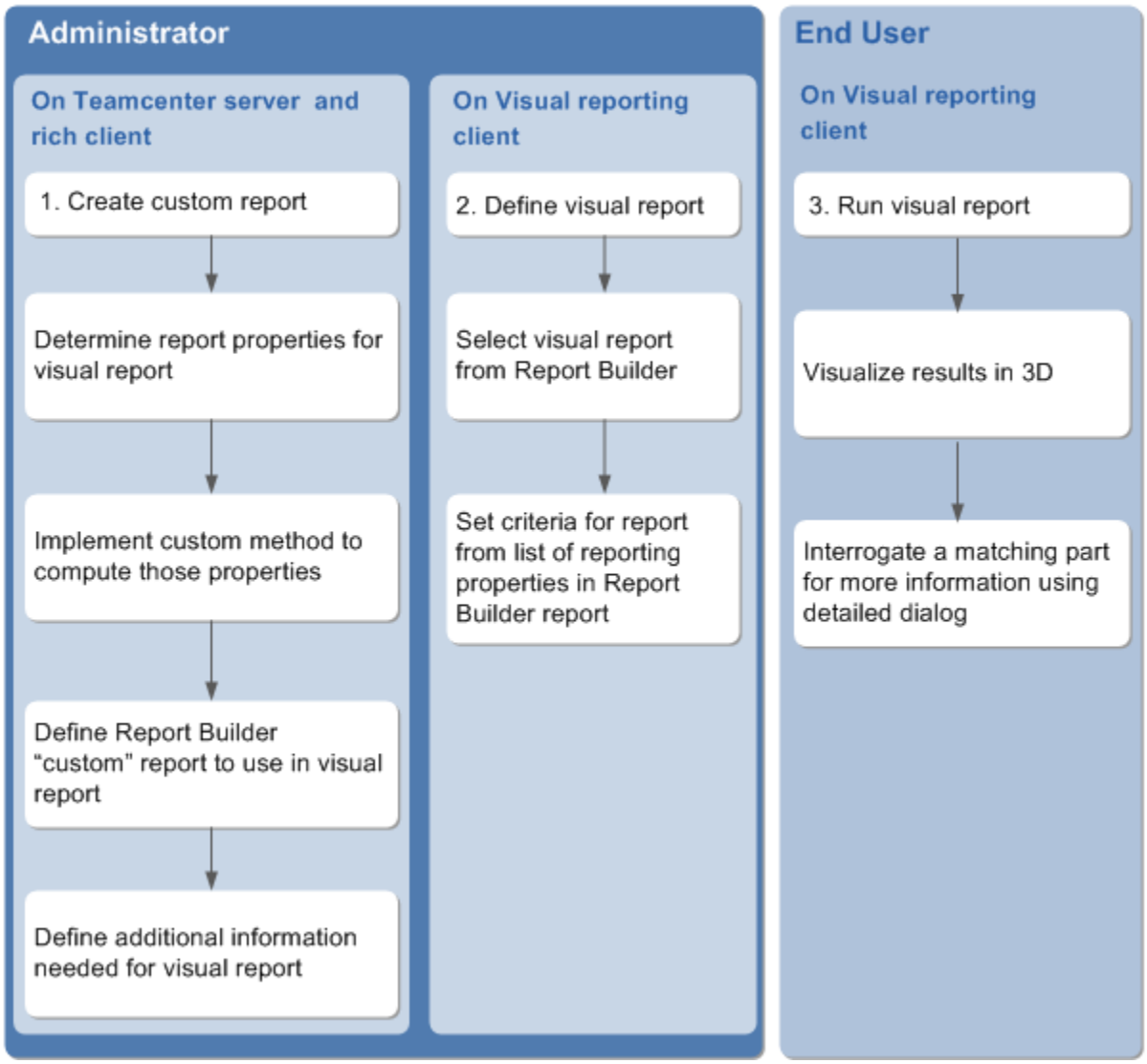
  Shows parts that have not fulfilled all their requirements. Colors all parts according to the requirement validation status.

  This report is not available in Report Builder. To use this report, import the **TC_9_0_00_VR_RPT_REQ_VAL_STATUS.xml** file into Teamcenter using the **import_export_reports** utility.

Creating complex visual reports

How to create complex visual reports?

The following diagram shows the process of creating complex visual reports.



Creating visual reports involves the following steps:

- Create custom method to compute the values of the visual report.
- Define a custom Report Builder report.
- Define the visual report in visual reporting client like NX.
- Run the report in a visual reporting client like NX.

For information about defining and running the visual report in NX, see the *Visual Reporting* guide in the NX help.

Creating a custom method

What are custom methods?

Methods are C/C++ functions that allow you to add functionality to Teamcenter.

Custom methods compute the set of properties supplied as input for the report using the context provided. Custom methods do not process closure rules and property sets that are specified in the report definition. The Report Builder framework processes the closure rules and property sets.

To create a custom method you must:

- Define the custom method.
- Build the custom method into Teamcenter.

When you create custom methods, use the following guidelines:

- When you generate a visual report from a visual reporting client, the custom method by itself does not gather the general parameters of the report. The general parameters are available in the report definition. Use the **SSR_get_property_descriptors** API in the custom method to get the general parameters.
- The custom method localizes property display names in the report output.

  Add report properties and custom property strings to the *TC_ROOT***\install\lang\textserver***language* **\report_customization_display_text_locale.xml** file, and use the **txtsvr** API to retrieve the localized strings.
- To retrieve the display names of Teamcenter properties, use the **AOM_UIF_ask_name** API.

  For more information about these Teamcenter Server APIs, see the *Integration Toolkit Function Reference* on Support Center.

Define a custom method

> Note:
>
> A sample custom method for visual reports is available in the *TC_ROOT***/sample/visual_report** directory.

The custom method code has the following parts:

- Method registration
- Execution method

Method registration

Methods must be registered against a type. For visual reports, the methods are registered against the **ReportDefinition** type. Registration is done in the initialization method for the **ReportDefinition** type.

Methods are registered using **METHOD_register_method**.

```
TCCORE_API int METHOD_register_method     (    const char * type_name
                                               const char * msg_name
                                               METHOD_function_t base_action
                                               TC_argument_list_t *
user_data
                                               METHOD_id_t * method )
```

You must follow the following conventions for custom method for visual reports:

- **type_name**

  Register method against the type **ReportDefinition**.

- **msg_name**

  Corresponds to the method name. Prefix the **msg_name** with **CrfGenerateVisualReport**.

  Example: For a custom method named **CostReport**, the **msg_name** value will be **CrfGenerateVisualReportCostReport**.

- **base_action**

  Corresponds to the custom function.

- **user_data**

  Value if **NULL**.

- **method**

  Corresponds to the output parameter.

Execution method

The execution method will have input parameters and computing code.

Signature of custom method

```
int sampleVisualReport (METHOD_message_t *msg, va_list args)
```

Input parameters

- **msg**

  The reference to the tag of a report definition.

- **args**

  The constituents of the **va_list** arguments.

- **int numObjects**

  The number of item revisions that are the context of the report.

- **tag_t *objects**

  The list of item revisions that are the context of the report.

- **int num_reporting_property**

  The number of reporting properties to be computed for the report.

- **parameter_descriptors_t *rpd**

  The list of reporting properties to be computed for the report.

- **int num_search_parameter**

  The number of search criteria or values to be used in refining the results of the report.

- **parameter_descriptors_t *spd**

  The list of search criteria or values to be used in refining the results of the report. Typically these parameters are a list of strings to be used to fill in values for a query that needs to be executed on the server.

- **int num_general_parameter**

  The number of any other parameters required by the visual report.

- **parameter_descriptors_t *gpd**

  The list of any other parameters required by the visual report.

- **int num_additional_parameter**

The number of any parameter that does not fit the previously described parameter types.

- **parameter_descriptors_t *apd**

  The list of any parameter that does not fit the previously described parameter types.

The following are the primary ITK APIs used in the custom methods.

For more information about ITK APIs and other APIs which support additional functions, see the *Integration Toolkit Function Reference* on Support Center.

- **SSR_create_visual_report_objects**
- **SSR_create_report_property_objects**
- **SSR_set_report_property_data**
- **SSR_create_related_objects**
- **SSR_set_related_object_data**
- **SSR_create_property_objects**
- **SSR_set_property_data**

Build the custom method in Teamcenter

You can build a custom method in the Teamcenter Server using a custom library.

Defining visual reports

Define the visual report in Teamcenter

> Note:
>
> Ensure you have added **Visual Report** as a value in the **Valid_ReportBuilder_Sources** preference.

1. Choose **File→Create Report**.

   The system displays the Create Report Definition Template wizard.

2. Select **Custom Report** and click **Next**.

3. Perform the following steps in the **Report Information** pane:

   a. In the **Report ID** box, type an ID or click **Assign** to automatically assign an ID.

b. In the **Name** box, type a short description of the report definition.

c. In the **Description** box, type a short description of the report definition.

d. From the **Source** list, select **Visual Report**.

The **Source** value is defined **Valid_ReportBuilder_Sources** in the preference.

4. From the **Closure Rule** list, select a closure rule to process the query definition. Closure rules specify how the data structure is traversed by specifying which relationships are of interest.

> Note:
>
> To see the values of closure rules, run the PLM XML/TC XML Export Import Administration application and select the closure rule in the left pane. The values of the selected closure rule are displayed in the right pane. You can also create your own closure rule using the PLM XML/TC XML Export Import Administration application.

5. From the **Property Set** list, select a set of properties to use for the report.

> Note:
>
> To see the values of property sets, run the PLM XML/TC XML Export Import Administration application and select the name of the property set in the left pane. The values of the selected property set appear in the right pane. You can also create your own property set using the PLM XML/TC XML Export Import Administration application.

6. Type the name of the custom method in the **Custom Method** box.

7. Click the **+** button on the right of the **Parameters** pane to add a new line, and type or select the parameter name, display name, parameter type, and value.

The following parameter types are available in their separate tabs:

- **Report Properties**

  Specifies the properties to be returned by the visual report.

- **Search Criteria**

  Specifies the context objects that should be filtered by Report Builder while delivering the output.

- **General**

  Any other parameters required by the visual report.

8. Click **Next**.

9. Perform the following steps in the **Reports Stylesheets** pane:

   a. Select the style sheet to use for the report by selecting it in the **Defined Stylesheets** pane and clicking the **+** button to move it to the **Selected Stylesheets** pane.

   > Note:
   >
   > Style sheets for visual reports are not available by default. You must create your own style sheets and import it in Teamcenter or use the available style sheets.
   >
   > To import a style sheet, click the **Import stylesheets** button  , located under the **+** and **–** buttons in the **Report Stylesheets** pane.
   >
   > The **Import Stylesheet** dialog box appears.
   >
   > In the **Dataset Type** box, select a style sheet type for the new style sheet: **HTML**, **MSExcel**, **MSWord**, **Text**, or **Xml**. This option is required if the end user wants to save the report as a dataset at the time the report is run.

   b. Click **Finish**.

      The new report is added to the **Teamcenter Reports** folder.

   c. Click **Close**.

Define the visual report in the visual reporting client

You must define the visual report in the visual reporting client to generate the reports.

For information about defining visual report in NX, see the *Visual Reporting* guide in the NX help.

Debug visual reports

> Note:
>
> Ensure you have added **Visual Report** as a value in the **Enable_ReportBuilder_Reports_By_Source** preference.

1. In My Teamcenter in the rich client, choose **Tools→Reports→Report Builder Reports**.

   The system displays the Report Generation Wizard.

   If no reports are listed, no reports have been created by your system administrator.

2. Select a custom report.

3. Click **Next**.

4. In the **Report Information** pane, add or remove parameter values:

   a. Click the **+** button in the right of the **Parameters** pane to add a new line, and type the parameter name and value.

   b. Select a parameter value and click the **+** button in the right of the **Parameters** pane to remove a parameter.

5. If you want to choose the format for the report, select a style sheet from the **Report Stylesheets** list.

   > Note:
   >
   > Style sheets for visual reports are not available by default. You must create your own style sheets and import it in Teamcenter or use the available style sheets.

6. If you want to save the report as a file, select the **Create Dataset** check box and type a name in the **Dataset Name** box.

7. Click **Finish**.

   The report is displayed in your default XML application.

   If you saved the report as a dataset, the report is saved in your default **Home** folder, for example, **Newstuff**.

Define the look and format of the report using style sheets

Create style sheets to define the look and format of reports

Each report consists of a report definition and an associated XSL style sheet. The report definition defines the properties to fetch from Teamcenter and the rules for importing the data. The XSL style sheets provide more control over the properties you want to display in the report and the display order. They also help define the look and format of the report.

As a reports administrator (with or without DBA privileges), you can create new style sheets or use existing style sheets provided with Teamcenter and customize them for different types of reports. You can add more than one style sheet to a report. The end user can then select different style sheets while generating the report. For example, an end user can generate a summary report in HTML or spreadsheet format if you define both style sheets for the summary report.

Tip:

For more information about sample Teamcenter style sheets, see the *TC_DATA***crf****Resources** directory. You can create a copy of any of the XSL style sheets and customize it for your report definition.

Modify an existing report to add a new property and change its associated style sheet

Let us assume that you want to modify the default **Admin - Item Ownership** summary report and add a new **Sequence ID** property for the report. By default, this report displays columns containing information such as the item ID, revision ID, item name, item type, and owning user.

The report definition for this summary report consists of:

- **Admin - Item Ownership** query source for extracting attributes from a target object in the Teamcenter database, using specific search criteria.
- **TC_2007_00_CrfReports** property set for including additional properties that are not part of the query search in the report.

  For example, the **Release Status** and **Date Released** properties in the report are not part of the search criteria in the query source.

Admin - Item Ownership

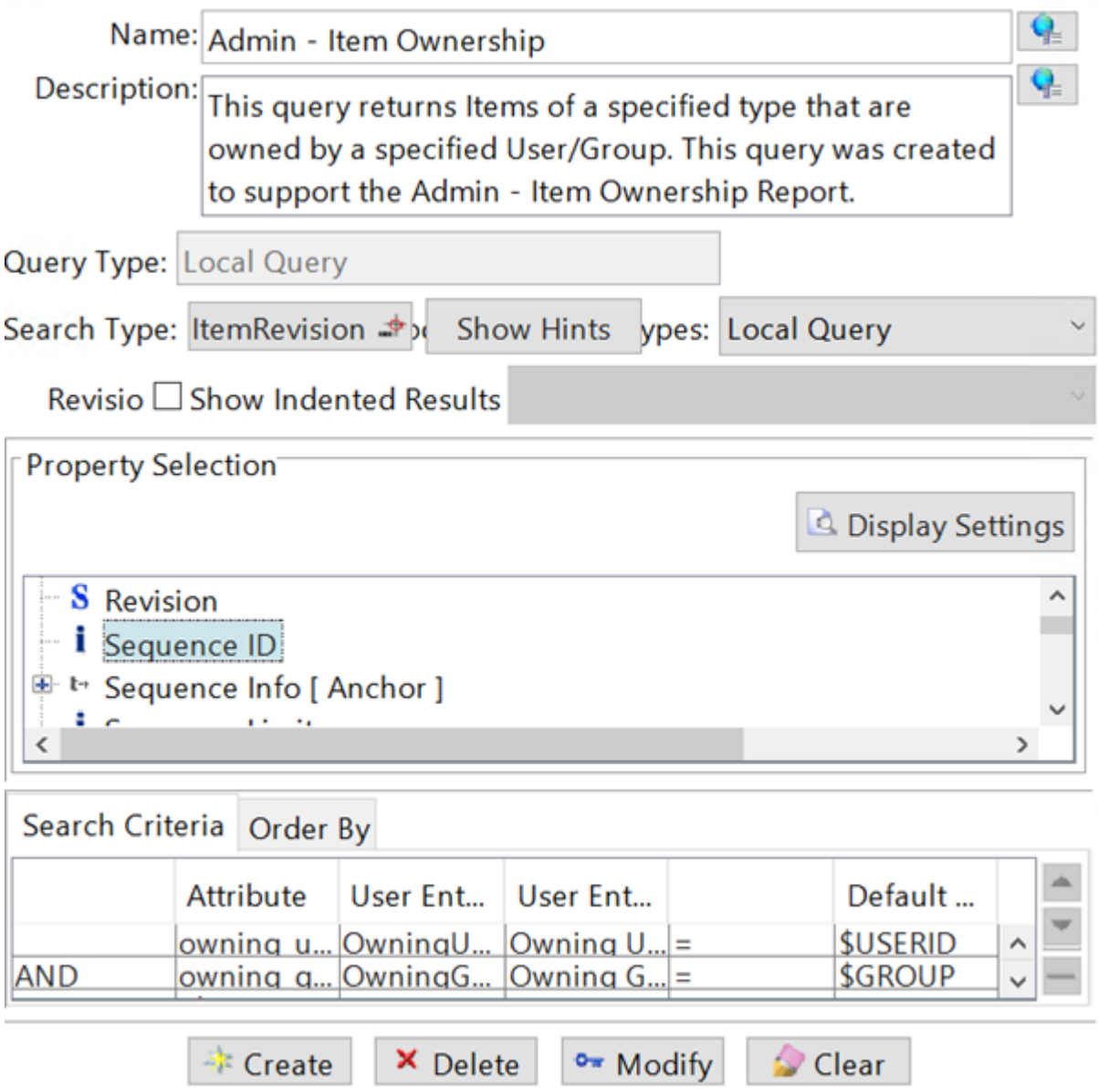
3 Search Criteria Used

Item ID	Revision ID	Name	Type	Release Status	Date Released	Owning User	Owning Group	Date Created	Last Modifying User	Last Modified Date
machining_knowledge	A	machining_knowledge	ItemRevision			infodba	dba	2020-12-29T09:44:32	infodba	2020-12-29T09:44:32
standard_thread_inch	A	standard_thread_inch	ItemRevision			infodba	dba	2020-12-29T09:44:20	infodba	2020-12-29T09:44:20
cap_screw_cs_inch	A	cap_screw_cs_inch	ItemRevision			infodba	dba	2020-12-29T09:44:20	infodba	2020-12-29T09:44:20
mill_planar_target_metric	A	mill_planar_target_metric	ItemRevision			infodba	dba	2020-12-29T09:44:28	infodba	2020-12-29T09:44:28
MillTurn_Express_target_inch	A	MillTurn_Express_target_inch	ItemRevision			infodba	dba	2020-12-29T09:44:28	infodba	2020-12-29T09:44:29
014505	A	ForOccEff_part48	ItemRevision	TCM Released	2015-04-04T00:33:25	infodba	dba	2015-04-04T00:05:15	infodba	2015-04-04T00:33:25
014483	A	ForOccEff_part30	ItemRevision			infodba	dba	2015-04-03T23:52:23	infodba	2015-04-03T23:52:23
014479	A	ForOccEff_part26	ItemRevision			infodba	dba	2015-04-03T23:51:54	infodba	2015-04-03T23:51:54
014478	A	ForOccEff_part25	ItemRevision			infodba	dba	2015-04-03T23:51:48	infodba	2015-04-03T23:51:48

1. Modify the query source to add the new property.

   a. Open the **Admin - Item Ownership** query source in Query Builder.

   b. To add the attribute to the search criteria, in the **Attribute Selection** area, double-click the **Sequence Id** attribute, and use the default values.

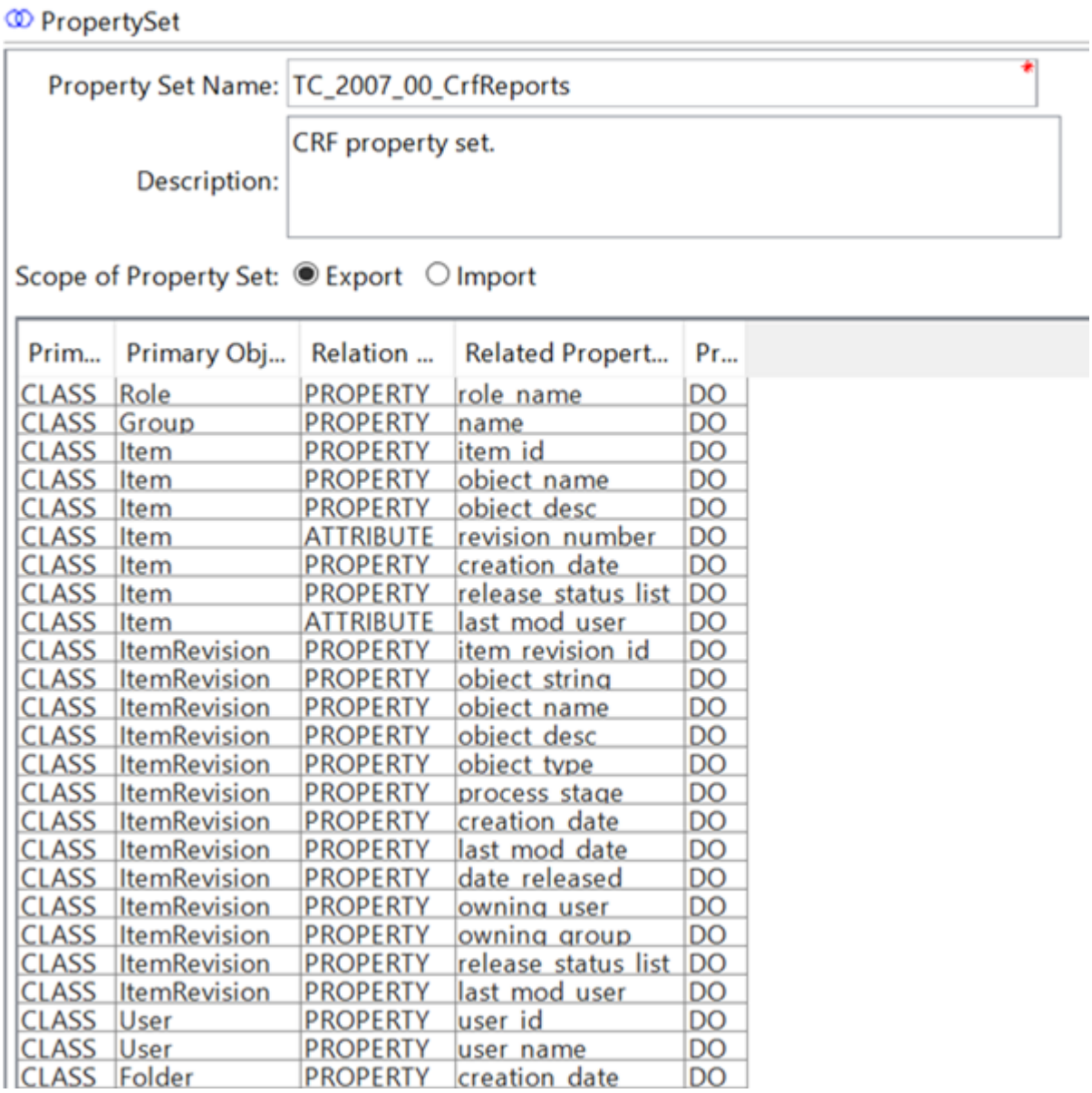


c. To modify the query source, click **Modify**.

2. Modify the property set.

a. Open the **TC_2007_00_CrfReports** property set in PLM XML/TC XML Export Import Administration.

b. To add an object, click the **Add** button and select **CLASS** for primary object class, type **ItemRevision** for primary object, select **ATTRIBUTE** for relation type, type **sequence_id** for a related property or object, and select **DO** as the property action type.



Prim...	Primary Obj...	Relation ...	Related Propert...	Pr...
CLASS	Role	PROPERTY	role_name	DO
CLASS	Group	PROPERTY	name	DO
CLASS	Item	PROPERTY	item_id	DO
CLASS	Item	PROPERTY	object_name	DO
CLASS	Item	PROPERTY	object_desc	DO
CLASS	Item	ATTRIBUTE	revision_number	DO
CLASS	Item	PROPERTY	creation_date	DO
CLASS	Item	PROPERTY	release_status_list	DO
CLASS	Item	ATTRIBUTE	last_mod_user	DO
CLASS	ItemRevision	PROPERTY	item_revision_id	DO
CLASS	ItemRevision	PROPERTY	object_string	DO
CLASS	ItemRevision	PROPERTY	object_name	DO
CLASS	ItemRevision	PROPERTY	object_desc	DO
CLASS	ItemRevision	PROPERTY	object_type	DO
CLASS	ItemRevision	PROPERTY	process_stage	DO
CLASS	ItemRevision	PROPERTY	creation_date	DO
CLASS	ItemRevision	PROPERTY	last_mod_date	DO
CLASS	ItemRevision	PROPERTY	date_released	DO
CLASS	ItemRevision	PROPERTY	owning_user	DO
CLASS	ItemRevision	PROPERTY	owning_group	DO
CLASS	ItemRevision	PROPERTY	release_status_list	DO
CLASS	ItemRevision	PROPERTY	last_mod_user	DO
CLASS	User	PROPERTY	user_id	DO
CLASS	User	PROPERTY	user_name	DO
CLASS	Folder	PROPERTY	creation_date	DO

c. To modify the property set, click **Modify**.

3. Copy an existing style sheet, modify it to add new properties, import it, and assign the style sheet to a report.

   a. Open the **Admin - Item Ownership** report in Report Builder and verify the name of the associated style sheet. The style sheet is called **admin_ownership_html.xsl**.

   b. Browse to the *TC_DATA***crf****Resources** directory and create a copy of the **admin_ownership_html.xsl** style sheet.

   c. Open the copy of the **admin_ownership_html.xsl** style sheet in an XML editor.

d. (Optional) To replace the logo, search for the **printlogo** element, look for the image source, and specify another logo source.

e. Add the new **sequence_id** property to the header cell in the table of the HTML report.

To add this property to the **Sequence ID** header column, search for the **sorttable** element, and add the new column name.

> Tip:
>
> The **<th>** element is used in XSL for header cells and they contain header information.

Example:

```
<body BGCOLOR="#FFFFFF" link="#0000FF" vlink="#660066">
  <div style="text-align:center">
    <table id="sorttable" border="1" style="width: 60%;margin:auto">
      <tr align="center" bgcolor="#B8CFEP">
        <!-- 1st row -->
        <th onclick="sortCells(0)" title="click here to sort">
                  Item ID</th>
        <th onclick="sortCells(1)" title="clich here to sort">
                  Revision ID</th>
        <th onclick="sortCells(2)" title="click here to sort">
                  Name</th>
        <th onclick="sortCells(3)" title="click here to sort">
                  Type</th>
        <th onclick="sortCells(4)" title="click here to sort">
                  Release Status</th>
        <th onclick="sortCells(5)" title="click here to sort">
                  Date Released</th>
        <th onclick="sortCells(6)" title="click here to sort">
                  Owning User</th>
        <th onclick="sortCells(7)" title="click here to sort">
                  Owning Group</th>
        <th onclick="sortCells(8)" title="click here to sort">
                  Date Created</th>
        <th onclick="sortCells(9)" title="click here to sort">
            Last Modifying User</th>
        <th onclick="sortCells(10)" title="click here to sort">
            Last Modified Date</th>
       </tr>
```

Add the following before the **</tr>** tag:

```
<th onclick="sortCells(10)" title="click here to sort">Sequence
ID</th>
```

f. To specify the variable name in the sorttable table, search for **variable name="occele"**, and add the sequence ID as a variable.

A variable is used to store a value especially when the value of the target source object is parsed from a long directory path. It can be a node or a final display name, such as the object name.

Example:

```
<xsl:template name="creCLext1">
<xsl:param name="occele"/>
<!--<xsl:variable name="occele" select="/plm:PLMXML/plm:ProductRevision
[@id=$occid]"/>-->
    <xsl:variable name="itemId" select="$occele/plm:UserData/plm:
        UserValue[@title='item_id']/@value"/>
    <xsl:variable name="itemRevId" select="$occele/plm:UserData/plm:
        UserValue[@title='item_revision_id']/@value"/>
    <xsl:variable name="itemName" select="$occele/@name"/>
    <xsl:variable name="objectType" select="$occele/plm:UserData/plm:
        UserValue[@title='object_type']/@value"/>
    <xsl:variable name="lastModDate" select="$occele/plm:UserData/plm:
        UserValue[@title='last_mod_date']/@value"/>
    <xsl:variable name="createdDate" select="$occele/plm:UserData/plm:
        UserValue[@title='creation_date']/@value"/>
    <xsl:variable name="userRef" select="substring-after($occele/
plm:UserData/
        plm: UserValue[@title='owning_user']/@dataRef,'#')"/>
    <xsl:variable name="userName"
        select="/plm:PLMXML/plm:User[@id=$userRef]/@userId" />
    <xsl:variable name="modifyingUserRef" select="substring-after($occele/
        plm:UserData /plm:UserValue[@title='last_mod_user']/@dataRef,'#')"/>
    <xsl:variable name="modifyingUserName" select="/plm:PLMXML/
        plm:User[@id=$userRef]/@userId" />
    <xsl:variable name="groupRef" select="substring-after($occele/
        plm:UserData/plm: UserValue[@title='owning_group']/
        @dataRef,'#')"/>
    <xsl:variable name="groupName" select="/plm:PLMXML/plm:Organisation
        [@id=$groupRef]/@name" />
```

Add a new **xsl: variable name** element for the sequence ID:

```
<xsl:variable name="sequenceid" select="$occele/plm:UserData/plm:
      UserValue[@title='sequence_id']/@value"/>
```

g. Create a standard cell to display data in the table for the new property.

To create a data cell for sequence ID, add the following before the **</tr>** tags in the **sorttable** table.

```
<td valign="top">
        <xsl:value-of select="$sequenceid"/>
        <xsl:text disable-output-escaping="yes">&amp;nbsp;</
xsl:text>
      </td>
```

> Tip:
>
> The **<td>** element is used for standard or data cells in XSL.

h. Save the XSL style sheet.

i. To import the style sheet, open the **Admin - Item Ownership** report in Report Builder, and in the **Report Stylesheets** area, click **Import Stylesheet**.

To specify the location of the new style sheet, select **HTML** as the **Dataset Type**, and then specify the location.

> Tip:
>
> If you want end users to generate the report as a spreadsheet and you have created a style sheet for Microsoft Excel, select **MSExcel** as the **Dataset Type**.

j. To assign the style sheet to the **Admin - Item Ownership** report, in the **Define Stylesheets** list, select the style sheet you imported, and add it to the **Selected Stylesheets** list for the specific report.

Create custom style sheets for various types of reports

You can create custom style sheets for various types of reports. For more information about sample Teamcenter style sheets, see the *TC_DATA***crf****Resources** directory. You can create a copy of any of the XSL style sheets and customize it for your report definition.

You can write the XSL function in the style sheet and refer some of the functions that are executed by Teamcenter. For example, open the *TC_DATA***crf****Resources****l10n_ps_bom_report_excel.xsl** file. This file contains the **setFile** external function:

```
<xsl:stylesheet version="1.0" xmlns:crf="http://ExternalFunction.setFile">
```

You can use the following functions for the custom style sheets you create.

setFile

```
/*
 *  Description: Performs the file copy from the source location to
the target location.
 *
 *  Function Return: void
 */
void Crf::setFile( char *setFileName, char * targetDir  );
```

getDate

```
/*
 *  Description: Gets the current date and the time with the time
zone.
 *
 *  Function Return: Current date and time.
 *
 */
char* Crf::getDate();
```

getDisplayName

```
/*
 * Description: Gets the localized name of the given key.
 *
 * Function Return: localized display name.
 *
 */
char* Crf::getDisplayName(char *key);
```

getPrefValue

```
/*
 * Description: Gets the Site level value of the given preference
 *
 * Function Return: Preference value.
 *
 */
char* Crf::getPrefValue(char *pref_name );
```

getCurrentDir

```
/*
 *  Description: Gets the current directory.
 *
 *  Function Return: Current directory path.
 *
```

```
 */
char* Crf::getCurrentDir();
```

getPropertyDisplayName

```
/*
 * Description: Gets the display name of the property in a given
locale.
 *
 * Function Return: Property display name.
 *
 */
char* Crf::getPropertyDisplayName(char* type_name, char* prop_name,
char* locale);
```

getPropertyDisplayValue

```
/*
 * Description: Gets the display value of a property in a given
locale.
 *
 * Function Return: Property display value.
 *
 */
char* Crf::getPropertyDisplayValue(char* object_uid, char*
prop_name, char* locale);
```

getTypeDisplayName

```
/*
 * Description: Gets the localized display name of a given type.
 *
 * Function Return: Type display name.
 *
 */
char* Crf::getTypeDisplayName(char* type_name, char* locale);
```

getReportDisplayText

```
/*
 * Description: Gets the localized display text of a given key.
 *
 * Function Return: Display text.
 *
 */
char* Crf::getReportDisplayText(char *key, char* locale);
```

writeDocument

```
/*
 * Description: Writes to the document using exsl:document extension
 *
 * Function Return: void.
 *
 */
/*
void Crf::writeDocument(const char* path,const char* content);
```

setRowIndex

```
/*
 * Description: Sets index value for a Row. Assigns a value to a
rowIndex variable.
 *
 * Function Return: void.
 *
 */
void Crf::setRowIndex( const int rowidx );
```

addRowIndex

```
/*
 * Description: Adds a given number to a row index variable.
 *
 * Function Return: void.
 *
 */
int Crf::addRowIndex(const int num);
```

setColIndex

```
/*
 * Description: Sets the index value for a column for a colIndex
variable.
 *
 * Function Return: void.
 *
 */
void Crf::setColIndex( const char* colidx );
```

addColIndex

```
/*
 * Description: Adds a given number to the column index variable.
 *
 * Function Return: void.
 *
```

```
 */
char* Crf::addColIndex(const char* colNum);
```

addMerger

```
/*
 * Description: Creates and adds merge information into mergers
vector.
 *
 * Function Return: void.
 *
 */
void Crf::addMerger( const char* startColIdx,const char*
startRowIdx,const char* colNum,const char* rowNum);
```

writeMerger

```
/*
 * Description: Writes merge information from the mergers vector.
 *
 * Function Return: final merged string
 *
 */
String Crf::writeMerger( );
```

writeAwUrl

```
/*
 * Description: Writes Active Workspace URL value from preference
'CRF_REPORT_AW_OBJECT_URL'
 *
 * Function Return: URL value from preference.
 *
 */
String Crf::writeAwUrl( const char* uid )
```

Configuring reports with a watermark

Defining reports with a watermark

You can optionally define a watermark for reports to identify copyright information and the status of the generated report. For example, you can add a watermark to specific types of reports to display your company name and the report status, such as *confidential*. Reports with watermarks help protect company proprietary information from unauthorized disclosure. The watermark is defined as a text entity in the style sheet associated with the report type. You can change the text, font size, location, and orientation of the watermark in a report by editing the style sheet.

By default, change notice objects and workspace objects reports have a predefined watermark with the text entity **Confidential**. You can also use the watermark in other types of reports by editing the style sheets associated with those reports.

Modify report definitions

Before defining a report definition, you must create a:

1. Query source to create and maintain customized searches for objects in the Teamcenter database.

   You use Query Builder to create a query source.

2. Closure rule to specify the structure for processing related items.

3. Property set to specify additional output properties.

4. Transfer mode to combine closure rules and property sets to define the context of the PLM XML import or export operation.

   You use PLM XML/TC XML Export Import Administration to create transfer modes.

The **Change Notice Objects Report** and **WSO Object Report** report definitions are provided with default query source, closure rules, property sets, and transfer modes as follows:

Rule, set, or mode	Change Notice Objects Report	WSO Object Report
Closure Rule	**CMClosureRuleDefault**	**WSOClosureRuleDefault**
Property Set	**CMPropertySetDefault**	**WSOPropertySetDefault**
Transfer Mode	**CMTransferModeDefault**	**WSOTransferModeDefault**

1. In Report Builder, open the **Teamcenter Reports** folder.

2. Select the **WSO Object Report** or the **Change Notice Objects Report** report definition and click the **Report Data** tab.

3. Change the information in the boxes as appropriate.

4. For item reports, select one or both options as appropriate.

   - **Run in Async** to allow users to print the report asynchronously. Clear this option to allow users to print reports synchronously.

Active Workspace users can continue to work on other tasks while the system generates the report in the background.

The **AsyncService** translator uses the Dispatcher framework to independently processes asynchronous requests from the Teamcenter server, using the Teamcenter SOA native C++ framework in the background mode.

- **Use for Print** to make this template available for Active Workspace **Print** functionality.

  > Note:
  >
  > This option is not enabled for **Change Notice Objects Report** report definition by default. Ensure that you select this option for this report definition.

5. To make this report template available to specific users or groups, select users or groups from the **Defined Group** box and add them to the **Selected Group** box.

   > Tip:
   >
   > The selection of groups impacts the visibility of the report template for Active Workspace **Print** functionality only.
   >
   > If you do not select a specific user or group, the system automatically assigns the template to all users and groups.

6. Click **Modify** to save your changes.

Customize the watermark by editing the style sheet associated with the report definition

You can customize the watermark for the **Change Notice Objects Report** and **WSO Object Report** report definitions by editing the associated style sheet.

- **Change_object_Sample_template.xsl** file to customize the watermark for **Change Notice Objects Report**.
- **WSO_Object_Sample_Template.xsl** file to customize the watermark for **WSO Object Report**.

You can change the content, font size, location, and orientation of the watermark.

1. Open and edit a style sheet associated with a report definition.

   a. In Report Builder, open the **Teamcenter Reports** folder.

   b. Select the report definition for which you want to customize the watermark, and click the **Report Data** tab.

c. In the **Defined Stylesheets** box, select the style sheet you want to edit, and click the **Open the selected stylesheet** icon.

Teamcenter exports the style sheet to a temporary directory.

d. Browse to the directory where you exported the style sheet, and open the XSL file in an XML editor.

2. Change the text or the content of the watermark.

a. Locate the **watermark_print** attribute.

Example:

```
<div id="watermark_print"  align="center">
      <p>Confidential&#169;</p>
    </div>
```

The default watermark text is *Confidential*.

b. Change as appropriate.

You can include a hardcoded string or add one or more property values of the objects you want to generate for the report.

3. Change the font size of the watermark.

The default font size is **25pt**.

a. Locate the **font-size: 25pt** attribute in the **style type** tag.

```
<style type="text/css">
          .myOtherTable { background-color:#C2C2C2;border-collapse:
                          collapse;font-size:15px;border:2; }
          .myOtherTable th { background-color:#ECECEC;color:white;
                          width:10%; border-bottom:2px dotted #BDB76B;}
          .myOtherTable td, .myOtherTable th { padding:5px;border:2;
                          border-bottom:2px; dotted #BDB76B;}
          .myOtherTable td { border-bottom:2px dotted #BDB76B; }
            .myOtherTable th { border-bottom:2px solid #BDB76B; }
          .pagebreak {page-break-before: always;}
          @media print {
          #watermark_print {
          display: block;
          position: fixed;
          right: 0;
          z-index: 5;
```

```
color: #B81C1C;
font-size: 25pt;
```

b. Change the font size as appropriate.

4. Change the location of the watermark.

You can change the **width** and **height** attributes of the watermark. By default, these attributes are set to **100%** and **50%**, respectively.

a. Locate the **width** and **height** attributes in the **style type** tag.

```
<style type="text/css">
          .myOtherTable { background-color:#C2C2C2;border-
collapse:
                          collapse;font-size:15px;border:2; }
          .myOtherTable th { background-
color:#ECECEC;color:white;
                          width:10%; border-bottom:2px dotted
#BDB76B;}
          .myOtherTable td, .myOtherTable th
{ padding:5px;border:2;
                          border-bottom:2px; dotted #BDB76B;}
          .myOtherTable td { border-bottom:2px dotted #BDB76B; }
            .myOtherTable th { border-bottom:2px solid #BDB76B; }
          .pagebreak {page-break-before: always;}
          @media print {
          #watermark_print {
          display: block;
          position: fixed;
          right: 0;
          z-index: 5;
          color: #B81C1C;
          font-size: 25pt;
          width: 100%;
          height: 50%;
</style>
```

b. Change the **width** and **height** attributes as appropriate.

5. Change the orientation of the watermark.

The **transform: rotate** attribute determines the orientation of the watermark.

The transform attribute applies a transformation and allows you to rotate an element. The attribute varies depending on the type of browser you use. For example, the **-webkit-transform: rotate** attribute is used for Mozilla Firefox, while the **-ms-transform:rotate** attribute is used to apply transformation in Internet Explorer.

a. Locate the **transform: rotate** attribute in the **style type** tag.

```
<style type="text/css">

          @media print {
          #watermark_print {
          position:absolute;
          bottom:0;
          width:100%;
          height:60px;
          background:#6cf;
          display: block;
          position: fixed;
          right: 0;
          z-index: 5;
          color: #B81C1C;
          font-size: 25pt;
          height: 50%;
          margin: 0;
          top:350px;
          left:125px;
          margin: auto;
          margin-right: auto;
          margin-left: auto;
          -webkit-transform: rotate(-40deg); /* FF3.5+ */
          -moz-transform: rotate(-40deg); /* Opera 10.5 */
          -o-transform: rotate(-40deg);  /* Saf3.1+, Chrome */
          -ms-transform:rotate(-40deg);
          transform: rotate(-40deg);
          filter:  progid:DXImageTransform.Microsoft.
                           BasicImage(rotation=3);  /* IE6,IE7 */
          -ms-filter: "progid:DXImageTransform.Microsoft.
                           BasicImage(rotation=3)"; /* IE8 */
          text-align: center;
          }
       </style>
```

By default, the watermark is set to **transform: rotate(-40deg);**, implying that the watermark text is in an anti-clockwise position at 40 degrees.

b. Change the **transform: rotate;** angle as appropriate.

If you do not specify any value, (for example, **transform: rotate;**) the watermark remains in a straight position. However, if you specify a positive value such as **transform: rotate(25deg);** the watermark rotates clockwise to 25 degrees.

6. Import the style sheet to an existing Teamcenter report definition.

a. In Report Builder, open the **Teamcenter Reports** folder.

b. Select the report definition for which you want to customize the watermark, and click the **Report Data** tab.

c. In the **Report Stylesheets** area, click the **Import stylesheets** button in the **Report Stylesheets** pane.

d. In the **Dataset Type** box, select a style sheet type for the new style sheet, for example, **HTML**. This option is required if the user wants to save the report as a dataset when the report is run.

e. Click the **Browse** on the **Location** box to locate the style sheet file. The style sheet must be in XSL format.

f. Click **OK**.

The style sheet is added to the **Defined Stylesheets** pane.

g. Select the new style sheet in the **Defined Stylesheets** pane, and click the **+** button to move it to the **Selected Stylesheets** pane.

h. Click **Modify** to save your changes.

Add the watermark to an existing report by editing the style sheet associated with the report

In addition to the default reports with the watermark, you can use the watermark in other types of reports. You can do this by editing the style sheets associated with that report.

The **Change_object_Sample_template.xsl** and **WSO_Object_Sample_Template.xsl** style sheets contain the watermark entity by default. You can open one of these templates and copy the watermark entity to the style sheet associated with the report definition for which you want to define a watermark.

1. **Open the XSL style sheet associated with the report for which you want to add a watermark**.

2. **Open the Change_object_Sample_template.xsl or WSO_Object_Sample_Template.xsl style sheet.**

   These style sheets have the watermark entity within the **style type** tags.

3. Copy the **style type** tags containing the watermark to the style sheet of the report to which you want to add the watermark.

   You can copy the **style type** tags inside the **html** tags after the **<head></head>** tags and within the **<body></body>** tags.

> Tip:
>
> Some reports have java scripts and the **<script type></script>** tags appear after the **<head></head>** tags and are followed by the **<body></body>** tags.

Tag to copy:

```
<style type="text/css">

          @media print {
          #watermark_print {
          position:absolute;
          bottom:0;
          width:100%;
          height:60px;
          background:#6cf;
          display: block;
          position: fixed;
          right: 0;
          z-index: 5;
          color: #B81C1C;
          font-size: 25pt;
          height: 50%;
          margin: 0;
          top:350px;
          left:125px;
          margin: auto;
          margin-right: auto;
          margin-left: auto;
          -webkit-transform: rotate(-40deg); /* FF3.5+ */
          -moz-transform: rotate(-40deg); /* Opera 10.5 */
          -o-transform: rotate(-40deg);  /* Saf3.1+, Chrome */
          -ms-transform:rotate(-40deg);
          transform: rotate(-40deg);
          filter:  progid:DXImageTransform.Microsoft.
            BasicImage(rotation=3);  /* IE6,IE7 */
          -ms-filter: "progid:DXImageTransform.Microsoft.
            BasicImage(rotation=3)"; /* IE8 */
          text-align: center;
          }
        </style>
```

After copying (example):

```
<html>
      <head>
        <title>Report Title</title>
      </head>
```

```
<body BGCOLOR="#FFFFFF" link="#0000FF" vlink="#660066">
  <style type="text/css">

    @media print {
    #watermark_print {
    position:absolute;
    bottom:0;
    width:100%;
    height:60px;
    background:#6cf;
    display: block;
    position: fixed;
    right: 0;
    z-index: 5;
    color: #B81C1C;
    font-size: 25pt;
    height: 50%;
    margin: 0;
    top:350px;
    left:125px;
    margin: auto;
    margin-right: auto;
    margin-left: auto;
    -webkit-transform: rotate(-40deg); /* FF3.5+ */
    -moz-transform: rotate(-40deg); /* Opera 10.5 */
    -o-transform: rotate(-40deg);  /* Saf3.1+, Chrome */
    -ms-transform:rotate(-40deg);
    transform: rotate(-40deg);
    filter:  progid:DXImageTransform.Microsoft.
      BasicImage(rotation=3);  /* IE6,IE7 */
    -ms-filter: "progid:DXImageTransform.Microsoft.
      BasicImage(rotation=3)"; /* IE8 */
    text-align: center;
    }
  </style>
  </body>
</html>
```

4. Copy the **watermark_print** division tags to the end of the **</html>** tag. This division tag contains the text of the watermark.

   Tags to copy:

```
<div id="watermark_print"  align="center">
      <p>Confidential&#169;</p>
    </div
```

After copying (example):

```
</html>
    <div id="watermark_print"  align="center">
      <p>Confidential&#169;</p>
    </div>
  </xsl:template>
```

5. **Customize the watermark by editing the style sheet associated with the report definition**.

6. **Import the edited style sheet**.

Create a report definition with a watermark

Before you can define a report definition, you must create a **query source, closure rule, property set, and transfer mode**.

1. In Report Builder, click **File→Create Report**.

2. Select **Item Report** report type, and click **Next**.

3. Click **Assign** to automatically assign a report ID, and then specify a name and description for the report.

4. Specify a query source, closure rule, property set, and then click **Next**.

5. Select a defined style sheet and click **Add** or **Import stylesheets**, and then click **Finish**.

6. **Specify a watermark for this report**.

7. **Customize the watermark by editing the style sheet associated with the report definition**.

8. **Import the edited style sheet**.

4. Configuring reports for Active Workspace

Organize reports in separate dashboards in Active Workspace

As a reports administrator, you can organize reports in separate dashboards in Active Workspace. After creating a dashboard, you can add active summary reports and active item reports and share the dashboard with all users or specific users, groups, and roles as appropriate.

Prerequisites

Only a reports administrator or a user with DBA privileges can create new dashboards.

Procedure

1. On the **Home** page, click the **Reports** tile.
2. To create a new tab on the **Dashboards** sublocation, click **Manage Dashboard** > **Add Dashboard**.

   My Dashboard and **Workflow** dashboards are available by default.

   For more information, see *View My Dashboard reports* in *Active Workspace Fundamentals*.
3. (Mandatory) Specify a name for the dashboard.
4. (Optional) Specify a description.
5. To add a report template, from **Templates**, choose **Add**, select a template, and click **Add Report**.

   You can add templates for active summary reports and active item reports. For active item reports, you must also specify the source object.

   You can also add the report templates after creating the dashboard (see **step 8**).
6. To add more report templates, repeat **step 5**.
7. To add the dashboard, click **Add**.
8. To add more report templates after creating the dashboard, click **Add Report** from the dashboard you created.
9. To share the dashboard, click **Manage Dashboard** > **Share Dashboard** and select all users or specific users, groups, and roles as appropriate.

When you share a dashboard with other users, they can add their own report templates to it. These templates are available only to the specific users who added them and are not accessible in the parent dashboard that was shared. Additionally, the reports administrator or a user with DBA privileges controls the base copy of the shared dashboard. When an admin user deletes the base copy of the shared dashboard, it is removed from all the shared locations.

10. To open a report, hover over the report title until a link appears, and click the title link to open it.

    Alternatively, click **More Commands** > **Open**.

11. To delete a report, click **More Commands** > **Remove**.

    When you delete a shared dashboard, it is removed from the current location and all the shared locations.

Enable preview for Excel reports in Active Workspace

To enable **Preview** for Excel reports from the **My Reports** tab in Active Workspace, modify the **AWC_defaultViewerConfig.VIEWERCONFIG** preference to add the **CrfOutputExcel.Awp0TcooViewer=excel** value.

Teamcenter Office integration is a prerequisite to preview Excel reports.

Enable the Reports tab in Active Workspace for the business objects for which to generate reports

Some business objects such as schedules do not have the **Reports** tab enabled by default. In such cases, reports cannot be generated for such objects. You can modify the style sheet associated with the business object to enable the tab.

For more information about modifying style sheets using the XRT Editor, see *Active Workspace Customization* on Support Center.

To enable the **Reports** tab for the **Schedules** business object:

1. Log on to Active Workspace.

2. Set **Group** to **dba**, **Role** to **DBA**, and **Workspace** to **Active Architect**.

3. Click the **XRT Editor** tile.

   The **XRT Editor** opens in a new tab.

4. Switch to the initial tab and set **Workspace** to **Default**.

5. On the **Home** page, click the **Schedules** tile and open a schedule.

6. Switch to the **XRT Editor** tab. It refreshes to display the style sheet associated with the business object.

7. To add the **Reports** tab, click **Start Edit** and add the following line to the end of the **rendering** element tag:

```
<inject type="dataset" src="Rb0InContextReportsSubLocation" />
```

Example of the style sheet for the business object after adding the above line before the closing **/ rendering** element tag:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
// @<COPYRIGHT>@
// ==================================================
// Copyright 2021.
// Siemens Product Lifecycle Management Software Inc.
// All Rights Reserved.
// ==================================================
// @<COPYRIGHT>@
-->
<!-- Default style sheet for displaying Schedule summary in the Show Object
Location. -->

<rendering xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="XMLRenderingStylesheet_Schema.xsd">
    <header>
        <image source="type"/>
        <property name="owning_user"/>
        <property name="last_mod_date"/>
        <property name="release_status_list"/>
        <property name="object_type"/>
        <property name="fnd0Schmgt_Lock"/>
    </header>
    <page titleKey="tc_xrt_ScheduleGantt" pageNameToken="scheduleNavigation"
    visibleWhen="ActiveWorkspace:SubLocation
    != com.siemens.splm.clientfx.tcui.xrt.objectNavigationSubLocation"/>
    <inject type="dataset" src="ScheduleSummaryProperties"/>
    <inject type="dataset" src="Rb0InContextReportsSubLocation" />
</rendering>
```

8. Save the style sheet.

9. To verify the changes, switch to the initial tab with the business object and refresh the page.

The **Reports** tab is added to the business object.

Import or export reports authored in Active Workspace

As a reports administrator, you can import or export active summary reports, active item reports, and report definitions authored in Active Workspace using the **aw_import_export_reports** utility. For more information, type **aw_import_export_reports -h** on the Teamcenter command prompt.

Allow users to add additional columns to report tables

As an administrator, you can allow users to add additional columns to tables in active summary reports in Active Workspace by setting the **REPORT_AW_ObjectType_Properties** site preference.

For information about retrieving a list of preferences, see *Where can I get a list of preferences?* in *Teamcenter Preferences*.

Allow users to add additional charts to reports

As an administrator, you can allow users to add additional charts to active summary reports and active item reports in Active Workspace by setting the **REPORT_AW_ItemReport_Objects_FilterProperties** site preference.

For information about retrieving a list of preferences, see *Where can I get a list of preferences?* in *Teamcenter Preferences*.

Control the display of dashboards

As an administrator, you can control the display of dashboards for users based on their groups and roles based on the **REPORT_Create_Dashboard_Allowed_GroupRoles** site preference.

For information about retrieving a list of preferences, see *Where can I get a list of preferences?* in *Teamcenter Preferences*.

5. Defining reports in Reporting and Analytics

Managing Reporting and Analytics report definitions

Reporting and Analytics is a stand-alone reporting application. When it is installed and deployed in a Teamcenter environment, it integrates with Report Builder and displays reports in the **TcRA Reports** folder.

Use the Reporting and Analytics Mapper and BuildNPlay applications to create and manage reports. Use Mapper to set user rights and to create the cubes (similar to property sets) that are displayed in reports. Use BuildNPlay to create and manage the Reporting and Analytics report definitions.

For more information about BuildNPlay and Mapper, see the Reporting and Analytics documentation.

The process for managing Reporting and Analytics reports is as follows:

1. In Mapper, **set rights** for users so they can create reports.
2. In Mapper, **create cubes** that define the data to be displayed in a report.
3. In BuildNPlay, **create reports and place them** in the **TcRA Reports** folder in Report Builder.
4. In Report Builder, **manage the reports** in the **TcRA Reports** folder.

Set rights to create and deploy reports

To create and deploy reports using BuildNPlay, a user must have the correct rights based on their role. As an administrator, you use the Reporting and Analytics Mapper application to set the user rights.

1. Log on to the Reporting and Analytics Mapper application.
2. Click the **Role Based Security** button  .

   The Mapper application displays the available roles:

   - **EQUBE_DEFAULT**

     BuildNPlay users can run Reporting and Analytics reports.

   - **EQUBE_SUPER**

BuildNPlay users can modify, save, delete, and deploy Reporting and Analytics report definitions.

3. To locate users, click the down arrow on the **View By** box and choose **Users**.

4. In the **User Name** box, type a string to search and click the **Search** button  .

5. To expand the search results, choose the **+** button by **All**. Select the user on the left side of the pane to see their roles in the right side of the pane.

6. To change a user's roles, select the boxes in the right side of the pane.

   You must assign the **EQUBE_DEFAULT** and the **EQUBE_SUPER** roles to users to allow them to create reports and deploy them to Teamcenter.

7. Click the **Save** button in the lower right of the pane.

Create a cube

If you want to create a report that has a specific set of data, you must first use Mapper to create a cube that contains that data. A *cube* defines the data displayed in a Reporting and Analytics report. For example, if a report has columns that list the item ID, description, and creation date, these columns are provided by the cube.

> Note:
>
> It is important to understand Teamcenter class and attributes structure (the POM schema) when you create property sets. Administrators can examine the POM schema in the Business Modeler IDE by using the **Classes** view, or by right-clicking a class in the **Classes** view and choosing **Open in UML Editor**.

1. Log on to the Reporting and Analytics Mapper application.

2. Choose **Cube→Define**.

   Mapper displays the **Define Cube** tab.

3. Perform the following steps on the **Define Cube** tab:

   a. Type a name for the cube in the **Cube Name** box.

   b. Leave **Assign Category** as **Default** and select **Use Semantic Mapping**.

   c. From the **Select Connection** list, choose **1-User Connection** for standard user or **2-User Connection** if you use Security Services single sign-on.

   d. Click **OK**.

Mapper displays the **Search** tab.

4. Perform the following steps on the **Search** tab:

   a. In the **Search classes** box, type the name of the class you want to use for the cube, or if you want to use a saved query enter **SQ_**.

   b. Press Enter or click the **Search** button  .

   c. Click the right arrow button to move the class or saved query to the right pane.

   d. Click the **Add to Tree** button.

      Mapper displays the **Objects and Steps** tab.

5. Perform the following steps on the **Objects and Steps** tab:

   a. Click the arrow on **Attributes**.

      Mapper displays the attributes on the class or saved query.

   b. Choose the attributes you want to use on the cube by clicking the right arrow button to move them to the right pane. For saved queries, do not use **SQ_** names.

   c. Click the arrow on **Filters**.

      Mapper displays the filters on the class or saved query.

   d. Remove any filters you do not want by clicking the left arrow button to move them to the left pane. For saved queries, only use those that are prefaced by **SQ_**.

      > Note:
      >
      > If use a class cube and you have chosen **owning_user** or **owning_name**, you must remove their filters and convert these attributes to strings.

   e. Click the **Apply** button.

6. Click the **Measure Placement** tab.

   Perform the following steps on the **Measure Placement** tab:

   a. Click the **+** button by the class or saved query name.

      The attributes appear.

b. Select the attributes you want on the cube and click the arrow button to move them to the right pane.

c. Click the **Submit** button.

Mapper displays the attributes on the columns, under **Msr1**, **Msr2**, and so on.

d. Change the names on the columns as desired, for example, change **Msr1** to something appropriate for the attribute below it, change **Msr2**, and so on. These are the columns displayed in the report.

7. Save the cube by clicking the **Save** button on the top toolbar.

Mapper displays a confirmation dialog box containing the following message:

`Cube is created with name:` *cube-name*

8. To send the cube to the server, click the **Update Server Metadata** button .

Create and deploy an ad hoc report

In BuildNPlay, you can create a Reporting and Analytics report from a cube. You can deploy it to Teamcenter so that it displays in the **TcRA Reports** folder. You must be assigned the **EQUBE_SUPER** role to create reports.

The following procedure provides the basic steps for creating an ad hoc report. For more detailed instructions, see the Reporting and Analytics BuildNPlay documentation.

1. Log on to the Reporting and Analytics BuildNPlay application.

2. Click **Ad Hoc** on the toolbar.

3. In the **Cube Categories** pane, select the cube you want to use for the report.

BuildNPlay displays the cube information in the right pane.

4. In the **Measures** pane, select the columns you want to display on the report.

5. If you want to filter the data to be displayed in the report, click the **Apply Filter** button .

BuildNPlay displays the filtering variables at the bottom of the window. Enter the operators and values to use for filtering. Only include the needed operators. For example, if you are using dates, only keep the **<=**, **==**, and **>=** operators, and, for saved queries, remove **NULL**.

6. Click the **Submit** button at the bottom of the window.

7. Click the **Save** button .

   BuildNPlay displays the **Save definition** dialog box.

8. Perform the following steps in the **Save definition** dialog box:

   a. In the **Name for this definition** box, type a name.

   b. In the **Description** box, type a brief explanation.

   c. In the **Category** box, select an area to store the report definition.

   d. Select **Save definition as redefinable template**. This means that you and those you authorize can change the report.

   e. Select **Deploy to Teamcenter**.

   f. From the **Teamcenter Type** list, select the type of report: **Item**, **Summary**, or **Custom**.

   g. Click the **Save** button.

      BuildNPlay displays a new dialog box with the following message:

      ```
      Please select the factors you want to redefine at the time of
      retrieval.
      ```

   h. Select the factor.

      BuildNPlay displays the attributes.

   i. Click the operator button to the left of the attributes to select the operators to use, for example, **==**, **LIKE**, and so on. By default, all operators are on the attribute.

      > Note:
      >
      > If you are using a saved query, you *must* only include the needed operators. For example, use only **==** to enter matching data. If you include operators that do not match the attribute, the data does not get retrieved.

   j. Click the **Save** button.

   k. Click the **Set Permissions** button.

   l. Select the users you want to give permission to run this report and click the **Add** button. To allow all users to run it, select **All**.

m. Click **Submit**. The report is submitted to Report Builder.

A dialog box displays the message:

```
Permissions Have Been Set Successfully
```

n. Click **OK**.

9. To verify that the report displays in Report Builder, log on to Report Builder as a user who has permissions to view the report and select the **TcRA Reports** folder.

The new report appears in the **TcRA Reports** folder.

Manage Reporting and Analytics report definitions from Report Builder

If you have permissions to work with the Reporting and Analytics reports, you can execute, view, edit, set permissions, and delete Reporting and Analytics report definitions from the **Report Data** tab in Report Builder. This method launches the Reporting and Analytics BuildNPlay application to manage the report definitions.

1. Start the Report Builder application in the Teamcenter rich client.

2. Select the **Report Home→TcRA Reports** folder.

3. Select a Reporting and Analytics report definition and click the **Report Data** tab.

Report Builder displays information about the report definition in the **Report Data** pane.

4. Click one of the following buttons in the **Report Data** pane:

- **Execute**

  Generates the report from within the Reporting and Analytics BuildNPlay application. In BuildNPlay, enter any additional filter information and choose **Submit** and then **Show Report**.

  Generating a Reporting and Analytics report this way is the same as running a Reporting and Analytics report from My Teamcenter by choosing **Tools→Reports→Summary Reports**.

- **View**

  Shows the report definition details in BuildNPlay.

- **Modify**

Opens the report definition for modification in BuildNPlay.

- **Delete**

  Removes the report definition from Reporting and Analytics and Teamcenter. When prompted, click **Yes** to confirm deletion.

- **Set Permissions**

  You can change rights to the report in BuildNPlay. You are allowed to set permissions on a report only if you have rights to work with the report definition by the report creator.

5. If prompted, log on to the Reporting and Analytics BuildNPlay application.

   The Reporting and Analytics BuildNPlay application launches in a Web browser.

6. Use BuildNPlay to perform the desired action, for example, modify a report definition or set permissions.

For more information about managing Reporting and Analytics reports using BuildNPlay, see the Reporting and Analytics BuildNPlay documentation.

Convert attributes to strings

When creating a cube using a class, you must convert typed reference attributes and Boolean attributes to strings. For example, if you have chosen **owning_user** or **owning_name**, you must convert them to strings. If you are using a saved query for the cube, you do not have to convert any attributes to strings.

1. Log on to the Reporting and Analytics Mapper application.

2. Click **Manage Views** on the toolbar.

   Mapper displays the **Views** tab.

3. Enter search criteria for the cube and click the **Search** button.

   Mapper displays available cubes.

4. Select the cube that contains attributes to be converted.

   The cube information displays in the **View** tab.

5. Click the **+** button in the **Attributes** column.

   Mapper displays the **Select Attribute** dialog box.

6. In the right pane, select the attribute you need to convert to a string (for example, **ItemRevision.owning_user**) and click the **fn** button in the upper right corner of the dialog box.

   Mapper displays the **Define Expression** dialog box.

7. Perform the following steps in the **Define Expression** dialog box:

   a. Select the attribute at the top of the **Define Expression** dialog box.

   b. In the **Functions** box, double-click **TO_STRING**.

   c. In the **Operators** box, double-click **(**.

      A **(** operator is placed to the left of the attribute at the top of the dialog box.

   d. Select the space to the right of the attribute and double-click **)**.

      A **)** operator is placed to the right of the attribute, enclosing it in parentheses. The final expression should look like the following:

      ```
      TO_STRING ( ItemRevision.owning_user )
      ```

   e. Click **Submit**.

8. In the **Select Attribute** dialog box, select the other attributes you need to convert to strings. When done, click **OK**.

9. Click the **Map View** button  .

   Mapper displays the cube in the **Map View** tab.

10. In the **Filters** box, double-click the converted attribute (for example, **owning_user**) to add it to **Column Name** and **Filter Name** in the right pane. You can change the display name for the attribute under **Column Name**.

11. Click **Save**.

12. To send the updated cube information to the server, click the **Update Server Metadata** button  .

6. Generating reports

Generating reports

You can generate Report Builder and Reporting and Analytics reports from Teamcenter rich client and Active Workspace.

For more information about generating reports in Active Workspace, see *Active Workspace Fundamentals*.

Generating Report Builder reports in the rich client

Using the rich client to generate Report Builder reports

You can generate Report Builder reports in My Teamcenter by choosing **Tools→Reports→Report Builder Reports** or by right-clicking an object and choosing **Generate Report**.

The process of generating reports differs slightly depending on the type of report you choose. If style sheets are available, they can be selected while generating a report. You also have the option to save the generated results as a dataset.

Generate a summary report

Summary reports are created from saved or dynamic queries. When you select a summary report from a list of available summary reports, you are prompted to input query criteria.

1. In My Teamcenter in the rich client, choose **Tools→Reports→Report Builder Reports**.

   The system displays the Report Generation Wizard.

   If no reports are listed, no reports have been created by your system administrator.

2. Select a summary report.

   > Note:
   >
   > If you choose a Reporting and Analytics report, only **Finish** is available. When you choose **Finish**, you are logged on to the Reporting and Analytics BuildNPlay application. Enter any additional filter information and choose **Show Report**.

3. Click **Next**.

4. Fill in criteria for the query. Only the objects that match the query are placed into the report.

> Note:
>
> You can enter multiple criteria separated by the delimiting character. By default the delimiting character is a semicolon (;). In cases where you want Teamcenter to ignore the delimiting character, use the escape character before the delimiting character. By default the escape character is tilde (~).

5. To select the display locale, select the locale from the **Report Display Locale** list.

6. If you want to choose the format for the report, select the style sheet from the **Report Stylesheets** list.

7. If the report is based on **Office Template** source, the **Live Integration** check box is seen.

   Select the **Live Integration** check box to generate the report in Live mode.

8. If you want to save the report as a file, select the **Create Dataset** check box and type a name in the **Dataset Name** box.

   > Note:
   >
   > If the report is based on **Office Template** source, the **Create Dataset** check box is disabled.
   >
   > The dataset generated for reports based on the **Office Template** source is based on the style sheet selected from the **Report Stylesheets** list.

9. If you want the report to be generated in the background, select the **Run in Background** check box.

   This check box is selected by default if the reports administrator has selected the **Run in Async** check box while configuring reports. You can deselect it if you do no want the report to be generated in the background.

   The **AsyncService** translator uses the Dispatcher framework to independently process asynchronous requests from the Teamcenter server in the background mode.

   After the report is generated, you get an email notification. The generated report is available from the **Newstuff** folder and you can double click the report to open it.

10. Click **Finish**.

    If you did not choose a style sheet, the report is displayed in your default XML application. If you chose an HTML spreadsheet, the report is displayed in a Web browser.

    If you saved the report as a dataset, the report is saved in your default **Home** folder, for example, **Newstuff**.

If you selected the **Live Integration** option, the report is in Bulk Live mode. Click the **Save To Teamcenter** button in the report to save the changes you make in the report to Teamcenter.

Generate an item report on a Teamcenter object

You can generate an item report from My Teamcenter in the rich client by choosing one or more Teamcenter objects, such as an item revision. Only objects with associated reports have report definitions displayed in the selection wizard. When you select a item report from a list of available reports, you may be prompted for additional parameters.

1. In My Teamcenter, choose a Teamcenter object, such as an item revision.

2. Right-click the object and choose **Generate Report**.

   The system displays the Report Generation Wizard.

   If no reports are listed, no reports have been created by your system administrator for the selected object type. Teamcenter has standard item report definitions for item revisions.

3. Select an item report.

   > Note:
   >
   > If you choose a Reporting and Analytics report, only **Finish** is available. When you choose **Finish**, you are logged on to the Reporting and Analytics BuildNPlay application. Enter any additional filter information and choose **Show Report**.

4. Click **Next**.

5. For BOM reports, a **Revision Rule** box is displayed. Click the arrow at the end of the **Revision Rule** box to select the revision to run the report on.

   > Note:
   >
   > The BOM report gets generated only for the first selected context object. BOM reports can be generated from Structure Manager.

6. To select the display locale, select the locale from the **Report Display Locale** list.

7. If you want to choose the format for the report, select a style sheet from the **Report Stylesheets** list.

8. If the report is based on **Office Template** source, the **Live Integration** check box is seen.

   Select the **Live Integration** check box to generate the report in Live mode.

9. If you want to save the report as a file, select the **Create Dataset** check box and type a name in the **Dataset Name** box.

   Note:

   If the report is based on **Office Template** source, the **Create Dataset** check box is disabled.

   The dataset generated for reports based on the **Office Template** source is based on the style sheet selected from the **Report Stylesheets** list.

10. If you want the report to be generated in the background, select the **Run in Background** check box.

    This check box is selected by default if the reports administrator has selected the **Run in Async** check box while configuring reports. You can deselect it if you do no want the report to be generated in the background.

    The **AsyncService** translator uses the Dispatcher framework to independently process asynchronous requests from the Teamcenter server in the background mode.

    After the report is generated, you get an email notification. The generated report is available from the **Newstuff** folder and you can double click the report to open it.

11. Click **Finish**.

    If you did not choose a style sheet, the report is displayed in your default XML application. If you chose an HTML spreadsheet, the report is displayed in a Web browser.

    If you saved the report as a dataset, the report file is saved in your default **Home** folder, for example, **Newstuff**.

    If you selected the **Live Integration** option, the report is in Bulk Live mode. Click the **Save To Teamcenter** button in the report to save the changes you make in the report to Teamcenter.

Generate a custom report

Custom reports address special cases such as complex processing or calculations done through custom code, or when data comes from external sources. Each custom report object is associated with a custom program. When a custom report is selected from a list, the server launches the program and the custom process.

1. In My Teamcenter in the rich client, choose **Tools→Reports→Report Builder Reports**.

   The system displays the Report Generation Wizard.

   If no reports are listed, no reports have been created by your system administrator.

2. Select a custom report.

3. Click **Next**.

4. Fill in the requested report information:

   a. In the **Process** box, type the custom process path.

   b. In the **Output** box, type the output file name.

   c. Click the **+** button in the right of the **Parameters** pane to add a new line, and type the parameter name and value.

5. To select the display locale, select the locale from the **Report Display Locale** list.

6. If you want to choose the format for the report, select a style sheet from the **Report Stylesheets** list.

7. If you want to save the report as a file, select the **Create Dataset** check box and type a name in the **Dataset Name** box.

8. If you want the report to be generated in the background, select the **Run in Background** check box.

   This check box is selected by default if the reports administrator has selected the **Run in Async** check box while configuring reports. You can deselect it if you do no want the report to be generated in the background.

   The **AsyncService** translator uses the Dispatcher framework to independently process asynchronous requests from the Teamcenter server in the background mode.

   After the report is generated, you get an email notification. The generated report is available from the **Newstuff** folder and you can double click the report to open it.

9. Click **Finish**.

   If you did not choose a style sheet, the report is displayed in your default XML application. If you chose an HTML spreadsheet, the report is displayed in a Web browser.

   If you saved the report as a dataset, the report is saved in your default **Home** folder, for example, **Newstuff**.

Generate reports in Excel format

You can generate reports in Excel (.xls) format by modifying the **CRF_Report_Allowed_Excel_Stylesheets** user preference.

The **.xlsx** version is not compatible with older versions of Excel. Some reports have both Excel (**.xsl**) and **_OOXML** (**xlsx**) as stylesheets and they are not affected.

However, some reports have only Excel (**.xsl**) as stylesheets. After generating such reports, they cannot be opened as they are corrupted. As an end user, you can edit the **CRF_Report_Allowed_Excel_Stylesheets** user preference to add the name of the stylesheet to this preference value to generate the report in the **.xls** format.

If you have reports that have only Excel (**.xsl**) as stylesheets, do the following.

Procedure

1. While generating the report, copy the name of the stylesheet, for example, **admin_group_role_excel.xsl**.

2. Edit the **CRF_Report_Allowed_Excel_Stylesheets** user preference, add the name of the stylesheet to the preference values, and save the preference.

3. Generate and view the report in **.xls** format.