

Deltek Costpoint® 7.1.1

Integration Console

July 19, 2019

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

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

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

This edition published July 2019.

© Deltek, Inc.

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

All trademarks are the property of their respective owners.

Contents

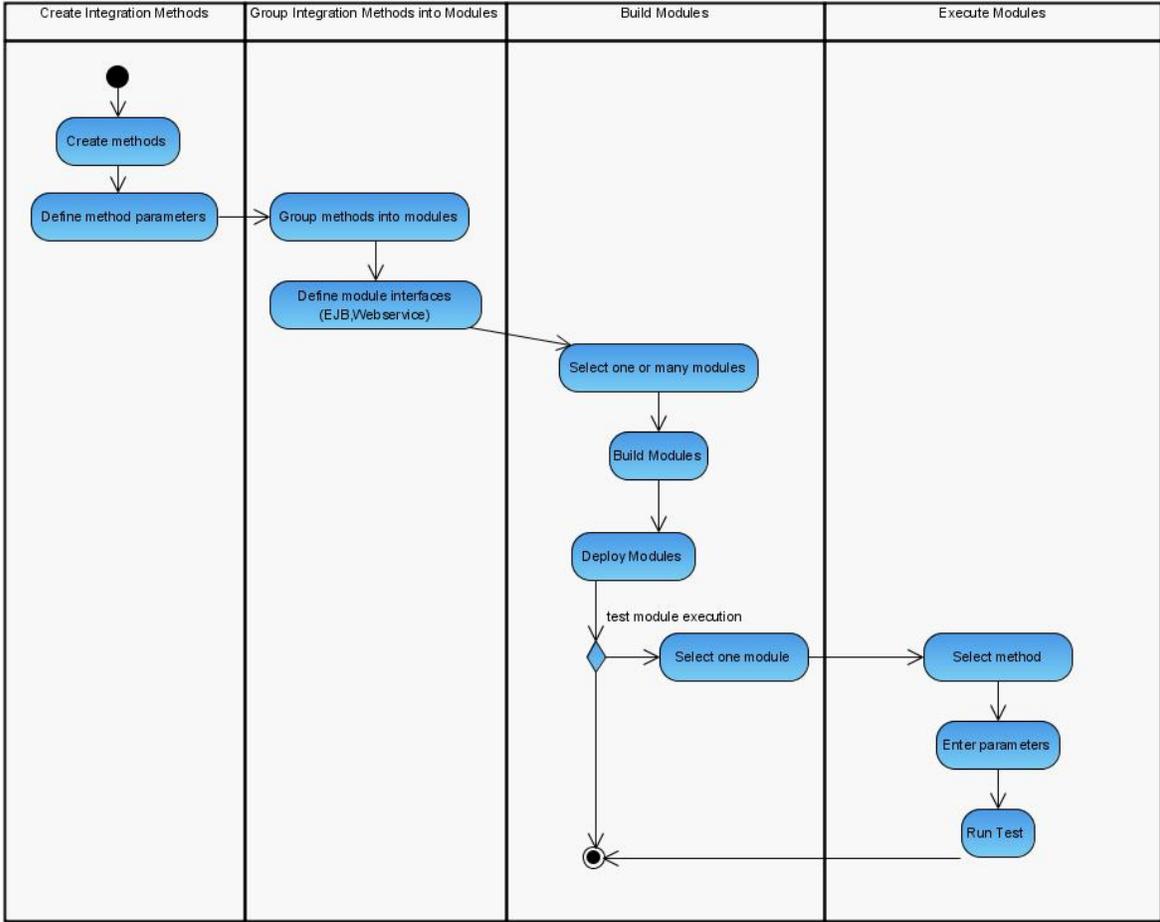
Costpoint Integration Console Overview	1
Procedures Overview	2
Step 1: Create Method	2
Step 2: Create Module	2
Step 3: Build Module	2
Step 4: Test Module	2
Step 5: Export Module	2
Step 6: Import Module	2
Start the Integration Console	3
Add/Edit Process Integration Methods	4
Process Integration Methods List Screen	4
Create a New Process Integration Method	5
Edit a Process Integration Method	8
Add/Edit Report Integration Methods	9
Report Integration Methods List	9
Enter a New Report Integration Method	10
Edit a Report Integration Method	12
Add/Edit Data Integration (Import or Export) Methods	13
Data Integration Methods List	13
Enter a New Data Integration Method	14
Edit a Data Integration Method	18
Add/Edit Data Integration Modules	20
Integration Module List Screen	20
Add/Edit an Integration Module	21
Build/Deploy Integration Modules	24
Integration Module Builder	24
Build a Module	26
Deploy a Module	27
Undeploy a Module	29
Export a Module as a Hotfix	30
Test an Integration Module	31
Select an Integration Module for Testing	32
Prepare to Test an Integration Module	33
Save and Reuse Testing Parameters	34
Preview - Testing Parameters	34

Specify the Query Parameters Used to Filter Integration Data	35
Report or Process Method Without Saved Parameters	35
Report or Process Method With Saved Parameters	37
Data Export Method	38
Data Import Method.....	41
Run the Test of an Integration Module	47
Modify Integration Console Properties	48
Weblogic Cluster Specific Test Server Properties	51
Appendix A: Generic Web Services	53

Costpoint Integration Console Overview

The Integration Console allows Costpoint system administrators or functional experts to expose any Costpoint processes (actions), reports, and data maintenance transactions via Web services, without writing any Java code. Currently, the console only exposes methods to be consumed by other systems. Future enhancement will include the ability to create components or facilities to consume Web service methods from other systems.

The following diagram presents a high-level overview of the process that you use to create, deploy, and test an integration method using this console.



Procedures Overview

Step 1: Create Method

You can create Web services for processes, reports, data export, and data import.

You can create processes and reports to be invoked synchronously or asynchronously.

Data from any maintenance/transaction applications can be exposed for export or import by selecting applications, then result sets and columns to be included.

Step 2: Create Module

You can select methods and group them into logical modules. A module can contain a mixture of action, report, or data methods.

Step 3: Build Module

After you create modules, you must build them. Building involves: creating Java files for the methods; compiling them into Java class; creating XML schemas and interfaces, including creating the Web Services Description Language (WSDL) file for the Web service interface; and packaging them into final JAR files.

An optional step is provided to allow you to deploy the JAR files to a Costpoint WebLogic application server.

Step 4: Test Module

The Integration Console provides you with the option to test a deployed module if the deployed module is on a server that is currently running. You select the modules you want to test, select the method in the module, enter the values for the parameters needed for the method, and execute the method. The method is executed and the response is returned, exactly as if it had been executed from an outside system.



The test is executed on the Costpoint server and database. Unless you want to run this method in a "live" environment, you should run the test on either a test application server or run it against test data.

Step 5: Export Module

Typically, you build and test Web services on a test server before deploying it onto production servers.

This step allows you to capture and package modules into a zip file that can be applied/deployed to another deployment.

Step 6: Import Module

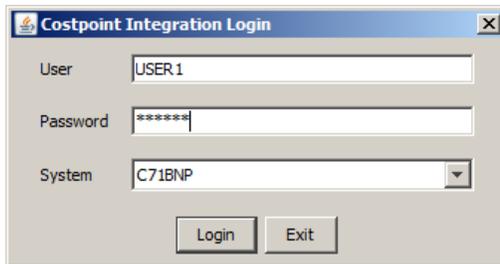
This is an optional step in which you import an integration module that has been exported from other deployments. In earlier releases of Costpoint, you had to use the Deploy Hotfix utility of the DB Wizard to import an integration module, but now you can perform this task directly from the Integration Console.

Start the Integration Console

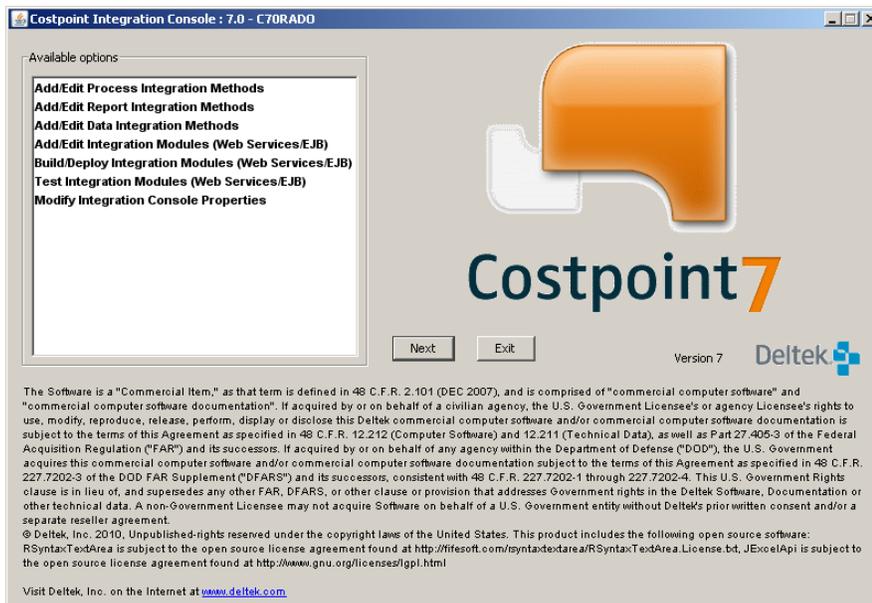
To start the Integration Console, complete the following steps:

1. Start the Integration Console using one of the following methods:
 - Run the batch file CPWebIntegrationUtility.cmd, located with all other Costpoint command files in the %deltek%\costpoint\71\bin folder.
 - From the **Start Menu** under the **Costpoint Menu** group click **Integration Console**.
2. On the Costpoint Intergration Logging dialog box, enter the following information:
 - **User:** Enter a Costpoint User ID that has rights to access the Integration Console. To allow user to access Integration Console, you need to select the **Allow Access To Integration Console Flag** option in the Maintain Users Application. Integration Console supports Database- and LDAP- (Acive Directory) based Authentication types.
 - **Password:** Enter the password associated with the User.
 - **System:** This drop-down field lists all the system names configured in Costpoint. These are the same systems that you would select if you were logging into the Costpoint application server via the Web browser.

The default system appears in the field. Select this or another system.



3. Click **Login** to begin your session. When you log in successfully, the Costpoint Integration Console screen displays. Use this screen to access all other screens.



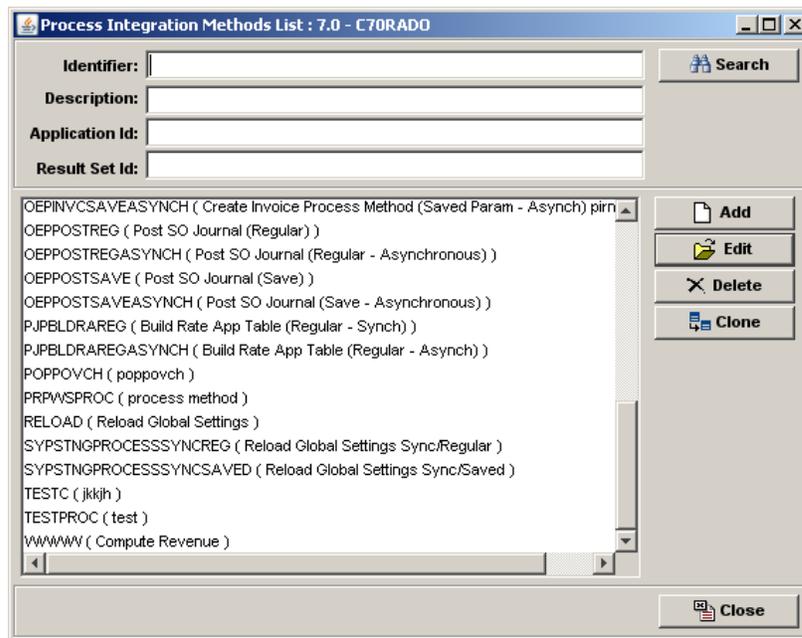
Add/Edit Process Integration Methods

Use the **Add/Edit Process Integration Methods** option to create integration methods for Costpoint processes. Any process (action) you can run interactively from the browser can be exposed as an integration method.

Process Integration Methods List Screen

Select **Add/Edit Process Integration Methods** to display the Process Integration Methods List screen.

On this screen, you can select existing methods to modify or create new methods.



Use the following buttons to perform your tasks:

- **Search:** If the methods list is large, use the Search feature to filter the search methods based on **Identifier**, **Description**, **Application Id**, or **Result Set Id**. Enter characters in any of these fields, and click **Search** to display the filtered methods list. All filters use the SQL LIKE condition.
- **Add:** Click **Add** to create a new integration method. When you click this button, the **Enter a New Process Integration Method** screen displays. Use this screen to enter the new method.
- **Edit:** Click **Edit** to modify the selected integration method. When you click this button, the **Edit Process Integration Method** screen displays. Use this screen to modify the selected method.
- **Delete:** Select an integration method from the list, and click **Delete** to delete an integration method.
- **Clone:** Select an integration method from the list, and click **Clone** to create a new integration method by cloning an existing method. When you click this button, the **Edit Process Integration Method** screen displays. Use this screen to clone the selected method.

- **Close:** Click **Close** to close the **Process Integration Methods List** screen and return to the Costpoint Integration Console screen.

Create a New Process Integration Method

Click **Add** on the **Process Integration Methods List** screen to create a new method.

Enter a new Process Integration Method : 7.0 - C70RADD

Identifier: CREATETESTSAVEDPARAMS

Description: Create Purchase Orderes

App: PPGPO (Create Purchase Orders) Id: Search

Result Set: PPGPO_PARAM (Create Purchase Orders)

Action: PPGPO_PRINTPROCESS (Print and Create Purchase Orders)

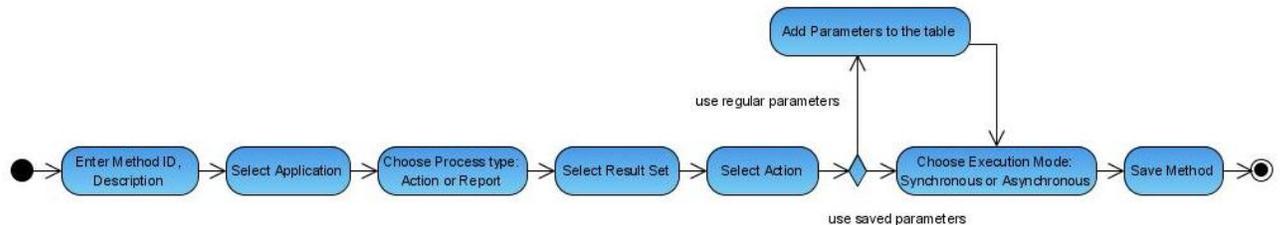
Synchronous

Object ID	Description	Parameter	Required	Data Type
SAVED_PARM_ID	Saved Parameter	SAVED_PARM_ID	✓	STRING (30)

Parameters: Regular Saved

Buttons: Save, Reset, Close

The following diagram illustrates the typical steps required to create a new process integration method.



Use these buttons at the bottom of the screen:

- **Save:** Click **Save** to save the integration method and return to the Process Integration Methods List screen.
- **Reset:** Click **Reset** to discard saved changes and load the last saved configuration.
- **Close:** Click **Close** to discard unsaved changes and return to the Process Integration Methods List screen.

Enter Data for the Method

Enter data in the following fields:

- **Identifier:** Enter a unique ID for this method. Only alpha characters are allowed. The ID you specify is used as the name of the integration method. For example, if you enter **CREATEPURCHASEORDER** in the **Identifier** field, the integration method name is:
createpurchaseorder
- **Description:** Enter a description for this method. The description is used for internal identification only and is not used elsewhere.
- **App:** Click the drop-down arrow in this field to see a list of all the Costpoint process and reporting applications to which actions are associated. Select the application that contains the end process that you want to expose for integration.
- **Id:** This field allows you to filter the list of applications based on ID. Enter the first couple of characters and click **Search**.
- **Search:** Click **Search** to filter the list of available applications.
- **Result Set:** This field automatically displays the top-level result sets that belong to the selected Costpoint application. Currently, each action application has only one result set, so this field will generally display only one result set.
- **Action:** This field contains a list of actions associated with the selected application. Most applications have only one action. The Posting application may have more than one action (Post or Print and Post). Click the drop-down arrow to select the action you want to use to create the integration method.
- **Synchronous:** Select this option to choose synchronous execution to launch a report or action. Synchronous execution requires the caller to wait until the method execution is completed before performing other Costpoint tasks. This choice is available in the interactive browser user interface as well as in this console.

If you do not select synchronous execution, the execution will be asynchronous. Asynchronous execution returns the command to the caller immediately after receiving the instruction. The method is queued up in a Costpoint message queue to be executed there. This is the same behavior that occurs when the method is launched interactively in the browser user interface.

A user can ask to receive an email notification when the server executes and completes the process; for this option to work, the user profile must contain an email address.

- **Parameters:** This option lets you specify the parameters for the method so that the caller of the integration method knows what to construct in the parameters when calling the method.

There are two ways to specify how parameters can be passed:

- **Regular :** Requires data to be specified for each individual parameter. You must construct and pass all the parameters when the method is called.
- **Saved :** Uses existing parameters that are already saved in the Costpoint UI application.

Use Saved Parameters

Select the **Saved** option if you want the method to use parameters that were previously saved.

In the Costpoint UI interactive mode, a user can execute an action by entering new parameters or retrieving parameters that were saved previously. Saving criteria for re-use is convenient if the action will be repeated in the future with the same parameters (for example, periodically posting vouchers).

You assign a unique parameter ID when selection parameters are saved in interactive mode.

If you use this option, the console creates a parameter in the integration method that only requires a parameter ID to be passed. At execution time, the selection parameters saved previously are automatically retrieved from the function parameter table, based on the ID.

Use this approach to expose a method that will execute on a previously saved parameter.

The following is an example of the method, using saved parameters:

```
createpousingsavedparams ( systemName As string , companyId As string , savedParamId As string ) As MethodResponse
```



System name and company ID are included as parameters in all methods because they are required to identify the correct system and company (similar to when you log in interactively via the browser UI).

Use Regular Parameters

Alternatively, you can create a method that requires data for each individual parameter. You must construct and pass all the parameters when calling such a method.

If you select the **Regular** option, you will see a list of all the parameters. Select parameters using check boxes or drop-down lists.

Object ID	Description	Parameter	Required	Data Type
<input checked="" type="checkbox"/>	BUYER_ID_FR	Starting Buyer	BUYER_ID_FR	STRING (12)
<input checked="" type="checkbox"/>	BUYER_ID_TO	Ending Buyer	BUYER_ID_TO	STRING (12)
<input checked="" type="checkbox"/>	BUYER_RANGE_CD	Buyer Range Option	BUYER_RANGE_CD	STRING (6)
<input checked="" type="checkbox"/>	NON_CONT_BUYER_FL	Buyer Non- Contiguous Ranges	NON_CONT_BUYER_FL	STRING (1)
<input checked="" type="checkbox"/>	NON_CONT_PO_FL	Purchase Order Non- Contiguous Ranges	NON_CONT_PO_FL	STRING (1)
<input checked="" type="checkbox"/>	NON_CONT_RQ_FL	Requisition Non- Contiguous Ranges	NON_CONT_RQ_FL	STRING (1)
<input checked="" type="checkbox"/>	NON_CONT_VENDOR_FL	Vendor Non- Contiguous Ranges	NON_CONT_VENDOR_FL	STRING (1)
<input checked="" type="checkbox"/>	PO_ID_FR	Starting Purchase Order	PO_ID_FR	STRING (10)
<input checked="" type="checkbox"/>	PO_ID_TO	Ending Purchase Order	PO_ID_TO	STRING (10)
<input checked="" type="checkbox"/>	PO_LN_SORT_CD	Purchase Order Line Sort	PO_LN_SORT_CD	STRING (1)
<input checked="" type="checkbox"/>	PO_RANGE_CD	Purchase Order Option	PO_RANGE_CD	STRING (6)
<input checked="" type="checkbox"/>	RQ_ID_FR	Starting Requisition	RQ_ID_FR	STRING (10)
<input checked="" type="checkbox"/>	RQ_ID_TO	Ending Requisition	RQ_ID_TO	STRING (10)
<input checked="" type="checkbox"/>	RQ_RANGE_CD	Requisition Range Option	RQ_RANGE_CD	STRING (6)
<input checked="" type="checkbox"/>	VEND_ID_FR	Starting Vendor	VEND_ID_FR	STRING (12)
<input checked="" type="checkbox"/>	VEND_ID_TO	Ending Vendor	VEND_ID_TO	STRING (12)
<input checked="" type="checkbox"/>	VEND_RANGE_CD	Vendor Range Option	VEND_RANGE_CD	STRING (6)

The following is an example of the final method, using specified parameters:

```
createpospecifyingparams ( systemName As string , companyId As string , BUYER_RANGE_CD As string , NON_CONT_BUYER_FL As string , NON_CONT_PO_FL As string , NON_CONT_RQ_FL As string , NON_CONT_VENDOR_FL As string , PO_LN_SORT_CD As string , PO_RANGE_CD As string , RQ_RANGE_CD As string , VENDOR_RANGE_CD As string ) As MethodResponse
```

The list of parameters provides the following information:

- **Object ID (Label):** This column displays the internal ID and description used in the application.
- **Description:** This column displays the description for this parameter. The console defaults this description from the application. You can change it to a different description, as needed. (This description currently has no use, but it may be used in the future to generate documentation.)
- **Parameter:** This column displays the parameter name in the method. The console defaults the internal object ID into this column. You can change it to a more descriptive name using the description that is displayed in the Description column.
- **Required:** This column indicates whether or not the parameter is required.
- **Data Type:** This column displays the parameter data type.

To make the column selection process more efficient, click the down arrow at the top left of the table.

Object ID	Description	Parameter	Required	Data Type
Starting Company	COMPANY_ID_FR			STRING (10)
Company Option	COMPANY_ID_RANGE		✓	STRING (10)
Ending Company	COMPANY_ID_TO			STRING (10)
Include Deactivated Users	INCL_TERM_USER_FL		✓	STRING (1)
Page Break	PAGE_BRK_FL		✓	STRING (1)

Choose from these options:

- **Include Only Editable Columns:** Select this option to include only editable columns.
- **Include Only Required Columns:** Select this option to include only required columns.

Edit a Process Integration Method

Click **Edit** on the Process Integration Methods List screen to modify an existing method. All options are the same as the options on the Enter a New Process Integration Method screen, described above. You can modify any options or parameters.

Object ID	Description	Parameter	Required	Data Type
BUYER_ID_FR	Starting Buyer	BUYER_ID_FR		STRING (12)
BUYER_ID_TO	Ending Buyer	BUYER_ID_TO		STRING (12)
BUYER_RANGE_CD	Buyer Range Option	BUYER_RANGE_CD	✓	STRING (6)
NON_CONT_BUYER_FL	Buyer Non- Contiguous Ranges	NON_CONT_BUYER_FL	✓	STRING (1)
NON_CONT_PO_FL	Purchase Order Non- Contiguous...	NON_CONT_PO_FL	✓	STRING (1)
NON_CONT_RQ_FL	Requisition Non- Contiguous Ra...	NON_CONT_RQ_FL	✓	STRING (1)
NON_CONT_VENDOR_FL	Vendor Non- Contiguous Ranges	NON_CONT_VENDOR_FL	✓	STRING (1)
PO_ID_FR	Starting Purchase Order	PO_ID_FR		STRING (10)
PO_ID_TO	Ending Purchase Order	PO_ID_TO		STRING (10)
PO_LN_SORT_CD	Purchase Order Line Sort	PO_LN_SORT_CD	✓	STRING (1)
PO_RANGE_CD	Purchase Order Option	PO_RANGE_CD	✓	STRING (6)
RQ_ID_FR	Starting Requisition	RQ_ID_FR		STRING (10)
RQ_ID_TO	Ending Requisition	RQ_ID_TO		STRING (10)
RQ_RANGE_CD	Requisition Range Option	RQ_RANGE_CD	✓	STRING (6)
VEND_ID_FR	Starting Vendor	VEND_ID_FR		STRING (12)
VEND_ID_TO	Ending Vendor	VEND_ID_TO		STRING (12)
VEND_RANGE_CD	Vendor Range Option	VEND_RANGE_CD	✓	STRING (6)

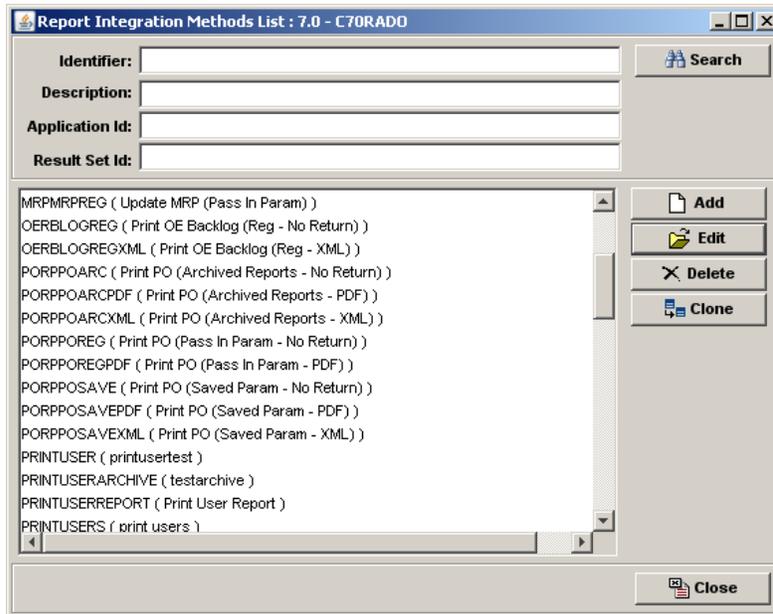
Add/Edit Report Integration Methods

Use the **Add/Edit Report Integration Methods** option to create integration methods for Costpoint reports. Any report that you can run interactively from the browser can be created as an integration method.

When you select this option, the Report Integration Methods List screen displays.

Report Integration Methods List

On this screen, you can select existing methods to modify, or you can create new methods.



The options on this screen are the same as those on the Process Integration Methods List screen.



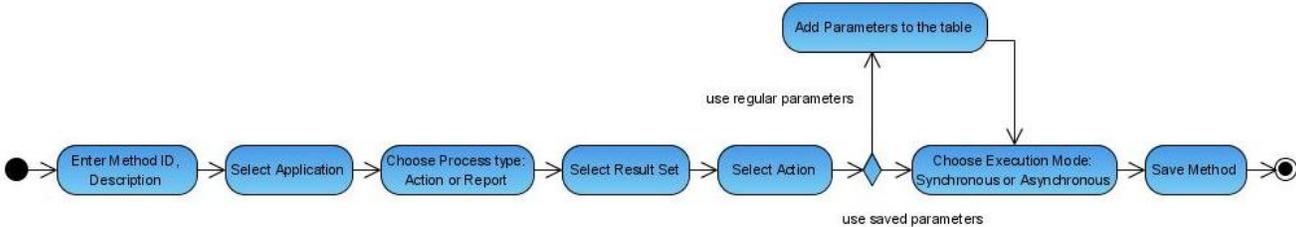
See the [Add/Edit Process Integration Methods](#) section for field and option descriptions.

Enter a New Report Integration Method

Click **Add** on the Report Integration Methods List screen to create a new method.

Object ID	Description	Parameter	Required	Data Type
SAVED_PARM_ID	Saved Parameter	SAVED_PARM_ID	✓	STRING (30)

The following diagram illustrates the typical steps required to create a new report integration method.



Enter Data for the Method

This screen has the same input fields and options as those on the Process Integration Methods List screen, plus several report-specific fields and options.

- Archived Reports:** If you select this option, under **Parameters**, the integration method will be generated with one string parameter, Archived Report ID. At runtime, the report will not be generated, but it will be uploaded from the archive using the Archive Report ID. For this to work, the report must be generated and archived using our standard browser UI.



You cannot use the Archived Reports option if you have chosen Synchronous execution.

Required	Data Type
	STRING (10)
✓	STRING (10)
✓	STRING (10)
✓	STRING (1)
✓	STRING (1)

Parameters

Regular

Saved

Archived Reports

- Return Report As:** Use this option to specify how a report should be delivered (print to a printer, email as an attachment, download as a PDF, and so on). Different reports have different delivery options.

Return report as: **NO_RETURN**

Max Number of Pages: **NO_RETURN**

HTML

PDF

EXCEL

Possible choices are:

- NO_RETURN:** The report is not returned as part of the integration method invocation, but you can set up a saved parameter with one of many available Delivery Options in the Print Options dialog box.



NO_RETURN is the only option if you have chosen Synchronous execution.

- HTML:** The report is returned as part of an HTMLReportResponse object as an HTML String along with the corresponding array of images referenced in the report.
- PDF:** The report is returned as part of a PDFReportResponse object as a binary array that represents the report in PDF format.
- EXCEL:** The report is returned as part of an EXCELReportResponse object as a binary array that represents the report in Excel format.
- Max Number of Pages:** This field allows you to limit the number of report pages you want to generate. Additional pages are truncated.



When you run a report from the normal browser UI, the report can print only if there is sufficient information in the Print Options dialog box, which contains settings that control how the report is printed on a printer, emailed, archived, downloaded, or saved as a file.

All of these options, other than printing to a printer, require additional settings that are not currently available as parameters for an integration report method. Unless you want to print the report to the server printer, you should select the **Saved Parameters** option and specify these settings in the saved parameters.

See the [Add/Edit Process Integration Methods](#) section for information about the **Saved Parameters** option.

Edit a Report Integration Method

Click **Edit** on the Report Integration Methods List screen to modify an existing method. All options are the same as the options on the Enter a New Report Integration Method screen. You can modify any options or parameters.

Object ID	Description	Parameter	Required	Data Type
SAVED_PARM_ID	Saved Parameter	SAVED_PARM_ID	✓	STRING (30)

Parameters

Regular
 Saved
 Archived Reports

Add/Edit Data Integration (Import or Export) Methods

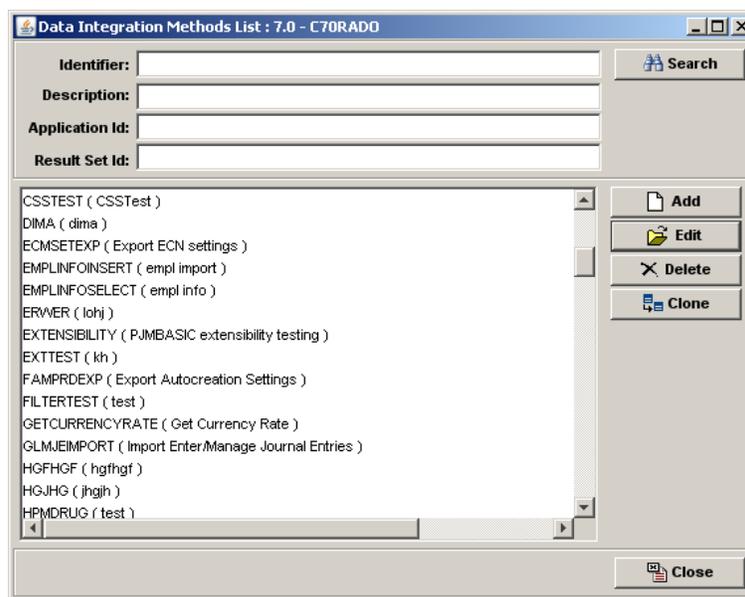
Use the **Add/Edit Data Integration Methods** option to create integration methods for Costpoint data export or import. Any application in Costpoint that displays data for viewing or accepts data for saving can be created as an integration method, including any configuration, maintenance, or transaction application.

When you select this option, the Data Integration Methods List screen displays. You can modify or delete an existing data integration method or create a new method from scratch or by cloning an existing method.

Data Integration Methods List

Select **Add/Edit Data Integration Methods** to display the Data Integration Methods List screen.

On this screen, you can select existing methods to modify, or create new methods.



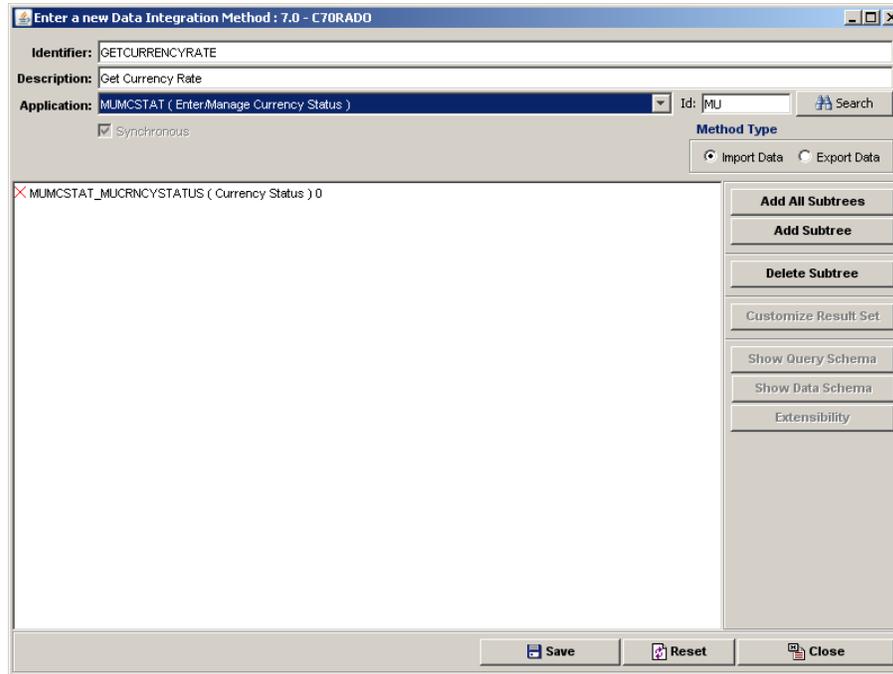
The options on this screen are the same as those on the Process Integration Methods List screen and the Report Integration Methods List screen.



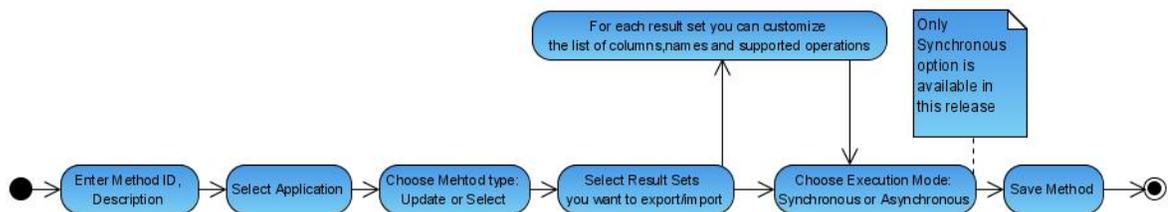
See the [Add/Edit Process Integration Methods](#) section for field and option descriptions.

Enter a New Data Integration Method

Click **Add** on the Data Integration Methods List screen to create a new method.



The following diagram illustrates the typical steps required to create a new data integration method.



Enter Data for the Method

Enter data in the following fields:

- **Identifier:** Enter a unique ID for this method. Only alpha characters are allowed. The ID that you specify is used as the name of the integration method. For example, if you enter **GETCURRENCYRATE** in the **Identifier** field, the integration method name will be:
getcurrencyrate
- **Description:** Enter a description for this method. The description is used for internal identification only and is not used elsewhere at this time.
- **Application:** Click the drop-down arrow in this field to see a list of all Costpoint applications that can export or import data. This list includes any configuration, maintenance, or transaction application.

- **Id** —This field allows you to filter the list of applications by Application ID by entering the first couple of characters and clicking **Search**.
- **Method type**: Select the type you want to create: **Import Data** or **Export Data**.

Result Sets

As soon as you choose an application, the result sets that belong to the application display in the window below. These result sets are the same result sets that you see and work with when you open this application in the browser UI.

Initially, all result sets display with the **Delete (X)** icon.

Use these buttons to control the result sets that are selected:

- **Add All Subtrees**: Click this button to add all result sets (and their children) to the tree of result sets you want to export or import. The **Delete** icon changes to a **Folder** or **Bullet** icon to signify that the result sets are now enabled.
- **Add Subtree**: Click this button to add the selected result sets (and their children) to the tree of result sets you want to export or import.
- **Delete Subtree**: Click this button to remove the selected result sets (and their children) from the tree of result sets you want to export or import.
- **Customize Result Set**: For each result set, you can select columns you want to include, customize their names, and enter extra data-specific options.



You cannot select a subtree if its parent subtree is not selected.

Customize the Result Set

Click the **Customize Result Set** button to display a listing of all columns in the result set. By default, all columns, both visible and hidden in the Costpoint UI, are listed here and available for selection. The column listing includes additional metadata to help you identify the columns that you want to select.

Id	Title	Name	Method PK	Server PK	Visible	Editable	Requ...	Data Type	Ext. Unit Id
<input checked="" type="checkbox"/>	ACCT_GRP_CODE	ACCT_GRP_CODE	<input type="checkbox"/>					STRING (3)	
<input checked="" type="checkbox"/>	ACCT_ID	ACCT	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (15)	
<input checked="" type="checkbox"/>	ACCT_NAME	ACCT Name	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (25)	
<input checked="" type="checkbox"/>	ACKN_DT	Acknowledgment Date	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		DATE	
<input checked="" type="checkbox"/>	ACKN_FL	Acknowledgment Required	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	STRING (1)	
<input checked="" type="checkbox"/>	ADDR_DC	Vendor Address	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (10)	
<input checked="" type="checkbox"/>	AUTO_DEF_HDRTXT...	Auto-Default Header Text	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	STRING (1)	
<input checked="" type="checkbox"/>	AUTO_VCHR_FL	Auto- Voucher	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (1)	
<input checked="" type="checkbox"/>	BILL_TO_ADDR_DC	Bill-To Address	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (10)	
<input checked="" type="checkbox"/>	BILL_TO_LOC_ID	Bill-To Location	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (12)	
<input checked="" type="checkbox"/>	BILL_TO_LOC_ID_DE...	Bill-To Location Description	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (30)	
<input checked="" type="checkbox"/>	BILL_TO_LOC_ID_NA...	Bill-To Location Name	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (40)	
<input checked="" type="checkbox"/>	BLKT_BAL	Blanket Amt Balance	<input type="checkbox"/>		<input checked="" type="checkbox"/>			NUMBER (1...	
<input checked="" type="checkbox"/>	BRNCH_ADDR_DC	Return Address Code	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (10)	
<input checked="" type="checkbox"/>	BRNCH_LOC_ID	Return Location	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (12)	
<input checked="" type="checkbox"/>	BRNCH_LOC_ID_DESC	Return Location Descripti...	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (30)	
<input checked="" type="checkbox"/>	BRNCH_LOC_ID_NAME	Return Location Name	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (40)	
<input checked="" type="checkbox"/>	BUYER_ID	Buyer	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	STRING (12)	
<input checked="" type="checkbox"/>	CHNG_DT	Last Change Date	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		DATE	
<input checked="" type="checkbox"/>	CIS_CD	CIS Code	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (6)	
<input checked="" type="checkbox"/>	CIS_VH_RT	CIS_VH RT	<input type="checkbox"/>		<input checked="" type="checkbox"/>			NUMBER (8...	
<input checked="" type="checkbox"/>	CITY_NAME	Vendor Address City	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (25)	
<input checked="" type="checkbox"/>	CNFRM_FL	Confirming PO	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	STRING (1)	
<input checked="" type="checkbox"/>	CNTACT_FIRST_NAME	Vendor Contact First Name	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (25)	
<input checked="" type="checkbox"/>	CNTACT_LAST_NAME	Vendor Contact Last Name	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (25)	
<input checked="" type="checkbox"/>	COUNTRY_CD	Vendor Address Country	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (8)	
<input checked="" type="checkbox"/>	DEL_TO_FLD	Deliver To	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		STRING (25)	

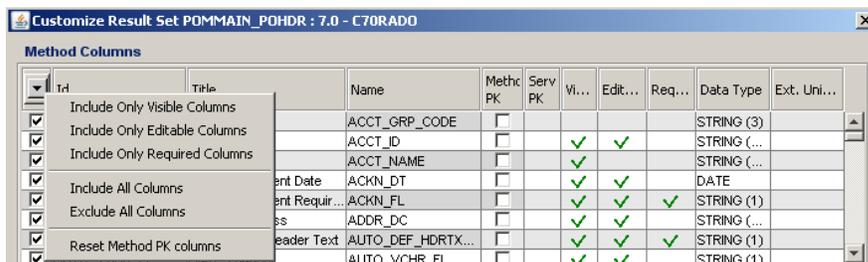
All Columns: 108 Selected Columns: 108 Supported operations: Select Insert Update Delete

Show Selected Only Save Cancel

The columns on this screen are:

- **Id:** This column displays the internal ID for the column.
- **Title:** This is the column title as you would see it in the Costpoint UI. If a table view title is available, it is displayed. If not, the form view title is displayed.
- **Name:** This column shows how the column name will be displayed in the integration method. In most cases, it will be the same as the column ID. If you have columns with the same name in several result sets, the result set name is added to the column name to keep it unique across result sets. The Integration Console lets you customize a column name, which can be convenient when a column name is too long .
- **Method PK:** This column is available only for data import integration methods, which require that you identify the columns that are primary keys. These are the columns that uniquely identify a row in the result set so that the integration method can properly perform the UPDATE, SELECT or DELETE operation. The exceptions to this rule are special types of result sets (Filter and Single Row) that can only have one row. In this case you don't have to specify method PK columns.
- **Server PK:** This column shows whether the column was defined as a primary key column.
- **Visible:** This column shows whether the column is visible in the Costpoint UI.
- **Editable:** This column shows whether the column is editable in the Costpoint UI.
- **Required:** This column shows whether the column is required in the Costpoint UI.
- **Data Type:** This column shows the column type.

Filtering Columns for Selection



To make it easier to select columns for inclusion in the method, click the down arrow to display some filtering options:

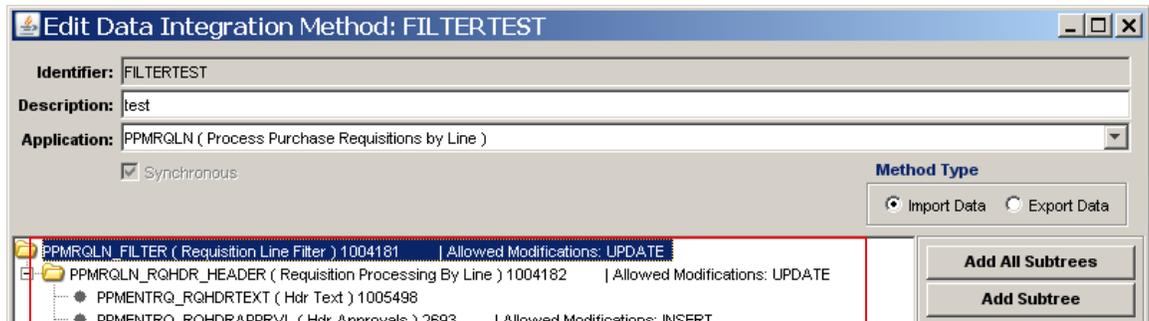
- **Include Only Visible Columns:** This option selects all columns that are visible in the Costpoint UI. This option is convenient for data export methods.
- **Include Only Editable Columns:** This option selects all columns that are editable in the Costpoint UI. This option is convenient for data import methods.
- **Include Only Required Columns:** This option selects all columns that are required in the Costpoint UI. This option is convenient for data import methods.
- **Include All Columns:** This option selects all columns.
- **Exclude All Columns:** This option deselects all columns.
- **Reset Method PK Columns:** This option puts a checkmark in the **Method PK** field for any column defined as a Server PK. This option is only available for data import methods. It usually selects the proper **Method PK** fields. However, in some rare cases, no columns

exist with a Server PK label. If this occurs, select those columns that are logical keys for this result set.

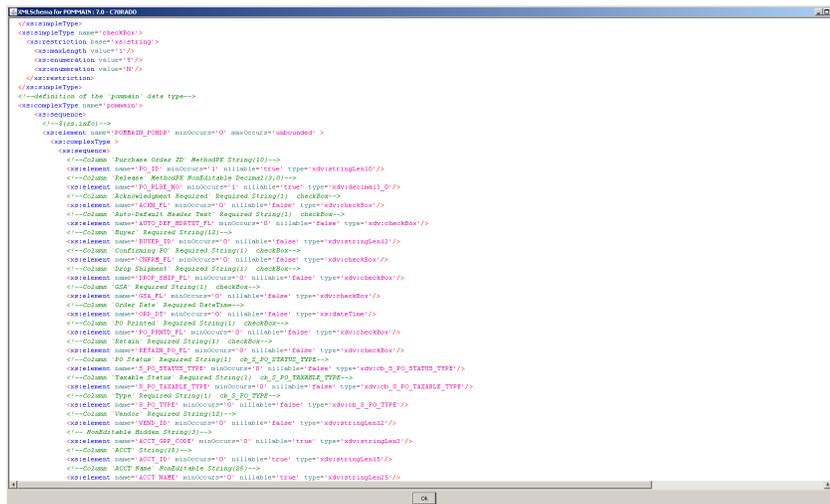


You may not always be able to identify logical keys. For help identifying logical key columns, see the *Deltek Costpoint 7.0 Data Dictionary*.

- **Show Selected Only/Show All Columns:** This option controls the display of columns on the Customize Result Set screen. You can choose to show all columns or only those selected. This can be convenient when a result set has many columns.
- **Supported Operations:** Select the **Select**, **Insert**, **Update**, and/or **Delete** check boxes to choose which operations will be supported by the integration method.
 - For Data export methods, only **Select** is available.
 - For Data import methods, **Select**, **Insert**, **Update**, and **Delete** options are available if Costpoint allows these operations to take place for a given result at runtime.
 - Some operations have extra requirements. For example, you usually need to select Method PK columns if you choose the **Select**, **Update**, or **Delete** operations.
 - If some operations are not allowed for a result set, the **Allowed Modifications** string is added to the end of the result set title. For example, in the method below, only the UPDATE operation is allowed for the top two result sets and only the INSERT operation is allowed for the “Hdr Approvals” result set.



- **Show Data Schema:** Click this button to display the XML schema for the data you want to import or export.



- Show Query Schema:** This option is available only for data export methods. This screen displays the XML schema for the query conditions that you can pass to the generated data method at run-time to filter the data that you are exporting.

For each result set, you can set one or many query conditions (which act as OR conditions in a SQL query). Each condition can contain one or many column conditions (which act as AND conditions in a SQL query) on the columns you are exporting. These columns should also be among the list of query-able columns in the original result set (those columns that you can select for query in the Costpoint UI). Each column condition is a combination of column name, relation, and value.

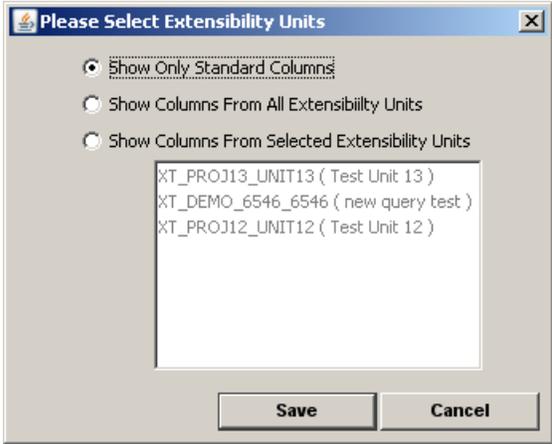
Edit a Data Integration Method

Click **Edit** on the Data Integration Methods List screen to modify an existing method. The options on this screen are the same as the options on the Enter a New Integration Method screen. You can modify any options or parameters.

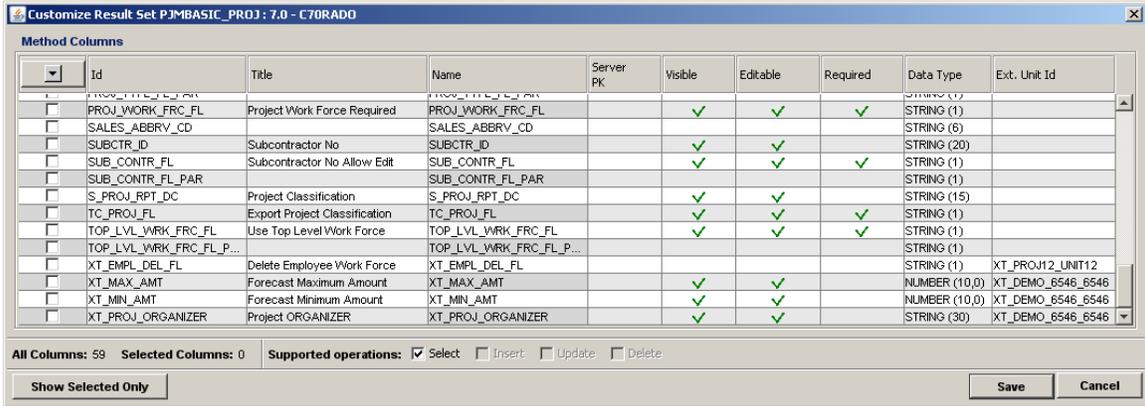
Extensibility Units

If you have some extensibility units in a selected application, the **Extensibility** button becomes enabled.

When you click **Extensibility**, the Please Select Extensibility Units dialog box displays. Use this dialog box to select the Extensibility Units that you want to include in the data method.



When you select an Extensibility Unit, columns from this unit become available on the Customize Result Set screen. You can select the columns to be used at runtime, as you would with any regular column.



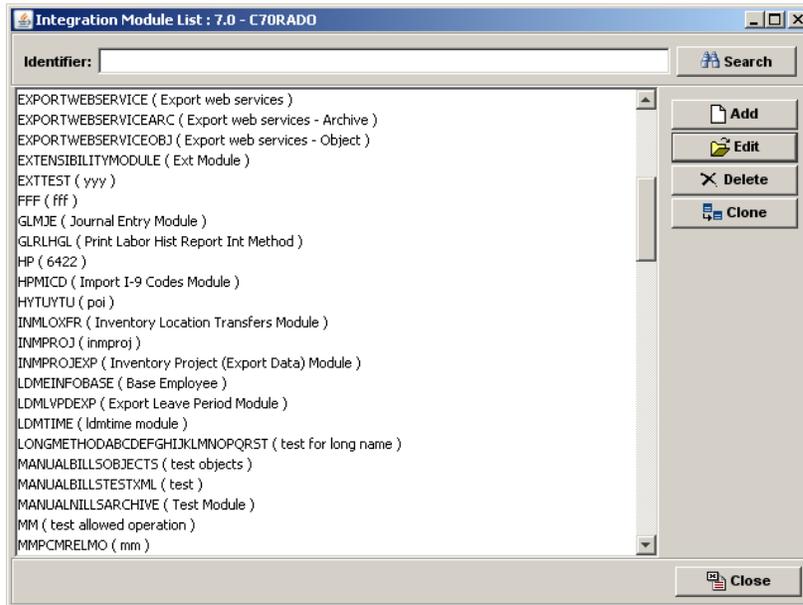
The **Ext. Unit Id** column displays the extensibility unit ID for columns that were created using the Extensibility Console. This column is empty for standard result sets.

Add/Edit Data Integration Modules

Select **Add/Edit Data Integration Modules** to display the Integration Module List screen.

Integration Module List Screen

On this screen, you can select existing modules to modify, or you can create new modules.



Use these buttons to control the modules that you select.

- **Search:** If the modules list is large, use the Search feature to filter the search modules based on module **Identifier**. Enter conditions and then click **Search** to display the filtered module list.
- **Add:** Click **Add** to create a new integration module. When you click this button, the Enter a New Integration Module screen displays.
- **Edit:** Click **Edit** to modify the selected integration module. When you click this button, the Edit Integration Module screen displays.
- **Delete:** Click **Delete** to delete the selected integration module.
- **Clone:** Select an integration module from the list and click **Clone** to create a new integration module by cloning an existing module. When you click this button, the Edit Integration Module screen displays. Use this screen to clone the selected module.
- **Close:** Click **Close** to close the Integration Module List screen and return to the Costpoint Integration screen.

Add/Edit an Integration Module

Enter/Edit Data for the Module

Enter or edit data in the following fields:

- **Module ID:** Enter a unique ID for this module. Only alpha characters are allowed.
 The ID you enter is used as part of the name of the JAR file created when the module is finally built. For example, if you enter **CURRENCY** in the **Module ID** field, the final JAR file's name will be **currencyws.jar** (the "ws" stands for Web service).
 The ID is also used as part of the WSDL file name. For example, if you enter **CURRENCY** in the **Module ID** field, the final WSDL name is **CurrencyService.WSDL**.
 You can edit this field only when adding a new module.
- **Description:** Enter a description for this module. The description is used for internal identification only.
- **Vendor and Version:** These fields are used by third-party software integrators who are creating integration methods for their end users. They are used to track versions of the released module. They are used for identification only and do not affect how the integration module functions.
- **Connection Type**
 - **Use SSL:** Select this option to use Secured Socket Layer (SSL) as the protocol that client software will use to access the deployed module.
 - **One Way:** Select this option to specify that the module should be invoked using One Way SSL (server authentication mode).
 - **Two Way:** Select this option to specify that the module should be invoked using Two Way SSL (mutual authentication mode).



Deltek recommends that all testing should be done first without requiring SSL since SSL setup and troubleshooting is a manual and very time consuming process. Deltek will support SSL-related inquiries only when the SSL certificate is issued and signed by a valid CA.

- **Web Service Authentication Type:** Select at least one option:
 - **Username Token:** Select this option to specify that this module will support Username Token as a way of providing the identity of the requestor executing the methods in this module. Username Token is a standard set by OASIS (Organization for the Advancement of Structured Information Standards).



For more information about Username Token, see <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>.

- **SAML Token:** Select this option to specify that this module will support SAML Token as a way of providing the identity of the requestor executing the methods in this module. SAML (Security Assertion Markup Language) is also a standard set by OASIS.
- **Generate:** Specify the type of interface system that will be generated for the selected module.
 - **Web service:** Select this option to specify that a Web service interface will be generated for the selected integration module. In order to start using this type of the interface, you will need to build and deploy a Web Service jar into Costpoint application which requires redeployment of the whole Costpoint.
 - **Generic Web service:** This is a new option in Costpoint 7.1. Select this option to specify that a Generic Web service interface will be generated for the selected integration module. This module is automatically ready for the usage and do not require build or redeployment of Costpoint application. Please note that these Web Services can be only String based. For more on Generic Web Services, see “Appendix A: Generic Web Services.”
- **Methods Properties:** These properties specify how data is passed for all methods in this module. These properties are available only for the **Web Service** interface type.
 - **Pass Data As (String or Object):** Specify whether you want to pass data and query conditions as a string or as a complex data structure.
 - **String Data Format (Archived Text or Clear Text):** If you select the **Pass Data As String** option, you can specify whether the system should use Archived Text or Clear Text.

The **Archived Text** option can greatly minimize the number of bytes required to pass data (reducing the data to about a tenth of its original size). It works by zipping the data using a standard java ZLIB library (compatible with GZIP and PKZIP) and passing compressed data over the wire. You can zip data using any tool that is compatible with GZIP or PKZIP, and then pass it as an array of bytes to the data import method. For a data export method, you still pass the query condition as a string, but you get an array of bytes as an integration method return. You can unzip this data using any tool that is compatible with GZIP or PKZIP.

You can use Deltek’s test Java client program as a template. Source code for the test Java client program contains a simple Java, as well as .NET C#, equivalent of zip/unzip functions.

- **Add Link:** Click this button to begin adding methods to this integration module. A new blank row displays in the **Assigned Methods** window. Click the drop-down button to select from methods (created in previous steps) to be included in the module.
- **Remove Link:** Click this button to remove an integration method from the module. (This does not delete the method itself).
- **Save Module:** Click this button to save the currently selected module.
- **Close:** Click this button to close the screen and return to the Costpoint Integration screen.

Build/Deploy Integration Modules

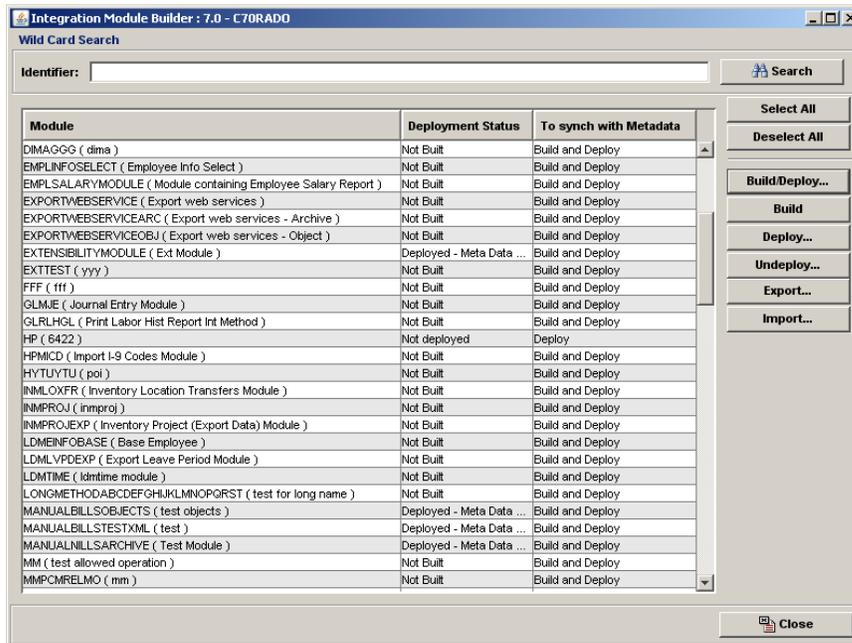
Select the **Build/Deploy Integration Modules** option to display the Integration Module Builder screen. Use this screen to select one or more modules to build and deploy.

Typically, you should test a Web service on a test server before deploying it onto production servers. You must build and deploy the Web service in order to test it.

After a Web service passes testing, you can either export it from the test server and import it onto the production server or deploy it straight to the production server.

This section will help you build and deploy a Web service.

Integration Module Builder



This screen displays all existing modules, with their deployment status and the action recommended to bring their status to the recommended state. The console compares the metadata created for the module with the module JAR file located in the working folder and in the application server enterprise folder to determine the build and deploy status and whether there is a metadata mismatch.

Select one or more module from the list, or perform a search to find the modules you want to build or deploy.

Click **Select All** to select all modules or **Deselect All** to deselect all modules

To search, enter a filter, based on the module ID, in the **Identifier** field, and click **Search**. The filter is applied using the LIKE condition.

Data about Modules

For each module on the screen, you can review the following information:

- **Module:** This is the the module ID and description.
- **Deployment Status/To Synch with Metadata**
 - **Undeployed:** This status can refer to three different situations:
 - The module has not been built. The recommended action is to build the module and deploy it to the server.
 - The module has been built, but not deployed to the server. The recommended action is to deploy the module to the server.
 - The module has been built, but not deployed to the server, and there is a mismatch in the metadata used to build it. The recommended action is to recreate the module, synchronize it with the metadata, and deploy it to the server.
 - **Deployed - Out of synch:** The module has been deployed to the server. However, the JAR file deployed to the server is not in sync with the metadata used to build the module. The recommended action is to recreate the module, synchronize it with the metadata, and deploy it to the server.
 - **Deployed - In synch:** The module has been built and deployed to the server and the JAR file deployed to the server is in sync with the metadata used to build the module. No further action is necessary.
 - **Deployed - No metadata:** The program detects a module JAR file deployed to the server, but there is no corresponding metadata used to build or rebuild the module. The recommended action is to undeploy the module from the server.

Options for Building and Deploying

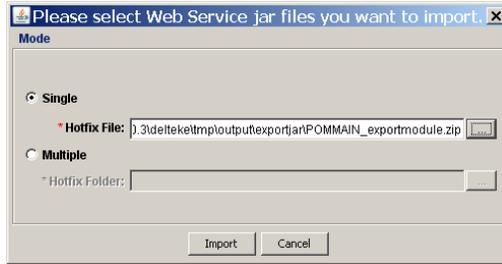
Use these buttons in the building and deploying process:



You must build modules before you can deploy them.

- **Build/Deploy:** Click this button to start the build process. Then deploy the module immediately after the build completes.
- **Build:** Click this button to start the build process. The Building Process dialog box displays, showing you the progress of the build.
- **Deploy:** Click this button to open the Deploy screen. Use this screen to deploy selected modules.
- **Undeploy:** Click this button to undeploy selected modules. The undeploy process removes selected modules from the server and updates the application.xml file.
- **Export:** Package the module and the metadata used to create this module into an installation zip file. This file can then be installed and deployed to another system.
- **Import:** Click this button to deploy a previously exported integration module. The Import process here works the same way as the Deploy Hotfix option in the Costpoint DB Wizard. However, the DB Wizard is a better way to deploy an integration module into a production system. The Import option here is a convenience feature that allows you to quickly deploy an integration module into a local/test system.

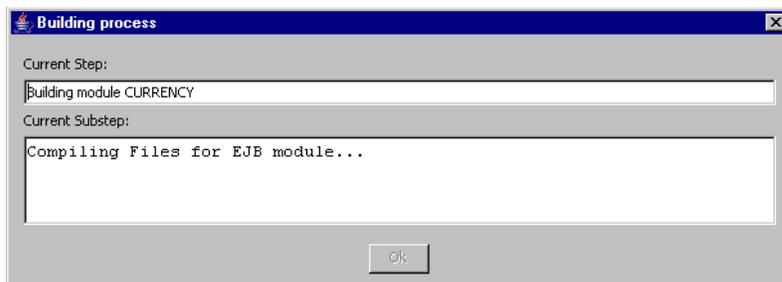
When you click the **Import** button, you will see a simple dialog box where you have to either choose the individual module that you want to import or a folder that contains several modules. Click **Import** to import them.



- **Close:** Click this button to close the screen and return to the Costpoint Integration screen.

Build a Module

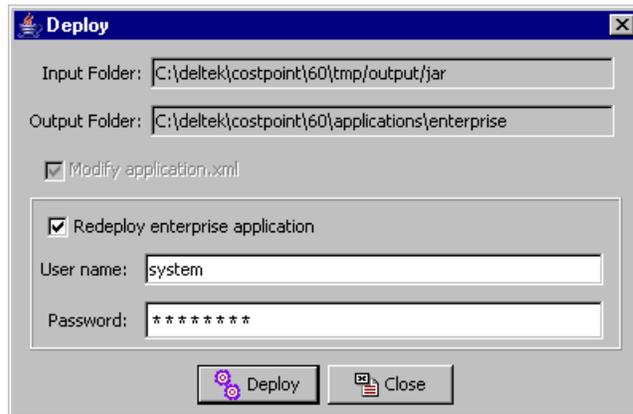
When you start the build process, the Building Process dialog box displays, showing the progress of the build. Building involves several steps, so it is important to wait until the process ends. When the build process is completed, the **Current Step** field displays “Process complete” and the **OK** button is enabled. Click **OK** to exit the dialog box.



Deploy a Module

The deploy process copies previously built modules from an input folder to an output folder. It then registers the modules on the WebLogic server.

If the Costpoint service is running, the process will redeploy the Costpoint application so that newly added modules are available immediately, without restarting the Costpoint service.



The Deploy dialog box contains the following fields:

- **Input Folder:** This field displays the folder where the console expects to find the previously generated modules. The location of the folder comes from configuration settings. You can modify the input folder using the Modify Build Properties screen.



For more information about the Modify Build Properties screen, see the [Modify Integration Console Properties](#) section.

- **Output Folder:** This field displays the folder to which the deploy process will copy the modules. The location of the output folder comes from the Costpoint Home folder set up using the Configuration utility.
- **Modify application.xml:** Select this option to indicate that the deploy process should modify the application.xml file. This option is selected by default.
- **Redeploy enterprise application** —. Select this option to deploy the Web service changes immediately. Weblogic server should be running for this option to work. In case of a clustered configuration, at least the Admin server should be running. With this option, changes are deployed to the server and the entire enterprise application goes through an internal restart process. Any active users or sessions running on Costpoint servers will lose their sessions and the overall process could take some time which is less than the time it takes to restart the service.

If you do not choose this option, configuration changes are made to the server and the new Web service runtime files are copied to the Costpoint home directory. Changes are not effective until the Costpoint service is restarted. Sometimes Weblogic will place a lock on the Web service runtime file and the tool will not be able to replace the file until the corresponding Weblogic server is stopped.

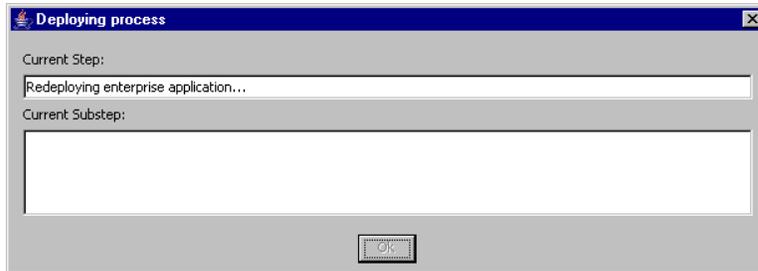
Clients using dedicated servers to host Web services should not select **Redeploy enterprise application**. They should stop the dedicated server (thus releasing any locks placed on the files), deploy the Web service changes using the Integration console, and then restart the dedicated server for the changes to take effect. This approach gives

flexibility in deploying Web service changes without impacting the rest of the cluster and the processes running on Costpoint servers.

- **User name:** Enter the user name that will be used to redeploy the enterprise application. This user must have sufficient rights to redeploy the application. Generally, this can be the same user who starts the application server.
- **Password:** Enter the user's password.

Run the Deployment

Click the **Deploy** button to start the deployment process. You will see the Deploying Process dialog box, which shows you the steps in the process as they are completed.



Locate Modules

After a successful deployment, a Web service module is registered under the sub URL webservices.

For example, if the current URL to log into Costpoint is:

<http://CPWebServer:7009/Costpoint.htm>

Then the Web service component is registered under:

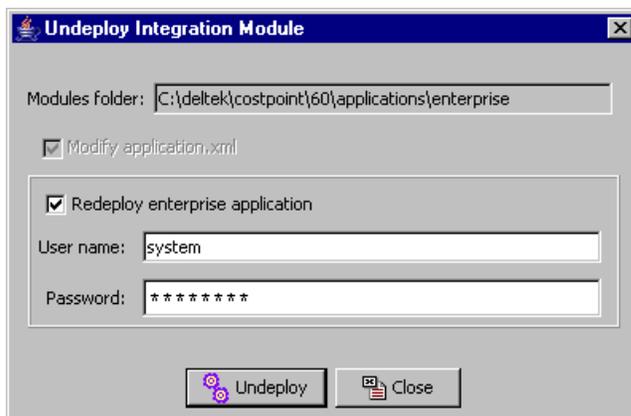
<http://CPWebServer:7009/webservices>

To browse the WSDL for a Web service with the module ID of CURRENCY, use the URL:

<http://localhost:7009/webservices/CurrencyWS?WSDL>

Undeploy a Module

Select **Undeploy** from the Integration Module Builder screen to delete previously deployed modules and un-register them on the WebLogic server. The undeploy process can send a command to the WebLogic server to redeploy the entire Costpoint application so that deleted modules are no longer available on the server.



The Undeploy Integration Module dialog box contains the following fields:

- **Modules Folder:** This field displays the location of deployed modules on the server.
- **Modify application.xml:** This check box indicates that the undeploy process will modify the application.xml file. This option is checked by default.
- **Redeploy enterprise application:** Use this option to instruct the undeploy process to redeploy the entire Costpoint application, to make the undeployed modules unavailable immediately. If you do not select this option, undeployed modules are still available until the application server is restarted.
- **User name:** Enter the user name that will be used to redeploy the enterprise application. This user must have sufficient rights to redeploy the application. Generally, this can be the same user who starts the application server.
- **Password:** Enter the user's password.

Export a Module as a Hotfix

The export process captures the metadata used to create the module and its integration methods, and creates a hotfix file (in the form of a zip file) that is ready to be applied to another system. This process is designed to allow users to create and test modules in a development/testing environment before applying them in a production environment. It is also designed to allow a third-party system integrator or Deltek consulting team to develop integration modules off-site and then redeploy the modules on site.

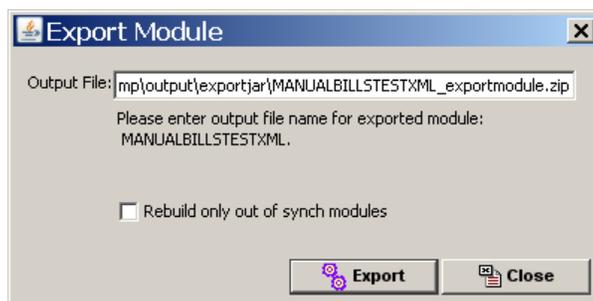
Any module can be selected for export as long as the metadata is not missing. It does not have to be built or deployed. If you select the **Rebuild only out of synch modules** option, the console will only rebuild out of synch modules. Otherwise, the console will rebuild all modules before exporting them.

The final hotfix file (in the form of a zip file) contains a manifest file and a JAR file. The manifest file contains basic information about the JAR file. The JAR file contains the SQL scripts for the metadata and the module JAR file.

The hotfix file is designed to be deployed to the production system via the Costpoint DB Wizard. Select the **Deploy Hotfix** option from the Costpoint DB Wizard menu to deploy the hotfix file. The Wizard will apply the metadata to the production system database and also deploy the module JAR file to the corresponding application server.

To export one or more modules, complete the following steps:

1. Select the modules to be exported, and click **Export**.
2. On the Export dialog box, accept or override the name and location of the hotfix file.
3. Click **Export** to create the hotfix file. One hotfix file will be produced for all modules selected.



Test an Integration Module

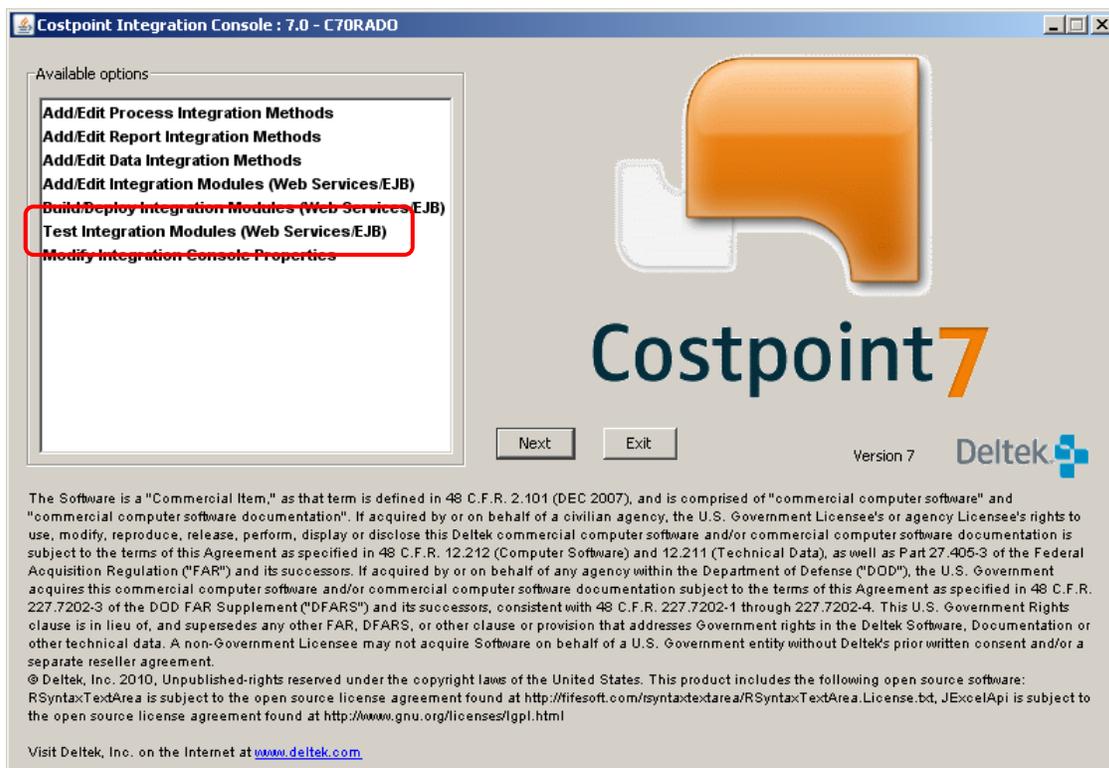
Before you can test an integration module, you must build and deploy it. Typically you deploy the module in a test environment, perform tests, and then deploy the module on a production server.

The main testing steps are:

1. Select the integration module you want to test.
2. Prepare for testing by specifying testing parameters.
3. Run the test.

These steps are described in the sections that follow.

To start the testing process, select **Test Integration Modules (Web Services/EJB)** from the main Integration Console screen.

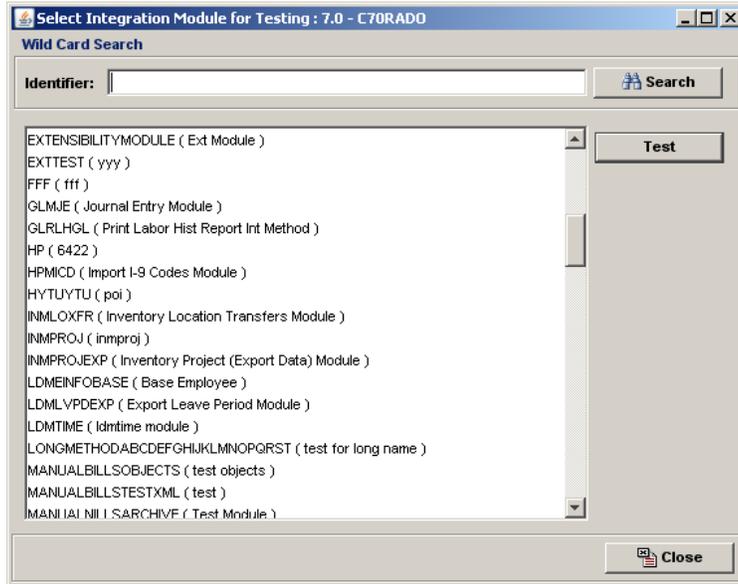


Select an Integration Module for Testing

To select an integration module for testing, complete the following steps:

1. Select **Test Integration Modules (Web Services/EJB)** from the main Integration Console screen, and click **Next**.

The **Select Integration Module for Testing** dialog box displays.



2. Use these fields to select a module for testing:
 - **Identifier:** If the module list is large, enter a filter, based on the module ID, in the **Identifier** field, and click **Search** to filter the list. The filter is performed using the LIKE condition.
 - **Test:** Select a module to test, and click this button to open the Test Module Execution dialog box. Use this dialog box to test/execute the methods in a module.
 - **Close:** Click this button to close the dialog box and return to the Costpoint Integration screen.
3. Click **Test**.



Deltek recommends that all testing should be done first without requiring SSL since SSL setup and troubleshooting is a manual and very time consuming process. Deltek will support SSL-related inquiries only when the SSL certificate is issued and signed by a valid CA.

Prepare to Test an Integration Module

Follow the steps in the previous section to select the integration module you want to test. The Test Integration Module Execution dialog box displays, showing data for that integration module.

Review and enter data on the Test Integration Module Execution dialog box:

- **Module:** This field displays the module ID and description.
- **Web service WSDL:** This field displays the location of the Web service WSDL file.
- **EJB JNDI Name:** This field is disabled and reserved for future use.
- **Method:** If the module contains multiple methods, use the drop-down list in this field to select the method to test.
- **Test:** In Costpoint 7, the only option is to execute the method via the Web service interface. This option is selected by default. In Costpoint 7.1.1 for Generic Web Services, two new ways to invoke WS were introduced: **Rest XML** and **Rest JSON**.
- **Use Default System:** Select this option to specify that the test should use the default system of the Costpoint application server for authentication purposes. The default system is the first system in the system name list in the enterprise.properties file.

In the **Method Parameters** section, you also need to specify the system that this method will be executed against. Generally, if you select this option, the default system should be the same as the system passed in using the method parameters, or the user must exist in both systems.

- **Method Parameters:** Use these fields to specify the values for the method parameters for execution. The first four parameters (**User**, **Password**, **System**, and **Company**) are system parameters that the console automatically generates. They are the same as those needed when you log into the Costpoint application using the Costpoint UI. The last parameter (**Query**) comes from the method definition.
 - **User:** Enter a Costpoint user ID. This user must have the necessary application rights to execute this method. (An integration method always originates from an

application.) In addition, this user must be authorized to execute an integration method.



See the online help for the **Allow Application Access via Integration Services** option in the Maintain Users application for more details.

- **Password:** Enter the Costpoint user password.
- **System:** Enter the name of the system to which you want to connect.
- **Company:** Enter the company ID that you want to use.
- **Query/Document:** Click the **View/Edit** button to display the Method Parameter Editor dialog box described under [Specify the Query Parameters Used to Filter Integration Data](#).

Save and Reuse Testing Parameters

You can save and reuse any set of testing parameters for future use.

- **Save Params:** Click this button to save all testing parameters (except the password) on the Test Integration Module Execution dialog box for subsequent execution or re-execution. The data is saved locally in the user's home folder on the workstation that runs the console. For Windows workstations, the data is saved at C:\Documents and Settings\- **Load Params:** Click this button to load a previously saved configuration.

Preview - Testing Parameters

- **Preview:** Click this button to preview the complete SOAP packet that will be submitted to the server. The **View/Edit** screen only shows a data portion of the message that client needs to submit to Cospoint. In the **Preview** screen, the complete message including security parameters, authentication parameters (such as user id, company, and system), and the data parameters are displayed.

Specify the Query Parameters Used to Filter Integration Data

The Integration Console lets you set up query filters to narrow down the pool of data on which the method will act. To set the query filters, click the **View/Edit** button on the Test Integration Module Execution dialog box to display the Method Parameter Editor screen.



The information you see on the screen depends on the method type that you are testing:

- Report and process methods
 - Without saved parameters: All of the parameters to run the report or process must be supplied through the Web service client.
 - With saved parameters: The method uses a set of parameters saved previously (via either the Costpoint UI or a Data Import Integration method).
- Data export method
- Data import method

Report or Process Method Without Saved Parameters

For this method, you can edit parameters using the Wizard or XML Editor tab on the Method Parameter Editor screen:

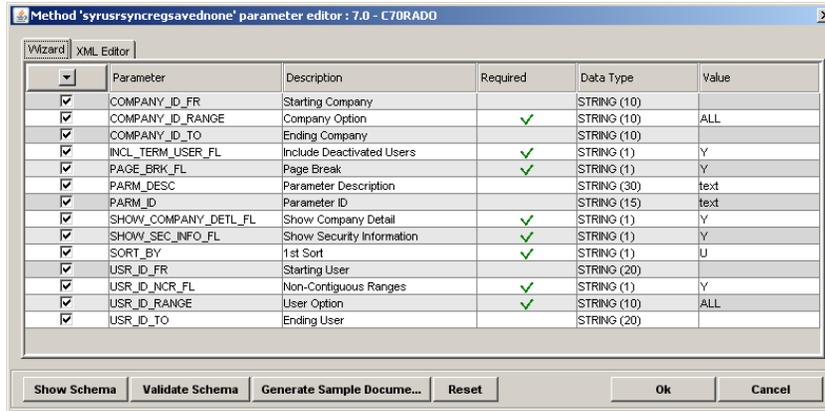
- The Wizard tab displays the parameters as a table that you can edit.
- The XML Editor tab displays the actual XML code and lets you edit the code.

Use the Wizard Tab

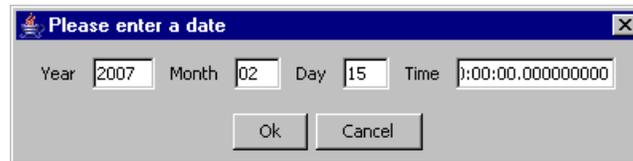
If you use the Wizard tab, complete the following steps:

1. Identify the parameters that you want to pass. For each of these parameters, you must enter a specific value in the **Value** column. If you do not pass a parameter, the system will set this parameter to a default value (if it has any) or to the null value.
2. To make the column selection process more efficient, click the down arrow at the top left of the table, which displays the following options:
 - **Include Only Editable Columns:** Select this option to include only editable columns.
 - **Include Only Required Columns:** Select this option to include only required columns.
3. Enter values for the parameters.

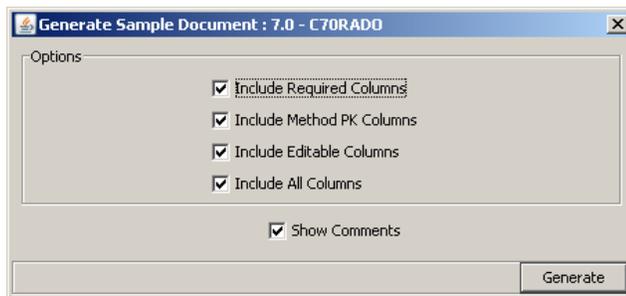
For convenience, the console provides a drop-down list for the **Value** field whenever possible. A drop-down is provided if, in the Costpoint UI, a field is of one of the following types: combo box, range control combo box, list box, check box, or radio button.



You must use a specific format to enter dates as values, because the console uses the `java.util.Calendar`, which accepts dates in the JDBC date format of `yyyy-mm-dd hh:mm:ss.ffffff`. To make it easier to enter dates in this format, the console displays the following dialog box.



4. (Optional) Click **Show Schema** to see the method XML schema.
5. Click **Validate Schema** to test the method XML schema using sample data. The sample data gets transformed into an XML document before validation.
6. (Optional) Click **Generate Sample Document** to generate a sample input document.



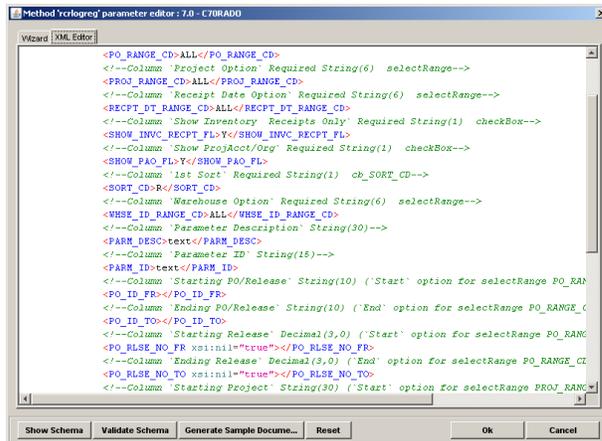
The Generate Sample Document screen lets you select from among several options, which the Integration Console will remember until you restart the console:

- **Include Required Columns:** Select this option to generate sample data for the required columns.
- **Include Method PK Columns:** Select this option to generate sample data for the method PK columns.
- **Include Editable Columns:** Select this option to generate sample data for the editable columns.
- **Include All Columns:** Select this option to generate sample data for all the columns.
- **Show Comments:** Select this option to include XML comments in the generated XML. The comments will be visible only when you are in the XML Editor mode.

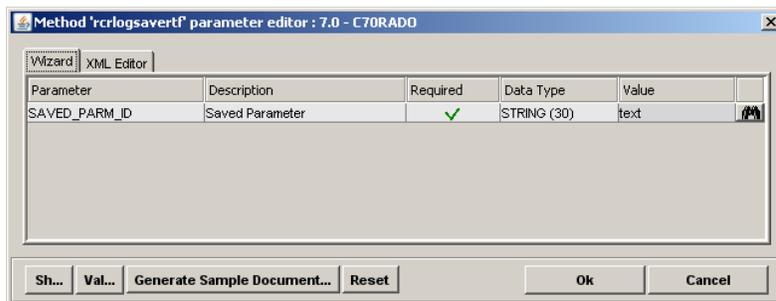
7. (Optional) Click **Generate** to generate the sample document.
8. Take one of the following actions:
 - Click **Ok** to save the generated query condition into the XML Editor tab. The query condition will be translated into an XML document.
 - Click **Reset** to reset all the changes since you opened the editor.
 - Click **Cancel** to discard all the changes and close the dialog box.

Use the XML Editor Tab

The XML Editor tab allows you to modify the XML document directly.

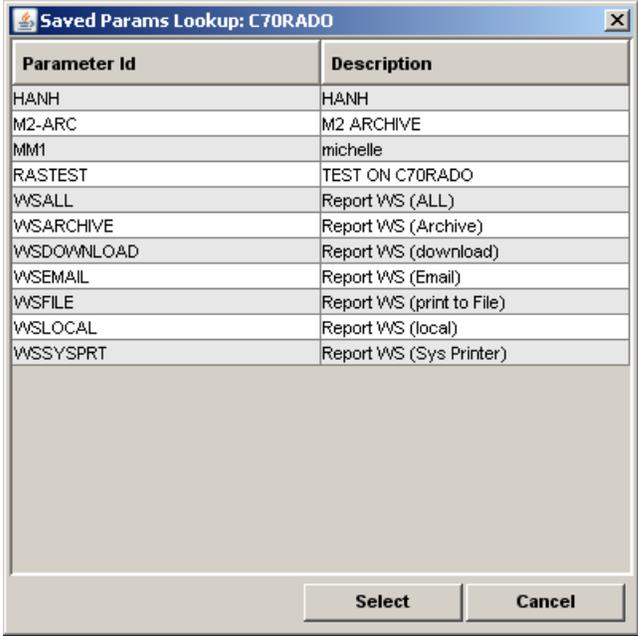


Report or Process Method With Saved Parameters



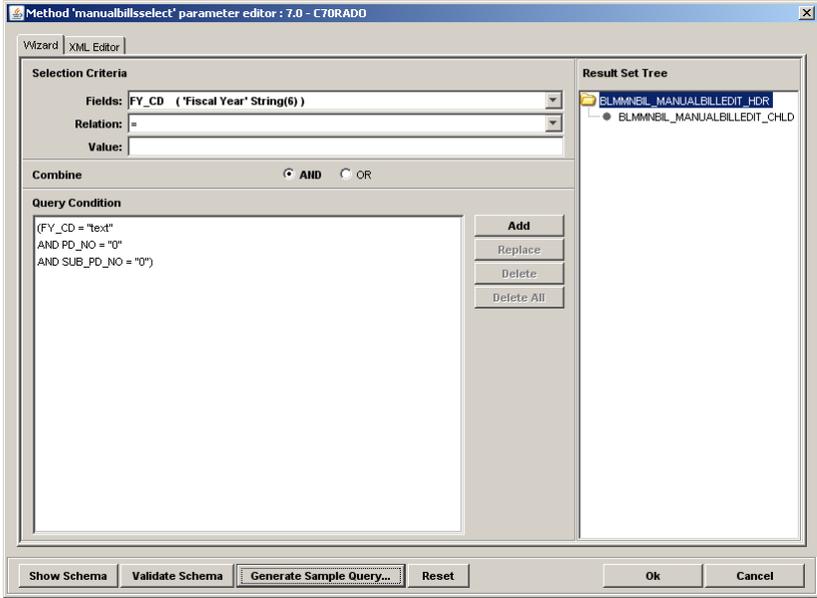
If you are testing a report or process method that uses saved parameters, a single parameter, named **SAVED_PARAM_ID**, displays in the Method Parameter Editor screen. In the **Value** column, enter the parameter ID to be executed. The parameter must have been saved previously (via either the Costpoint UI or a Data Import Integration method).

Click the binoculars icon to display previously saved parameters. Select a parameter ID from this lookup.



Data Export Method

If you are testing a data export method, you must construct a query string based on the columns available for export.



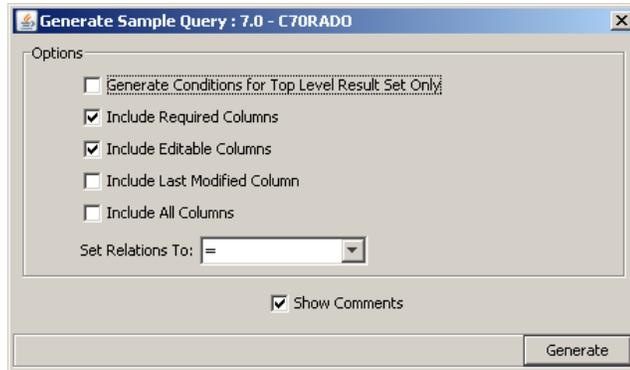
You can build the query using the Wizard or XML Editor tab on the Method Parameter Editor screen:

- The Wizard tab provides a series of query-building fields that let you create the query.
- The XML Editor tab displays the query condition string and lets you directly edit the code. The changes you make on this tab are synchronized back to the Wizard tab, so you can use both methods in the same session.

Use the Wizard Tab

To build the query using the Wizard tab, complete the following steps:

1. (Optional) Click **Show Schema** to see the method XML schema.
2. (Optional) Click **Validate Schema** to test the method XML schema using sample data. The sample data gets transformed into an XML document before validation.
3. Click **Generate Sample Query** to generate a sample query. The Generate Sample Query dialog box lets you select from among several options, which the Integration Console will remember until you restart the console:



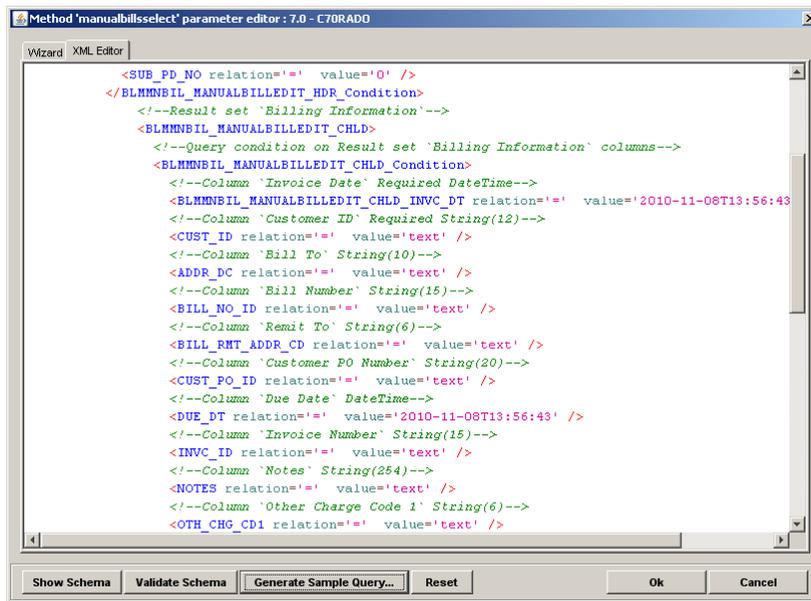
- **Generate Conditions for Top Level Result Set Only:** If you have several result sets, select this option to generate query conditions only for the top level result set. Leave this option cleared to generate query conditions for all result sets.
 - **Include Required Columns:** Select this option to generate sample query conditions on required columns.
 - **Include Editable Columns:** Select this option to generate sample query conditions on the editable columns.
 - **Include All Columns:** Select this option to generate sample query conditions on all columns.
 - **Include Last Modified Column:** Select this option to generate a sample query condition on the Last Modified Pseudo Column if a particular result set has one.
 - **Set Relations To:** Any condition on a result set column has several parts: column, relation operator, and value. Several relation operators (for example, **is null**) do not accept values. Select the **Set Relations To** option to choose which relation operator to use in generating sample query conditions. If you select **random**, the console will use a random algorithm when generating a relation operator. You can also select one of the common operators for all data type relation operators (for example, = (equal)).
 - **Show Comments:** Select this option to include XML comments in the generated XML. The comments will be visible only when you are in the XML Editor mode.
6. Click **Generate** to generate the sample query.
 7. If your export method involves more than one result set, select the result set from the **Result Set Tree**. Then build the **Query Condition** for the result set using the **Selection Criteria** fields:
 - **Fields:** From the drop-down list, select the fields available in the export method.
 - **Relation:** From the drop-down list, select the relation condition.

- **Value:** Enter the corresponding value for the selected field and relation.
 - **Combine:** Use the **AND** and **OR** options to add additional conditions.
 - **Add:** Click this button to add the condition to the **Query Condition** box.
 - **Replace:** Highlight an existing condition in the **Query Condition** box and click this button to replace it with the new condition.
 - **Delete:** Highlight an existing condition in the **Query Condition** box and click this button to delete it.
 - **Delete All:** Click this button to clear all conditions in the **Query Condition** box.
8. Take one of the following actions:
- Click **Ok** to save the generated query condition into the **Query** field on the Test Module dialog box. The query condition will be translated into an XML document.
 - Click **Cancel** to discard all the changes and close dialog box.

Use the XML Editor Tab

To review or edit the query using the XML Editor tab, complete the following steps:

1. Click **View/Edit** on the Test Integration Module Execution dialog box.
2. Click the XML Editor tab, which displays the query condition string.



3. Review and, if necessary, edit the query. Any changes you make on this tab are synchronized back to the Wizard tab, so you can use both methods in the same session.

Data Import Method

If you are testing a data import method, you must specify the data that will be imported. You do this by directly editing the XML document that contains the data for your import method.

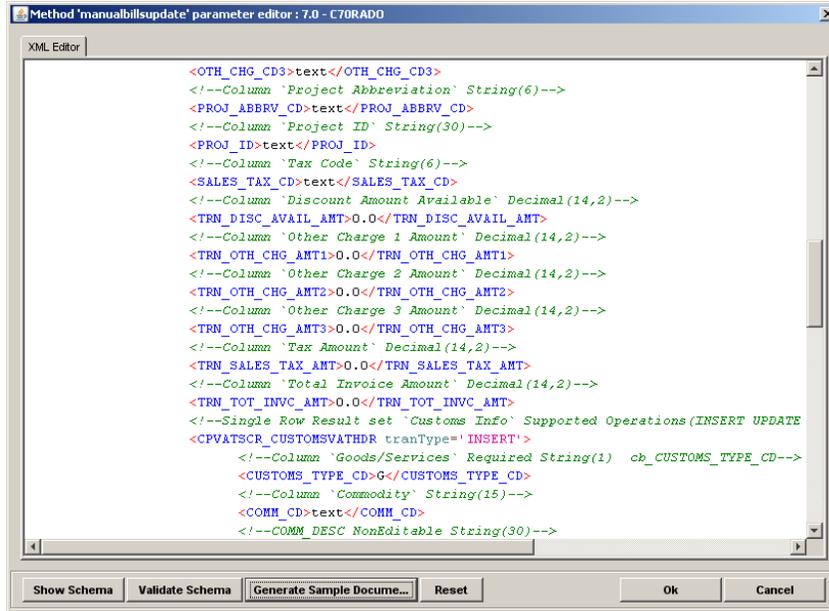
To specify the data that will be imported, complete the following steps:

1. Click **View/Edit** on the Test Integration Module Execution dialog box.
2. (Optional) On the XML Editor tab, click **Generate Sample Document** to generate a sample XML document to edit. See step 6 for more information about generating sample documents.
3. Edit the XML document that contains the data for your import method.

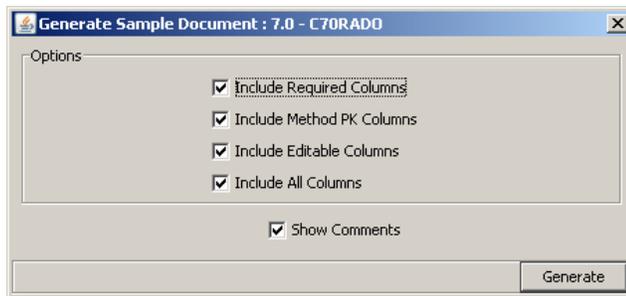
You can use four operations with the import method: **INSERT**, **UPDATE**, **DELETE**, **MERGE**, and **SELECT**. Each result set that you specify has a tranType attribute. Set this attribute to the operation that you want to perform in this test. If you omit this attribute, the console assumes you are specifying an INSERT operation.

For each column included in the import method, include the tag and the value to be used:

- **INSERT**— All required columns (as indicated in the XML schema) should be present and their values included in the document. A required column is indicated with the attribute **nillable='false'**.
- **UPDATE** operation: Only columns that need to be updated should be included in the document. If a column needs to be updated to a blank value, include the tag in the document but omit the value. For example, enter **<NOTES></NOTES>**.
- **MERGE** operation: Use this to INSERT new records or UPDATE existing records, depending on whether or not a record already exist.
- **DELETE** operation: All columns specified as primary keys (**PK**) when you created the import method must be present and have values.
- **SELECT** operation: All columns specified as primary keys (**PK**) when you created the import method must be present and have values. You must use the SELECT operation when you want to update a child result set only. In this case, you use the SELECT operation to select a row in the parent result set, and then you use INSERT, DELETE, or UPDATE to modify a child row.



4. (Optional) Click **Show Schema** to see the XML schema created for this import method (with all the selections and settings) when the import method was created.
5. Click **Validate Schema** to validate your document to see if it conforms to the XML schema for this method.
6. Click **Generate Sample Document** to generate a sample input document. The Generate Sample Document screen lets you select from among several options, which the Integration Console will remember until you restart the console:

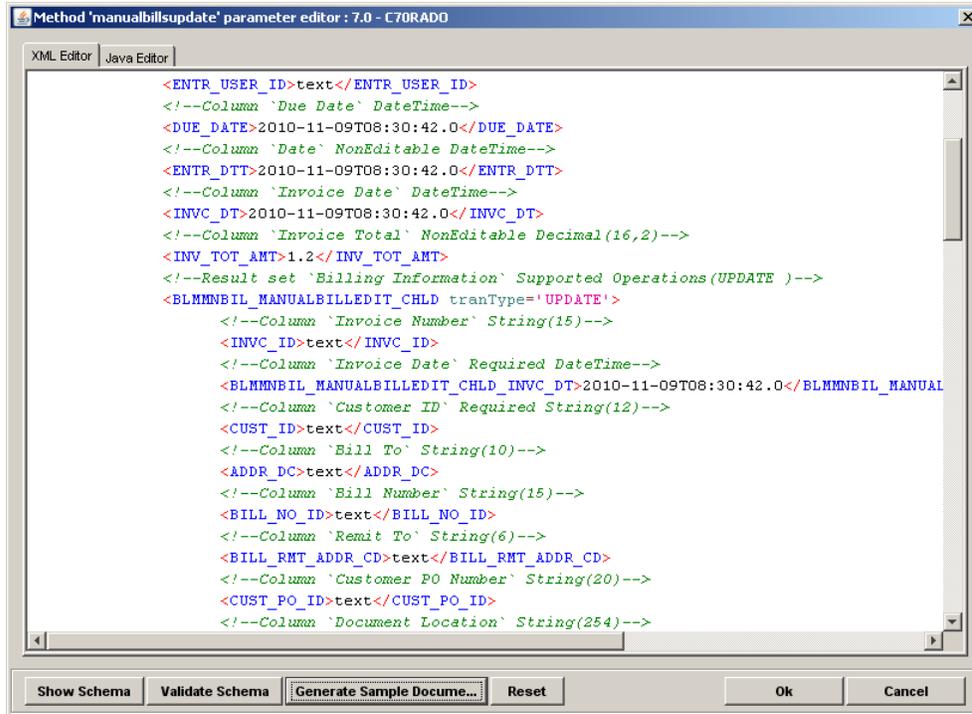


- **Include Required Columns:** Select this option to generate sample data for required columns.
 - **Include Method PK Columns:** Select this option to generate sample data for method PK columns.
 - **Include Editable Columns:** Select this option to generate sample data for editable columns.
 - **Include All Columns:** Select this option to generate sample data for all columns.
 - **Show Comments:** Select this option to include XML comments in the generated XML. The comments will be visible only when you are in the XML Editor mode.
7. Click **Generate**.

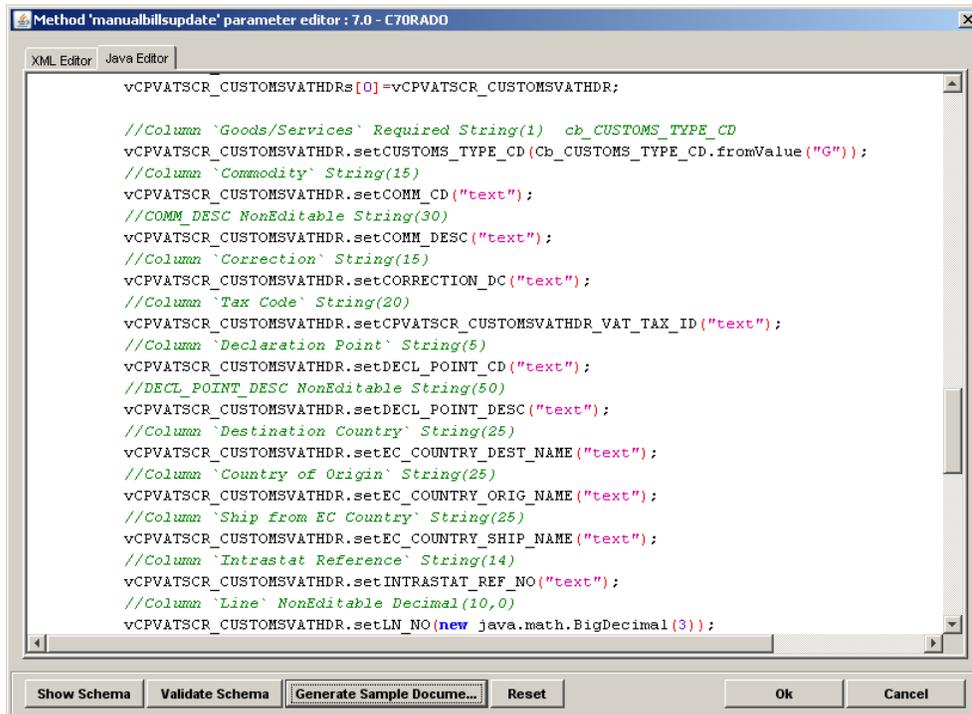
The Integration Console generates one row for each result set that you included in the method.

If you decided to pass data as an object, the Method Parameter Editor screen displays. The options on the screen depend on the method type. Plus, there is an extra view that represents your query condition or input document as a Java Class that will be invoked by the test method program to generate query condition objects and pass them to the integration method.

The following is an XML Editor view of the data import method editor.



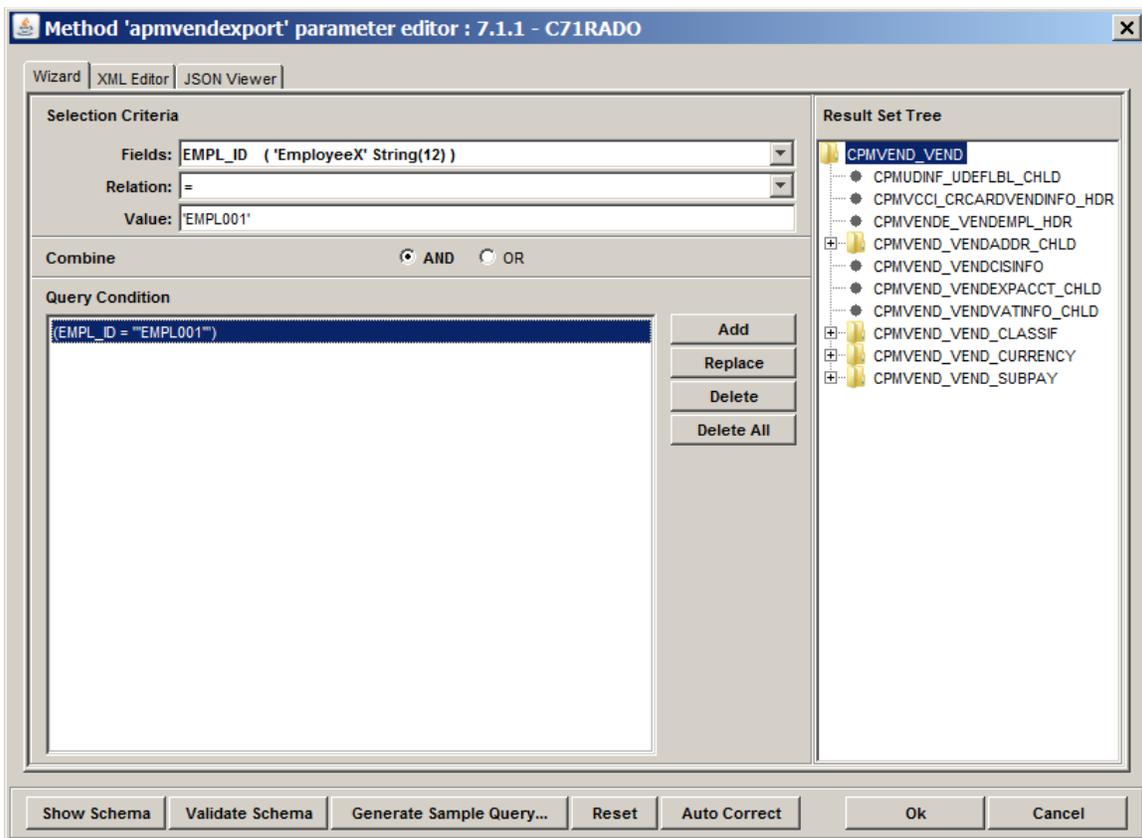
The following is a Java Editor view of the data import method editor.



The buttons at the bottom of the screen let you perform these actions:

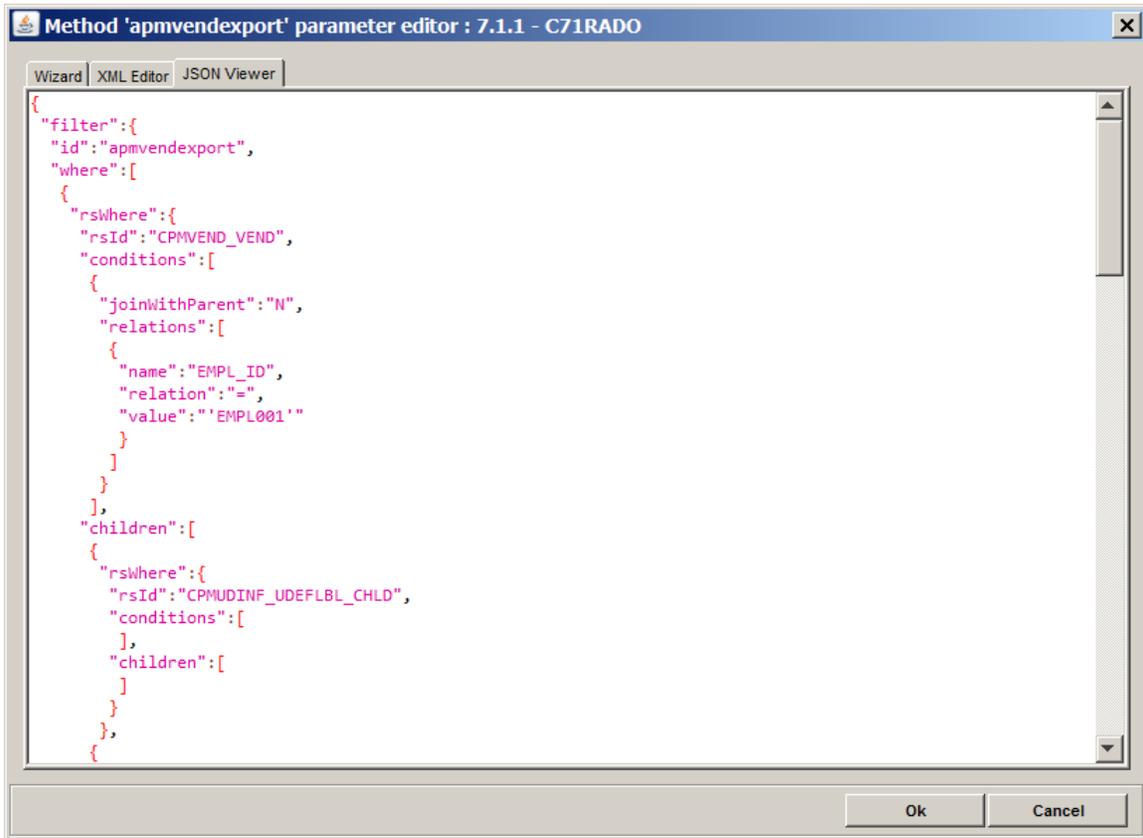
- **Show Schema:** Click this button to display the data schema for your method.
- **Validate Schema:** Click this button to validate your Java Class by compiling it.
- **Generate Sample Document:** Click this button to generate a sample document.
- **Ok:** Click this button to close the Editor screen after validating the Java Class by compiling it and updating your query condition. If you are currently in the XML View, the document will be translated into Java Class format before validation.
- **Cancel:** Click this button to close the screen without updating your query condition.

In addition to the standard **SOAP XML** format, the Generic Web Services product supports invoking Web Services in **Rest XML** and **Rest JSON** formats. To learn more about these options, Deltek recommends using the View/Edit Query dialog box, which can show you how to construct XML/JSON for an entered query condition.

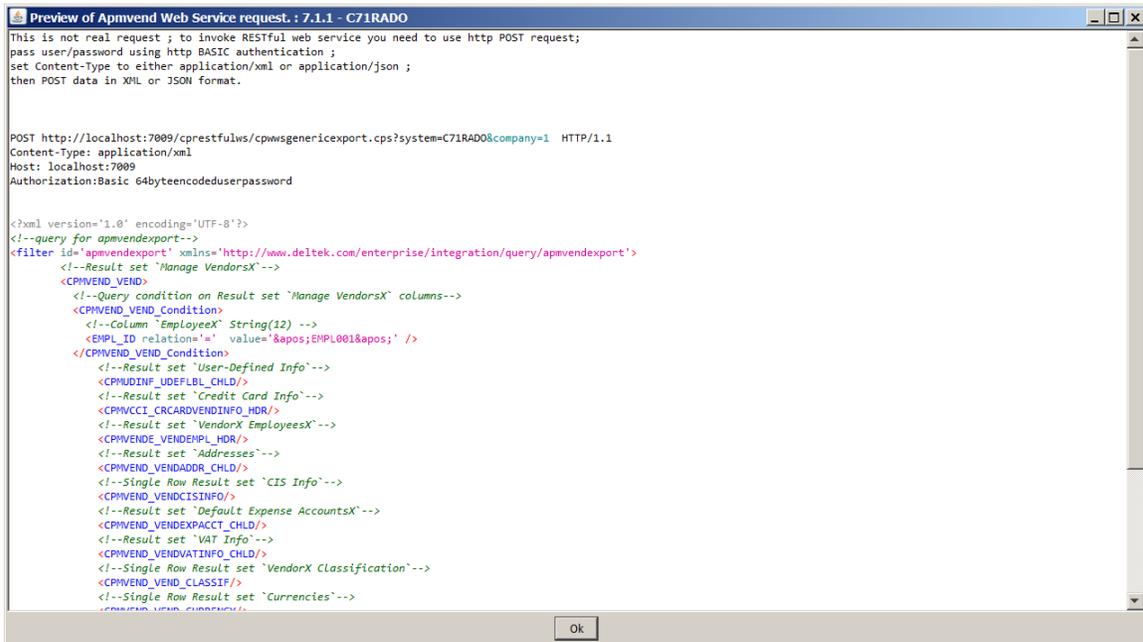


Specify the Query Parameters Used to Filter Integration Data

On the XML Editor/JSON Viewer tab:



And use the Preview dialog box for Rest XML:



Or **Rest JSON** format:

```

Preview of Apmvend Web Service request. : 7.1.1 - C71RADO
This is not real request ; to invoke RESTful web service you need to use http POST request;
pass user/password using http BASIC authentication ;
set Content-Type to either application/xml or application/json ;
then POST data in XML or JSON format.

POST http://localhost:7009/cprestfulws/cpwsgenericexport.cps?system=C71RADO&company=1 HTTP/1.1
Content-Type: application/json
Host: localhost:7009
Authorization:Basic 64byteencodeduserpassword

{
  "filter":{
    "id":"apmvendexport",
    "where":[
      {
        "rsWhere":{
          "rsId":"CPMVEND_VEND",
          "conditions":[
            {
              "joinWithParent":"N",
              "relations":[
                {
                  "name":"EMPL_ID",
                  "relation":"=",
                  "value":"EMPL001"
                }
              ]
            }
          ]
        },
        "children":[]
      }
    ],
    "children":[]
  }
}
    
```

Ok

Those Preview dialog boxes contain more information on how to invoke Web Service in the desired format.

Run the Test of an Integration Module

Click **Test** on the Test Integration Module Execution dialog box to start method execution. The console performs basic parameter validation checks to ensure that:

- Required parameters are not null
- Entered parameters represent valid values according to their data types (double, calendar, string).

If no errors are found, the Testing Process status screen displays, showing the progress of the execution.

For Web Services troubleshooting purposes, you can generate special Debugging Web Services log file that can help you troubleshoot Web Services invocation problems. This Option can be turned on or off in the Configuration Utility » Logging tab.



For more information please see *Deltek Costpoint Configuration Utility* guide.

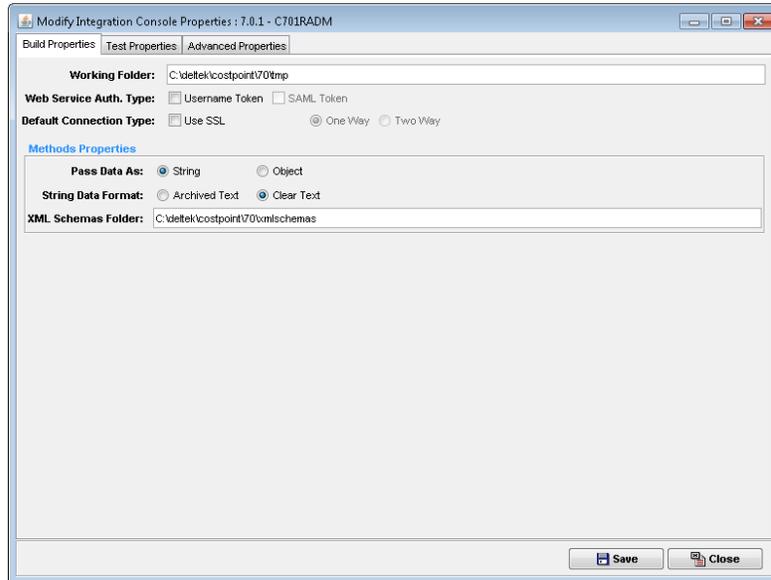
Modify Integration Console Properties

Environment settings used by the Integration Console are stored in Costpoint configuration files. This section describes the settings that can be modified in this tool.

Select **Modify Integration Console Properties** from the main Integration Console screen, and click **Next**.

BuildProperties Tab

Use this tab to modify the default settings used in the build process.



The fields on the screen are:

- **Working Folder:** Enter the folder where the Integration Console puts temporary files and folders.
- **Web Service Authentication Type:** This is the default authentication token supported for a Web service. Select **User Name Token**. It is the only option available in this release.
- **Default Connection Type:** For an SSL connection, select this option and indicate if you will use SSL **One Way** or **Two Way**. Do not select this option for a non-SSL connection.
- **Methods Properties:** Specify the default settings used when you create a new method. You can change any default when you create the method.
 - **Pass Data As (String or Object):** Specify whether you want to pass data and query conditions to the generated integration module's methods as a string parameter or as a complex data structure.
 - **String Data Format (Archived Text or Clear Text):** Select **Archived Text** to take advantage of the archive option, which greatly minimizes the number of bytes required to pass data.



See the [Build/Deploy Integration Modules](#) section for more details.

- **XML Schemas Folder:** Specify the location where you want to store generated XML schemas for all data integration methods. This folder automatically becomes a subfolder for each module created.

Test Properties Tab

Use this tab to modify default settings used in the test process.

The fields on the screen are:

- **User:** Enter the default user name. This is the user ID that is used to connect to Costpoint Server to run the test. This user must be authorized (in the Costpoint Manage Users application) to access Costpoint via Integration Services.
- **Password:** Enter the password for the default user.
- **Company:** Enter the default company name. The Integration Console needs this information to execute modules.
- **Use Default System:** Selecting this check box specifies that the test will connect to the default system of the Costpoint application server for authentication purpose. The default system is the first system in the system name list in the enterprise.properties file. You still need to specify, in the method parameters, the system that this method should be executed against. Generally, if you select this option, the default system should be the same as the system passed in the method parameters.
- **SSL Client Properties:** If SSL is the chosen protocol, these settings are used by the console during module deployment and execution.
 - **Trust Keystore:** Specify the SSL properties that are used in both One Way and Two Way SSL protocols. Supported Trust Keystore options are:
 - Use Standard Java Trust
 - Use WebLogic Demo Trust
 - Use Custom Trust

Depending on the chosen Trust Keystore, some additional fields in this section may become enabled and editable.

- **Type:** The file format of the Trust Keystore (for example, jks)
- **Name:** The name of the Trust Keystore file

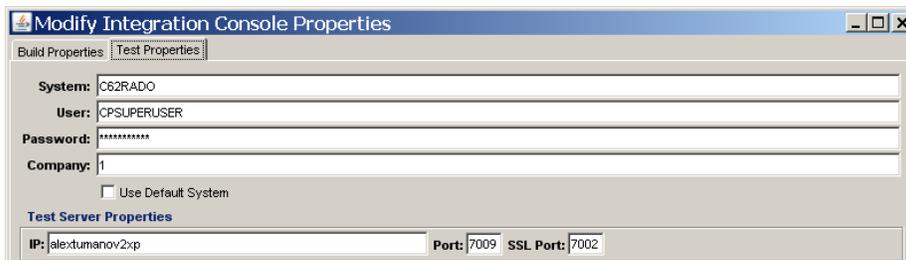
- **Password:** The password of the Trust Keystore file (default password for java key store is **changeit**).
- Identity Keystore (For Two Way SSL Only)

If you are using the Two Way SSL protocol, you must specify the following parameters, which are used only in Two Way SSL:

 - **Type:** The file format of the Identity Keystore (for example, jks)
 - **Name:** The name of the Identity Keystore file
 - **Password:** The password of the Identity Keystore file (the default password for java key store is **changeit**).
- **Additional Parameters:** Use this field to pass any additional SSL parameters.

Weblogic Cluster Specific Test Server Properties

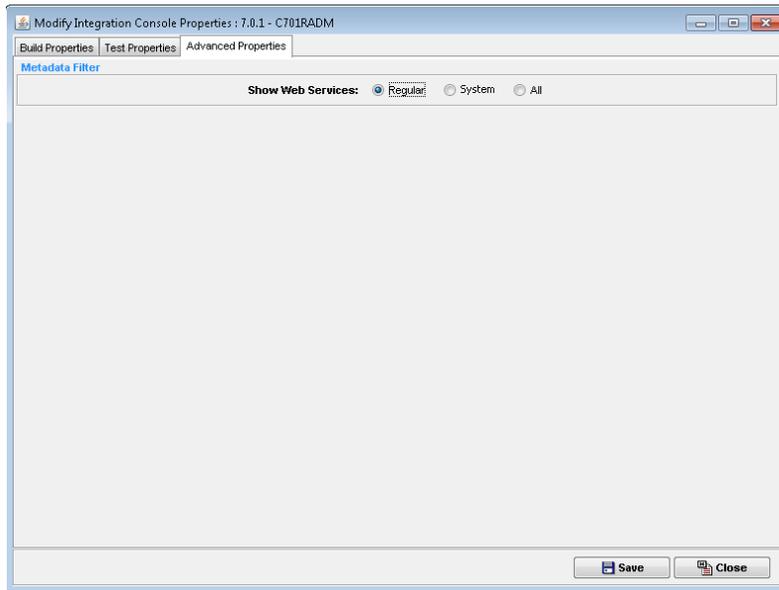
If you deploy integration modules into a Weblogic cluster, the default test server properties used in a single server configuration do not work. In a single server configuration, the integration console should point to the single WebLogic server. In a cluster configuration, the integration console should point to one of the node servers in the cluster.



These fields appear at the bottom of the Test Properties tab of the Modify Integration Properties screen:

- **IP:** The IP address of the server in the cluster to be used to test the integration module
- **Port:** The port of the server in the cluster to be used to test the integration module
- **SSL Port:** The SSL Port of the server in the cluster to be used to test the integration module if it has been configured to use SSL

Advanced Properties Tab



Costpoint 7.01 contains a few prebuilt Web services that are included in the standard installation. These Web services provide interfaces to Deltek partnering products and are considered part of the system. The definition of those modules can be viewed in the Integration Console, but they are not shown by default. Use this screen to control that settings.

- **Show Web Services:** Select from the options:
 - **Regular:** This is the default option. Only the client-defined Web services are displayed in the Integration Tool.
 - **System:** Use this option to display only the prebuilt Web services built by Deltek. You should not modify these prebuilt Web services as it could break the prebuilt functionality.
 - **All:** Use this option to display all client-defined and prebuilt Web services.

Appendix A: Generic Web Services

Costpoint supports object- and string-based Web services. String-based Web services pass data as XML documents encoded as strings while object-based Web services pass XML documents directly. This means that for string-based Web services, WSDL does not fully describe the structure of input/output documents; they are just strings. To generate a string Web service client, program developers should look at separate XML schema documents that describe the structure of input/output documents. At the same time, it allows you to keep WSDL simple and you can make some changes to the input/output documents structure without changing WSDL.

Currently, Costpoint requires you to rebuild/redeploy an object-based Web service after any change to the method/module it contains. For a string-based Web service, Costpoint does not require you to rebuild/redeploy the Web service as long as you only make minor changes to the method structure. But, if you make major changes to the method structure (the one that affects Web service WSDL—for example, add/remove the method), you need to rebuild/redeploy the Web service.

For either type of Web service, Costpoint currently generates a “real” Web service that you need to deploy to the application server using the Deploy Hotfix utility of the DBWizard or the deploy/import functionality of the Integration Console. Along with the Web service jar, costpoint also maintains Web service metadata that it stores in several ADMIN tables and uses in design/run time. This means that while Costpoint supports multiple systems on the single Weblogic server, this module does not work with Web services. Costpoint declares a Web service in the ADMIN schema of a single system, but when you deploy a Web service, costpoint makes it a part of the enterprise application, so it is effectively deployed to all systems. This complicates Web service management in the multisystem environment.

To get around these limitations, Deltek introduced new type of generic Web services. Two special string-based Web services were created to handle all requests to the generic Web services:

- Cpwwsgenericmodule
- Cpwwsgenericmodulesec

These handle all generic Web service requests over HTTP and HTTPS (One Way SSL). Each module has several methods that handle particular types of Web service requests :

- cpwwsgenericexport: Supports data export Web services
- cpwwsgenericimport: Supports data import Web services
- cpwwsgenericprocess: Supports process Web services
- cpwwsgenericreportnoretur: Supports report Web services that do not return the report
- cpwwsgenericreporthtmlreturn: Supports report Web services that return report in HTML format
- cpwwsgenericreportpdfreturn: Supports report Web services that return the report in PDF format
- cpwwsgenericreportexcelreturn: Supports report Web services that return the report in Excel format

You define Web service methods the same way as before, but when you create a Web service module, you have an option to create a **Generic Web Service** with string-based parameters passed as clear text. After you define the **Generic Web Service**, you can test it right away without the need to build or deploy it. This is possible because generic web services do not generate ‘real’ Web services. Costpoint only stores metadata in the system tables that describe the Web service, but to invoke such a Web service, you need to invoke methods of a predefined system Web service (Cpwwsgenericmodule or Cpwwsgenericmodulesec). At run time, Costpoint

looks at the incoming document to figure out which generic web service it belongs to. After that, Costpoint proceeds as if you invoked a normal string-based Web service. When you are ready to move the Web service to the production environment, you need to export it from the test system using the Export functionality of the Integration Console and then deploy it to the production Weblogic server using the Deploy Hotfix utility (or use the Import functionality of the Integration console). If you export only generic web services, the Deploy Hotfix utility will not ask you to restart Weblogic (if it is currently running). You will need to use the Rebuild Global Settings application instead and run the Reload global settings action to refresh the Web service metadata after deploying the Web services.



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

deltek.com