Deltek

Speed. Clarity. Control.

Costpoint Extensibility Designer User Guide

Costpoint Extensibility Designer User Guide

26 March 2026

Deltek

Contents

Deltek

Costpoint Extensibility Designer User Guide

Overview

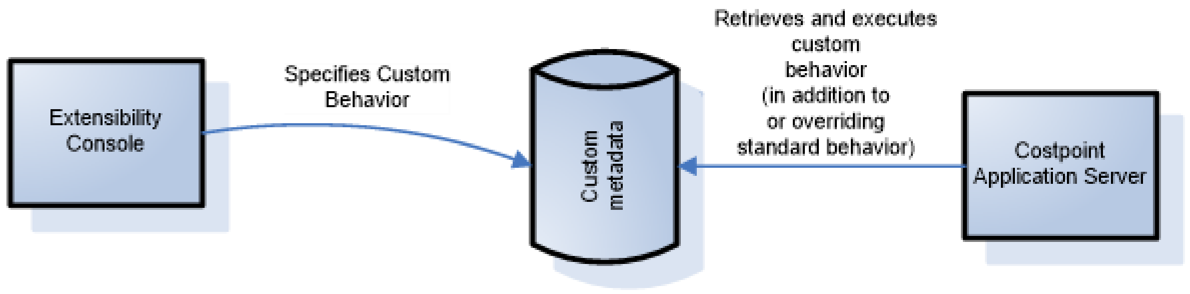
The Costpoint Extensibility Designer (or Extensibility Console) allows a custom developer to create changes to screen objects and layout and register additional logics to be executed for the application. Changes are stored in custom metadata schema and can be retrieved and packaged in an installation package using this same tool.

> Note: Customers who would like to take advantage of the Extensibility functionality will need to get a new License that will enable this Extensibility functionality. This additional license is free, but will require the customer to sign an additional agreement with Deltek. For more details, customers should contact their Account manager at Deltek.

This tool must be run from the server by a developer/consultant with administrative rights to the Costpoint Server. It is not intended for production environment.

> Note: For all system objects like result sets, objects, actions, reports, and so on, the only characters allowed are uppercase and lowercase letters (A through Z), numbers (0 through 9), and underscore (_).

> Tip: For most data entry text fields, increase the size of the dialog box by double-clicking with the right mouse button.


Extensibility Console
Specifies Custom Behavior
Custom metadata
Retrieves and executes custom behavior (in addition to or overriding standard behavior)
Costpoint Application Server

Process/Workflow Diagram

The following denotes basic steps or events related to usage of the Extensibility Console

1. A user creates an extensibility project in this tool. A project consists of one or more units.

2. Each unit constitutes a set of custom behavior to be applied to a UI profile. If all users should have the same behavior, only one unit is needed. If the behavior is different by company/user group or role, multiple units are needed for each unique set of behavior.

3. A unit can contain customization for one or more applications.

4. The user selects a particular application screen and designs the changes. Changes are saved separately in the ADMIN metadata schema.

5. After designing the changes, the user can create an installation package for the project which can be installed to another Costpoint system.

References

Before beginning the process of extending Costpoint Applications, Deltek recommends reviewing the following manuals to better understand overall architecture of Costpoint and what types of objects (for example, applications, result sets, result set trees, actions, reports) are present in Costpoint and what types of links exist between those objects:

- Costpoint 8.2 Screen Customization And Business Logic Extensibility
- Costpoint 8.2 Extensibility Designer Quick Start Guide
- Costpoint 8.2 Extensibility Designer Coding Guide

The Extensibility Designer tool is a client server program created using the standard Java Swing interface and requires the Java Runtime Engine (JRE).

The following tools must be installed on the client machine where you plan to run the Extensibility Designer tool:

- Access to Oracle's WebLogic server installation (typically located under the C:\Oracle\Middleware12.2.1.4)
- Full access to the Costpoint installation folder (typically located under C:\deltek\costpoint\82)
- Access to the CPWebExtDesigner.cmd utility (typically located under C:\deltek\costpoint\82\bin)

Enable a Costpoint User to Run the Extensibility Console

The Extensibility Console requires a login credential. You must enable a user in Costpoint to have access to the utility.

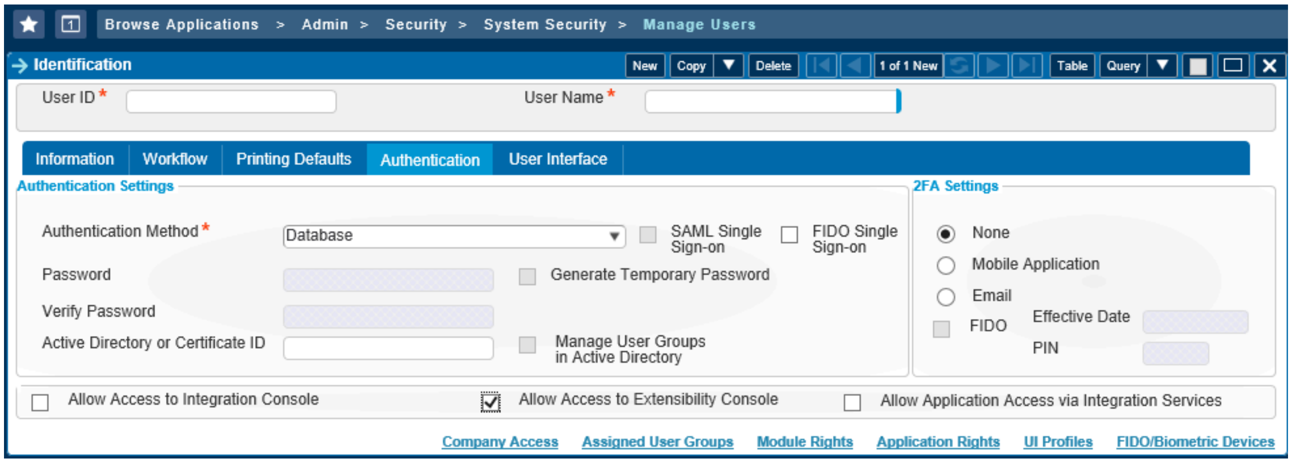
To enable a user to run the Extensibility Console:

1. In Costpoint, go to the Manage User application.

2. Query and select the user that will be accessing the Extensibility Console.

3. Click the **Authentication** tab.

4. Select the **Allow Access to Extensibility Console** check box.



User ID *
User Name *

Authentication Settings

Authentication Method * Database
SAML Single Sign-on
FIDO Single Sign-on
Password
Generate Temporary Password
Verify Password
Active Directory or Certificate ID
Manage User Groups in Active Directory

2FA Settings

[✓ None] Mobile Application Email
FIDO
Effective Date
PIN

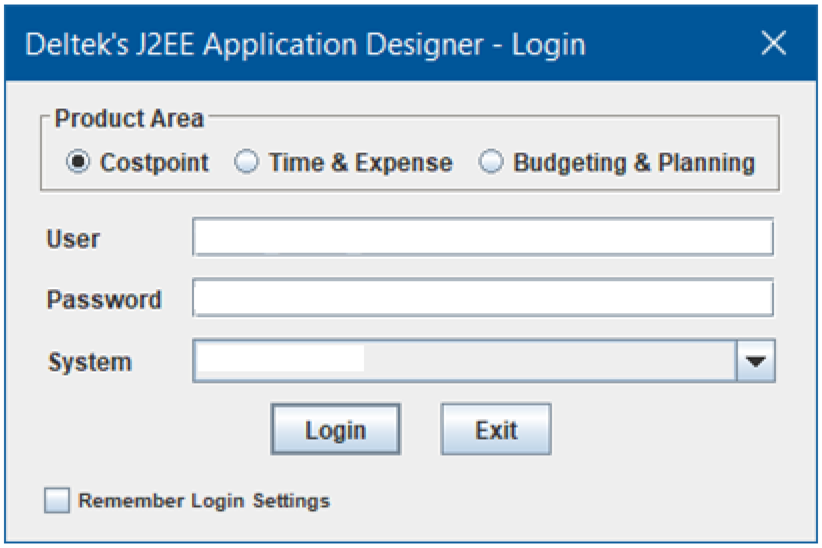
Allow Access to Integration Console
[✓ Allow Access to Extensibility Console]
Allow Application Access via Integration Services

Extensibility Console supports Database- and LDAP- (Active Directory) based Authentication types.

To start the Extensibility Designer, run the CPWebExtDesigner.cmd batch file. It should be located with the other Costpoint command files in the \deltek\costpoint\82\bin folder.

To log into the Extensibility Designer:

1. Enter the username and password you use for Deltek Costpoint® Web in the **User** and **Password** fields.



Product Area
[✓ Costpoint] Time & Expense Budgeting & Planning
User
Password
System
Remember Login Settings

2. Select a system from the **System** drop-down list.

   Tip: Optionally, you can select the **Remember Login Settings** check box to save the username and selected system. You will always be prompted to enter your password.

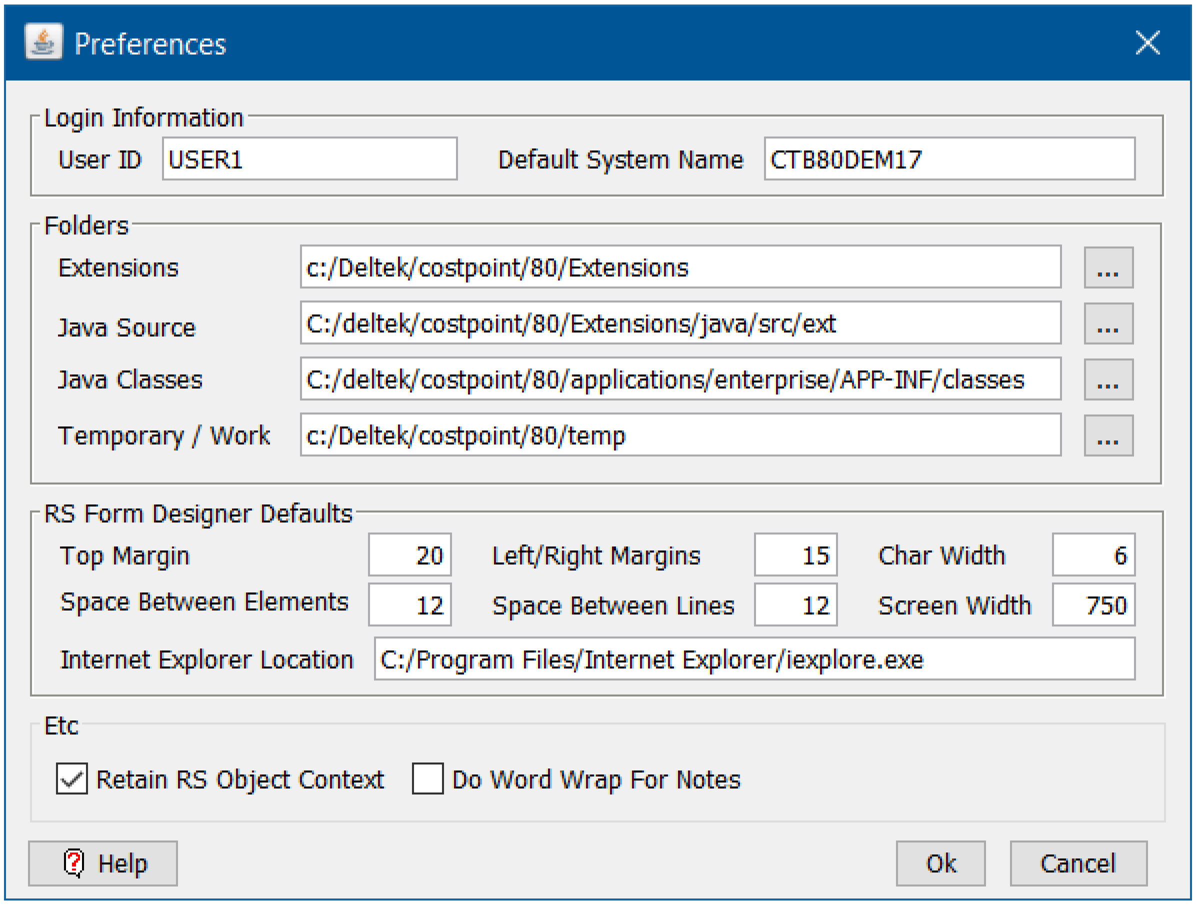
3. Select **Product Area**.

   Costpoint Area is selected by default, but if you are planning to customize Applications and Result Sets from Time & Expense or Budgeting & Planning, it is important to select the correct **Product Area**. For example, if you are doing T&E app customization, select Time & Expense as it will allow proper usage of the T&E Result Set Object Templates.

4. Click **Login** or press **Enter**.

Note: If you encounter a warning about the license upon startup, you will need to request a new license before the Extensibility feature will work in Costpoint. The license is free but you are required to sign a contract addendum with Deltek. For more information, contact your account manager at Deltek.

After you log in to the tool for the first time, you will get a message stating that Java Folders are not set and you should go into Preferences dialog box to enter and set the correct folder values. Click **Menu » Tools » Preferences** to display the dialog box.


Preferences

Login Information

User ID: USER1

Default System Name: CTB80DEM17

Folders

Extensions	c:/Deltek/costpoint/80/Extensions
Java Source	C:/deltek/costpoint/80/Extensions/java/src/ext
Java Classes	C:/deltek/costpoint/80/applications/enterprise/APP-INF/classes
Temporary / Work	c:/Deltek/costpoint/80/temp

RS Form Designer Defaults

Top Margin	20	Left/Right Margins	15	Char Width	6
Space Between Elements	12	Space Between Lines	12	Screen Width	750

Internet Explorer Location: C:/Program Files/Internet Explorer/iexplore.exe

Etc

[✓] Retain RS Object Context [ ] Do Word Wrap For Notes

You can use this dialog box to enter or change the default User ID and Default System Name that you use to log into the tool. You can also change the default values of the following fields and options, which includes the folders where the extensibility information will be extracted to and where it will reside until the Extensibility Packager packs your extension.

- Extensions Folder: Enter the main folder where all extension work will be created.
- Extensions Java Source Folder: Enter the root Java folder under which all extension Unit Java source files should be placed.
- Extensions Java Classes Folder: Enter the root Java classes folder under which all extension Unit Java class files should be placed. If this folder is not pointing to the same folder where all the regular application classes are stored, runtime will be unable to load them unless you modify the classpath for the enterprise application, which generally is not recommended unless you are familiar with Weblogic and Java class loaders. Deltek recommends that this folder point to /Deltek/costpoint/82/applications/enterprise/APP-INF/classes.
- Temporary / Work Folder: Enter the work folder that the tool will use for work file operations.
- RS Form Designer Defaults: Deltek recommends that you accept the default values until you are familiar with the Form Designer and better understand how to position fields on the screen.
- Retain RS Object Context: Clear this check box if you want the tool not to keep the same RS Object context in the RS Editor screens.

When you have finished editing the fields and options, click **Ok** to save the changes. If any of the folders do not exist, you will be prompted to create them.

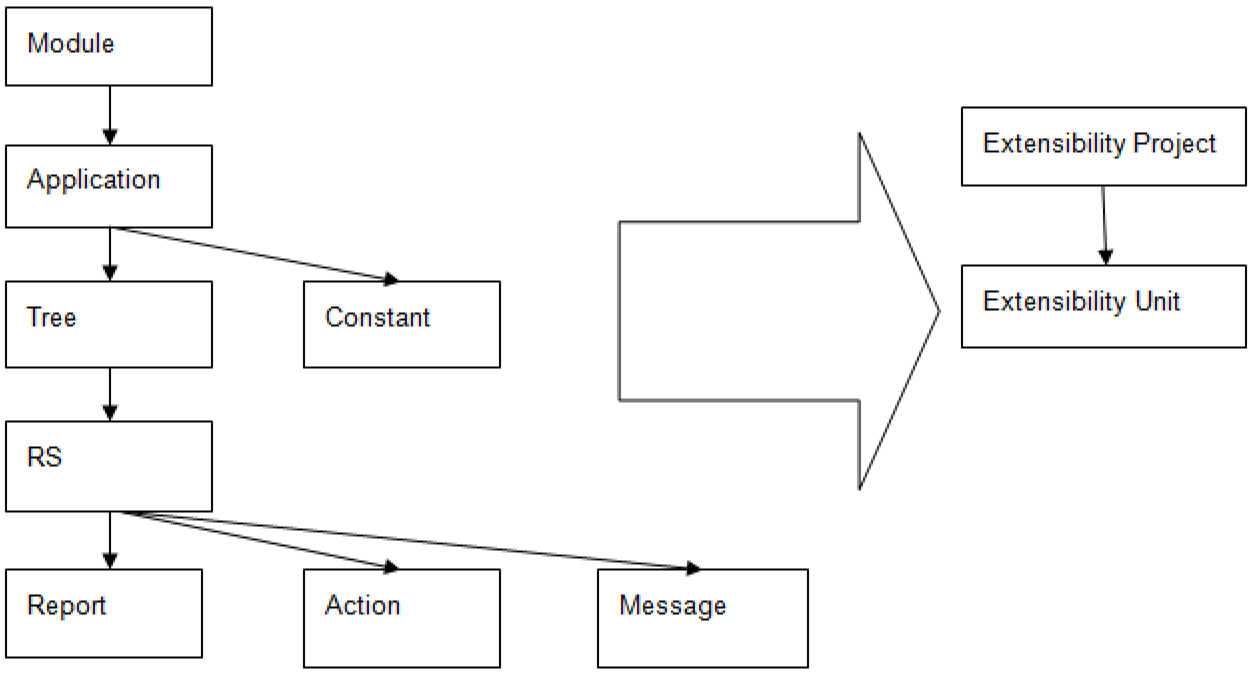
The following table shows the steps involved in creating and deploying customization for Costpoint clients:

Step	Description
1	Gather requirements for your Extensibility Project.
2	Enable the Extensibility feature for your system using the Configuration Utility. You only have to enable the feature once. There is no need to enable it again for consequent Extensibility projects.
3	Create an Extensibility Project.
4	Create one (or more) Extensibility Unit(s) depending on your requirements.
5	Customize/Extend Costpoint objects and create new Custom Objects for each Extensibility Unit according to the requirements. Multiple Extensibility Units are needed only when customized application should behave differently for different users/user groups. In that case, you will need to create one Extensibility unit for each type of behavior.
6	Assign Extensibility Unit(s) to UI profiles (and consequently to Costpoint User(s)) and test customized Costpoint Application(s) to make sure requirements are met.
7	Export Extensibility Project Database Script(s).
8	Package all Extensibility Project Files (including DB scripts, Java classes, DB Stored Procedures) with the help of Extensibility Packager.
9	Deliver Extensibility package to the production system.
10	Install Extensibility Project using DBWizard Tool.
11	Activate and assign Extensibility Units to appropriate Users/User profiles in Enter/Manage User Interface Profiles (SYMPROF) Application and test customized Costpoint Application(s) to make sure requirements are met.

Attention: See the Costpoint 8.2 Extensibility Designer Quick Start Guide for illustrative examples.

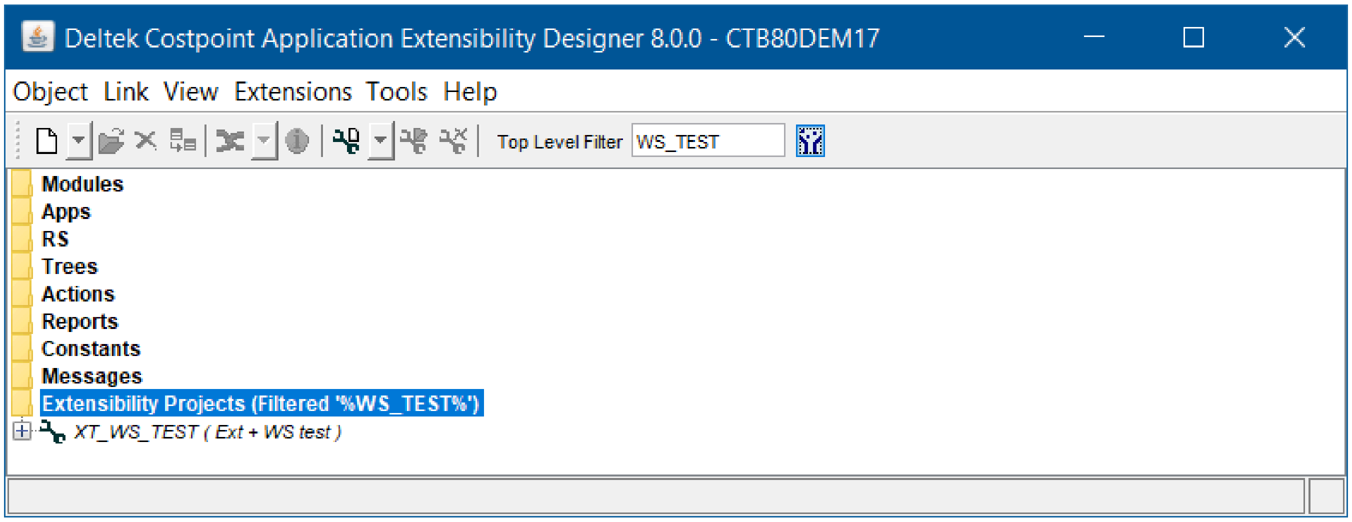
After you enter a valid **User** and **Password** combination, the main window that opens displays folders containing **Modules**, **Apps**, **RS**, **Trees**, **Actions**, **Reports**, **Constants**, **Messages** and **Extensibility Projects** that relate to the system that is currently being used. All these objects are components used in the application.

The following diagram illustrates the relationship between these objects.



Main Window

The dialog box contains an explorer tree where you can drill down to the related components from any of the top-level folders. Double-click the folder to the left of the text description to expand an item.


Deltek Costpoint Application Extensibility Designer 8.0.0 - CTB80DEM17
Object Link View Extensions Tools Help
Top Level Filter WS_TEST
Modules
Apps
RS
Trees
Actions
Reports
Constants
Messages
Extensibility Projects (Filtered '%WS_TEST%')
XT_WS_TEST (Ext + WS test)

Tip: When drilling down, each component also includes a backward link to the parent component for ease of navigation. Therefore, a tree can recursively display a list of linked or assigned objects of other types.

If you are not familiar with the structure of a particular application, start by navigating through the regular object lists that are assigned that application. This will help you understand the structure of the application, links between different objects in the application, and if objects from the application are reused in other parts of Costpoint.

Modules

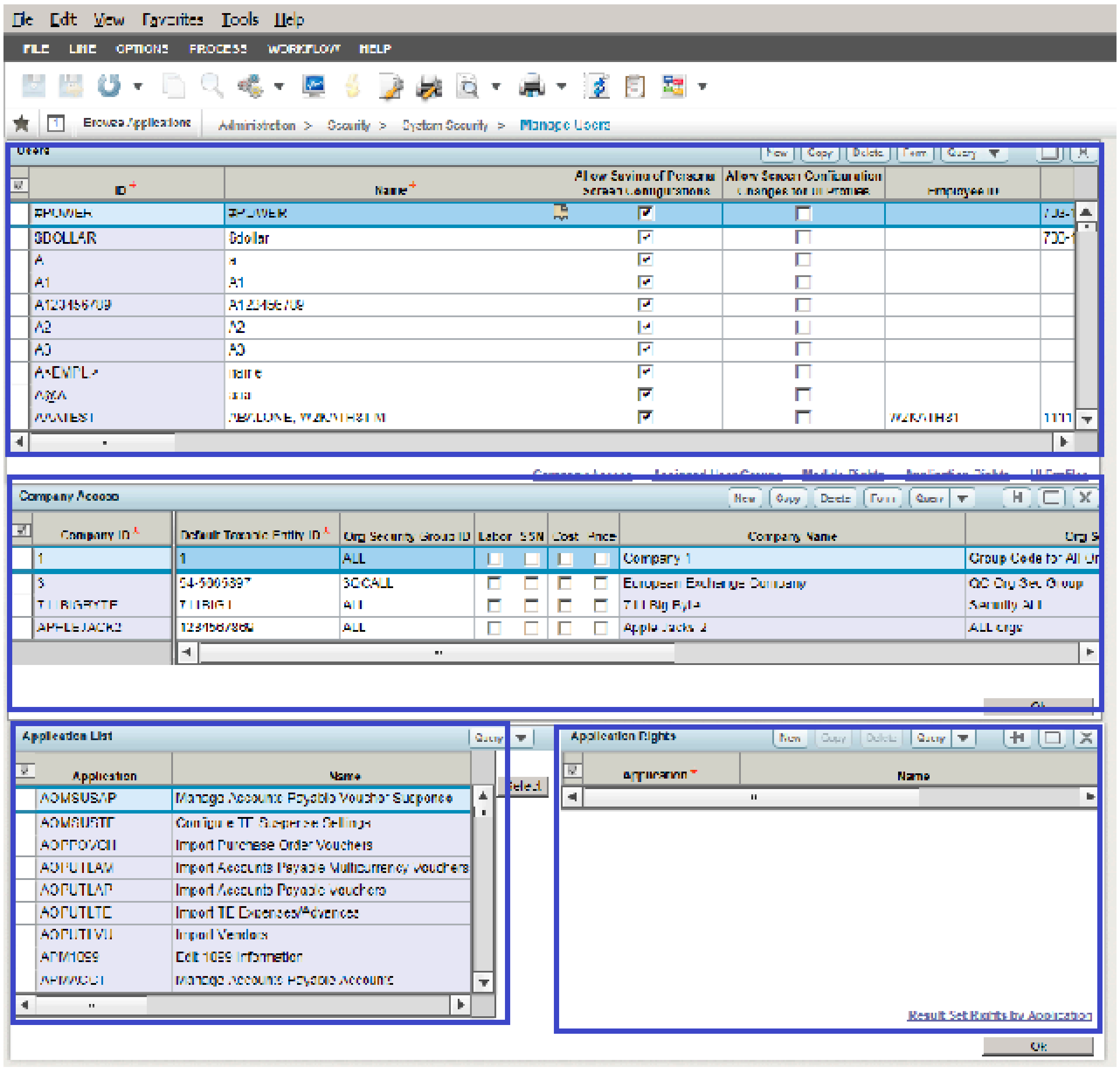
The Modules folder contains a list of standard modules that are being used in Costpoint. A module is a group of Costpoint Applications related by functionality and used to simplify rights management in Costpoint. You cannot customize or add a Costpoint module. They are shown for identification and navigation only. Drill down to find the application you want to customize.

Apps

The Apps folder contains a list of all applications in Costpoint. An application consists of core components (for example, result sets, reports, actions). Generally, when you customize an application, you actually customize its components, not the application object itself. The only element you can customize the application object is the title of the application, which will be used on the menu.

Result Set

The result set (RS) folder contains a list of result sets (screens/subtasks) in Costpoint. There are thousands of result sets in Costpoint. An application can include one or more result sets. As shown in the following figure, each of the blocks is a result set.


File Edit View Favorites Tools Help
FILE LINE OPTIONS PROCESS WORKFLOW HELP
Browse Applications
Administration > Security > System Security > Manage Users
New Copy Delete Form Query
Select
Result Set Rights by Application
OK

Users

	ID	Name	Allow Saving of Personal Screen Configurations	Allow Screen Configuration Changes for UI Profiles	Employee ID	
	#POWER	#POWER	☑	☐		700-1
	$DOLLAR	$dollar	☑	☐		700-1
	A	a	☑	☐		
	A1	A1	☑	☐		
	A123456789	A123456789	☑	☐		
	A2	A2	☑	☐		
	A3	A3	☑	☐		
	AAEMPL…	[illegible]	☑	☐		
	AAA	[illegible]	☑	☐		
	AAATEST	ABCLONE, WARNTHEIM	☑	☐	WARNTHEIM	1111

Company Access

	Company ID	Default Taxable Entity ID	Org Security Group ID	Labor	SSN	Cost	Price	Company Name	Org S…
	1	1	ALL	☐	☐	☐	☐	Company 1	Group Code for All Ur…
	3	54-5005397	3QCALL	☐	☐	☐	☐	European Exchange Company	QC Org Sec Group
	711BIGBYTE	711BIG1	ALL	☐	☐	☐	☐	711 Big Byte	Security All
	APPLEJACK2	123456789	ALL	☐	☐	☐	☐	Apple Jacks 2	ALL orgs

Application List

	Application	Name
	AOMSUSAP	Manage Accounts Payable Voucher Suspense
	AOMSUSTE	Configure TE Suspense Settings
	AOPPOVCH	Import Purchase Order Vouchers
	AOPUTLAM	Import Accounts Payable Multicurrency Vouchers
	AOPUTLAP	Import Accounts Payable Vouchers
	AOPUTLTE	Import TE Expenses/Advances
	AOPUTLVN	Import Vendors
	APM1099	Edit 1099 Information
	APMACCT	Manage Accounts Payable Accounts

Application Rights

	Application	Name

A result set is a group of fields that are coming from one (or more) database tables and usually grouped together on the screen as one table (in case there is a Table View of this RS). Each Result set can be shown in Form or Table View. For each result set, you can customize its appearance by rearranging the position of the fields. You can change field labels, tab order, editability, and visibility. You can also add additional validation on the objects or lines; or add additional logic to be executed when you populate or save data.

Trees

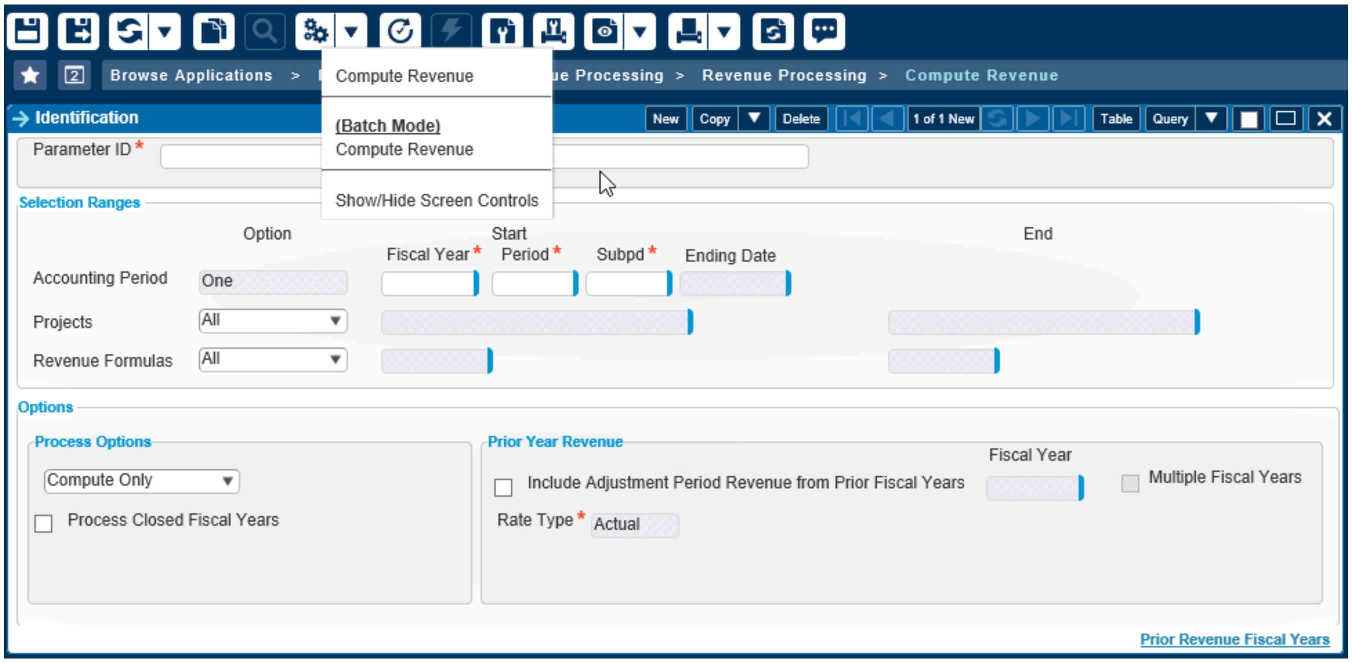
The Trees folder displays the relationship of the result sets in an application. When an application only contains one result set, there is no need for a tree. However, if it contains more than one result set, the relationship between the result sets are described in a tree. The tree is then linked (assigned) to the application.

You can customize existing a RS Tree (properties of the link) or add a New RS (subtasks) to the existing Trees..

Actions

The Actions folder displays the list of processes (usually called actions) in Costpoint. Each action describes the logic to be executed when a user clicks action icons or buttons on the interface. Actions are not called on standard events such as query, save, add new line, and so on. The logic for these standard events (plug-ins) is handled inside the result set which can be extended in the result set.

An example of action is **Compute Revenue** action in the Compute Revenue application. These are long running processes.



Parameter ID *

	Option	Start Fiscal Year *	Start Period *	Start Subpd *	Ending Date	End
Accounting Period	One					
Projects	All					
Revenue Formulas	All					

Compute Only

☐ Process Closed Fiscal Years

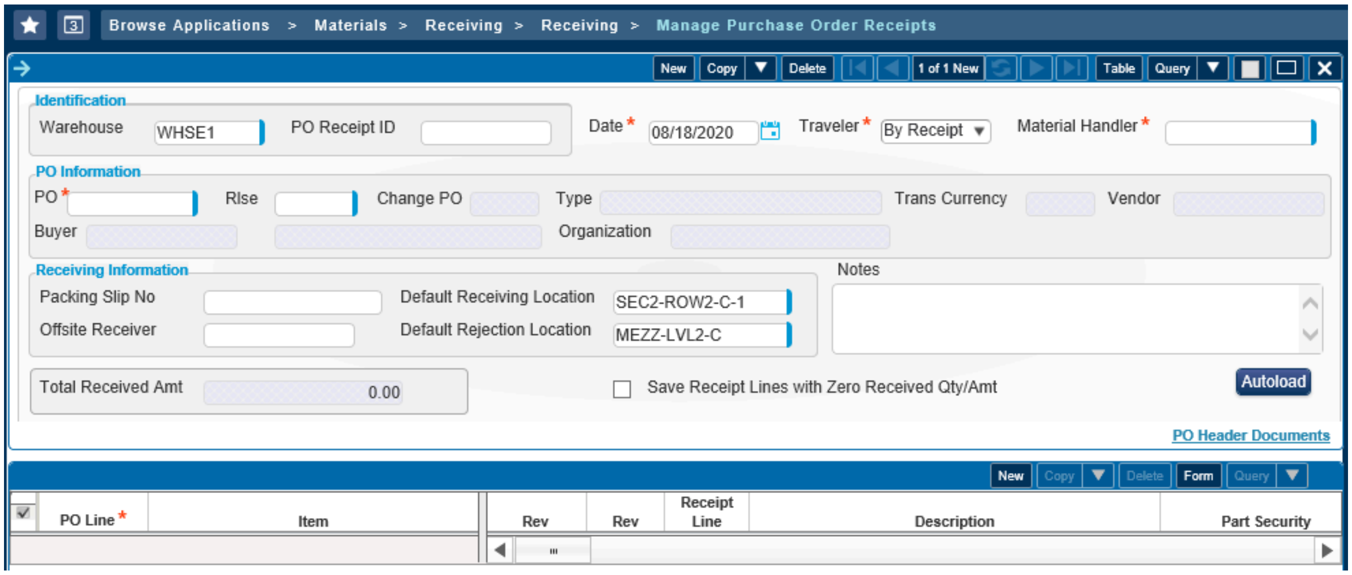
☐ Include Adjustment Period Revenue from Prior Fiscal Years

Fiscal Year

☐ Multiple Fiscal Years

Rate Type * Actual

An action can also be on a maintenance or transaction application. An example is **Autoload** receipt lines on the Manage Purchase Order Receipt application. These actions typically do not run long processes. They just aid users in data entry.


Browse Applications > Materials > Receiving > Receiving > Manage Purchase Order Receipts
New
Copy
Delete
1 of 1 New
Table
Query
Autoload
PO Header Documents
New
Copy
Delete
Form
Query

Identification

Warehouse WHSE1 | PO Receipt ID | Date * 08/18/2020 | Traveler * By Receipt | Material Handler *

PO Information

PO * | Rlse | Change PO | Type | Trans Currency | Vendor

Buyer | Organization

Receiving Information

Packing Slip No | Default Receiving Location SEC2-ROW2-C-1

Offsite Receiver | Default Rejection Location MEZZ-LVL2-C

Notes

Total Received Amt 0.00

Save Receipt Lines with Zero Received Qty/Amt

PO Line *	Item	Rev	Rev	Receipt Line	Description	Part Security

You can customize an action by customizing the title or status text of the action. You can add additional logic to be executed in addition to the standard logic performed by the action either before or after standard Action. Also, you can add new custom actions.

Note: The status text feature is now obsolete and is no longer used by the run-time environment. This change aligns with the removal of the status bar in most modern browsers, including Microsoft Edge, Chrome, and Firefox. References to status text remain in this document for historical context and apply only to buttons/actions, not regular fields.

Reports

The Reports folder displays a list of reports in Costpoint. You can customize a report to show different titles/ labels or to include additional data fields. Customizing a report for label changes require extending the report object in this tool and the report template. Adding fields to the report also requires extending the result set that contains the data feeding into the report. These are called Reporting Data result set. These do not have any UI presentation and are used only for feeding data to the report. To customize how a report data is laid out on paper you must customize its report template.

To customize report Templates, you will need to download, install, and learn the report template editor called Business Intelligence and Reporting Tools (or BIRT) This is a free tool produced by Open Source BIRT Project (http://www.eclipse.org/birt/)

Constants

The Constants folder displays a list of constants used in Costpoint. Constants are data in the database that can be uniquely identified with the login session (that is, the company ID of the current session, the login ID of the user). As long as it is associated with only one value, it can be created as a constant. Once it is created, it can be used to display on the screen as label or used in status text or reports. The majority of Costpoint settings are represented as Constants.

You cannot customize an existing constant. However, you can add a new constant.

Messages

The Messages folder displays a list of messages used in Costpoint. Messages or text are used in Costpoint to display information, warnings, or errors.

To avoid confusion, you cannot change standard Costpoint messages. However, you can add a new message for an existing validation or new validation that you are adding for a result set or action.

Extensibility Projects

The Extensibility Projects folder displays a list of extensibility projects that have been created in Costpoint. Each extensibility project contains an Extensibility Units folder that displays a list of extensibility units within the project, which in turn contains a list of customized objects under this unit.

Menu Bar

This section describes the function of each menu bar item.

Object Menu

Menu Item	Description
New	Click this menu item to create a brand-new extensibility object (for example, extensibility project, extensibility unit, message, constant, or action, RS, and so on). For new extensibility objects, select the extensibility project and unit where the new object will be created. See the following sections for details on how to add new extensibility objects: ▪ Extensibility Projects ▪ Extensibility Units ▪ Messages ▪ Constants ▪ Add New Actions/Processes ▪ Add New RS
Open	Click this menu item to open a selected object in the main window. See the following section for details on how to edit extensibility objects:

Menu Item	Description
	▪ Extensibility Projects ▪ Extensibility Units
Clone (Duplicate)	Click this menu item to duplicate a selected object in the main window. Duplicating an object does not copy all the sub-objects. You can only clone objects that you have created. You cannot clone regular Costpoint objects.
Delete	Click this menu item to delete a selected object in the main window. On the Confirm Delete dialog box that appears, you must click **Yes** to confirm the deletion or **No** to cancel the request.
Usage Information	Click this menu item to view information about where a selected object (in the main window) is used in the whole system. This can help you find or verify the impact of the changes in that object to other related objects. If you use this feature on an extensibility project, the Extensibility Console will generate an Excel spreadsheet that lists all properties that were created or customized in the selected project.
Exit	Click this menu item to close the Extensibility Console.

Link Menu

Menu Item	Description
Assign/Unassign	Click this menu item to open a dialog box where you can manage links between objects.

View Menu

Menu Item	Description
Refresh	Click this menu item to refresh filter results in the main window. **To filter items in the main window:** 1. Select any folder in the main window.

Menu Item	Description
	2. Enter the text by which to filter items in the **Top Level Filter** field. 3. Click **View » Refresh** to view the filter results. Filtering the list of objects is applicable only to the top-level objects and is performed as a wild card search. For example, if you are looking for the PJMBASIC application, you can enter 'PJMB' or 'BASIC' in the Top Level Filter field and the list will display only application that have 'PJMB' or 'BASIC'. This search is not case-sensitive and is done based on both object ID and name.

Extensions Menu

Menu Item	Description
Extend	Use this menu item to create extension objects (for example, **RS**, **Action**, **Report**, **App**).
Open Extension	Click this menu item to open a selected extension object in the main window. This is the same as the **Object » Open** menu item except that it is only enabled if the selected object in the main window is an extension object. You cannot open a regular object using this feature.
Delete Extension	Click this menu item to delete a selected extension object in the main window. This is the same as the **Object » Delete** menu item except that it is only enabled if the selected object in the main window is an extension object. You cannot delete a regular object using this feature.
Export Script	Click this menu item to export the SQL metadata script for a selected extensibility project or unit in the main window. The script contains SQL insert statements that can be used to back up your work and deploy the extension objects to another system. Deltek strongly recommends that you export scripts on the project level since the extensibility project script is required to create an extensibility package (see Appendix C – Extensibility Packager). This project script contains metadata changes for all enclosed units. Exporting a script at extensibility unit level should be done generally for backup or quick review purposes. Keep in mind that if you generated scripts at unit level, you must remove the unit script before you start packaging the extensibility project.

Tools Menu

Menu Item	Description
Extensibility Packager	Click this menu item to open the Extensibility Packager. See Appendix C – Extensibility Packager for more information about the Extensibility Packager.
Extensibility Deployer	Click this menu item to open the Extensibility Deployer. This dialog box can be used to quickly deploy an already packaged Extensibility project file to the system you are logged in. It will apply DB scripts to a given system and copy report and compiled java classes to a given product deployment, but it will not re-deploy the Costpoint application. If your Extension contains java classes, you will need to restart WebLogic manually to make those new Java classes effective. Keep in mind that the full functionality is provided by DBWizard » Deploy Extensions menu item and this dialog box serves as a quick way to test deployment of the Extensibility project that you've created using Extensibility packager.
SQL Editor	Click this menu item to open the SQL Editor.

Help Menu

Menu Item	Description
About	Click this menu item to display a dialog box that contains the Extensibility Console's version and copyright information.

Toolbar

This section describes the functions of the different icons on the Toolbar as well as the **Top Level Filter** field.

Icons

The toolbar icons perform the same actions as the menu items and are explained below.

Tip: Hover your cursor over a toolbar icon to view a short description of its function.

Icon	Menu Item or Description
	Object » New
	Object » Edit
	Object » Delete
	Object » Duplicate
	Object » Usage Information
	Link » Assign/Unassign
	Extensions » Extend
	Extensions » Open Extension
	Extensions » Delete Extension
	View » Refresh

Top Level Filter

The **Top Level Filter** field filters the ID of the first level list of modules, applications, result sets, trees, actions, reports, and extensibility projects.

- The **Top Level Filter** field allows the entry wildcard characters to narrow choices.
- Click the **Filter** icon located to the right of the text box to display the results.
- To display all result, remove all text and click the **Filter** icon.

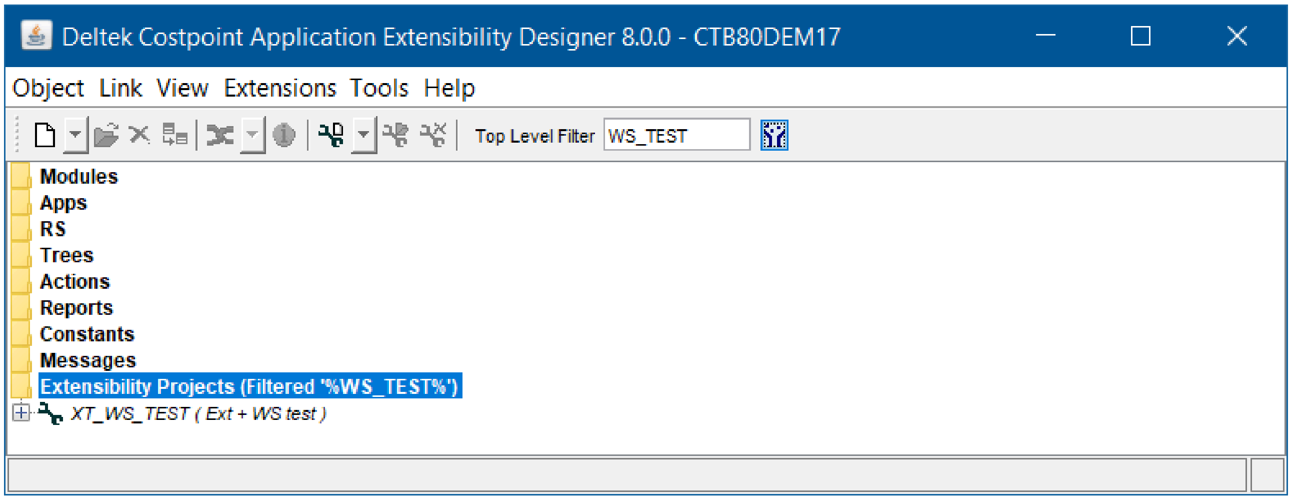
This section contains instructions on how to add, modify, or delete an extensibility project.

Add an Extensibility Project

To add a new extensibility project:

1. On the **Object** menu, click **New » Extensibility Project** or click the **New** ( ) icon and then click **Extensibility Project**. The **New Extensibility Project** dialog box displays. A project by itself contains no definition or logic for customization. It is just an umbrella for extensibility objects so we can organize, track and maintain extensibility objects.

Deltek


Deltek Costpoint Application Extensibility Designer 8.0.0 - CTB80DEM17
Object Link View Extensions Tools Help
Top Level Filter WS_TEST

Modules
Apps
RS
Trees
Actions
Reports
Constants
Messages
Extensibility Projects (Filtered '%WS_TEST%')
XT_WS_TEST (Ext + WS test)

2. Complete the text fields as applicable for the project and click the **Ok** button. The **Id** and **Name** fields are required. Note that the new project now displays under the **Extensibility Projects** folder.

   - **Id**: Enter a unique ID for the extensibility project. Cloud Clients must include the five-digit customer ID number after the XT_ prefix. This field is required.

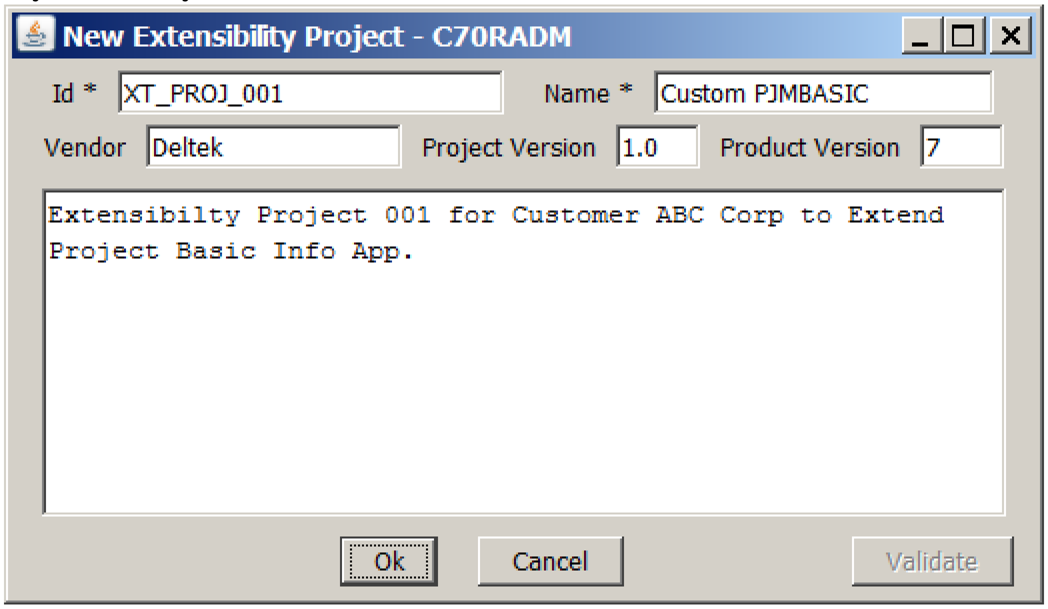
     Note: All extensibility project IDs must begin with 'XT_'. All new extensibility objects inside this project will have this project ID as prefix. This separates extensibility objects from regular objects and helps track what objects belong to what project.

     Each extensibility project ID must be unique. This ensures that customers can apply multiple extensibility projects to their environments.

     Cloud Clients: If you do not follow the required naming format, your extension request will be rejected.

   - **Name**: Enter the unique name for the project. This informational descriptor is a required field and is limited to a total of 13 characters.
   - **Vendor**: Enter the name of the company that is creating this project.
   - **Project Version**: Enter the project version. Deltek highly recommends that you increment the version when making updates or changes to the project for easier version tracking. It will help you determine which version of the project is deployed when testing or troubleshooting your project.
   - **Product Version**: Enter the version of the currently installed Costpoint system for which the extension is being developed.
   - **Notes**: Enter miscellaneous information about this project that will help you to understand project's purpose and functionality. Deltek highly recommends that you don't leave this field blank but use it to describe your project.
   - **Validate Button**: Click this button to validate extensibility project customizations against currently deployed standard Costpoint metadata. If some of the changes are not permitted anymore or done on

objects that no longer exist in Costpoint, a dialog box will appear describing those invalid customization changes. You must modify your customization to remove those no longer valid customizations. Only valid extensibility units can be assigned and executed at run-time. Costpoint will not load and execute any extensibility unit that is invalid.



Id *	XT_PROJ_001	Name *	Custom PJMBASIC		
Vendor	Deltek	Project Version	1.0	Product Version	7

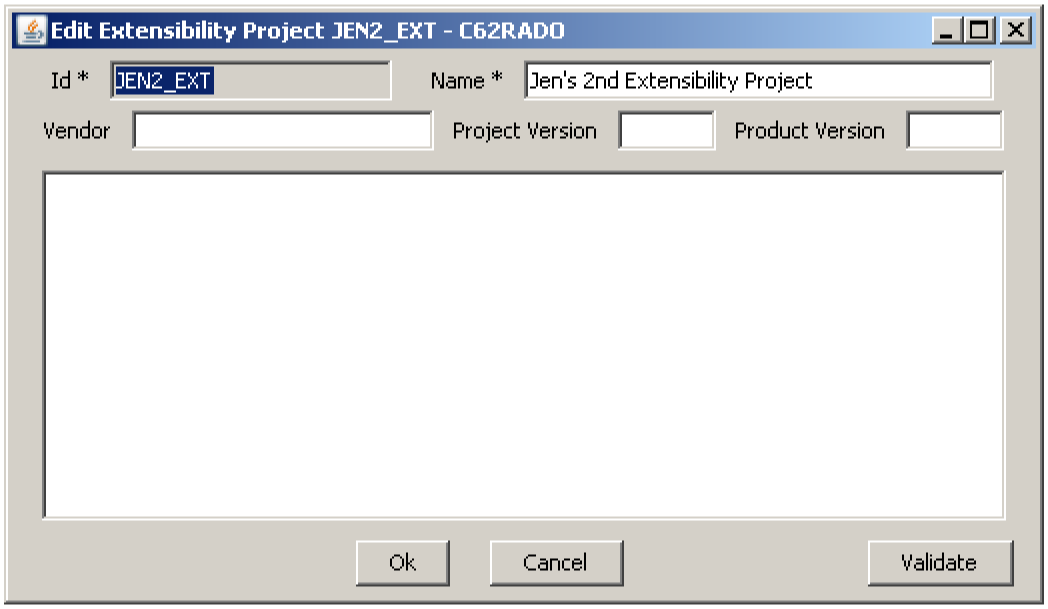
```
Extensibilty Project 001 for Customer ABC Corp to Extend
Project Basic Info App.
```

Modify an Extensibility Project

To modify an extensibility project:

1. Double-click the **Extensibility Projects** folder to expand it.

2. Click the project to modify and click **Object » Edit** from the menu or click the **Open** toolbar icon. The **Edit Extensibility Project** dialog box displays.


Edit Extensibility Project JEN2_EXT - C62RADO
Id *
JEN2_EXT
Name *
Jen's 2nd Extensibility Project
Vendor
Project Version
Product Version
Ok
Cancel
Validate

3. Update and complete the text fields as applicable for the project and then click **Ok**. You cannot modify the **Id** field and the **Name** field is required.

Delete an Extensibility Project

To delete an extensibility project:

1. Double-click the **Extensibility Projects** folder to expand it.
2. Select the project that needs to be deleted and select **Object » Delete** from the menu or click the **Delete** toolbar icon. Please note that deleting Extensibility Project will delete all metadata customizations done under this Project. This will not delete any files in your file system.

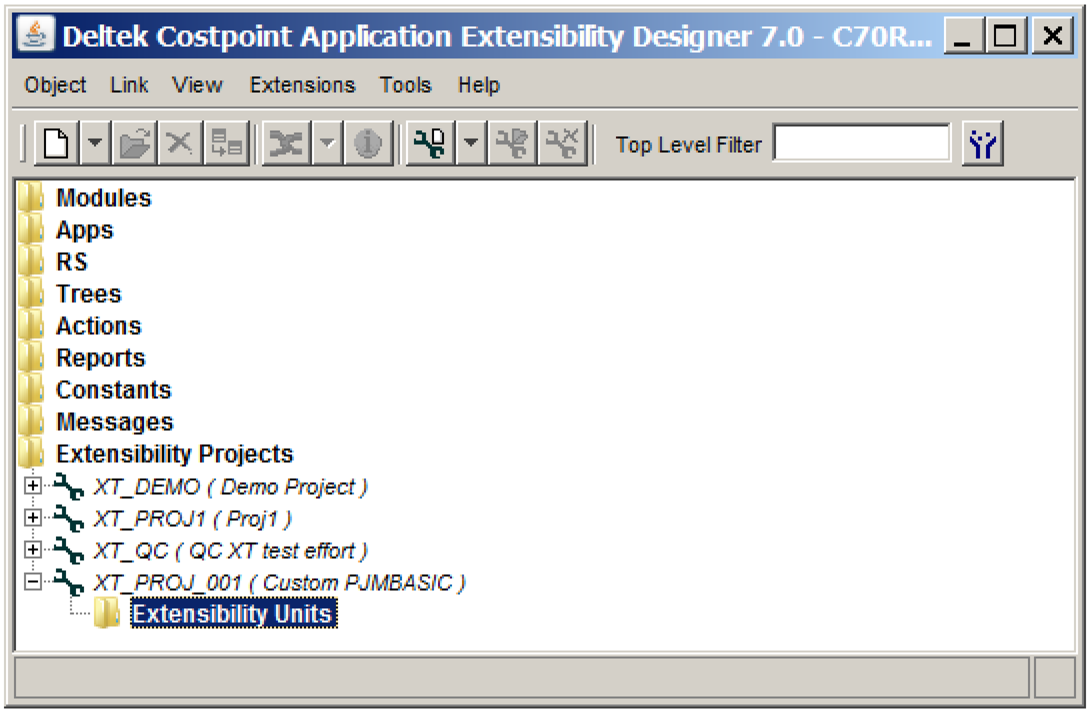
> **Note:** Deleting an extensibility project will delete all metadata customizations done under the project. This will not delete any files in your file system.

This section contains information on how to add, modify, or delete extensibility units.

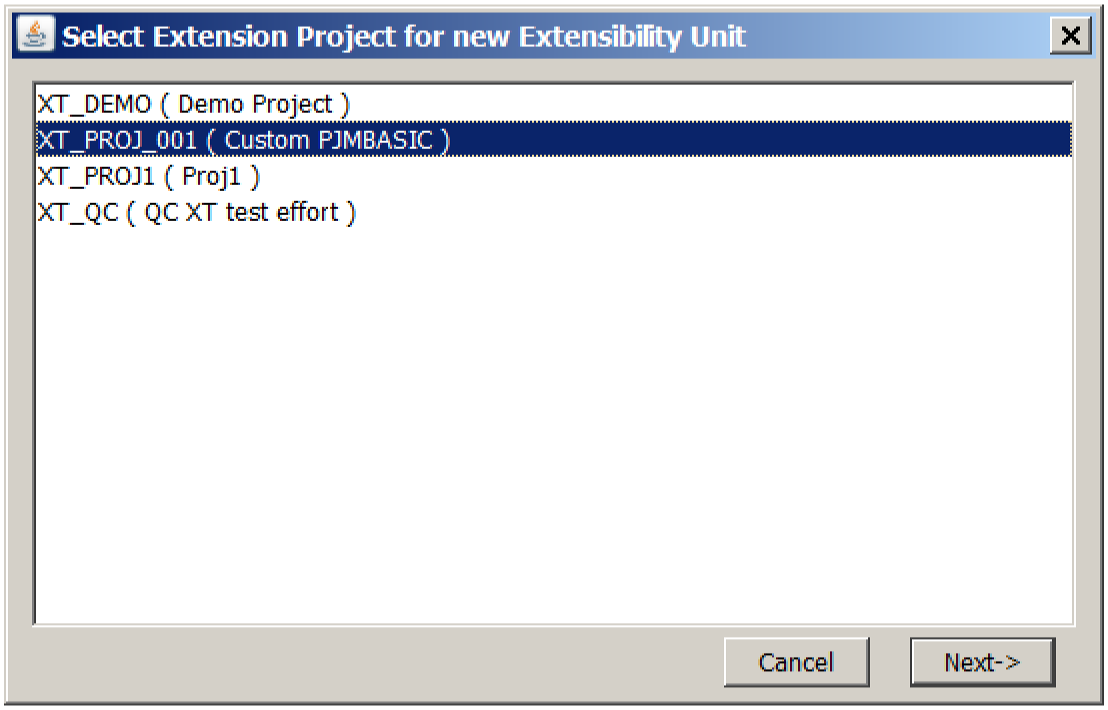
Add an Extensibility Unit

To add an extensibility unit:

1. Expand the existing extensibility project. A new folder (Extensibility Units) displays below the project name.



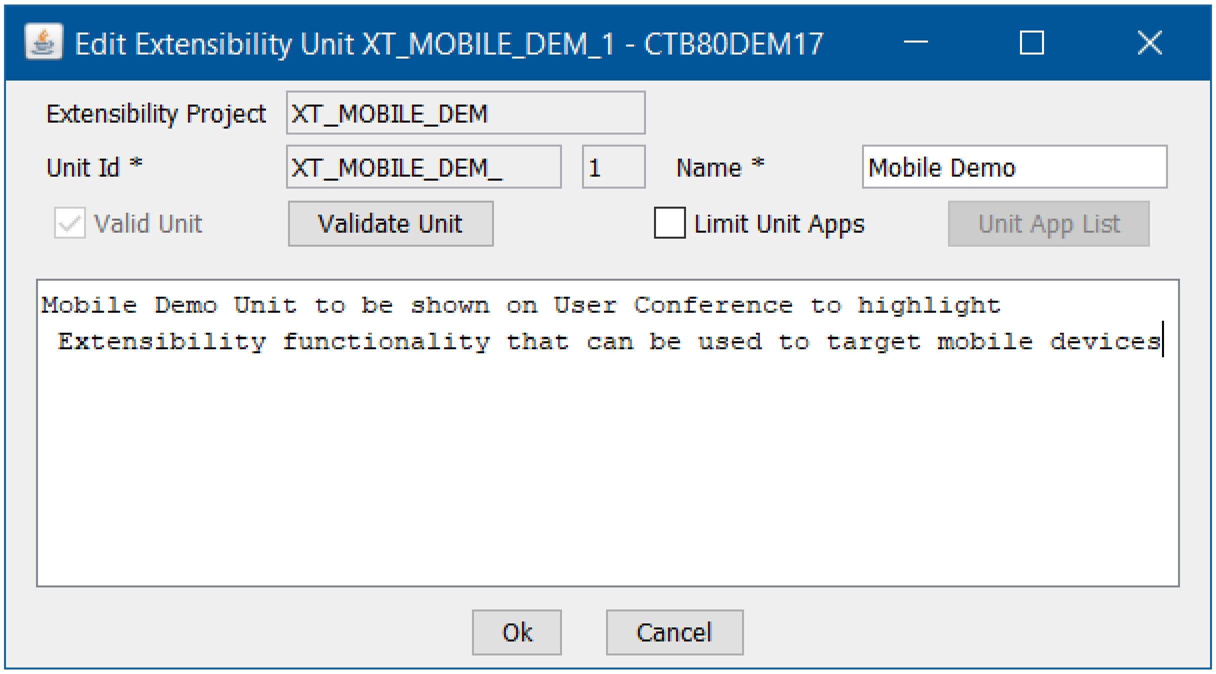
2. Click **Object** » **New** » **Extensibility Unit** on the Menu Bar or click and then click **Extensibility Unit**. The Select Extension Project for new Extensibility Unit dialog box displays.


Select Extension Project for new Extensibility Unit
XT_DEMO (Demo Project)
XT_PROJ_001 (Custom PJMBASIC)
XT_PROJ1 (Proj1)
XT_QC (QC XT test effort)
Cancel
Next->

3. Select an extensibility project where the new unit will be created and then click **Next**. The New Extensibility Unit dialog box displays.

A unit contains no definition or logic for customization. It is just an umbrella for extensibility objects so that it helps you organize, track, and maintain extensibility objects. However, each unit contains the set of object extension that go together when assigning to profiles inside Costpoint. Behavior that should be assigned to a different profile must be implemented in different units. However, if custom behavior should be the same for all Costpoint users, you only need one extensibility unit.

Multiple extensibility projects and units can significantly increase the burden of maintaining and assigning them to UI profiles for Costpoint administrators. Deltek recommends that you create a minimal set of extensibility projects and units.


Edit Extensibility Unit XT_MOBILE_DEM_1 - CTB80DEM17

Extensibility Project: XT_MOBILE_DEM

Unit Id *: XT_MOBILE_DEM_ 1

Name *: Mobile Demo

Valid Unit: ✓

Limit Unit Apps: ☐

```
Mobile Demo Unit to be shown on User Conference to highlight
 Extensibility functionality that can be used to target mobile devices
```

4. Complete the text fields as applicable for the unit and then click **Ok**. The new unit displays under the **Extensibility Units** folder within the project.

- **Unit Id**: Complete the unit ID. It automatically starts with the **Extensibility Project** ID and an underscore (_).
- **Name**: Enter a descriptive name for the unit.
- **Valid Unit**: This check box indicates if the metadata for objects within the unit is valid.
- **Validate Unit**: Click this button to validate the metadata for objects within the unit. This process makes sure that the metadata in objects within the unit are valid with the current Costpoint regular metadata. For example, after you customize a field on a screen to make it hidden, Deltek has made a change to the same object and makes it required via a hot fix. This will invalidate the customization since a required object cannot be customized to be hidden. The extension object must be corrected to be valid.
- **Notes**: Enter any notes to describe this extensibility unit that will help you to understand the unit's purpose and functionality. Deltek highly recommends that you do not leave this field blank but use it to describe your unit.
- **Limit Unit Apps**: Select this check box to explicitly specify a list of applications where this customization will

be applied and limit customized functionality only to selected applications that can be selected when clicking the **Unit App List** button. This functionality should be used when you customize a reusable Object (for example, a Result Set), which is used in more than one application and you want to limit the customization to just a subset of applications where the result set is used. For example, assume that you want to customize the Lookup account RS CPM_ACCT_LKP that is used in dozens and dozens of different applications all over Costpoint. But, you want to limit your customization of this Lookup to just the AOPMICDN app, where this lookup is used to select a range of accounts to process. In this case, you would select the **Limit Unit Apps** check box and select only the AOPMICDN app in the list of applications.

Note: All extensibility unit IDs must begin with the parent extensibility project's ID prefix. All extensibility objects inside this unit will have the unit ID as its prefix. This separates extensibility objects from regular objects and objects from other units.

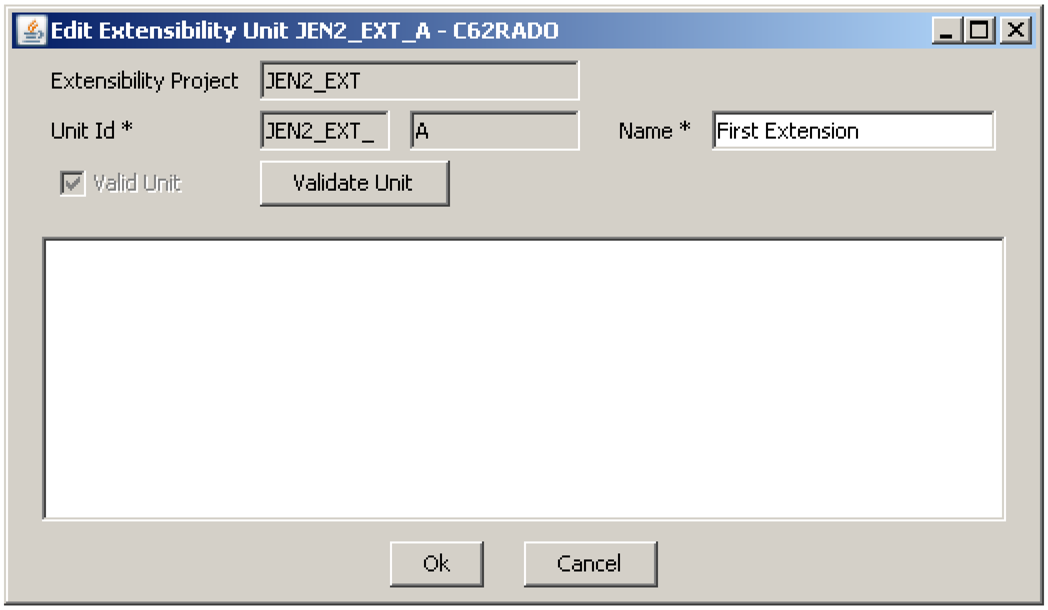
Note: You must create more than one extensibility unit only if you require different behavior for different user groups/roles. Otherwise, you only need a single unit for a given project.

Modify an Extensibility Unit

To modify an extensibility unit:

1. Expand the **Extensibility Projects** folder, the project to modify, and the **Extensibility Units** folder.

2. Select the unit to modify and then click **Object » Edit** on the Menu Bar or click the **Open** toolbar icon. The Edit Extensibility Unit dialog box displays.


Edit Extensibility Unit JEN2_EXT_A - C62RADO

Extensibility Project: JEN2_EXT
Unit Id *: JEN2_EXT_ A
Name *: First Extension
Valid Unit: ✓

3. Update and complete the text fields as applicable for the unit and then click **O**. You cannot modify the **Extensibility Project** and **Unit Id** fields.

Delete an Extensibility Unit

To delete an extensibility unit:

1. Double-click the **Extensibility Projects** folder to expand it.

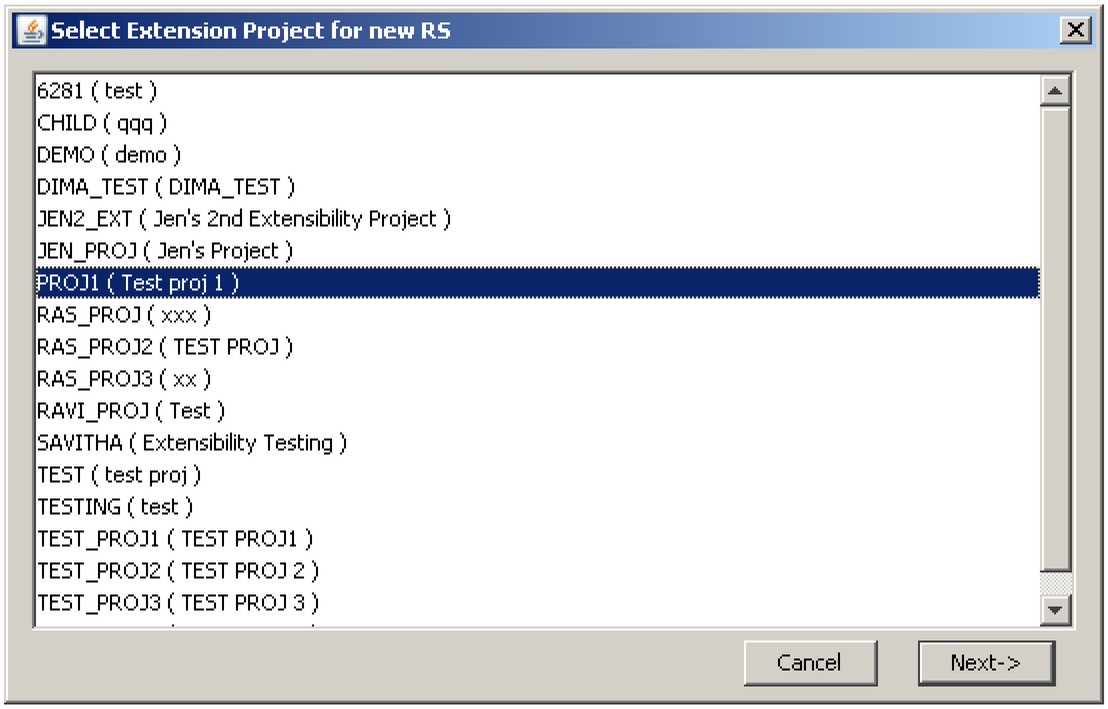
2. Select the unit to delete and then click **Object » Delete** or click the **Delete** toolbar icon. Please note that deleting Extensibility Unit will delete all metadata customizations done under this Unit from the database. This will not delete any files in your file system.

> **Note:** Deleting an extensibility unit will delete all metadata customizations done for the unit. This will not delete any files in your file system.

A Result Set (RS) is typically a header or a subtask in the Costpoint application. It is a group of related fields that usually come from a single table or multiple tables and is used to display (and enter) values into those fields. Depending on your needs, you can either customize (Extend) an Existing RS or create a new RS.

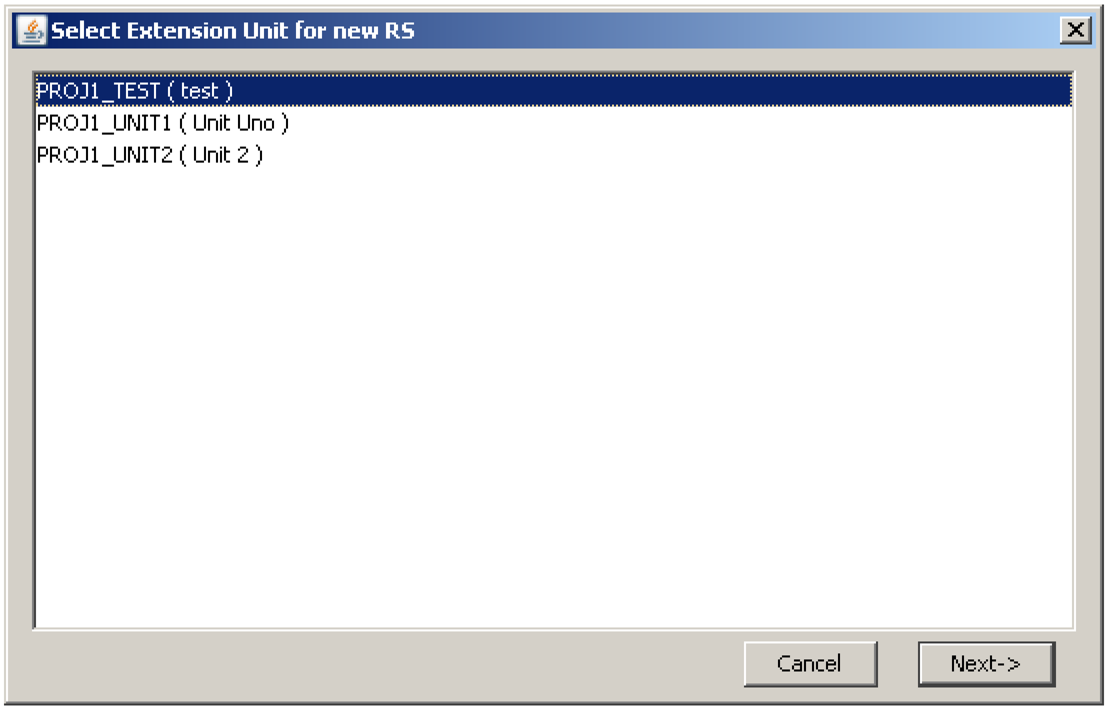
To extend a result set:

1. Click **Extensions » Extend » RS** (or click and then click **RS**). The Select Extension Project for new RS dialog box displays.


Select Extension Project for new RS

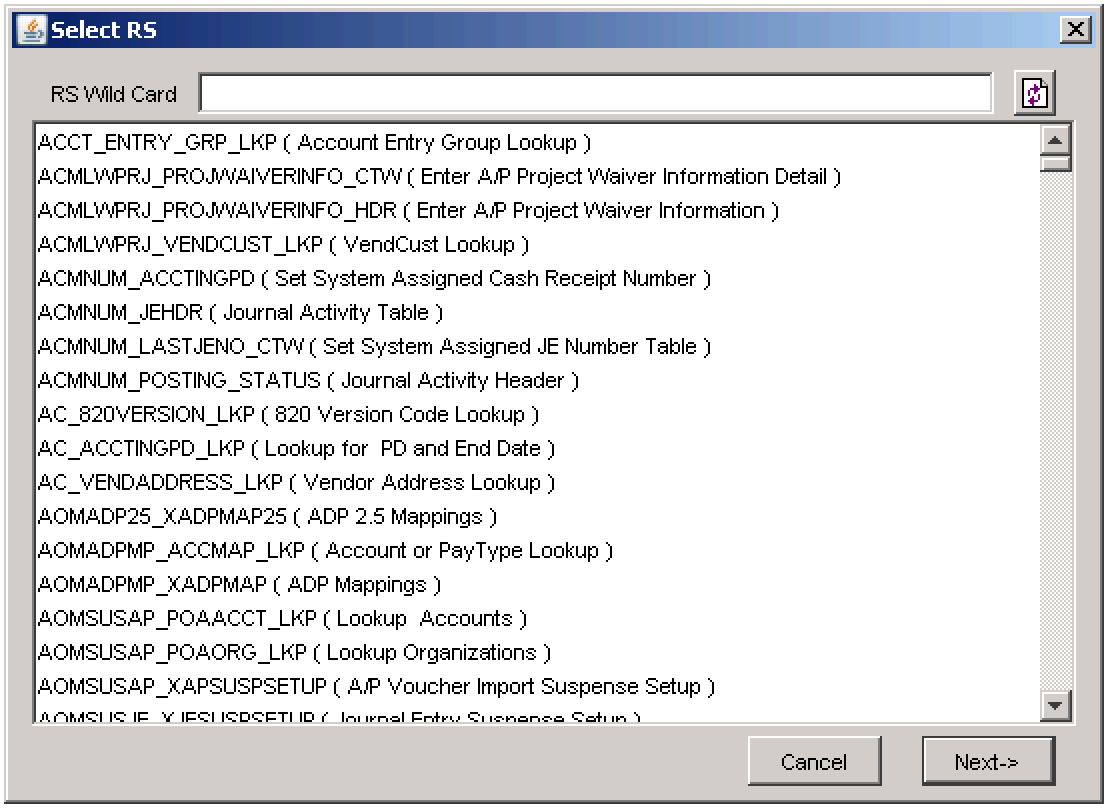
6281 (test)
CHILD (qqq)
DEMO (demo)
DIMA_TEST (DIMA_TEST)
JEN2_EXT (Jen's 2nd Extensibility Project)
JEN_PROJ (Jen's Project)
PROJ1 (Test proj 1)
RAS_PROJ (xxx)
RAS_PROJ2 (TEST PROJ)
RAS_PROJ3 (xx)
RAVI_PROJ (Test)
SAVITHA (Extensibility Testing)
TEST (test proj)
TESTING (test)
TEST_PROJ1 (TEST PROJ1)
TEST_PROJ2 (TEST PROJ 2)
TEST_PROJ3 (TEST PROJ 3)

2. Select a project and then click **Next**. The Select Extension Unit for new **RS** dialog box appears.


Select Extension Unit for new RS

PROJ1_TEST (test)
PROJ1_UNIT1 (Unit Uno)
PROJ1_UNIT2 (Unit 2)

3. Select an extension unit and then click **Next**. The Select RS dialog box appears.



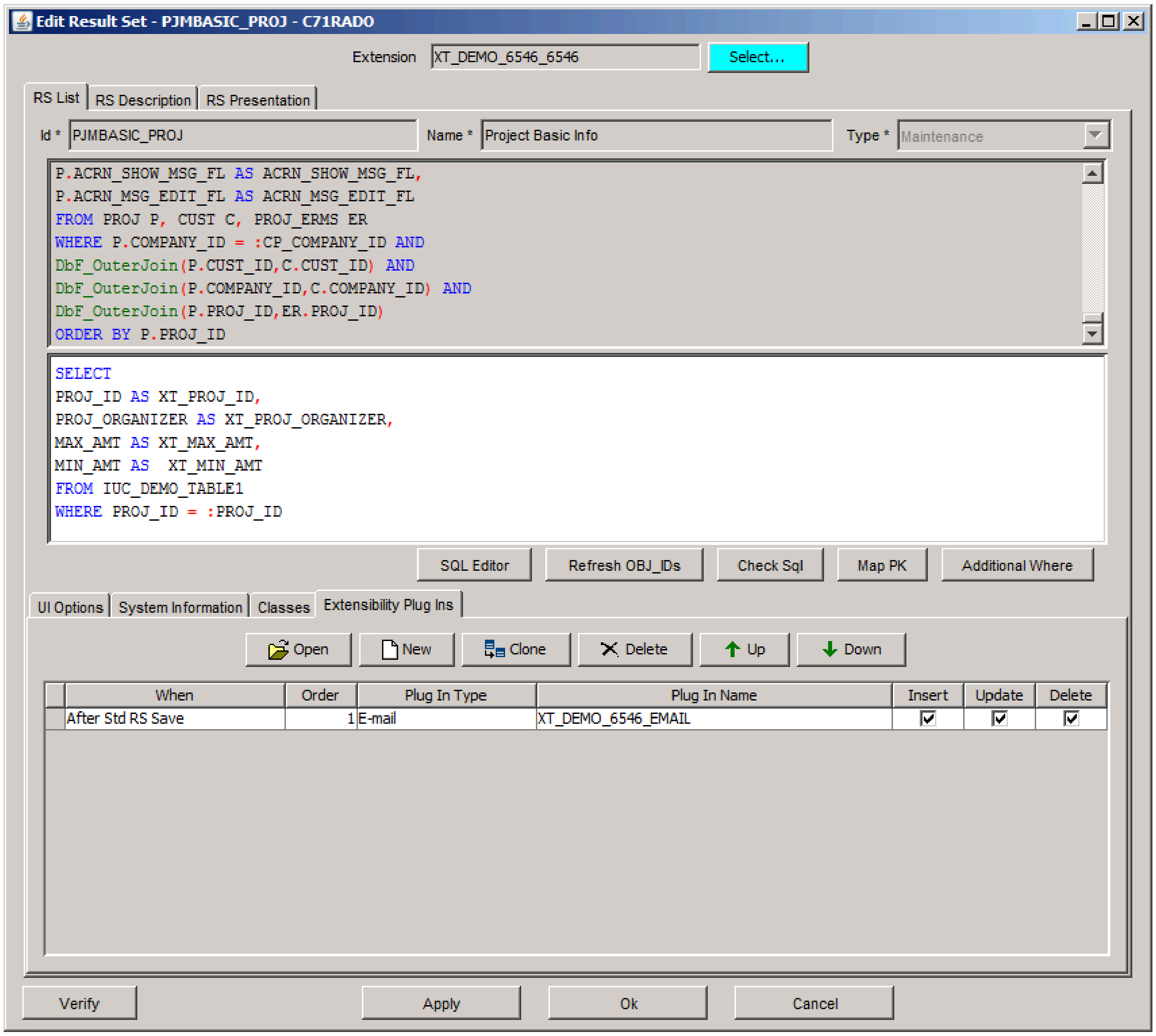
Select RS

RS Wild Card

ACCT_ENTRY_GRP_LKP (Account Entry Group Lookup)
ACMLWPRJ_PROJWAIVERINFO_CTW (Enter A/P Project Waiver Information Detail)
ACMLWPRJ_PROJWAIVERINFO_HDR (Enter A/P Project Waiver Information)
ACMLWPRJ_VENDCUST_LKP (VendCust Lookup)
ACMNUM_ACCTINGPD (Set System Assigned Cash Receipt Number)
ACMNUM_JEHDR (Journal Activity Table)
ACMNUM_LASTJENO_CTW (Set System Assigned JE Number Table)
ACMNUM_POSTING_STATUS (Journal Activity Header)
AC_820VERSION_LKP (820 Version Code Lookup)
AC_ACCTINGPD_LKP (Lookup for PD and End Date)
AC_VENDADDRESS_LKP (Vendor Address Lookup)
AOMADP25_XADPMAP25 (ADP 2.5 Mappings)
AOMADPMP_ACCMAP_LKP (Account or PayType Lookup)
AOMADPMP_XADPMAP (ADP Mappings)
AOMSUSAP_POAACCT_LKP (Lookup Accounts)
AOMSUSAP_POAORG_LKP (Lookup Organizations)
AOMSUSAP_XAPSUSPSETUP (A/P Voucher Import Suspense Setup)

4. Use the **RS Wild Card** field to narrow the list and then select a result set.

5. Click **Next**. The Edit Result Set dialog box appears.



Extension: XT_DEMO_6546_6546

Id *: PJMBASIC_PROJ
Name *: Project Basic Info
Type *: Maintenance

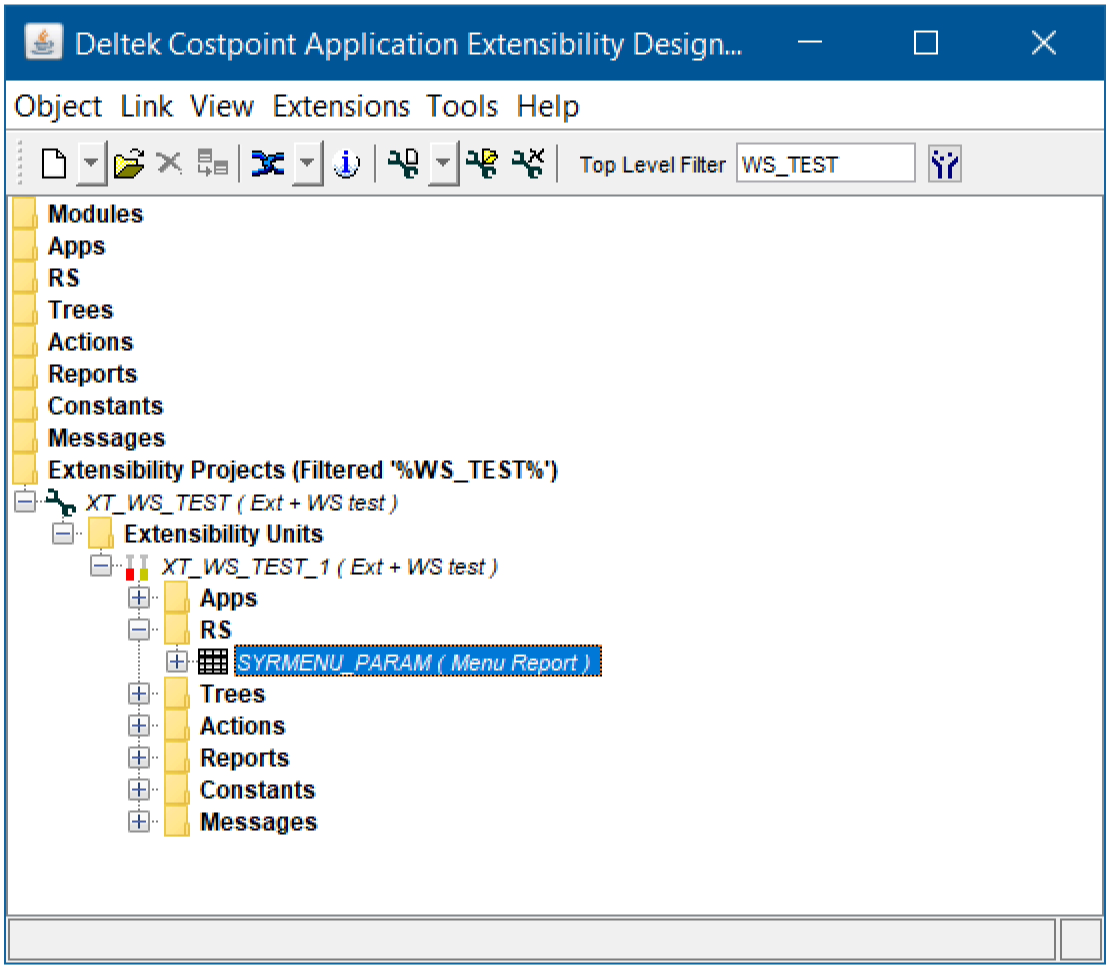
```
P.ACRN_SHOW_MSG_FL AS ACRN_SHOW_MSG_FL,
P.ACRN_MSG_EDIT_FL AS ACRN_MSG_EDIT_FL
FROM PROJ P, CUST C, PROJ_ERMS ER
WHERE P.COMPANY_ID = :CP_COMPANY_ID AND
DbF_OuterJoin(P.CUST_ID,C.CUST_ID) AND
DbF_OuterJoin(P.COMPANY_ID,C.COMPANY_ID) AND
DbF_OuterJoin(P.PROJ_ID,ER.PROJ_ID)
ORDER BY P.PROJ_ID
```

```
SELECT
PROJ_ID AS XT_PROJ_ID,
PROJ_ORGANIZER AS XT_PROJ_ORGANIZER,
MAX_AMT AS XT_MAX_AMT,
MIN_AMT AS  XT_MIN_AMT
FROM IUC_DEMO_TABLE1
WHERE PROJ_ID = :PROJ_ID
```

When	Order	Plug In Type	Plug In Name	Insert	Update	Delete
After Std RS Save	1	E-mail	XT_DEMO_6546_EMAIL	✓	✓	✓

6. Customize the behavior of the result sets. The Edit Result Set dialog box has 3 major tabs. The **RS List** tab contains properties that represent a result set as a whole, while the **RS Description** and **RS Presentation** tabs contain the properties of each RS object.

7. When you are done, click **Ok** to save the RS extension and close the Edit Result Set dialog box. The result set displays under the RS folder under the selected extensibility unit.


Deltek Costpoint Application Extensibility Design...
Object Link View Extensions Tools Help
Top Level Filter WS_TEST
Modules
Apps
RS
Trees
Actions
Reports
Constants
Messages
Extensibility Projects (Filtered '%WS_TEST%')
XT_WS_TEST (Ext + WS test)
Extensibility Units
XT_WS_TEST_1 (Ext + WS test)
Apps
RS
SYRMENU_PARAM (Menu Report)
Trees
Actions
Reports
Constants
Messages

Please keep in mind that you can't have multiple Extensions of the same regular object (for example, RS, or Application or Action) that modify presentation part of that object. So, if you have multiple Extensibility Units that change the same object, please make sure they are assigned to different users/user groups and never to the same user.

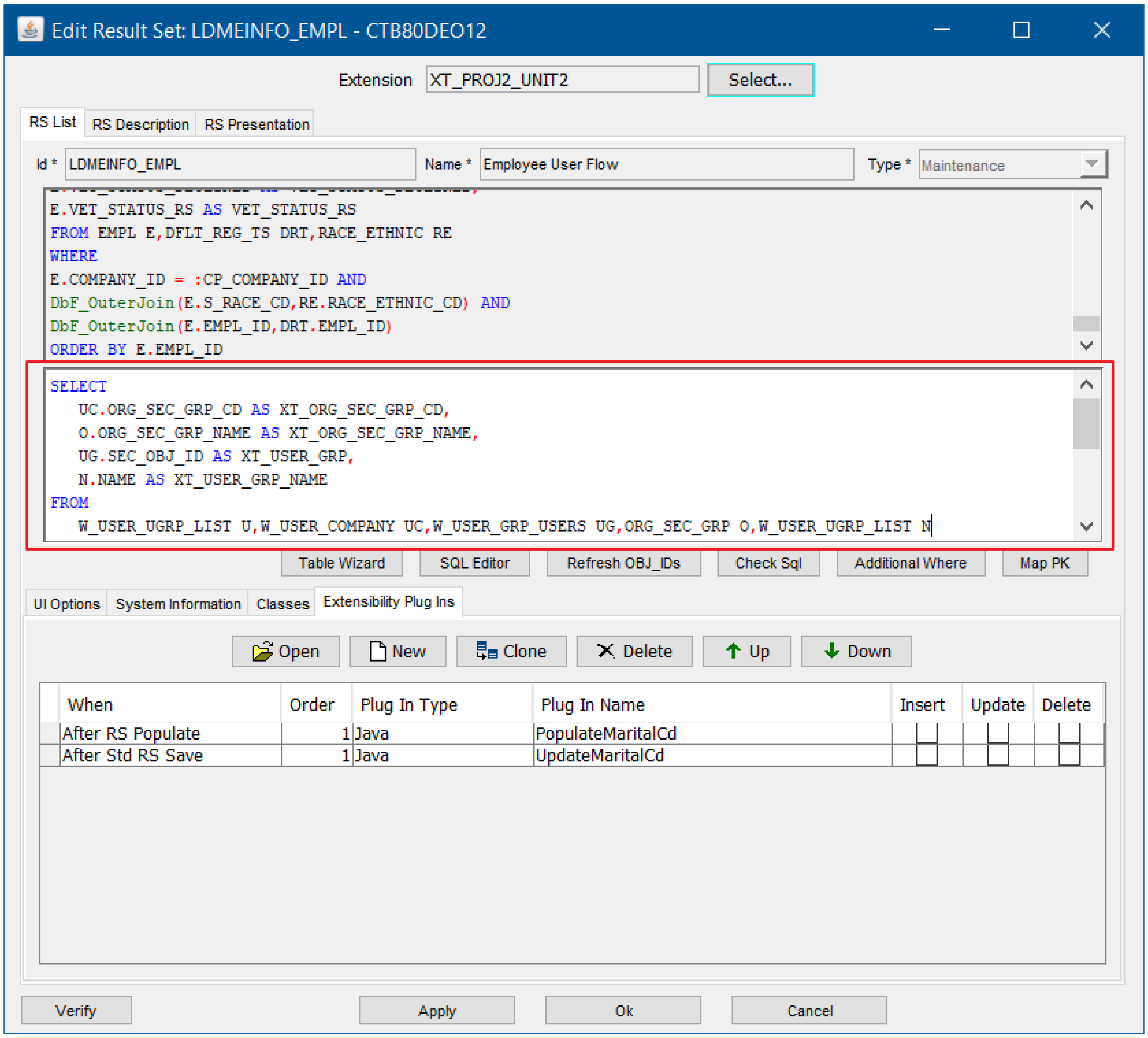
Also, when extending Existing RS, Deltek recommends that you minimize the number of changes you are making and minimize the list of customized standard Objects to make sure your extension will not be affected by future changes to standard RS Objects by Deltek's developers when they fix bugs or expand RS functionality. A good rule of thumb should be to leave all properties of standard RS objects unaffected unless you absolutely have to achieve your customization goals.

RS List Tab

When you extend a result set, the RS List tab displays the properties that were set by a Costpoint developer. These properties are greyed out and are displayed for information purposes only.

You can change the title of the RS that end users see on the screen in the **RS List** tab. To do this, click the UI Options tab and then enter an **Extensibility Title**. If you decided to return to the original RS title, clear that field.

If you need to add additional fields for a given RS, use the SQL Select extension box (under standard SELECT statement) to add the additional extensibility select statement:



```
E.VET_STATUS_RS AS VET_STATUS_RS
FROM EMPL E,DFLT_REG_TS DRT,RACE_ETHNIC RE
WHERE
E.COMPANY_ID = :CP_COMPANY_ID AND
DbF_OuterJoin(E.S_RACE_CD,RE.RACE_ETHNIC_CD) AND
DbF_OuterJoin(E.EMPL_ID,DRT.EMPL_ID)
ORDER BY E.EMPL_ID
```

```
SELECT
   UC.ORG_SEC_GRP_CD AS XT_ORG_SEC_GRP_CD,
   O.ORG_SEC_GRP_NAME AS XT_ORG_SEC_GRP_NAME,
   UG.SEC_OBJ_ID AS XT_USER_GRP,
   N.NAME AS XT_USER_GRP_NAME
FROM
   W_USER_UGRP_LIST U,W_USER_COMPANY UC,W_USER_GRP_USERS UG,ORG_SEC_GRP O,W_USER_UGRP_LIST N
```

When	Order	Plug In Type	Plug In Name	Insert	Update	Delete
After RS Populate	1	Java	PopulateMaritalCd	☐	☐	☐
After Std RS Save	1	Java	UpdateMaritalCd	☐	☐	☐

The extension SELECT can reference standard OBJECT_IDs in the WHERE clause. This extension SELECT statement will be executed for each row of data returned by the standard SELECT (that is displayed above and grayed out). Fields in the extension SELECT statement can also be saved if settings for them are set appropriately in the RS Description tab (see next section).

Tip: The naming convention for new extensibility object IDs is as follows. Each Object ID should start with "XT_" prefix. This can be achieved by using the **as** predicate in your SELECT statement:

SELECT COLUMN_1 as XT_COLUMN_1 from YOUR_CUSTOM_TABLE

You can use **SQL Editor** to write, examine and execute your SELECT statements.

Here is how your system is going to address saving of data in Custom tables for RS that has extension SQL: A typical use case that we are addressing is when the custom table contains several fields that needs to be maintained along with the main entity (for example, Project, Account, Bill).

To address this use case, the developer of the extension must:

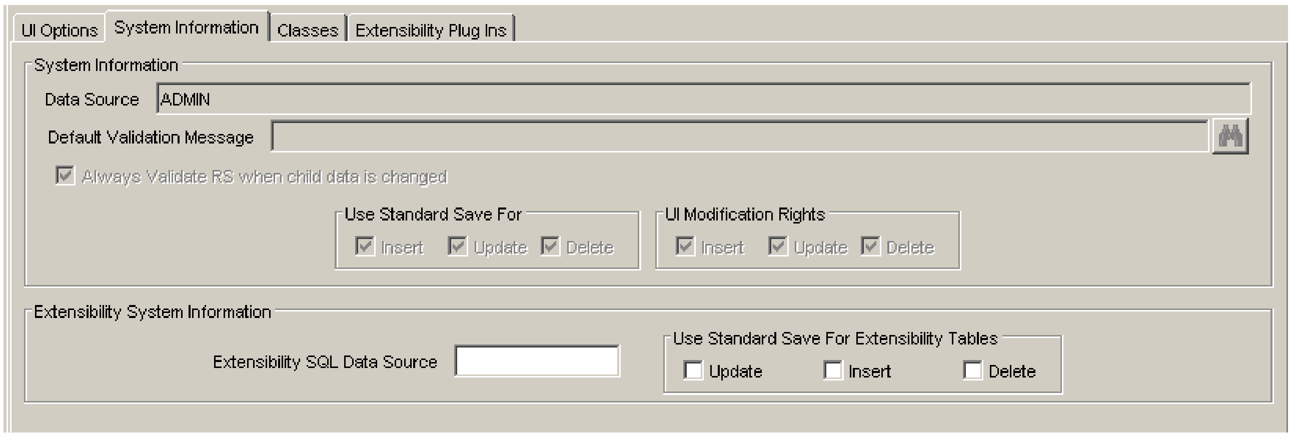
1. Enter SQL for custom table including WHERE clause that references Objects from standard RS. For example, if the standard object is 'ORG_ID' your select statement will look like this:

   ```
   SELECT ORG_NAME as XT_ORG_NAME, ORG_TOP_FL as XT_ORG_TOP_FL where ORG_ID =
   :ORG_ID
   ```

2. The extensibility data source in the System Information tab (for the extensibility SELECT statement) can be entered if your custom data table is located in segment other than DATA(previously DELTEK DB schema). If your custom table(s) is in the same regular Transaction schema, then leave the **Data Source** field blank.

   Please note that when extending the existing RS, your custom Data Source must be on the same DB platform (Oracle or MS SQL Server) as your main RS. If you have a situation where custom data is being stored in the DB on a different platform, you will need to create a separate RS/subtask for such custom data.

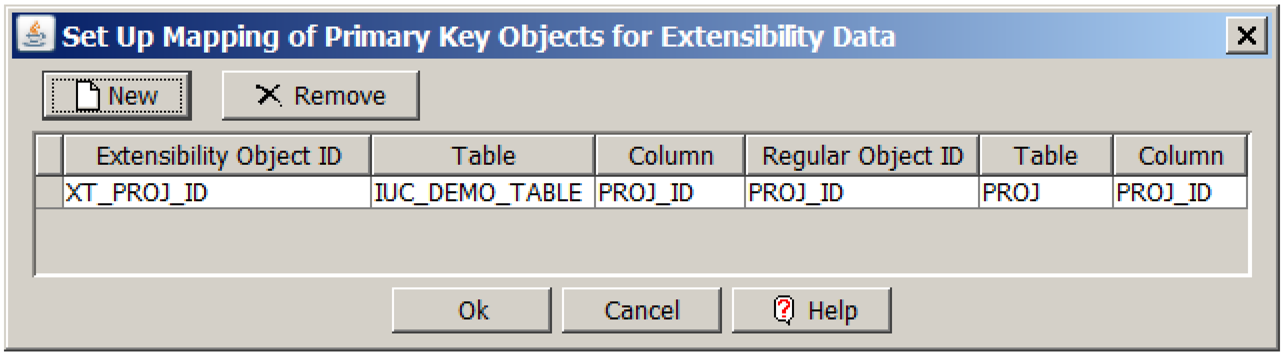
3. Select the check boxes in the **Use Standard Save For Extensibility Tables** group box to automatically save data in the extensibility field. Alternatively, you can leave the checkboxes cleared and write your own custom saving data logic in the Before or After RS Save plug-in(s).



Data Source: ADMIN
Default Validation Message:
[✓] Always Validate RS when child data is changed
Use Standard Save For: [✓] Insert [✓] Update [✓] Delete
UI Modification Rights: [✓] Insert [✓] Update [✓] Delete
Extensibility SQL Data Source:
Use Standard Save For Extensibility Tables: [ ] Update [ ] Insert [ ] Delete

4. Specify primary key fields for custom tables in the RS Description tab (see the RS Description Tab section for more information).

5. Map non-editable PK fields from Custom tables to fields in the main RS. Values in the mapped fields will be synchronized with values in main RS after standard 'Before Save' logic is executed by the system. Mapping of primary key objects can be done by clicking **Map PK**, which is right below the **Extensibility Select** field. This opens the following dialog.


Set Up Mapping of Primary Key Objects for Extensibility Data
New
Remove
Ok
Cancel
Help

Extensibility Object ID	Table	Column	Regular Object ID	Table	Column
XT_PROJ_ID	IUC_DEMO_TABLE	PROJ_ID	PROJ_ID	PROJ	PROJ_ID

6. In the Set Up Mapping or primary Key Objects for Extensibility Data dialog box, create as many rows as you have pairs of PK columns you want to map and specify the pairing by selecting an **Extensibility Object ID** and **Regular Object ID** from the drop-down lists.

Note: When you select a check box in the **Use Standard Save For Extensibility Tables** group box, the system maintains data in custom table(s) using the following rule:

When one (or more) of the editable Extensibility fields is not empty and a row in custom table(s) is not present, a new row will be inserted into custom tables. If all of the editable custom fields are empty, a corresponding row will be deleted from custom table in case it exists. In other words, if a physical row exists in main table, but not in custom table and user enters something in one of the custom fields, a new row will be inserted into custom table. If a user clears all the editable custom fields from existing row in the custom table, the corresponding row will be deleted from custom table(s) by the system. D

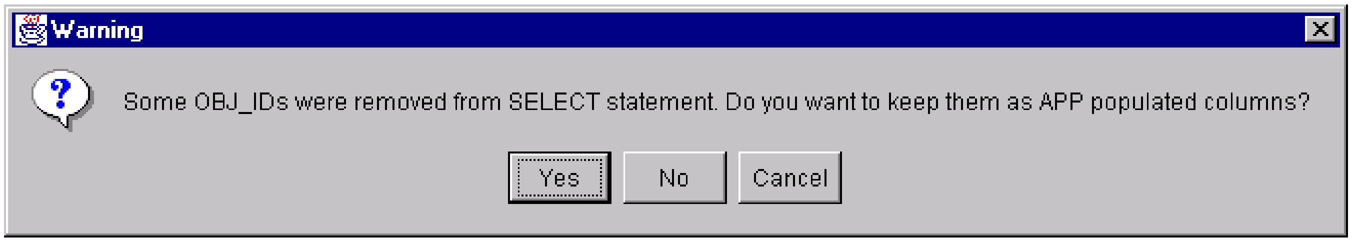
Defaulting functionality will be executed by the system only when user actually clicks a **New Row** button in the browser and will not be executed when a corresponding row is missing from Extensibility tables.

Refresh OBJ_IDs

You must click **Refresh OBJ_IDs** after you are done with entering an extensibility SELECT statement. The system will compare the SELECT SQL statement with existing rows in the RS Description and RS Presentation tabs and on the screen.

Any additional columns added to the SELECT statement will be populated into new RS description and presentation rows.

If there are fields in the RS description row that are orphaned (that is, no longer in the SQL statement), the system will issue a warning statement regarding the mismatch. Click **Yes** to keep the orphaned fields intact, **No** to delete all orphaned fields, or **Cancel** to abort the operation.


Warning

Some OBJ_IDs were removed from SELECT statement. Do you want to keep them as APP populated columns?

The Refresh OBJ_IDs feature does not save the change into the database. It just speeds up your data entry by populating the RS description and presentation rows for you. You still must click **Apply** or **OK** to save them into the database.

Note: Always remember to press **Check SQL** before refreshing OBJ IDs.

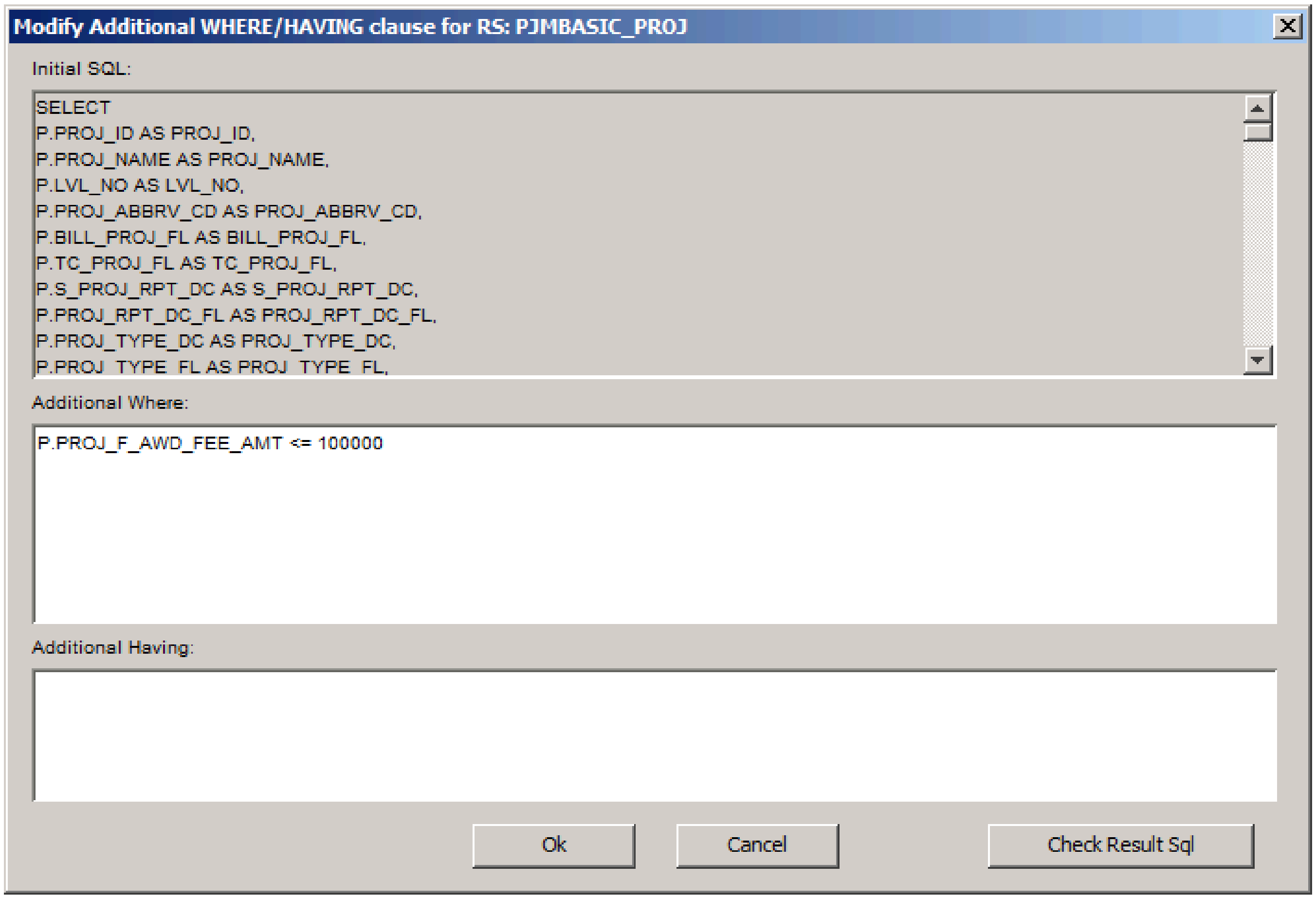
Adding Additional WHERE/HAVING Clauses

You can filter or limit rows returned by a regular query if the regular Result Set has the following

- Either a select statement or Set Select class
- At least one query-able object in the Result Set

To filter or limit the returned rows, you need to enter a custom WHERE and/or HAVING clause that will be added to the main RS regular select statement WHERE/HAVING clause when the query is executed at run-time.

Be careful with this feature, as end users will not be able to see or query filtered rows. To allow end users to see or query filtered rows, click the **Additional Where** button. A dialog box displays in which you will see the standard RS Select and you will be able to enter a custom WHERE and/or HAVING clause.


Modify Additional WHERE/HAVING clause for RS: PJMBASIC_PROJ

Initial SQL:

```
SELECT
P.PROJ_ID AS PROJ_ID,
P.PROJ_NAME AS PROJ_NAME,
P.LVL_NO AS LVL_NO,
P.PROJ_ABBRV_CD AS PROJ_ABBRV_CD,
P.BILL_PROJ_FL AS BILL_PROJ_FL,
P.TC_PROJ_FL AS TC_PROJ_FL,
P.S_PROJ_RPT_DC AS S_PROJ_RPT_DC,
P.PROJ_RPT_DC_FL AS PROJ_RPT_DC_FL,
P.PROJ_TYPE_DC AS PROJ_TYPE_DC,
P.PROJ_TYPE_FL AS PROJ_TYPE_FL,
```

Additional Where:

```
P.PROJ_F_AWD_FEE_AMT <= 100000
```

Additional Having:

After entering WHERE/HAVING clauses (no need to enter key words WHERE/HAVING), you can click the **Check Result Sql** button to verify if combined result is correct.

Table Wizard

If you need to create a new table to store your custom data, you can use the **Table Wizard** button on RS List tab. Clicking the button will open a dialog box that can help you create a simple table for your extensions.

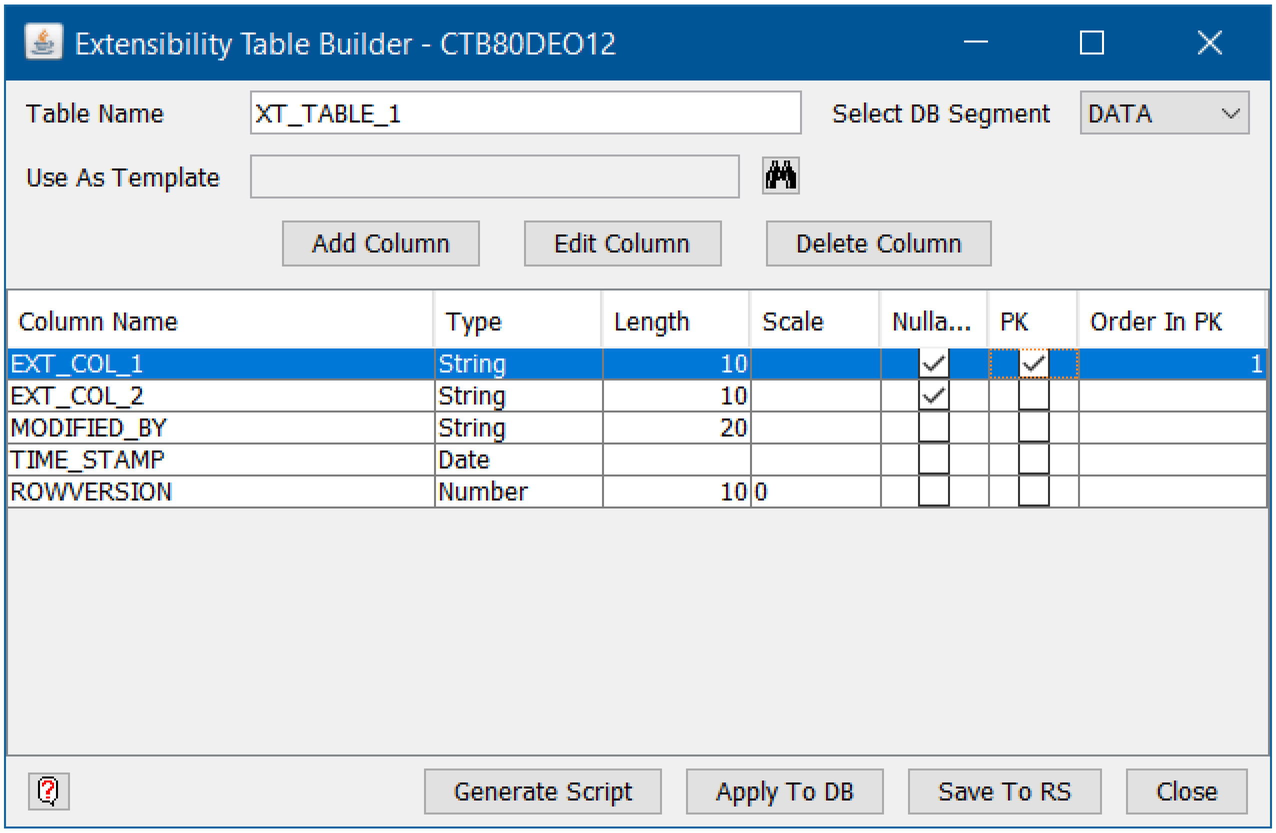


Table Name: XT_TABLE_1
Select DB Segment: DATA
Use As Template:

Column Name	Type	Length	Scale	Nulla...	PK	Order In PK
EXT_COL_1	String	10		✓	✓	1
EXT_COL_2	String	10		✓		
MODIFIED_BY	String	20				
TIME_STAMP	Date					
ROWVERSION	Number	10	0			

This dialog box is a simple new table builder intended to create simple tables. It is not intended for complex situations and should not be considered as a substitute for a dedicated SQL tool like Microsoft Sql Management Studio or Oracle's Sql Developer.

To use the Extensibility Table Builder dialog box:

1. Enter or modify the table name.

2. Take one of the following actions:

- Automatically create a table by selecting a table template.
- Manually create a table by adding the needed columns, and formatting the column properties (**Type**, **Length**, **Scale**, and so on).

3. When done with all columns, click the **Generate Script** button to create a SQL script to create the table.

   You can manually modify the script if needed.

4. Click the **Apply to DB** button to run the script and create the table in the database.

Note: The script saved under the Extensibility Project folder will be packaged with the other files inside the Extensibility Project zip file (as it is recommended).

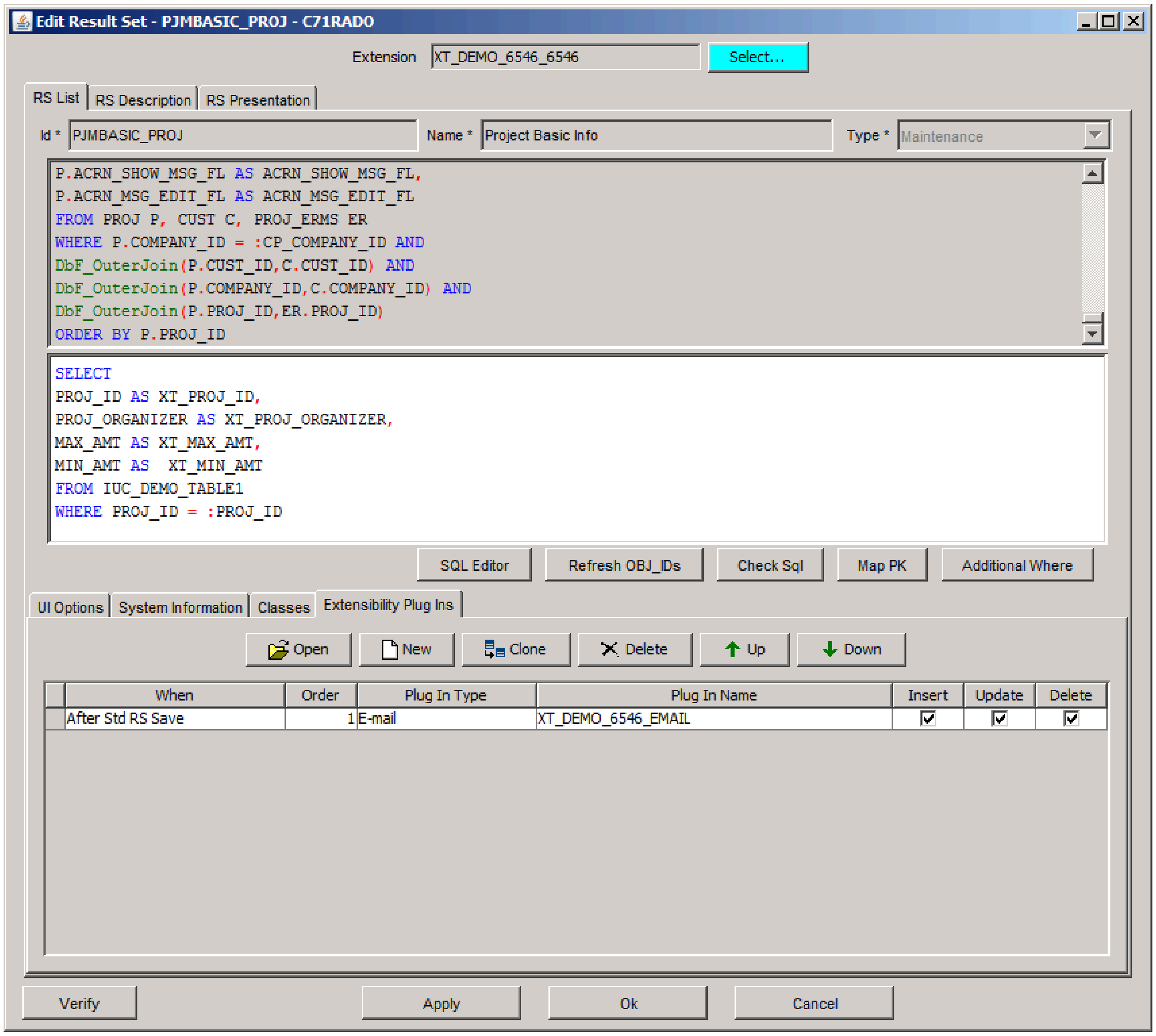
Extensibility Plug-Ins for Result Sets

For Result Sets, you can create one or more extensibility plug-ins to be executed on the following life events of RS:

- After RS Open
- After RS Populate
- Before Line Validate
- After Line Validate
- Before RS Validate
- After RS Validate
- Before RS Save
- After RS Save

To create an extensibility plug-in:

1. On lower half of the Extensibility Plug Ins tab, click **New**.



Extension: XT_DEMO_6546_6546

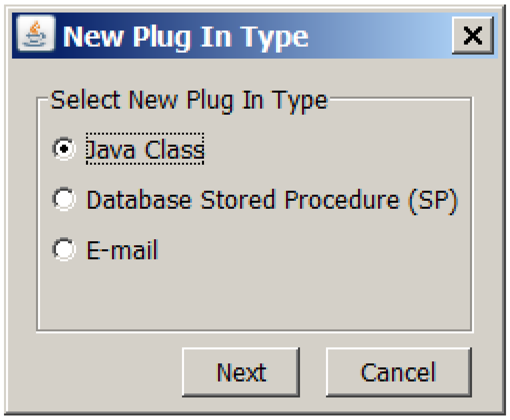
Id *: PJMBASIC_PROJ
Name *: Project Basic Info
Type *: Maintenance

```
P.ACRN_SHOW_MSG_FL AS ACRN_SHOW_MSG_FL,
P.ACRN_MSG_EDIT_FL AS ACRN_MSG_EDIT_FL
FROM PROJ P, CUST C, PROJ_ERMS ER
WHERE P.COMPANY_ID = :CP_COMPANY_ID AND
DbF_OuterJoin(P.CUST_ID,C.CUST_ID) AND
DbF_OuterJoin(P.COMPANY_ID,C.COMPANY_ID) AND
DbF_OuterJoin(P.PROJ_ID,ER.PROJ_ID)
ORDER BY P.PROJ_ID
```

```
SELECT
PROJ_ID AS XT_PROJ_ID,
PROJ_ORGANIZER AS XT_PROJ_ORGANIZER,
MAX_AMT AS XT_MAX_AMT,
MIN_AMT AS  XT_MIN_AMT
FROM IUC_DEMO_TABLE1
WHERE PROJ_ID = :PROJ_ID
```

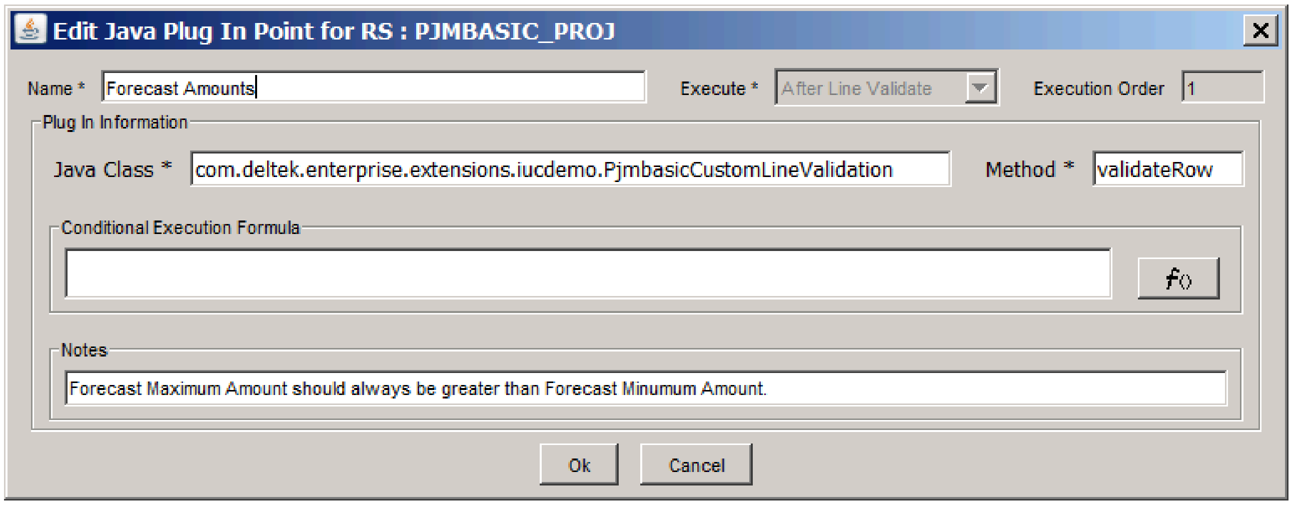
When	Order	Plug In Type	Plug In Name	Insert	Update	Delete
After Std RS Save	1	E-mail	XT_DEMO_6546_EMAIL	☑	☑	☑

The following dialog box displays that will ask you to select what type of plug-in you want to create.



(●) Java Class
( ) Database Stored Procedure (SP)
( ) E-mail

2. You have a choice of creating a java class, database-stored procedure, or email plug-in. For more on entering each type of plug-ins, see Appendix A - Plug-Ins.


Edit Java Plug In Point for RS : PJMBASIC_PROJ

Name * Forecast Amounts
Execute * After Line Validate
Execution Order 1

Plug In Information

Java Class * com.deltek.enterprise.extensions.iucdemo.PjmbasicCustomLineValidation
Method * validateRow

Conditional Execution Formula

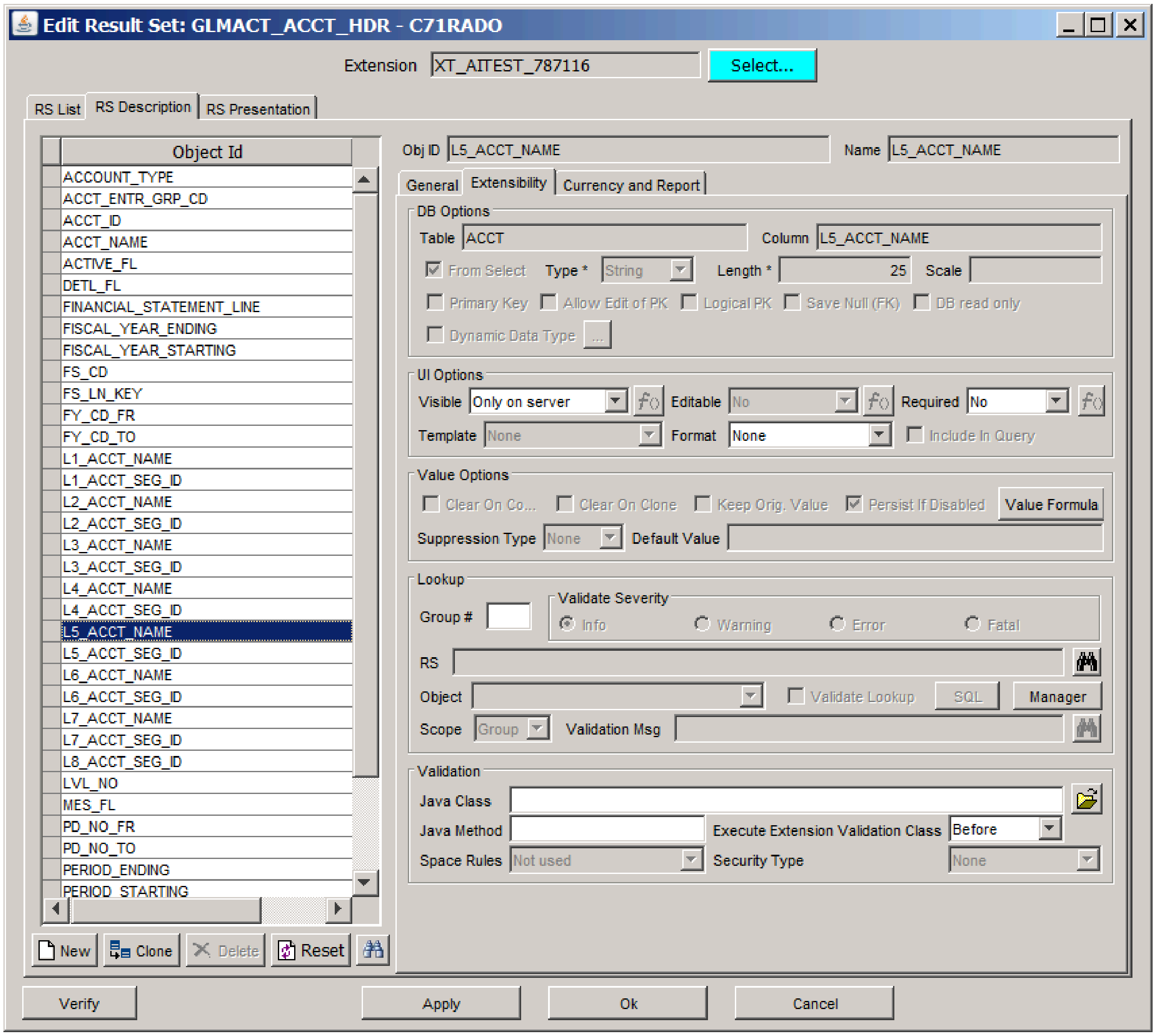
Notes
Forecast Maximum Amount should always be greater than Forecast Minumum Amount.

3. On the same Extensibility Plug Ins tab, you can edit existing plug ins, change the order in which they will be executed, or delete plug ins that were added before and are no longer needed.

RS Description Tab

On the left side, the RS Description tab contains a list of all objects (fields) (visible and hidden) in the current RS. This tab contains metadata information corresponding to each object that describes how Costpoint will handle each object. By selecting a particular object on the left side list, you can see its properties (that is, type, length, visibility, if field is required or not, has lookup or not) displayed on the right side of the screen. Majority of the Object properties are grouped together and are self-explanatory.

RS description rows are populated via the **Refresh OBJ_IDs** button (from the SQL select statement). For additional columns or fields that are not from the select statement, you can add them manually via the **New** button. These are called app-populated fields. When executing data read, the Costpoint system does not populate or save values from app-populated fields, so your extensibility plug-in classes should take care of those tasks if you add them.


Edit Result Set: GLMACT_ACCT_HDR - C71RADO

Extension: XT_AITEST_787116

Tab: RS Description

Object Id
ACCOUNT_TYPE
ACCT_ENTR_GRP_CD
ACCT_ID
ACCT_NAME
ACTIVE_FL
DETL_FL
FINANCIAL_STATEMENT_LINE
FISCAL_YEAR_ENDING
FISCAL_YEAR_STARTING
FS_CD
FS_LN_KEY
FY_CD_FR
FY_CD_TO
L1_ACCT_NAME
L1_ACCT_SEG_ID
L2_ACCT_NAME
L2_ACCT_SEG_ID
L3_ACCT_NAME
L3_ACCT_SEG_ID
L4_ACCT_NAME
L4_ACCT_SEG_ID
L5_ACCT_NAME
L5_ACCT_SEG_ID
L6_ACCT_NAME
L6_ACCT_SEG_ID
L7_ACCT_NAME
L7_ACCT_SEG_ID
L8_ACCT_SEG_ID
LVL_NO
MES_FL
PD_NO_FR
PD_NO_TO
PERIOD_ENDING
PERIOD_STARTING

Obj ID: L5_ACCT_NAME — Name: L5_ACCT_NAME

Tab: Extensibility

DB Options — Table: ACCT; Column: L5_ACCT_NAME; From Select: checked; Type *: String; Length *: 25; Scale: (blank); Primary Key, Allow Edit of PK, Logical PK, Save Null (FK), DB read only, Dynamic Data Type: unchecked

UI Options — Visible: Only on server; Editable: No; Required: No; Template: None; Format: None; Include In Query: unchecked

Value Options — Clear On Co..., Clear On Clone, Keep Orig. Value: unchecked; Persist If Disabled: checked; Suppression Type: None; Default Value: (blank)

Lookup — Group #: (blank); Validate Severity: Info (selected), Warning, Error, Fatal; RS: (blank); Object: (blank); Validate Lookup: unchecked; Scope: Group; Validation Msg: (blank)

Validation — Java Class: (blank); Java Method: (blank); Execute Extension Validation Class: Before; Space Rules: Not used; Security Type: None

To edit a row, highlight the entry on the left and edit the attributes on the right. There are three major tabs:

- General: This tab has all the settings that Costpoint developer specified for a given object. These are all greyed out. This tab is not available for the Extensibility Only RS objects.
- Extensibility: This tab is almost an exact copy of General tab, but here you can change some properties of the standard objects or fully control all the properties available for Extensibility Only objects.
- Currency and Report: This tab displays standard and extensibility properties corresponding to currency and report options of each object.

Note: There are 2 differences between what can be done on the General and Extensibility tabs. You cannot use Java script validations with extensibility. In addition, when setting your **Validation Java**

Class and Method for standard objects, you must specify if this validation method should be called before or after the call to standard validation class.

Fields and Options

Object ID List

The table on the left represents all the data containing objects in the RS. It is a combined list of standard and extensibility objects and is presented listed in alphabetical order.

The **Object Id** column contains the name of the object ID.

The **From DB** column (scroll right to see) is read only. If selected, it means this object comes from the Select statement (versus object populated by the application java class).

Object ID

Object ID for SQL select columns is automatically formed by the system. Additional objects added in the Description tab (or Presentation tab) can be named by the user.

Note: All Extensibility Objects should start with prefix "XT_"

The object ID is unique per result set. It can be up to 72 characters. It is used to transport data back and forth between various parts of the system. (HTML, Web Tier, App Tier). It must be upper case and contain only alphanumeric characters. Except for underscore character, no special character is allowed (for example %, &, (), space, etc.)

Avoid all Java, JavaScript, and HTML key words. Also, avoid the following key words used by system JavaScript.

"actDiv"	"bCan"	"blankRow"	"cbInput"	"cbTxt"
"dTbl"	"firstCol"	"firstRec_E"	"firstRec_N"	"frmvw"
"fTbl"	"fTblRw"	"H1"	"H2"	"hideform"
"hp1"	"hp2"	"hScr1"	"img"	"lastRec_E"
"lastRec_N"	"maxOld"	"midOld"	"midRow"	"minOld"
"pd_E"	"pd_N"	"pDiv"	"popHdrBtn"	"popHdrDiv"
"popRef"	"pu_E"	"pu_N"	"rsLine"	"rsLine2"

"rsLineTable"	"rsTblLine"	"SEL_ALL"	"selcol"	"subtaskBtn"
"subtaskSpan"	"tbHdrDiv"	"tblvw"	"tbTbl"	"track1"
"values"	"vp1_E"	"vp1_N"	"vp2_E"	"vp2_N"
"vScroll1_E"	"vScroll1_N"			

Avoid words that start with the following prefixes.

"col"	"edit_cd__"	"edit_pic"	"metadata"	"rh__"
"sel_cd"	"tabedge"	"tabsp"	"Row" followed by a number	"Row" followed by an underscore.

Note: The system will now validate and disallow the use of these reserved words automatically.

Naming Convention

Use ALIAS in SQL statement to shorten the object ID or to standard object reference. An alias name is the name you use in the SELECT SQL statement for this RS. For example:

```
SELECT PROJ_ID AS XT_PROJ_ID FROM PROJ would create and ID of XT_PROJ_ID
```

You should use alias name whenever a column name is commonly used in more than one table.

For Non-SQL Columns (Added Manually)

Enter a unique ID within this result set up to 72 characters. Always include the db column name as part of the ID. Use a descriptive name for the object ID. Avoid names such as DUMMY_COL, HIDDEN_COL, SYSTEM_COL, and so on. Use a descriptive name for its purpose such as XT_LINE_NO_SUM, XT_LINE_NO_CONST, XT_UPDATE_MODE_FL

Name

Since all labels in the system are internationalized and customizable, the object name is created so that we can have a fixed point of reference. They will be used for trouble shooting, tech support, and so on.

Table/Column

This area displays the database table and column of this object. These are filled in automatically via the SQL statement and upon clicking **Refresh OBJ_IDs**. Those added outside the SQL statement will always have these blank.

Type

Select the data type that matches the database type. Percent should be entered as N.

- S: String
- D: Date
- N: Number

If this row is created using the **Check SQL** or **Refresh OBJ_ID** button, the type will be defaulted in from the database.

> **Note:** If the object type is **Date**, format is **Date**, and no lookup is specified for this control, the system will automatically place the calendar control next to this field at run time.

Length or Precision

For string field, this is the max number of characters allowed.

For numeric field, this is the total number of digits (including decimal places but not including decimal point). The system will automatically check at run time if an entry has exceeded this length. For example, a max field in database of 9999.99 will have precision of 6.

Scale

Number data type only. Enter the scale (number of digits to the right of decimal point). For example, a max field in database of 9999.99 will have scale of 2.

From SELECT

This field Indicates that the column is included in the SQL statement.

- New DB fields must be added via the SQL statement/Refresh OBJ_IDs.
- App populated fields added via the **New** button are not 'From Select'. For example, columns showing calculated values or coming from a complicated select that could not be included in the SQL statement. (The values for these columns must be populated in application specific java code)

The Auto Complete feature for a lookup relies on this field. If a field is 'From Select', the framework can execute the auto complete. If no field is 'From Select' (for example, populated from RS populate), this feature will not be available. So even if you have no SQL in design tool but you have a Select plug in class, you must see if this still applies. If it does, check this box to ensure Auto Complete will be available on field using this RS for lookup.

Primary Key

Check if this column is a primary key. For a result set marked with database rights as **Insert**, **Update** or **Delete**, there must be at least one column marked as primary key per table. For multiple tables in the select, indicate primary key for each table that you want insert/update/delete to apply. Since DB activities (insert/update/delete) are set at the RS level (not at the table level), you cannot specify different activities for different tables.

Allow Edit of PK

Check if the primary key can be changed (in update mode).

Due to SQL Server triggers, this currently can only be used on columns not declared as physical PK (for example, if the column only has a unique index). It also shouldn't be used where declarative FK relationships exist, or where logical FK relationships exist. It can be used on logical (non-physical FK) if the application takes care of updating the dependent data.

Currently, the only usage for this is on the Non-contiguous Range (NCR) subtasks on the FROM column.

Save Null (Foreign Key)

If this is selected and the user does not fill the field at run time, system will insert a 'null' instead of a space. You should have it set the same way it is set in Client Server code.

- Select this check box if the column is a logical foreign key (whether or not it is set as a physical foreign key in the database).
- Select this check box for any field where you particularly don't want to save the default value (a space or zero) for any particular reason.

DbRead Only

The Select SQL Statement coded in the S_RS_LIST table will be used to derive the Insert and Update statement for a result set. If a column should be excluded from derived SQL statement, check this box.

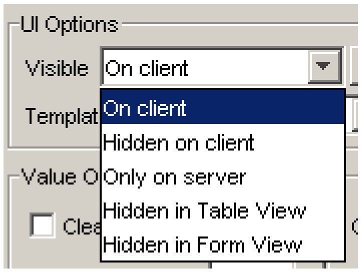
> **Note:** A table that is all read should not have any column marked as primary key (which is used to derive update and delete where clause statement.)
>
> Vice versa, a column marked as DB Read Only cannot be marked as primary key.

Dynamic Data Type

A particular Object can have values of different data types: String, Numeric, or Date. If so, you will need to select this check box. When selected, the Type will contain just the default or typical data type, but your RS Populate Plug-in can add values of different type(s) as well. In this case, you would need to click the ellipses button next to the **Dynamic Data Type** check box and select the object that contains the Object that is determining (driving) data type of the selected Object. Typically, such an object will be hidden, but the value will determine the data type, which in turn drives value formatting, validation, and so on.

Note: This is an advanced feature that is not recommended for use by beginners.

Visible



- B = **On client**: Exists at the client and visible. Formula can be used to conditionally set to invisible. If formula returns false, control will be hidden on both form view and table view. Use the formula only on variables that does not change throughout the life of the session (for example, application constants like Multi Currency license) to avoid objects showing and hiding in the same session (see Hidden in Table View and Hidden in Form View below).
- N = **Hidden on client**: Exists at the client but hidden. Hidden fields can be used to calculate other fields on the screen, used as defaults or used in query criteria. Use of hidden fields should be limited since classifying the field as Only on server can significantly improve performance.
- S = **Only on server**: Exists at the server only. For example, non-null fields that are defaulted with values by the specific application logic. These fields are not required at the client but must be at the server so insert can be completed.
- D = **Hidden in Table View**: Conditionally hidden based on Visible formula. When formula returns false, field will be hidden in Table View. In Form View, the field will be disabled instead hidden (to avoid blank spaces). This code is most applicable when the condition is based an application constant. Therefore, switching from row to row does not change the formula and columns in Table View will not expand or shrink.
- F = **Hidden in Form View**: Conditionally hidden based on Visible formula. When formula returns false, field will be hidden in Form View. In Table View, the field will be disabled (instead of hidden) to avoid table columns shrinking and expanding based on data on each row (as focus is set on each row). This code is most applicable when the condition is based on dynamic data per row.
  For example, you have a combo box to select one of the entity codes (for example, Vendor, Voucher or, Pay Vendor) and the Range Control objects for those that will hide (or show) in Form View appropriately. Range Control objects for these mutually exclusive choices should be stacked up on the Parameter screen for hiding.

Note: When there is no context row, all fields will be hidden in form view (Report and Process parameter screen only.)

Visible formula

This formula applies when the Visible code is **On client**, **Hidden in Table View**, or **Hidden in Form View**.

The formula is an expression that returns true or false. If return is true, the field is visible. If return is false, the field is hidden. If the return value of the formula is number, in this case 0 is treated as false, everything else as true. In case that the return value is String, only empty String "" considered as false. All non-empty strings are treated as true.

Note: Visible formula should be set using fields that are outside the current row or current result set. For example, it should depend on a constant or something in a parent row. Otherwise, it would be confusing when user sets focus on a row and suddenly the field disappears (if the formula is dependent on row data).

Formula evaluation timing

Evaluation of a visibility formula depends on the validation frequency currently in effect.

If the user has logged in using a validation frequency of 'Field', a visibility formula is evaluated when focus changes on the field(s) referred to in the formula. For example, suppose a field has a visibility formula of Parent.ACTIVE_FL=="Y" and the parent ACTIVE_FL field is a check box. If the user clears the ACTIVE_FL check box, the field with the example formula will not be hidden until the user clicks on another field or tabs away from the check box. Likewise, the child result set field will not reappear until the user selects the parent result set ACTIVE_FL check box and then moves focus away from the check box.

If the user has logged in using any validation frequency other than 'Field', a visibility formula is evaluated only when focus changes on the record containing the field(s) referred to in the formula. For example, suppose the user logs into Costpoint with a validation frequency of 'Application'. Continuing with the ACTIVE_FL example, if the user clears the ACTIVE_FL check box in the parent result set, the field with the visibility formula will not be hidden until the user moves to another record in the parent result set. Likewise, the child result set field will not reappear until the user selects the parent result set ACTIVE_FL check box and then moves focus away from the record containing the check box.

Editable

- Select **Yes** if this field is always enabled/editable or **No** if this field is always disabled.
- If a field is conditionally enabled or disabled, set the choice to **Formula** and click the formula button to enter the formula.

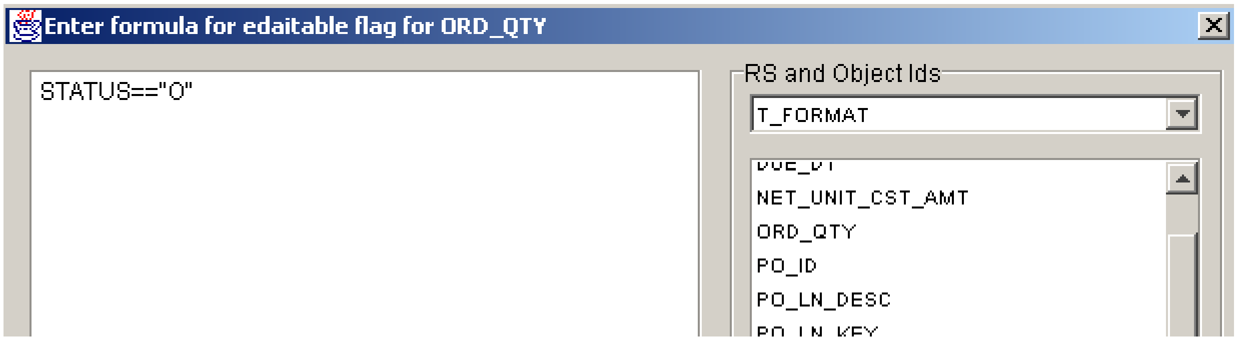
- If a field is only enabled in new row and disabled in existing row mode, choose **New Row Only**.

Formula

The formula is an expression that returns true or false. If return is true, the field is editable. If return is false, the field is not.

> Note: If the return value is a number, 0 is treated as false and everything else is true. If the return value is String, empty String "" is treated as false and everything else is true.

For example:



```
STATUS=="O"
```

The above editable formula for ORD_QTY returns true if Status equals O. In such case, ORD_QTY will be editable. Else if status is not O, ORD_QTY is not editable.

Remember, logic to disable fields at the UI does not substitute logic to validate fields at the server. If a field is to be disabled or blanked out based on other fields, this logic must also exist at the server as a validation.

Test for object or RS existence

- Test for value.
  You can use the key word 'Parent', 'Parent.Parent' or 'Parent.Parent.Parent' to refer to an object at the higher level. For example:
  `Parent.ACTIVE_FL=="Y"`
- Test for existence of an object.
  You can surround the object with the # sign to simply test for the existence of the object in a result set. For example:
  - #PROJ_ID# returns true if this object exists in the current result set.
  - #Parent.PROJ_ID# returns true if this object exists in the parent RS. You can have up to 3 levels of Parent. (That is, Parent.Parent.Parent.ObjID).
- Test for existence of a parent RS. For example:
  - #Parent# returns true if a parent RS exist (that is, the current RS is not in a tree having a parent).

Note: This check for existence can be used in application JavaScript via method IsValidObjectId. However, only OBJECTID is supposed in JavaScript. Parent levels are not supported. For example:

If IsValidObjectId("PROJ_ID") {........

Test for row flag

You can use the following functions to test the row flag:

- **isRowNew**: Test if row is new
- **isRowModified**: Test if row is modified
- **isRowDeleted**: Test if row is marked deleted
- **isValidObjectId**: Test if object Id exist in a result set.

For example, disable a line if row is not new

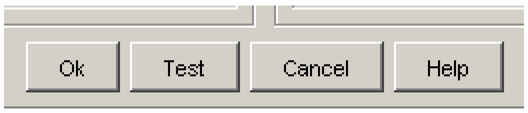
```
Formula = !isRowNew()
```

The row at the parent (or above) can also be tested by passing 'Parent' in the parameter. Do not use Parent.isRowNew(). This is not supported.

For example:

```
isRowNew("Parent") or isRowDeleted("Parent.Parent")
```

Test button



The test button now also tests the formula syntax in Java. The system will generate the java file temporarily, compiles the java class. If there is syntax error, such error will be reported instantly.

Formula evaluation timing

A field's editability formula is evaluated when the field gets focus, either by the user clicking on the field with the mouse or navigates into the field with the keyboard. Unlike visibility and label formulas, the timing of a field's editability formula does not vary depending upon the user's current validation frequency; the behavior is the same no matter whether Field, Record, Application, or Web service modes are in effect.

Deltek

Required

- **No**: Not required
- **Yes**: Required. The system will check the required entry at the client (system's JavaScript) to avoid unnecessary trip to the server.

  > **Note:** A field marked as Required must not be marked as Not Editable since the user will not be able to fill the value if it is blank.

- **Server Only**: Required at the server but not at the client (for example, auto-assigned code that needs to be performed at the server). The system will check for required field at the server after it has given the application the chance to complete the entries in the application's validation steps.
- **Formula**: Similar to Visibility or Editability formulas. This formula is an expression that returns true or false. If return is true, the field is required on the client. If return is false, the field is not.

If control type is **Check box** or **Radio button**, system will default Required to **Yes**.

Template

If an object is of a common type (for example, Project Id, Account Id, Org Id), select the template object available from the **Template** box. Several default entries will be defaulted from the templates to speed up data entry. Certain templates also have special treatment (see list below)

> **Note:** Entries will be defaulted only if the field is blank. If you previously selected a template object and then change to another template object, entries will not be defaulted again unless they were blank.

The following table describes the available templates.

Template	Description
ACCT_ID	Segment formatting is automatic (see note on segment formatting). Validation called on client and server to check for non-alphanumeric characters, excluding the delimiter "-".
ALTPROJ_ID	Segment formatting is automatic (see note on segment formatting). Validation called on client and server to check for non-alphanumeric characters, excluding the delimiter ".".
ASSET_ID	Validation is called on client and server to verify that the field is zero padded if the field contains all numeric values and check that the field is set to upper case

Template	Description
	if the field contains any non-numeric values. NOTE: This validation assumes a maximum field length of 10 characters.)
DOC_ID	System will check for this template and the revision format to determine if a space should be retained when selecting data for the result set.
EMAIL	Validation called on client and server to check for invalid email address.
EMPL_ID	If Lab SETTINGS is to use SSN for EMPL_ID, SSN formatting is automatic. Validation called on client and server to check for non-alphanumeric characters.
FILE_LOCATION_ID	**Added in Beta8**: For location ID for file uploading or file processing using a shared network folder where the application server can have access. If this template is used, cell validation for this object will be invoked and line validation will also be invoked regardless of whether the cell or the line has been modified.
FILE_NAME	Used in conjunction with FILE_LOCATION_ID.
FY_CD	**Added during Beta8**: Fiscal Year
INVC_ID	Invoice ID
INVT_ABBRV	Inventory Abbreviation
ITEM_ID	Item ID
LOC_1, LOC_2	For use in conjunction with WHSE_1, WHSE_2. Do not use this for Asset Location. Segment formatting is automatic if LOC_1 is used and another field in the result set containing the warehouse value has template of WHSE_1. (Similarly for LOC_2). You can have multiple LOC_1 for one WHSE_1 field. However, only one WHSE_1 is allowed per result set. If you have additional warehouse, use WHSE_2. Currently, there should not be any screen where there are more than 2 warehouse fields. Validation called on client and server to check for non-alphanumeric characters, excluding the delimiter "-".
ORG_ABBRV	Org Abbreviation. (See note on Synonym template.)
ORG_ID	Segment formatting is automatic (see note on segment formatting) Validation called on client and server to check for non-alphanumeric characters, excluding the delimiter ".".
PAYROLL_YEAR	Payroll Year as a number. The format default from this template is Year. When a number is entered, the system will validate that Payroll Year is always between

Template	Description
	1901 and 2078. When two-digit values are entered, a conversion is performed as follows: Values entered between 0 and 49 are converted to the years 2000 to 2049, while values entered between 50 and 99 are converted to the years 1950 to 1999. The template will be used on the client and server sides for validation purposes. The format will only be used on the client for formatting purposes.
PROJ_ABBRV	Project Abbreviation (See note on Synonym template.)
PROJ_ACCT_ABBRV	Project Acct Abbreviation
PROJ_ID	Segment formatting is automatic (see note on segment formatting) Validation called on client and server to check for non-alphanumeric characters, excluding the delimiter ".".
REF1_ID, REF2_ID	Segment formatting is automatic. Validation called on client and server to check for non-alphanumeric characters, excluding the delimiter ".". Reference labels on the client are retrieved and set by the system.
REF1_NAME, REF2_NAME	Segment formatting is automatic. Reference labels on the client are retrieved and set by the system.
REORG_ID	Segment formatting is automatic. Validation called on client and server to check for non-alphanumeric characters, excluding the delimiter ".".
REVISION	Item or Part Revision. Do not use this for Fixed Asset Template Revision. System will check for this template and the revision format to determine if a space should be retained when selecting data for the result set.
WHSE_1, WHSE_2	See LOC_1, LOC_2

Synonym Template

Synonym refers to the pair of Org Id/Org Abbreviation, Project Id/Project Abbreviation and Account/Project Account Abbreviation.

The normal behavior is when user enters one field, the other is completed by the system.

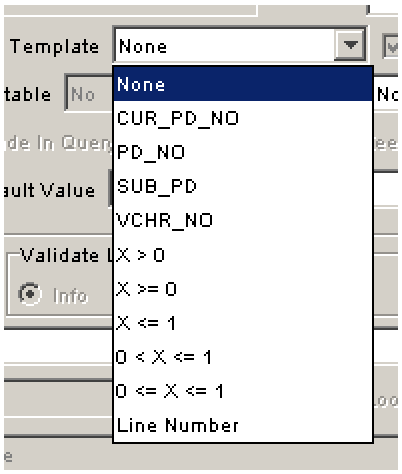
If you use template for synonym fields, when user changes or blanks out one field, the system will blank out the other field. Then if you use the same lookup and lookup group for both fields, value of one will be filled from the entry of the other.

Developer must make sure to assign scope=Field for both columns. This will ensure that lookup will not use the other field for filtering.

Note: Blanking out fields is done as part of the UI. If UI is disabled (on login), this will not happen. Therefore, it is possible to still have conflicting ID and Abbrev. Validation is supposed to be done to validate for valid combination if both fields are present.

Numeric Templates

If the field type is numeric, the following templates are available.



Numeric Validation Templates
PD_NO (validate number is > 0)
SUB_PD (validate number is > 0)
VCHR_NO (validate number is > 0)
X > 0
X >= 0
X != 0 (added in Beta9)
X <= 1
0 < X <=1

Numeric Validation Templates
0 <= X <=1
Line Number template: Validate number cannot be less than 1. Also, when cloning is done, the field will not be cleared (even when design setting for Clear on Clone is set to Y).

Automatic validation (at the client and the server) is done for these template types.

Do not write Java or JavaScript validation for these numeric templates as it is done automatically by the system.

Format

Different selection for format will appear depending on the type.

Number

- CUR: Currency
- PCT: Percent
- PAD: Zero Padded
  - Example 1
    precision: 5
    scale: 0
    User enters: 123
    Field should display: 00123 when user tabs out of the field.
  - Example 2
    precision: 5
    scale: 2
    User enters: 1.23
    Field should display: 001.23 when user tabs out of the field.
- YEAR: When a number is entered, system will validate that Pay Roll Year is always between 1901 and 2078. Add to 2000 if the value entered is between 0 to 49, Add to 1900 if the value entered is between 50 to 99

Date

- D: Date (will display month day and year)
- DMD: Month and Day (MM/dd)
- T: Time
- DT: Date/Time

For date format, user can enter 6 or 8 numbers; any other number of numbers is an error.

If the user enters any delimiters we do not delimit any part of the value.

If the user enters no delimiters we delimit the value.

String

- **UP**: Upper case,
- **LO**: Lower case,
- **PASS**: Password,
- **SSN**: SSN
- **IR**: Revision
- **PAD**: Zero Padded

Password fields are automatically masked with * when user type data in the field. SSN are automatically masked with hyphens when displayed.

Zero padded fields are padded with zeroes if the field contains all numeric values. If the field contains non-numeric values the field is set to upper case.

Include in Query

Select if the column should be included in query criteria selection. If none of the column is selected, the result set will not display the **Query** button.

Include in Query is not available if the column is App Populated (not from Select SQL).

Clear on Copy

Select if the value should be cleared (not duplicated) on copy/repeat line function.

> **Note:** If there is default value (for new line), when a line is duplicated, the default value will be used after it has been cleared.
>
> If the field has a **Line Number** template and the user clones the whole document, the field will not be cleared even when this is selected. The reason is we want to retain all the line numbers from the original document in the cloned document. The field will be cleared, however, if user only duplicates one line (versus clone the header which includes many lines).

Clear on Clone

Check if the value should be cleared on clone function.

Note: If field is at the server only and there is default value (for new line), the default value will be used after it is cleared on clone. It's important to have this check box set properly regardless if the node has not been set for clone in the tree link.

Keep Original Value

Select this if you want the system to make the original value (on select) available to your java class. In your server validation (Java code), you can compare new value to the original value to perform certain validation (via system's RowSetInterface.getOrigObjectId method).

Persist if Disabled

When business rule dictates that a field should not be changed in update mode due to some data condition, the developer enforces the rule by setting the field disabled using the Editable formula.

This is fine but since all business rules must be implemented at the server, how can the system easily perform this check without developer duplicating this code at the server.

One approach is automatically disregarding any changes ('Do not Persist') when a field is evaluated to disabled. This would be the default behavior.

If the developer wishes to override this behavior and persist the value ("Persist If Disabled"), select this box. Certain logic should then be added to the application's server-side code to validate this changed value. For more details, refer to PersistingData_New.doc under doc\system

Attention: See the "Editable Formula/Java Generation" section for more information on how this is implemented.

Note: This should be selected for non-editable description field such as **Name**, **Description**, and so on when it is derived from an ID field on the screen. Typically, the description field in these cases will be DB Read only, and the ID - Description pair will be in the same lookup group with the description field set to Field scope and the ID field set to Group scope.

Since the Description field is permanently disabled, there may be a temptation to clear this flag. But, if you clear the flag, there will be a minor UI problem:

The new value will not be kept when the row is pushed (persisted) to the app server (not db server). Therefore, if the user changes the ID (either through lookup or by typing in the value -- in Field mode), a new description is retrieved. However, when user scrolls the row out from client cache (row is persisted to server), and then scrolls it back in, we would see the old value in the description field, not the new value.

For non-editable fields that are saved to the database (for example, Entry User ID), you should have server code set them and do not check this box so that data will not be accepted from input.

Value Formula

To enter value formula for a field, click **Value Formula** to open the formula dialog box. If the field currently has formula, the button caption will show a different color.

Formula is entered on the left pane. The right pane is provided for convenience so you can see what object ID is available from what result set.

Click **Test** to check your formula. This does not guarantee successful execution of the formula. It just checks to see if the syntax is valid.

General

- Formula is executed on the client browser so there is no database access and all variables are from fields available at the client.
- Formula is a UI mechanism for user convenience only. It can be used to show a running total. It can execute a default calculation or a required calculation. If it executes a required calculation, such calculation must also exist on the server side.
- Formula cannot be executed for server only fields. Such logic must be added to the server side application code.

Elements

- You can use object Id from the same result set by using the Object Id in the formula
- Or from parent result set by using Parent.<object id> or Parent.Parent.<object id>
- You can use object id from a child result set by qualifying the id with the RS id of the child result set. Child object ID can only be used in the formula if it is in the SUM format (since child result set can only be looked at by the parent in total).
- You can also use application constants. Constants must be assigned to your app before they can be used. Constants can be referred to in formula by pre-pending the constant with the keyword CP.

Expression

- A formula has to be a valid Java expression

- Clicking on the Test button will perform a syntax check
- You can use arithmetic expressions such as +, -, *, etc.
- String functions are available such as length(), substr() , etc.
- You can use conditional expression such as (object1>0?object2:object3). This formula evaluates: if object 1 > 0 then = object 2, else object 3.
- You can use SUM for a child result set: SUM(<child RS ID>.<child Object Id>). For example, SUM(RQ_LN_ACCT.RQ_LN_ACCT_AMOUNT). Note: You cannot use multiple (*) or divide (/) with a SUM.

Event

- Formula is called when value is changed in a related field and user moved focus away from such field.
- Unlike validation, Formula is always called when the field is changed. Validation is not called if the field is changed to blank.
- Formula in a child result set using "parent" is not called when a parent field is changed. This is because only rows in context are recalculated.
- Formula in a parent result set is called when rows in related children result set is added, deleted or changed.
- Formula is called by the system after client side java script. It is also called after server side java validation (if in fast or medium mode).
- Formula is not called when the result set is first populated. Therefore, if the field is not part of the Select SQL, it must be populated via App Populate Java Interface on data loaded.

Order of execution

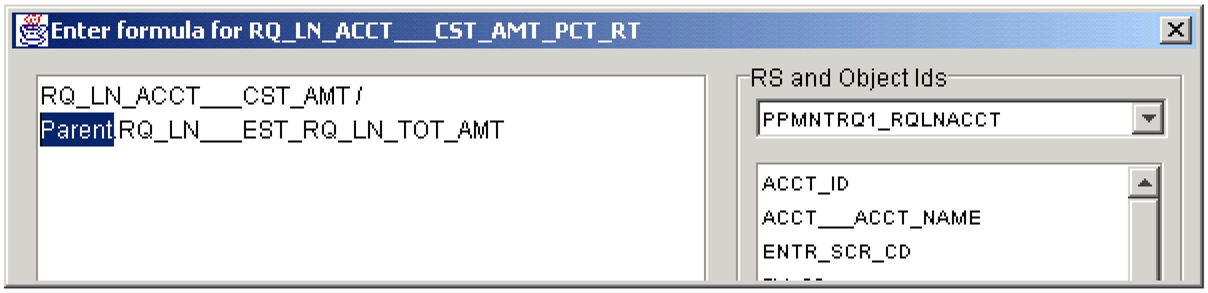
- When a column is modified, all formulas that directly depend on this column will be recomputed with columns that belong to the same result set as the modified column being computed first.
- Avoid circular reference (that is, Field A depends on Field B and Field B depends on Field A). The presence of circular reference will make the re-calculation order unpredictable.
- Columns from the parent result set linked to the modified column through SUM() is computed next.
- Then all formulas that depend on columns that were programmatically recomputed (current result set first and then parent result set).

Example 1: Clearing a field when another field is cleared

```
(ORG_ID == ""||ORG_ABBRV_CD==""?"":O___ORG_NAME)
```

This formula in the object O___ORG_NAME uses the 'if' expression to clear the ORG_NAME field when ORG_ID or ORG_ABBRV_CD field is cleared. Otherwise, leave the ORG_NAME as is.

Example 2: Referencing a parent column with the Parent keyword

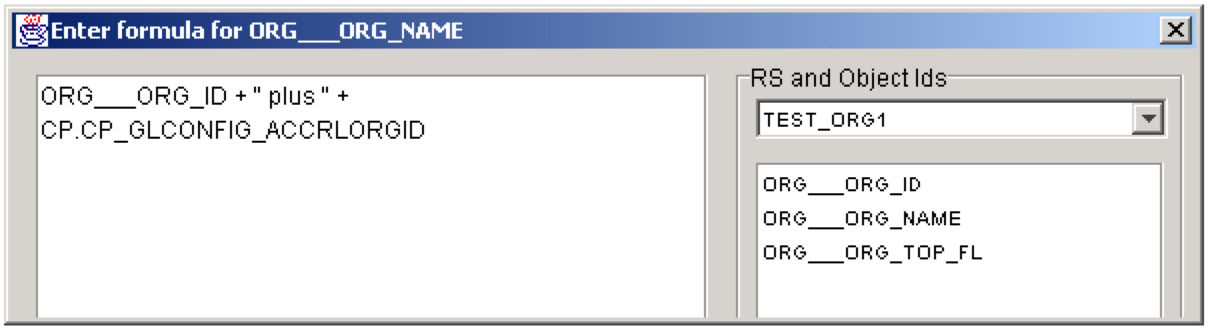


```
RQ_LN_ACCT___CST_AMT /
Parent RQ_LN___EST_RQ_LN_TOT_AMT
```

- This formula calculates the cost rate by dividing the child Cost Amount column with the Parent Total cost column.
- This formula is called when the child Cost Amount column is changed but is NOT called when the Parent Total cost column is changed.

Example 3: Using an application constant

This formula appends the ORG_ID to an application constant and put the resulting value in the ORG_NAME field.



```
ORG___ORG_ID + " plus " +
CP.CP_GLCONFIG_ACCRLORGID
```

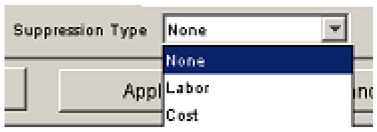
Suppression Type

Select the appropriate suppression type for this field.

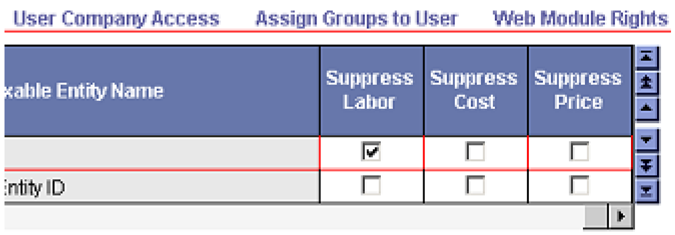
- Labor
- Cost
- Price

If the user has been flagged for suppression in the User Company assignment screen, the corresponding field suppressed here will not show the data.

The field is still visible but no data will be shown.

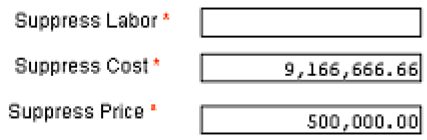


Below shows where suppress flags are set when company is assigned to such user. (Maintain user app - SYMUSR).



xable Entity Name	Suppress Labor	Suppress Cost	Suppress Price
	☑	☐	☐
ntity ID	☐	☐	☐

At run time, the field is disabled and no data will be shown



Suppress Labor *	
Suppress Cost *	9,166,666.66
Suppress Price *	500,000.00

Default Value

Enter the default static value on init/new. The system will fill the column with this value on new line at the client.

Default value can be a literal or a JavaScript expression. The expression can use literal or values from data available at the browser such as Object ID reference.

Literal/Number Default

Enter it with surrounding single quote (even if it's a number). For example, **Y** or **Pending**. For number values, **0** for default of zero.

Expression Default

If the default is a JavaScript expression, enter without the single quotes. Reserved word Parent, Parent.Parent or Parent.Parent.Parent (up to 3 levels) can be used to refer to Object ID in the result sets above this current result set. (If there is a need for additional level above the 3^rd^ level, please submit request to the Architecture team).

Note: Do not use JavaScript method new Date() as default for current date (which uses the workstation date). You should use the application constant CP.CP_CURRENT_DATE. This will default with the server date instead.

1. Using keyword Parent: (No surrounding quotes). Parent.Parent and Parent.Parent.Parent expressions are also supported. For example, default to the parent RS's PROJ_ID value:

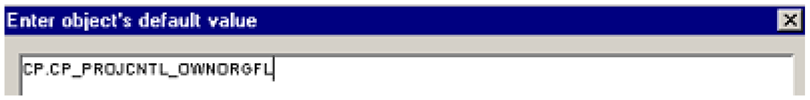
   `Parent.PROJ_ID`

2. Using App constants.

   App constants can also be used as defaults. App constants are created in the screen Constants/Constant List, and then assigned to an app in the Constants/App to Constants Links.

   After a constant has been assigned to the app, you can use it for default by pre-pending the id with the key word CP.

   Example below will default the value of the field to the value stored in table PROJ_CNTL, column OWN_ORG_FL



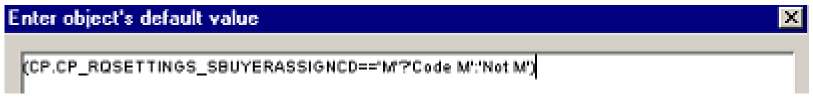
```
CP.CP_PROJCNTL_OWNORGFL
```

3. Test for object or RS existence

   Attention: See Test of object or RS existence in Editable Formula earlier.

4. Using JavaScript expression:

   Example below will default the value of the field to 'Code M' if in table RQ_SETTINGS, column S_BUYER_ASSIGN_CD is M. Otherwise; the default is 'Not M'.



```
(CP.CP_RQSETTINGS_SBUYERASSIGNCD=='M'?'Code M':'Not M')
```

Server Default

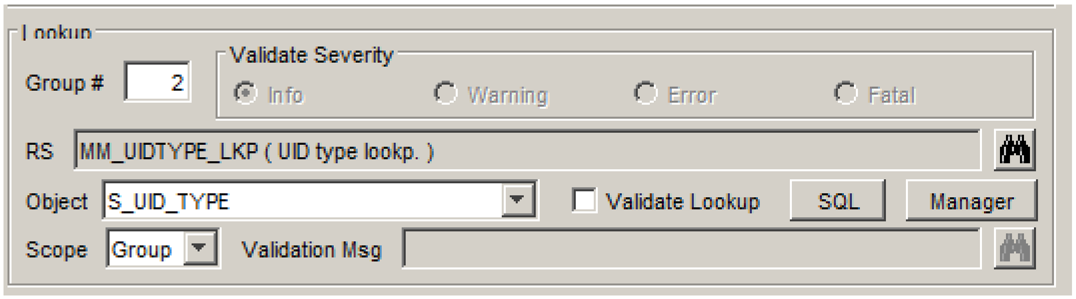
If Visible code = Server Only, system will fill the column with this value if it is blank. This is done whenever the new line is persisted back to the server the first time.

Note: Only these values can be defaulted at the server. All other values or expression will be ignored.

- Only literal value (begins with single quote)
- Application constant (begins with CP)
- Or value from parent or higher RS (begins with the keyword Parent)

The system will also persist default values at the server if the row has been created not via a browser client (for example, Clone which is done entirely at the server).

Lookup Info



Group # 2
RS MM_UIDTYPE_LKP (UID type lookp.)
Object S_UID_TYPE
Scope Group

Complete this section if the field is a lookup field.

Note: Lookup is available even if the field is not editable (so that other related information can be shown when lookup is shown). However, in this case, user will not be able to change the value of the disabled field.

Group No

If lookup is used, you must enter a number greater than zero.

Group No is used to group fields using the same result set as lookup. Fields with the same lookup RS ID and Group no will be brought back together as a group when a lookup is done for any field in the group.

For example, Acct ID and Acct Name fields use the same lookup CPM_ACCT_LKP. If you assign the same Group no, lookup Acct ID will also bring back Acct Name and vice versa. If you assign them different Group no, lookup Acct ID will not bring back Acct Name or vice versa.

If you want to bring back some values but do not want to use the same lookup, see SPECIAL SITUATION section at the end of this lookup section.

Lookup RS

Select Stand Alone RS for lookup.

Tip: Searching for an existing lookup

Before creating a new lookup, search for an existing lookup and see if there is a common lookup that has already been created for sharing.

Deltek strongly recommends using only "Lookup" RS Type Result Sets as lookups as well as recommends fully testing a particular lookup RS to make sure it is working as intended.

Validate Lookup

Use Validate lookup to do object validation whenever possible to avoid duplication of application code. The system will perform the validation to make sure entry is valid in this result set. Developer does not need to write additional code.

Note:

- A lookup RS that has no SQL select statement (design time or run time) cannot be used to do validate lookup. Make sure you review the lookup result set to check if the SQL select statement is there in either the tool or in the java class.
- If a column is being populated with data in the java class, such a column cannot be used as validate lookup.
- Do not use validate lookup if the lookup value is dependent on another field that is not part of the lookup group (that is, the lookup SQL contains references to the parent RS). This is because validate lookup is called in object validation and if user changes the value of parent columns, this lookup field will not be re-validated. (Same rule applies for any object validation. It should not depend on the value of another field).

If Validate Lookup is selected, select the level of severity if this validation fails (for example, ERROR, FATAL, etc.).

- **Result Set**: Select from the list box (or from lookup) the result set that will be used as lookup for this field.
- **Object ID**: Select from the list box the object from the result set that corresponds to this field.

Scope

The system always applies the LIKE condition filtering on a lookup field (when such lookup field has existing value).

When there is a lookup group involved (2 or more fields sharing the same lookup and values are brought back together upon looking up any of them), the scope code controls the filtering applied to the other field.

Select from the following scope codes:

- G: Group. When looking up other fields in the same group, the filter for the lookup where clause will include this field
- F: Field. When looking up other fields in the same group, the filter will not include this field.

Use G for parent field and F for child field.

Filtering on other fields is always set to EQUAL condition, and filtering on own field is always set to LIKE condition.

For example, two fields on the screen sharing the same lookup and group no: State Code and Zip Code.

- State Code has scope of G (parent field)
- Zip has scope of F. (child field since it can be filtered on state code)

When you look up State, the WHERE clause will NOT include Zip in the filter.

When you look up Zip, the WHERE clause WILL include State in the filter.

In addition, when a G field is being blanked out, the related F field will also be blanked out (since G field controls the group).

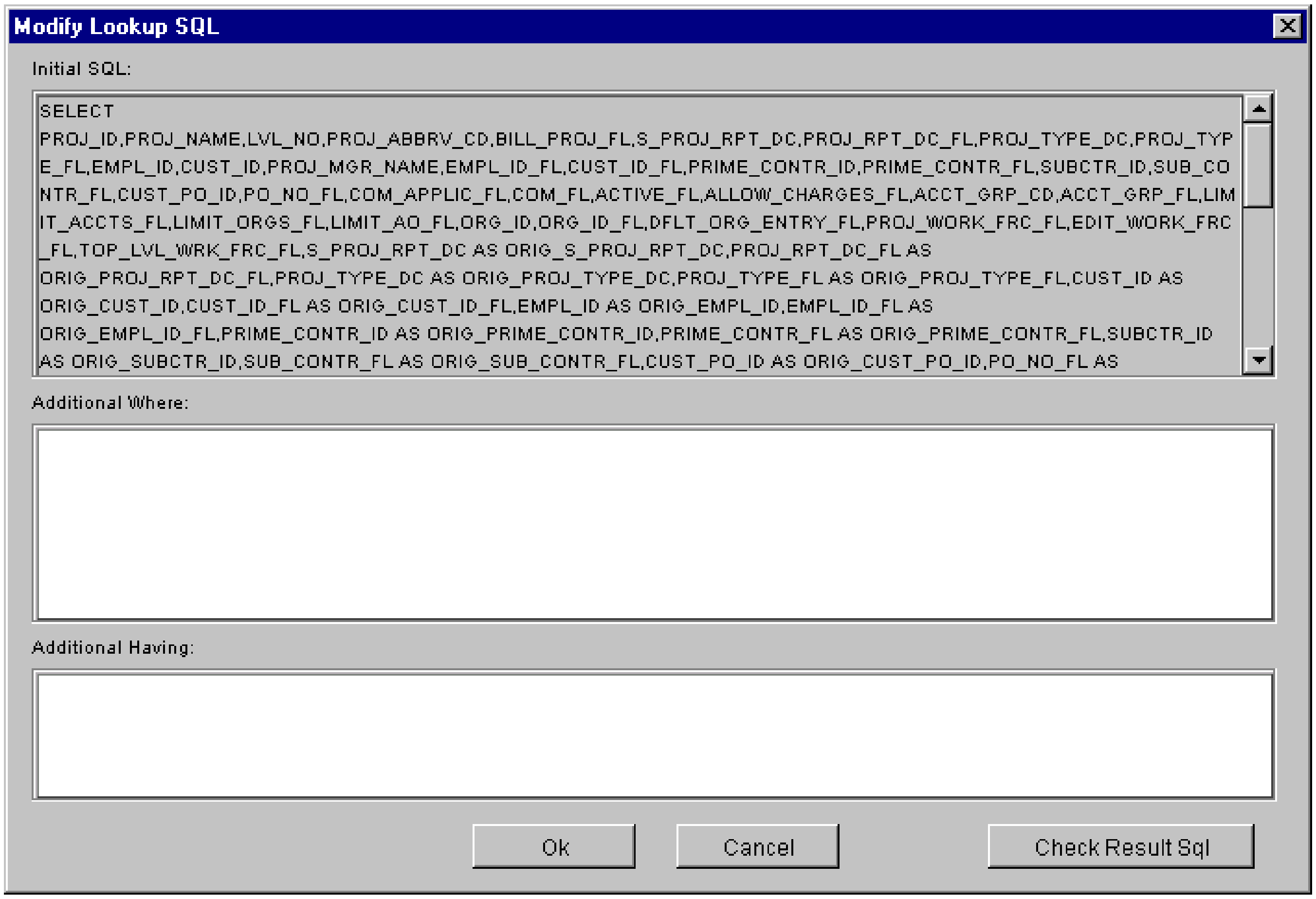
Validation Message

If Validate Lookup is checked, choose the message Id that will be used to issue the error message if validation fails.

If this is blank, the system will use the default validation message that is assigned to that lookup RS.

SQL Button

Click **SQL** to see the SQL used in the lookup and to add additional filter for the SQL Where clause or Having clause.



```
SELECT
PROJ_ID,PROJ_NAME,LVL_NO,PROJ_ABBRV_CD,BILL_PROJ_FL,S_PROJ_RPT_DC,PROJ_RPT_DC_FL,PROJ_TYPE_DC,PROJ_TYP
E_FL,EMPL_ID,CUST_ID,PROJ_MGR_NAME,EMPL_ID_FL,CUST_ID_FL,PRIME_CONTR_ID,PRIME_CONTR_FL,SUBCTR_ID,SUB_CO
NTR_FL,CUST_PO_ID,PO_NO_FL,COM_APPLIC_FL,COM_FL,ACTIVE_FL,ALLOW_CHARGES_FL,ACCT_GRP_CD,ACCT_GRP_FL,LIM
IT_ACCTS_FL,LIMIT_ORGS_FL,LIMIT_AO_FL,ORG_ID,ORG_ID_FL,DFLT_ORG_ENTRY_FL,PROJ_WORK_FRC_FL,EDIT_WORK_FRC
_FL,TOP_LVL_WRK_FRC_FL,S_PROJ_RPT_DC AS ORIG_S_PROJ_RPT_DC,PROJ_RPT_DC_FL AS
ORIG_PROJ_RPT_DC_FL,PROJ_TYPE_DC AS ORIG_PROJ_TYPE_DC,PROJ_TYPE_FL AS ORIG_PROJ_TYPE_FL,CUST_ID AS
ORIG_CUST_ID,CUST_ID_FL AS ORIG_CUST_ID_FL,EMPL_ID AS ORIG_EMPL_ID,EMPL_ID_FL AS
ORIG_EMPL_ID_FL,PRIME_CONTR_ID AS ORIG_PRIME_CONTR_ID,PRIME_CONTR_FL AS ORIG_PRIME_CONTR_FL,SUBCTR_ID
AS ORIG_SUBCTR_ID,SUB_CONTR_FL AS ORIG_SUB_CONTR_FL,CUST_PO_ID AS ORIG_CUST_PO_ID,PO_NO_FL AS
```

This dialog box contains the following:

- **Initial SQL**: The initial SQL is the exact SQL statement coded for the lookup RS.

> **Note:** Additional filtering for Org Security is done automatically by the system if the lookup RS contains fields coded for Org Security.

- **Additional Where**: To add additional condition for the Where clause for this lookup, enter the additional condition here. For example:
  - ACTIVE_FL = 'Y'
  - ORG_ID = :Parent.A___ORG_ID (A___ORG_ID is an object ID on the current RS). You use Parent to designate the RS that calls this lookup RS.
    Do not include the word "WHERE" in the text.

> **Note:** Do not add where clause for org security. This is handled by the system automatically on the lookup RS.

- **Additional Having**: To add additional Having clause for this lookup, enter it here.

- **Check Result SQL**: Click this button to verify the syntax of the SQL statement

If you want to bring back some value from a lookup to other fields but do not want to use the same lookup in those fields, read the example below

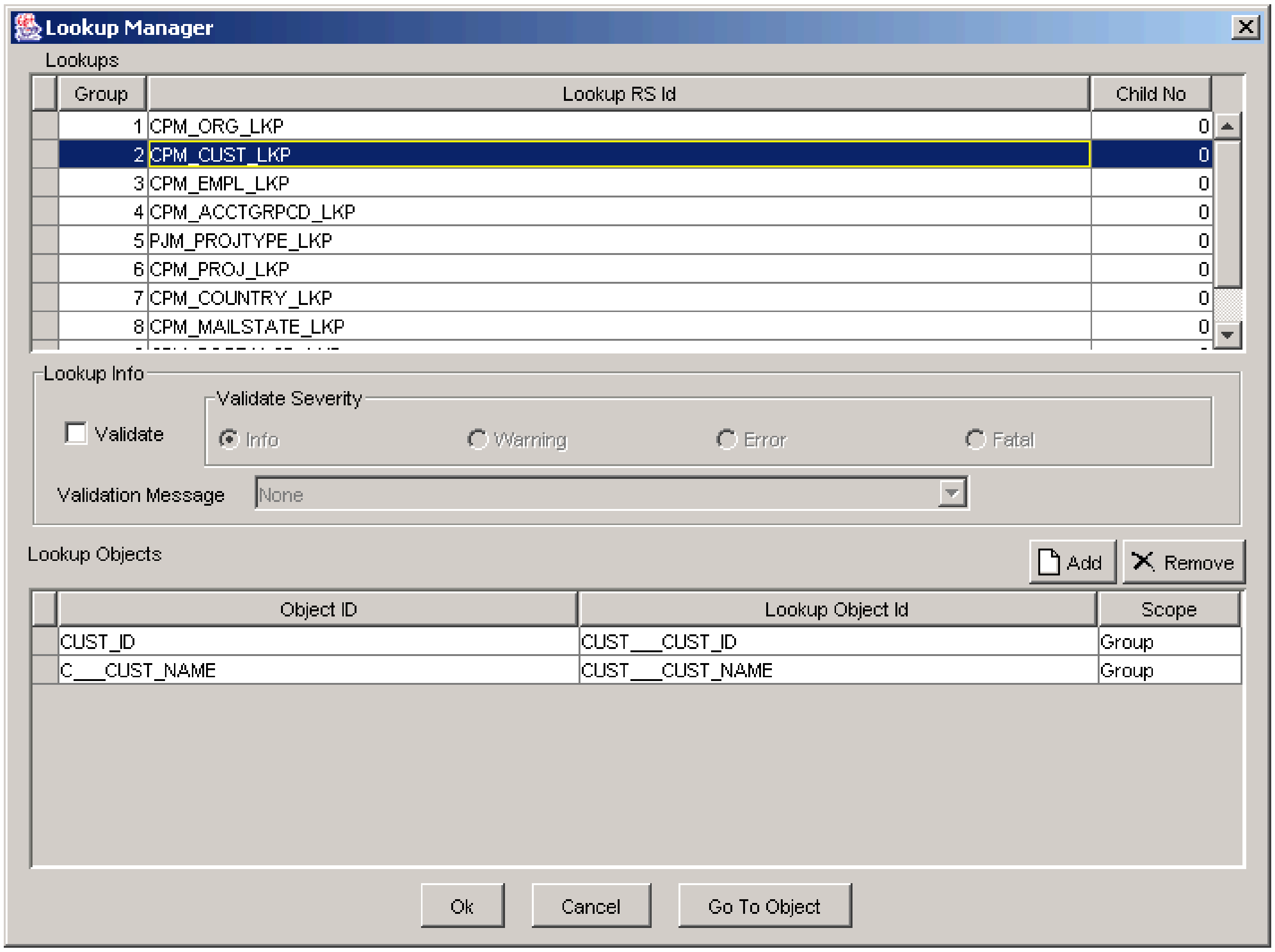
For example, on lookup of Item ID (ITEM table), bring back default Unit of Measure (U/M) from the ITEM table (ITEM.DFLT_UM_CD) to the UM_CD field. But on lookup of UM_CD, select from UM_CD table, not from ITEM table.

There are two columns named ITEM_ID and UM_CD.

- You must create a hidden (at the client) column named UM_CD_HIDDEN.
- Select the ITEM lookup for both ITEM_ID and UM_CD_HIDDEN and assign them both the same group number so that the values are brought back together.
- Select the UM lookup for UM_CD
- In the value formula field for UM_CD, put this formula to set it to UM_CD_HIDDEN: (UM_CD_HIDDEN).
- When user looks up ITEM, UM_CD_HIDDEN will be brought back and will fill the value of UM_CD via formula.
- Lookup UM_CD is now independent of lookup ITEM_ID.

Lookup Manager

Lookup Manager allows user to navigate and manage all lookups used in the whole result set. It can be accessed from the **Manager** button in the lookup section for any RS Description field.



Lookups

Group	Lookup RS Id	Child No
1	CPM_ORG_LKP	0
2	CPM_CUST_LKP	0
3	CPM_EMPL_LKP	0
4	CPM_ACCTGRPCD_LKP	0
5	PJM_PROJTYPE_LKP	0
6	CPM_PROJ_LKP	0
7	CPM_COUNTRY_LKP	0
8	CPM_MAILSTATE_LKP	0

Lookup Info

[ ] Validate

Validate Severity: [✓ Info] Warning Error Fatal

Validation Message: None

Lookup Objects

Object ID	Lookup Object Id	Scope
CUST_ID	CUST___CUST_ID	Group
C___CUST_NAME	CUST___CUST_NAME	Group

The screen shows all the lookup used in this result set.

For each lookup, you can set it as validated lookup and define the validation severity in here.

Also, the corresponding objects using the lookup is shown in the lower section. You can add or remove these objects from the lookup here. Or change the corresponding Lookup ID for each object. To go to the related object on the RS Description tab, highlight the object click **Go To Object**.

Val JS Class/JS Method

JavaScript is available for standard Costpoint methods only. They are currently not available for extensibility.

Java Class

If applicable, enter the name of the java class where validation is done for this object.

This name should be fully qualified with package name. For example:

com.deltek.enterprise.extensions.xt_project_x.pjmbasic.PjmbasicObjectValidation

Server object validation is executed only when there is a data change in this object and the new value is not blank. For Field connection mode, it will be executed when user leaves focus to another field. For Record connection mode, it will be executed when user leaves the line. For Application or web service mode, it will be executed when the transaction is saved.

Tip: See the Costpoint 8.2 Extensibility Designer Coding Guide for the class naming convention.

Java Method

Enter the name of the method for object validation.

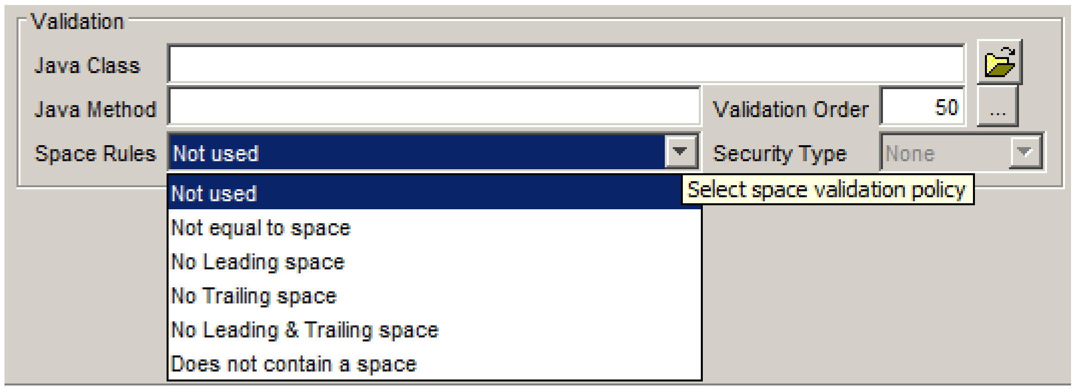
Note:See the Costpoint 8.2 Extensibility Designer Coding Guide for the method naming convention.

Validation Order

- This controls the order of Object validation when called by the system (in Record, Application or web service mode).
- If the user selects Field mode, validation will occur when user tabs out of the field. In other modes, object validation is not called until the line validation is called. When called, Object Validation will happen in the order of this number from low to high.
- It is important to place the critical objects first and dependent objects later so meaningless validations are avoided.
- Order number does not have to be unique. For example, if you do not care about the order among objects, put them in a group by giving them the same number. The system will process all those with zeros first, then ones, twos, etc. But within each group in no particular order.

Space Rules

Select one of the following to trigger the corresponding validation rule done by the framework.


Validation
Java Class
Java Method
Validation Order 50
Space Rules Not used
Security Type None
Not used
Not equal to space
No Leading space
No Trailing space
No Leading & Trailing space
Does not contain a space
Select space validation policy

- Not equal to space (EQ): Fails validation if the field contains only spaces.
- No Leading space (LD): Fails validation if there are leading spaces.
- No Trailing space (TR): Fails validation if there are trailing spaces.
- No Leading & Trailing space (LDTR): Fails validation if there are leading spaces or trailing spaces.
- Does not contain a space (CNT): Fails validation if the field contains space.

Security Type

If this field should have a particular security type attached to it, select the type from the combo box:

- O: Organization
- P: Project
- E: Employee

At run time, the system will automatically add a filter on the SQL select to only select the rows where this security type is allowed for this field for the current user. Only one field per result set should be marked for security type.

Note: Do not use security type for Report or Process Parameter result sets since there is no need to filter parameters. Use it for data result set.

CURRENCY and REPORT TAB

The currency fields are used only when you need to identify objects that are used in multi-currency conversion.

Currency Object ID

Enter the source object id that is used to derive the value of this given "currency" amount field.

Deltek

Currency Field Type

If the field is a "multi-currency" field, enter one of the following codes to identify what type of currency object the given field is:

Rate Group Id (grp1)	(EXCHGRP1_RATEGRPID)
Exchange Rate Date (grp1)	(EXCHGRP1_EXCHRTDATE)
Primary Rate (grp1)	(EXCHGRP1_PRIMARYRT)
Secondary Rate (grp1)	(EXCHGRP1_SECONDRT)
Primary Rate Flag (grp1)	(EXCHGRP1_PRIMARYRTFL)
Source Currency (grp1)	(EXCHGRP1_FRCRNCYCD)
Destination Currency (grp1)	(EXCHGRP1_TOCRNCYCD)
Rate Group Id (grp2)	(EXCHGRP2_RATEGRPID)
Exchange Rate Date (grp2)	(EXCHGRP2_EXCHRTDATE)
Primary Rate (grp2)	(EXCHGRP2_PRIMARYRT)
Secondary Rate (grp2)	(EXCHGRP2_SECONDRT)
Primary Rate Flag (grp2)	(EXCHGRP2_PRIMARYRTFL)
Source Currency (grp2)	(EXCHGRP2_FRCRNCYCD)
Destination Currency (grp2)	(EXCHGRP2_TOCRNCYCD)
Exchange Rate Group1	(MUAMT_EXCHGRP1)
Exchange Rate Group2	(MUAMT_EXCHGRP2)

Note: There are two sets of codes "grp1" & "grp2" which allows for two sets of exchange rates to be maintained.

The **Keep Original Value** check box will automatically be selected by the tool if any Currency Type is selected (except for the last 2 type MUAMT_EXCHGRP1 and MUAMT_EXCHGRP2).

Report Tab

The Report Tab is only used for fields identified as data for the report such as report details or selection criteria on the selection parameter.

Report Variable

Enter the Report Variable name. This is the report variable name as defined in the BIRT Report template and also defined in the Report screen (see Report section). Or click the lookup button to find the report and report variable name.

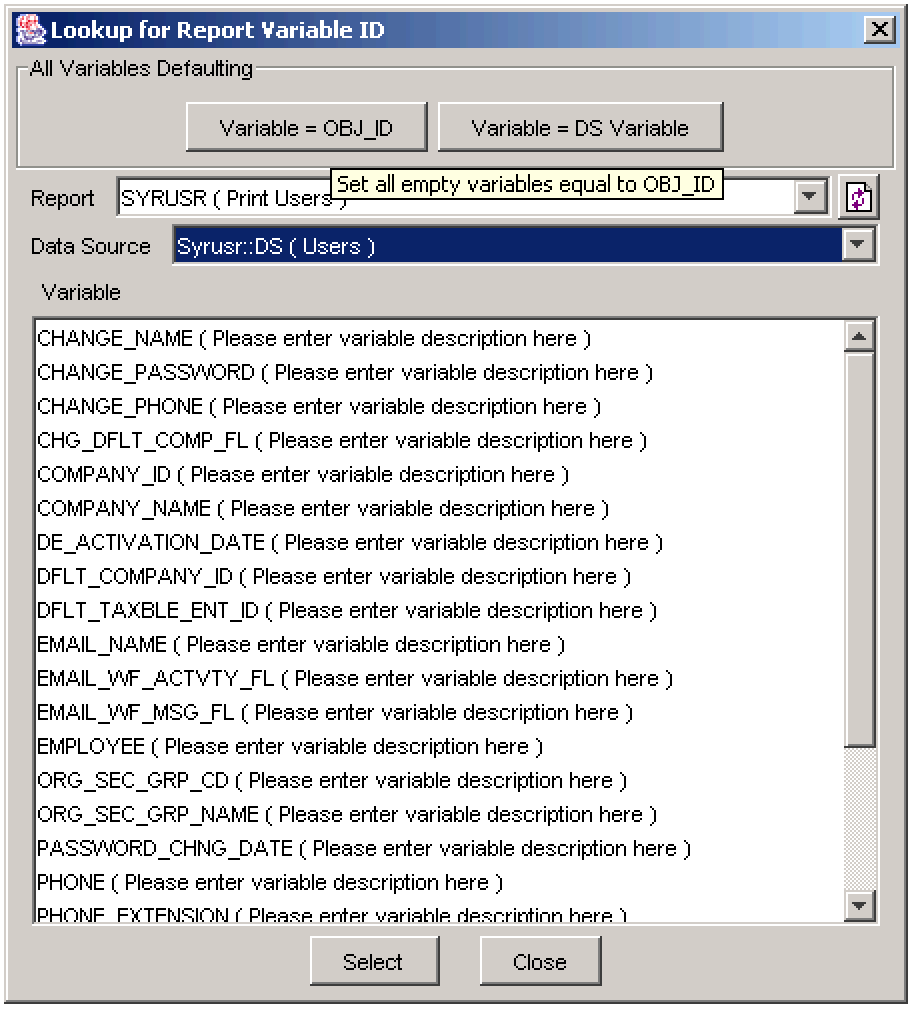
Button Variable =OBJ ID

Click this button to set all empty (unassigned) objects as Report variables with Report variable name = the object ID of itself. This will set all columns in this RS as report variables and the report variable name the same as object name.

Button Variable = DS Variable

Click this button to set all empty (unassigned) objects as Report variables with the Report variable name = the DS variable ID. The Report and Data Source fields must be completed for this action.

This action is slightly different than the one above. This one only pulls the report variables identified in the report definition. Therefore, if the RS has 10 variables and only 6 was listed in the Report DS, only 6 objects will be set here.



Report: SYRUSR (Print Users)

Data Source: Syrusr::DS (Users)

Variable

```
CHANGE_NAME ( Please enter variable description here )
CHANGE_PASSWORD ( Please enter variable description here )
CHANGE_PHONE ( Please enter variable description here )
CHG_DFLT_COMP_FL ( Please enter variable description here )
COMPANY_ID ( Please enter variable description here )
COMPANY_NAME ( Please enter variable description here )
DE_ACTIVATION_DATE ( Please enter variable description here )
DFLT_COMPANY_ID ( Please enter variable description here )
DFLT_TAXBLE_ENT_ID ( Please enter variable description here )
EMAIL_NAME ( Please enter variable description here )
EMAIL_WF_ACTVTY_FL ( Please enter variable description here )
EMAIL_WF_MSG_FL ( Please enter variable description here )
EMPLOYEE ( Please enter variable description here )
ORG_SEC_GRP_CD ( Please enter variable description here )
ORG_SEC_GRP_NAME ( Please enter variable description here )
PASSWORD_CHNG_DATE ( Please enter variable description here )
PHONE ( Please enter variable description here )
PHONE_EXTENSION ( Please enter variable description here )
```

For each report variable, it's sufficient to assign the report variable name (report ID not needed) since the report has been assigned to result set (in the Report section). When the report is run, the system looks for the Result set assigned to the report and then look for the variable name in here.

If the result set is assigned to more than one report, it is important to name the variable in the reports the same so it can be assigned to one field in the result set.

Report Parameter

For Report Parameter result set, if this field needs to be accessed in the report (for printing or used in report formula), select this check box (for example, Sort code, range selected).

New

To add a new Object to the Description tab, click **New**. An Object Id will be assigned automatically by the system. Any time before save, you can change the ID name.

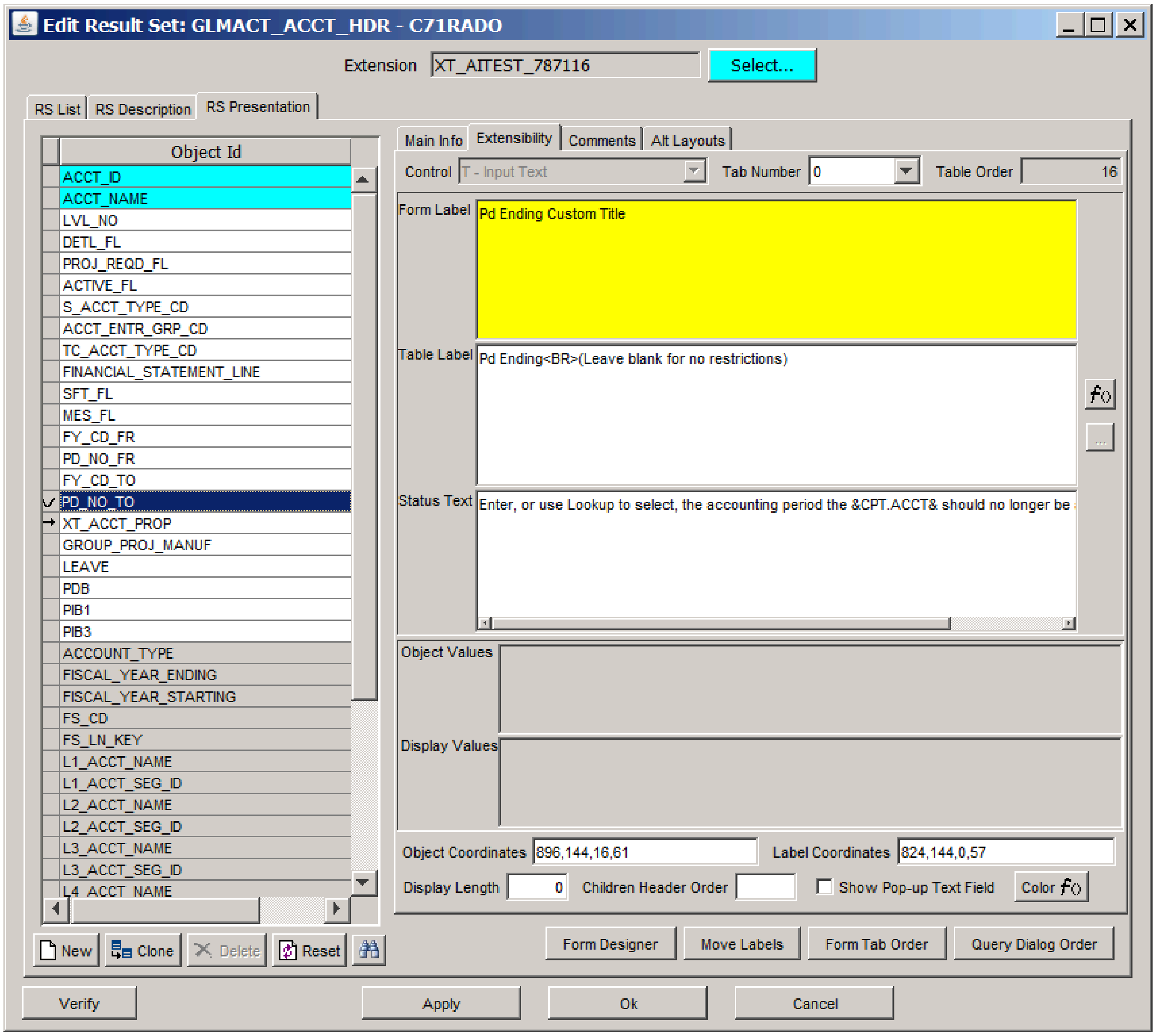
When a new row is added to the Description tab, the system automatically adds the same Object on the Presentation tab. If you change the ID name, the ID name in the Presentation tab will also be changed as soon as you tab out of the ID name in the Description tab.

Delete

- To delete an object from the Description tab, highlight the row and click on **Delete**.
- You can only delete an item that is not part of the SQL statement (that is, the From DB column is not checked).
- You cannot delete an item that is part of the SQL statement (that is, the From DB column is checked). To delete these items, delete them from the SQL Statement and click on Refresh OBJ IDs.

RS Presentation Tab

RS Presentation tab contains metadata describing presentation (HTML) properties (such as control type, titles, status text, coordinates) of each Object / Field in the Result Set. On the left side, RS Presentation tab contains a list of all data Objects / Fields (visible and hidden) in the RS Description tab plus any presentation only elements like labels, tabs, group boxes and lines. In this tab, objects are sorted by data entry Tab Order. In other words, they are presented in the same order the end users see when tabbing through fields in this result set inside Costpoint. Visible Objects are followed by presentation only elements (such as group foxes, labels, etc.) that do not have Tab Order and finally all hidden/server only objects that are grayed out in the table.


Edit Result Set: GLMACT_ACCT_HDR - C71RADO

Extension XT_AITEST_787116 Select...

RS List | RS Description | RS Presentation

Main Info | Extensibility | Comments | Alt Layouts

Control: T - Input Text; Tab Number: 0; Table Order: 16

Form Label: Pd Ending Custom Title

Table Label: Pd Ending<BR>(Leave blank for no restrictions)

Status Text: Enter, or use Lookup to select, the accounting period the &CPT.ACCT& should no longer be

Object Values:

Display Values:

Object Coordinates: 896,144,16,61; Label Coordinates: 824,144,0,57

Display Length: 0; Children Header Order: ; Show Pop-up Text Field

Object Id
ACCT_ID
ACCT_NAME
LVL_NO
DETL_FL
PROJ_REQD_FL
ACTIVE_FL
S_ACCT_TYPE_CD
ACCT_ENTR_GRP_CD
TC_ACCT_TYPE_CD
FINANCIAL_STATEMENT_LINE
SFT_FL
MES_FL
FY_CD_FR
PD_NO_FR
FY_CD_TO
PD_NO_TO
XT_ACCT_PROP
GROUP_PROJ_MANUF
LEAVE
PDB
PIB1
PIB3
ACCOUNT_TYPE
FISCAL_YEAR_ENDING
FISCAL_YEAR_STARTING
FS_CD
FS_LN_KEY
L1_ACCT_NAME
L1_ACCT_SEG_ID
L2_ACCT_NAME
L2_ACCT_SEG_ID
L3_ACCT_NAME
L3_ACCT_SEG_ID
L4_ACCT_NAME

To edit a row, highlight the entry on the left and edit the attributes on the right. There are three major tabs:

- The **Main Info** tab has all the options that Costpoint developer specified for a given Object (field). They are all greyed out. This **Main Info** tab is, obviously, not available for the Extensibility Only objects.
- The **Extensibility** tab is almost an exact copy of the first one, but here you can change some properties of standard objects or fully control all the properties available for Extensibility Only objects.
- The **Comments** tab displays standard comments and allows you to enter Extensibility comments for each Object. You can enter notes that are useful for remembering the purpose or some special behavior of the objects.
- The **Alt Layouts** tab displays both regular and custom information about Alternative layouts for a given Result Set if such information is entered.

Tip: For more on positioning fields on the screen in Form View please, see Appendix D - Form Designer.

Object ID List

- The table on the left represents all the Objects in the RS, both data and presentation only.
- The Object Id column contains the Id assigned to the Object.
- The Data Element column (scroll right to see) is a read only column. If selected, it means this object comes from (correspond to) the Description Tab. Else, it is an additional column added for presentation purpose only (for example, group box, label, button.).
- Object for PRESENTATION is always listed in the following order.
  - First, all visible data objects in RS in the table order.
  - Second, all non-data (presentation-only) elements (for example, tabs, lines, images)
  - Third, all non-visible(hidden) objects from Description tab.

New

To add a presentation only element, click **New**. A default Object ID will be assigned automatically by the system. Any time before save, you can change the ID.

Object ID

While this is not critical, try to give the Object ID some meaningful ID. For example, if it's for a status group box, name it something like XT_STATUS_GRP_BOX.

Other suggested suffixes:

- XT_ENT_BOX
- XT_NOTES_LABEL

Naming Convention

See naming convention for ID under the RS Description section.

Delete

- To delete a Presentation only object, highlight the row and click **Delete**.
- You can only delete item that is presentation in this tab (that is, the Data Element column is not selected).
- To delete an item that is a data element, you must delete it from the Description Tab or from SQL select statement.

- You cannot delete an object created by Deltek. Only objects that were added by you for extensibility.

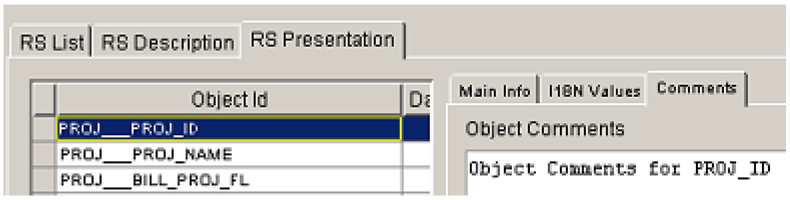
Main Info Tab

Contains all information related to the control and the first language supported (English).

Comments Tab

Enter any comment regarding this field. Does not affect field functionality.

Comments must be entered if the field is hidden at the client and is set to Required.



Object Comments for PROJ_ID

Control Type

If it is an input field created from the SQL statement or created on the RS Description tab, select one of the following codes:

- T: Input Text
- X: Check Box
- R: Radio Button (See Object Values). A Radio Button set must be set as required field in the DESC tab.
- C: Combo Box.

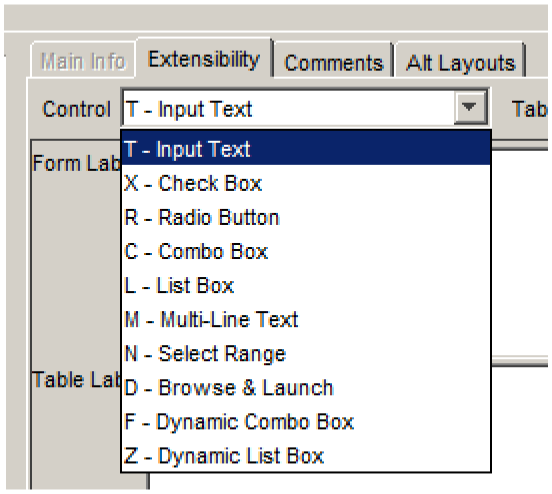
> Note: A permanent non-editable combo box (versus non-editable via formula) will be presented like a text box. This is designed so that you can use combo box to display translated text from a database value (for example, display S as System, U as User, etc.). Do not translate code to text in java class or in SQL statement since this does not work when the system is internationalized.

If the combo box is set as required field, then a value must be selected from the combo box at run time. The option list will start with the line that says "SELECT" followed by all the valid choices.

If the combo box is not a required field, the option list will start with the line that says "-None-" which is a valid option.

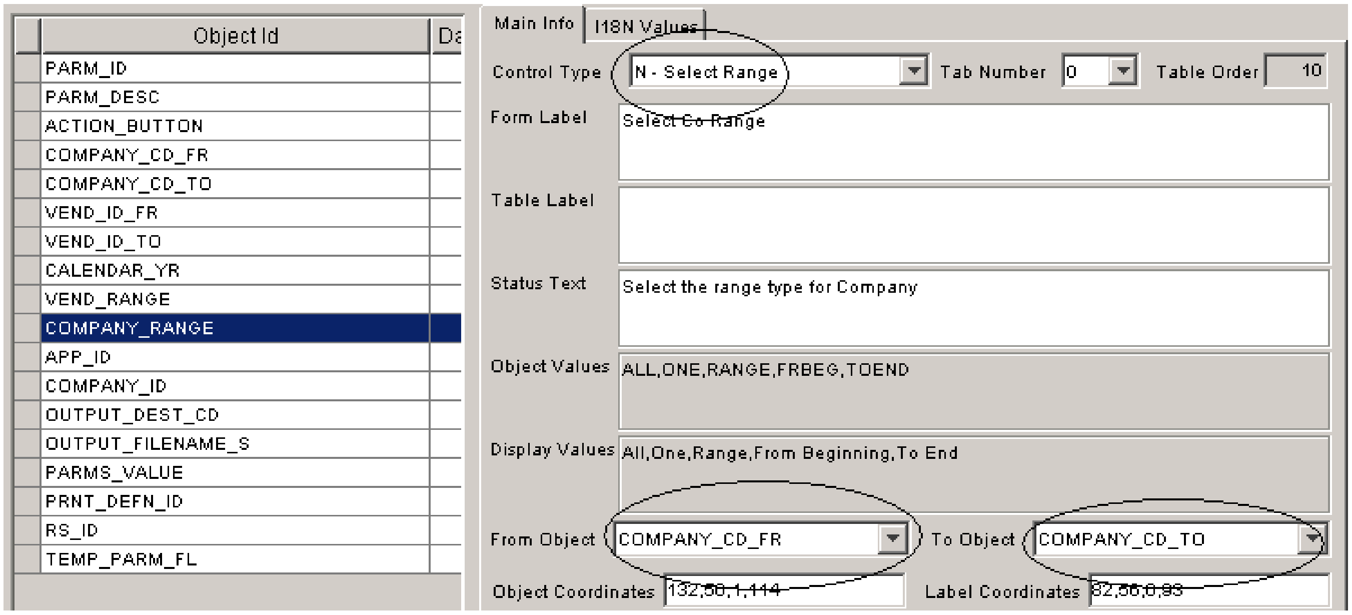
- L: List Box
- M: Multi Line Text
- N: Select Range (See Select Range Control below)

- D: Browse & Launch
- F: Dynamic Combo Box: This control type is similar to a regular Combo-Box, but the list of values is dynamic. Possible values are coming from an assigned lookup column, so it is a requirement that for such a Control Type, a Lookup should be assigned to a selected Object. Please be careful with this control type because if a list of values is too large (dozens or even hundreds of records), the UI can become unresponsive or even crash.
- Z: Dynamic List Box: This control type is similar to a Dynamic Combo Box type in terms of the list of values, but it is represented by a list (not combo box) and allows for a selection of multiple values.



Select Range Control

If the control is a Select Range Control, specify the FROM Object ID and the TO Object ID below.



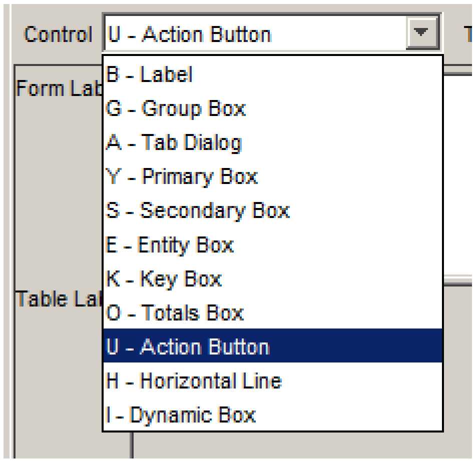
Object Id
PARM_ID
PARM_DESC
ACTION_BUTTON
COMPANY_CD_FR
COMPANY_CD_TO
VEND_ID_FR
VEND_ID_TO
CALENDAR_YR
VEND_RANGE
COMPANY_RANGE
APP_ID
COMPANY_ID
OUTPUT_DEST_CD
OUTPUT_FILENAME_S
PARMS_VALUE
PRNT_DEFN_ID
RS_ID
TEMP_PARM_FL

Control Type: N - Select Range
Tab Number: 0
Table Order: 10
Form Label: Select Co Range
Table Label:
Status Text: Select the range type for Company
Object Values: ALL,ONE,RANGE,FRBEG,TOEND
Display Values: All,One,Range,From Beginning,To End
From Object: COMPANY_CD_FR
To Object: COMPANY_CD_TO
Object Coordinates: 132,50,1,144
Label Coordinates: 82,50,0,93

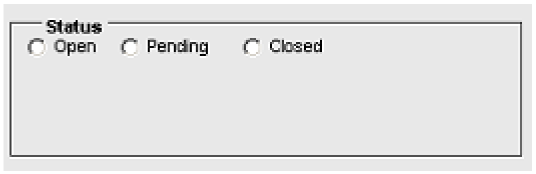
- At run time, the system automatically controls the enable/disable of the From/To objects depending on the range code entered (All, One, Range, etc.).
- Validation is also done at the server to ensure data is present and consistent with the range code selected (that is, To Value is greater than From Value). This is done for String and Numeric objects only, not for Date objects.
- If there are multiple objects for FROM and TO fields (for example, Part From/Revision From and Part To, Revision To), enter additional objects separated by comma. For example:
  From Object = PART_FROM,RVSN_FROM
  To Object = PART_TO,RVSN_TO
- There is no system validation for multiple object range controls for the values entered. For example, the system will not verify that the From Values is less than the To Values since this depends on the table and the objects involved. Developers must perform this validation in the application java code. For example:
  For FY/PD/SUB PD composite values, call RangeControl's method valFyPdSubRange to validate the composite value (that is, From must not be greater than To)

Non-Input Control Type

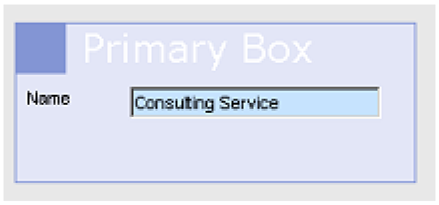
If this is an extra presentation item created on the Presentation tab, select one of the following codes:



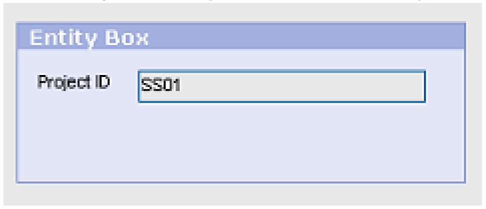
- **B - Label**: Additional label not associated with a control
- **G - Group Box**: A rectangular area with label included. Designer is responsible for placing the controls within the box.



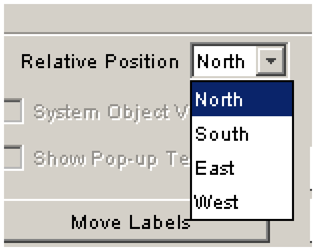
- **A - Tab Dialog**: See Tab Number below on how to add a tab control
- **Y - Primary Box**: A group box to designate an area as Primary information block



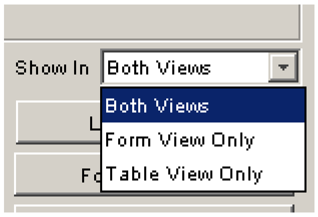
- S - Secondary Box: A group box to designate an area as Secondary information
- E - Entity Box: A group box to designate an area as Entity information block



- K - Key Box: A group box to designate a primary key information block
- O - Totals Box: A group box to designate a total block
- U - Action Button: If control is an Action Button, enter the label for the buttons. The label for Form view is used both in Form view and Table view.
  Buttons can be positioned anywhere for Form view in the form designer. In table view, however, you can only indicate which side of the table they will be placed on.
  Relative Position

  Show In
  You can select to show this button in Form View, Table View or both.

> Note: Action buttons cannot be programmatically enabled or disabled. To achieve similar functionality, check inside the action if it can be run or not and if not then display an informational message to a user why it cannot be run.

- H - Horizontal line: Create a horizontal line on the screen.
- I - Dynamic Box: A rectangular area with label included. Similar to Group Box, it can shrink vertically to the lowest visible Controls/Objects when other Objects/Controls that were placed inside this Dynamic Box become not visible.

Tab Number

If the form has tabs, Tab Number indicates which tab this control resides on the form.

To add tab number, you must first add a separate item having a control type of A for tab dialog. Add the label for this tab in the Form Label section.

Once you have added a tab control type, the Tab Number combo box will show the tab value available (starting with 1). If you add a 2nd tab control, the Tab number combo box will show 2 as a valid choice and so on. You can start assigning the number to other Presentation items. Any items that has tab zero assigned will be shown on all tabs.

Up to 8 tabs can be supported. The labels on the tab may be truncated if they are long and there are too many tabs. Make sure you review the tab labels at run time to see if they fit.

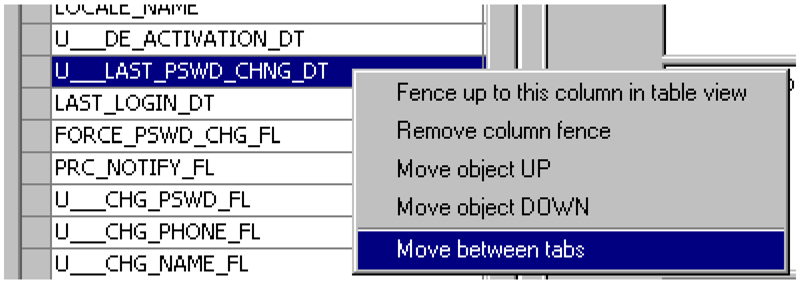
Move multiple objects to a tab

To make it easy to move objects from one tab to another, you can now select multiple objects and move them to a new tab.

To move multiple objects to a tab:

1. Activate the dialog by highlighting any row in the presentation tab.

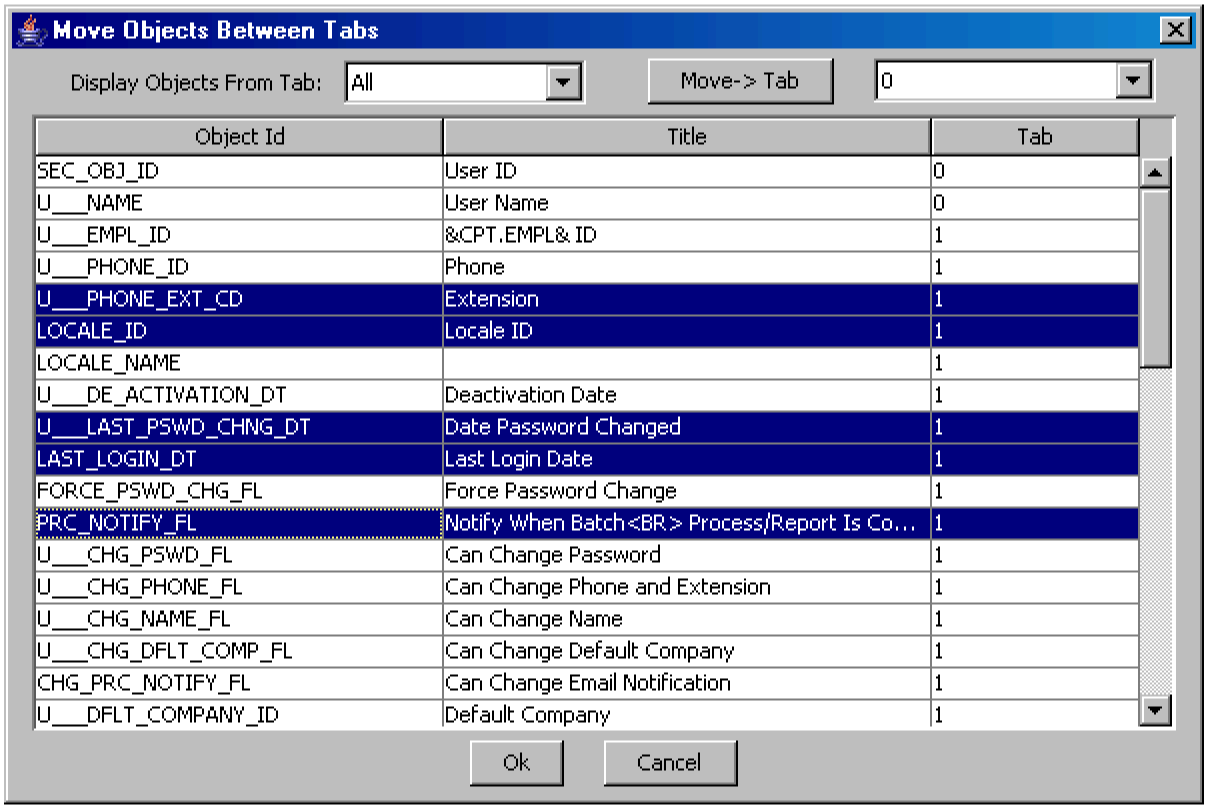
2. Right-click and select the **Move between tabs** option. A dialog will appear.



3. While holding down CTRL (or SHIFT to select a contiguous range), click to select multiple objects.

4. Select the tab number on the top right hand side and then click **Move » Tab**.

Tip: Use the combo box on the top left side to filter objects by their current tab number.



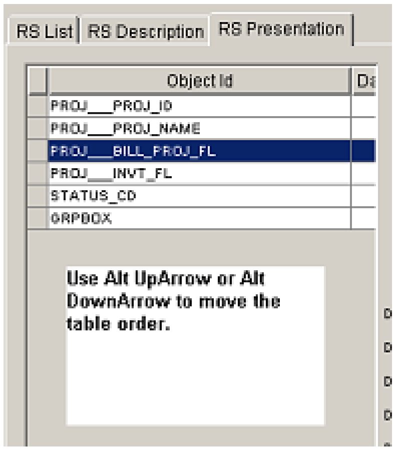
Object Id	Title	Tab
SEC_OBJ_ID	User ID	0
U___NAME	User Name	0
U___EMPL_ID	&CPT.EMPL& ID	1
U___PHONE_ID	Phone	1
U___PHONE_EXT_CD	Extension	1
LOCALE_ID	Locale ID	1
LOCALE_NAME		1
U___DE_ACTIVATION_DT	Deactivation Date	1
U___LAST_PSWD_CHNG_DT	Date Password Changed	1
LAST_LOGIN_DT	Last Login Date	1
FORCE_PSWD_CHG_FL	Force Password Change	1
PRC_NOTIFY_FL	Notify When Batch<BR> Process/Report Is Co...	1
U___CHG_PSWD_FL	Can Change Password	1
U___CHG_PHONE_FL	Can Change Phone and Extension	1
U___CHG_NAME_FL	Can Change Name	1
U___CHG_DFLT_COMP_FL	Can Change Default Company	1
CHG_PRC_NOTIFY_FL	Can Change Email Notification	1
U___DFLT_COMPANY_ID	Default Company	1

Table Order

Table order is the order of columns in table view.

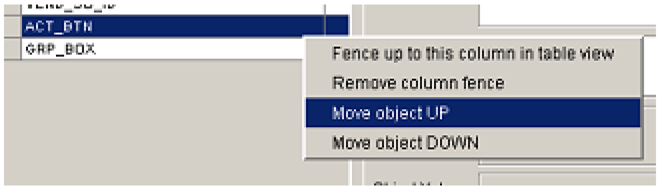
Note: This is also the input field tab order on Form View (unless override using the Form view order -- explained later). Form View tab order can be set via the **Form Tab Order** button at the bottom of the Presentation tab.

If you want to change the table order, highlight the item on the left and use the Alt Up Arrow or Alt Down Arrow to rearrange the tab order.



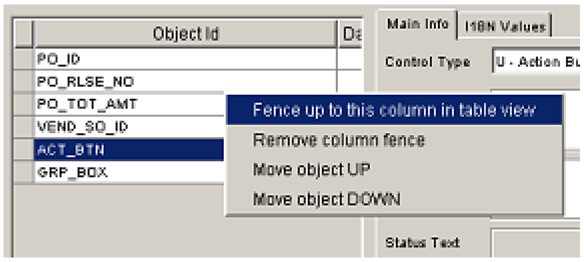
Use Alt UpArrow or Alt DownArrow to move the table order.

Object can be moved up or down by right clicking the object and then clicking **Move object UP** or **Move object DOWN**.



Columns to Fence

Fencing is not available for Extensibility. It can only be controlled by Deltek.



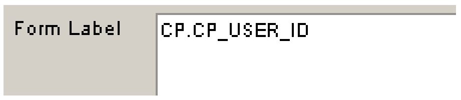
Form Label

Enter the text to be used as label for the input field. Labels are internationalized with English in Label 1.

Refer to UI standard for how label should be worded. Certain standard applies for common labels (that is, Abbrev instead of Abbrv).

Labels can be set programmatically from a list of predefined labels via label formula. (See Label Formula later).

You can also use application constants in the label. The constant must be assigned to your app unless it is one of the automatic constants (See Extensibility Coding Guide).

> Note: If a constant is present in either the form label, table label, or status, the constant must be present in all three. This is because system JavaScript 'eval' them together as a group. Failure to include the constants in the other fields (that is, only have text or blank in other fields) will result in "undefined" error in JavaScript.
>
> 
>
>
> CP.CP_USER_ID

You can use the <br>, <sup>, and <sub> HTML tags in form labels as well as links and link image tags.

For example,

```
<a href="http://www.w3schools.com"><img src="logo_w3s.gif" alt="W3Schools" borde
r="0" width="100" height="100" /></a>
```

Table Label

Enter the text to be used as label for the column in table view (if different than form view).

Labels can be set programmatically via label formula. (See Label Formula later)

Like form label, you can also use application constants in the table label.

> Note: If constant is present in either the form label, table label or status, constant must be present in all three. This is because system JavaScript 'eval' them together as a group. Failure to include the constants in the other fields (that is, only have text or blank in other fields) will result in "undefined" error in JavaScript.

The header is fixed at 2 rows and bottom center aligned. If you want column heading to break, you must explicitly place the tag "<BR> where you want the text to break.

To use this feature, use the standard HTML symbols for line break `<BR\>` in the table heading text of the result set designer.

Example: Customer `<BR\>` Name

Displays:

Customer

Name

Object Values

Valid values for this field in the database

For combo box or radio buttons, check box or any input text field (also see Display Values above.)

For combo boxes and radio buttons:

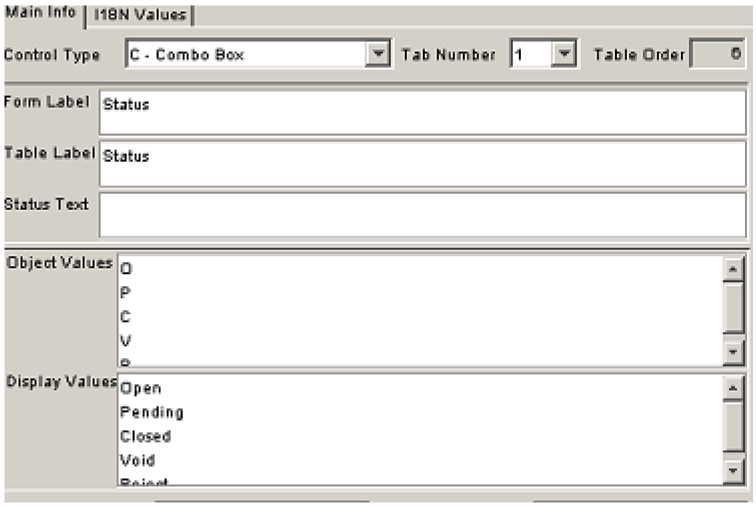
Enter all possible database values associated with this combo box or group of radio buttons in a comma-separated string. This comma-separated string must match the elements in the Display Values field in the exact order.

For example: The PO status can have these possible values:

- O: Open
- P: Pending
- C: Closed
- V: Void
- R: Reject

In this field enter each value on a separate line. In the field for display value, enter each item on a separate line in exactly the same order.

Combo Box example:

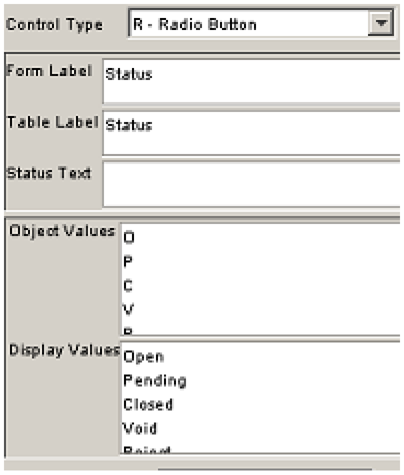


Control Type: C - Combo Box
Tab Number: 1
Table Order: 0
Form Label: Status
Table Label: Status
Status Text:
Object Values:
O
P
C
V
R
Display Values:
Open
Pending
Closed
Void
Reject

Note: Combo box is for known static values. Values from database such as those in S_ tables can be coded as combo box only if the values are not subject to possible future change.

Radio button example:

Radio buttons use the same Display Values and Object Values as combo boxes. In fact, radio buttons are turned into combo boxes in table view (since radio buttons can't be esthetically shown in table view).



Control Type	R - Radio Button
Form Label	Status
Table Label	Status
Status Text	
Object Values	O P C V
Display Values	Open Pending Closed Void

Check boxes

For check boxes, the Object Values field automatically contains Y, N values.

Display Values

Used for combo box or radio buttons only.

Enter the text associated with all the values in Object Values field (see examples in Object Values below).

Beginning in Costpoint version 8.2, there is a new functionality that allows you to hide specific values for combo boxes, list boxes, and radio buttons. This functionality can be achieved by using the reserved value **_HIDE_ME_**. Please note that the value starts and ends with underscores (_).

This reserved value can be used in regular or alternative layouts and will result in the Framework removing the corresponding value from the list of possible selections. Keep this in mind as it will be your responsibility to make sure hidden values do not exist in the database for any record. Otherwise, the end user will get an error stating that the queried value is not valid.

Object Coordinates/Label Coordinates

Use the Form Designer to position the object and label on the HTML page. Coordinates are in pixels and are stored as comma separate values in this format: Left, Top, Height, Width.

Keep in mind that in Auto-Positioning mode, the absolute coordinates will be converted in relative object positions as the system will try to fill all available real estate on the HTML page. The end result may look slightly different from the original page design in absolute coordinates. In addition, check-boxes in Auto-Positioning mode are always displayed with the label to the right of the check-box itself.

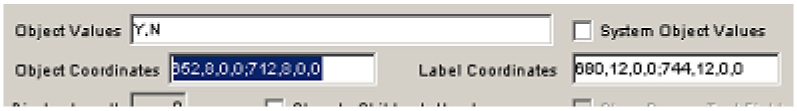
For Radio Buttons:

Object Coordinates and Label Coordinates are for the whole radio group, not for each single radio button. Each radio group (not button) has a full set coordinate of Left, Top, Height, Width. Each set is separated by a semicolon (;).

For example, one radio button group has 2 radio buttons for Yes and No.

- Enter the labels for radio buttons in Display Values as Yes, No
- Enter the values associated with the labels in Object Values as Y, N (no space)
- Enter Object coordinates for the group in sets separated by semicolon. Each set is for each radio button and has 4 coordinates (that is, below: Radio buttons have coordinates as 652,8,0,0;712,8,0,0).
- Enter Label coordinates for the group in sets separated by semicolon. Each set is for each radio button label and has 4 coordinates (that is, below: Radio buttons labels have coordinates as 680,12,0,0;744,12,0,0

Use the screen designer to create these coordinates.



Object Values: Y,N

System Object Values: ☐

Object Coordinates: 652,8,0,0;712,8,0,0

Label Coordinates: 680,12,0,0;744,12,0,0

The Form Designer allows the designer to move the control, hence, affecting the Left and Top coordinate. Most controls will have the Height and Width controlled automatically by the system. The exceptions are the Multi Line Text and List Box controls, which can be resized in the screen designer.

Display Length

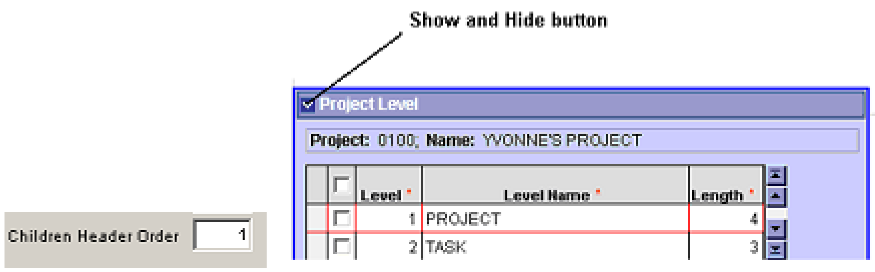
Enter the width of the field (in characters) to be displayed on the screen. The system will default this for you when the control is created.

Generally, for string field, this should be the same as Precision. For number field, leave it as zero and system will calculate based on Precision. For date field, leave it as defaulted by the system.

Children Header Order

If you want to show this field in the child RS, enter a number other than zero. The table label and value will be automatically displayed on all the children RS (and grandchildren, etc.) of this RS.

Multiple fields can be displayed. The order will be the order entered in this field.



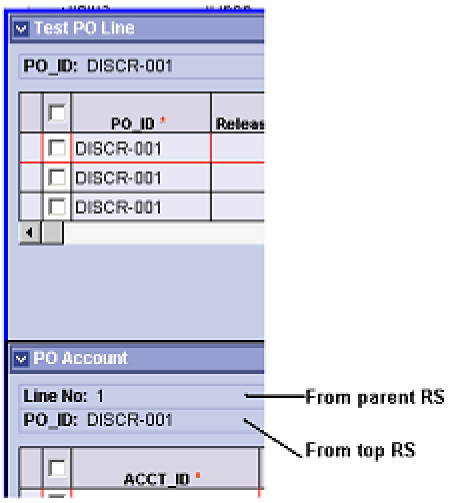
Children Header Order: 1

Project: 0100; Name: YVONNE'S PROJECT

Level *	Level Name *	Length *
1	PROJECT	4
2	TASK	3

At run time, you can click on the top left corner arrow button to hide or show the parent information.

If the tree contains more than 2 levels, the lowest RS will show both the parent and grandparent information.



PO_ID: DISCR-001

PO_ID *	Releas
DISCR-001	
DISCR-001	
DISCR-001	

Line No: 1
PO_ID: DISCR-001

ACCT_ID *

Show Pop-up Text Field

If control type is **Input Text**, select this check box if you want the ability to expand this field at run time in a pop up text dialog for easier editing/review.

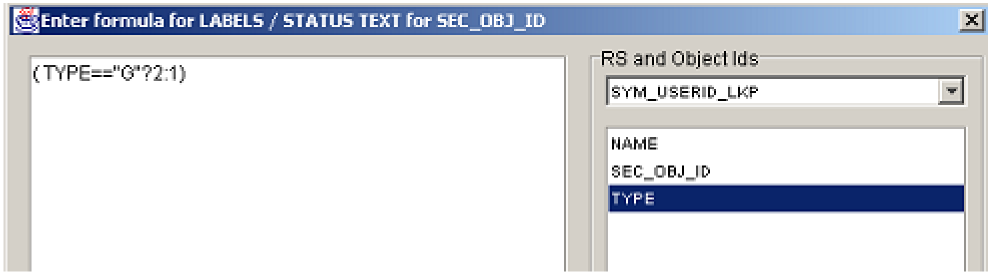
> Note: This flag is disabled for Multi Line Text controls since these controls can be sized in the form designer. However, in table view, the **Pop-Up** button is automatically available for Multi Line Text column.

Label Formula

Click to enter Formula for Labels. There is only one formula that applies to all (Form Labels, Table Labels).

Formula returns a number 1 or 2 or 3 or however many possibilities there are. For each possibility, there must be a corresponding text for labels.

Example:


Enter formula for LABELS / STATUS TEXT for SEC_OBJ_ID

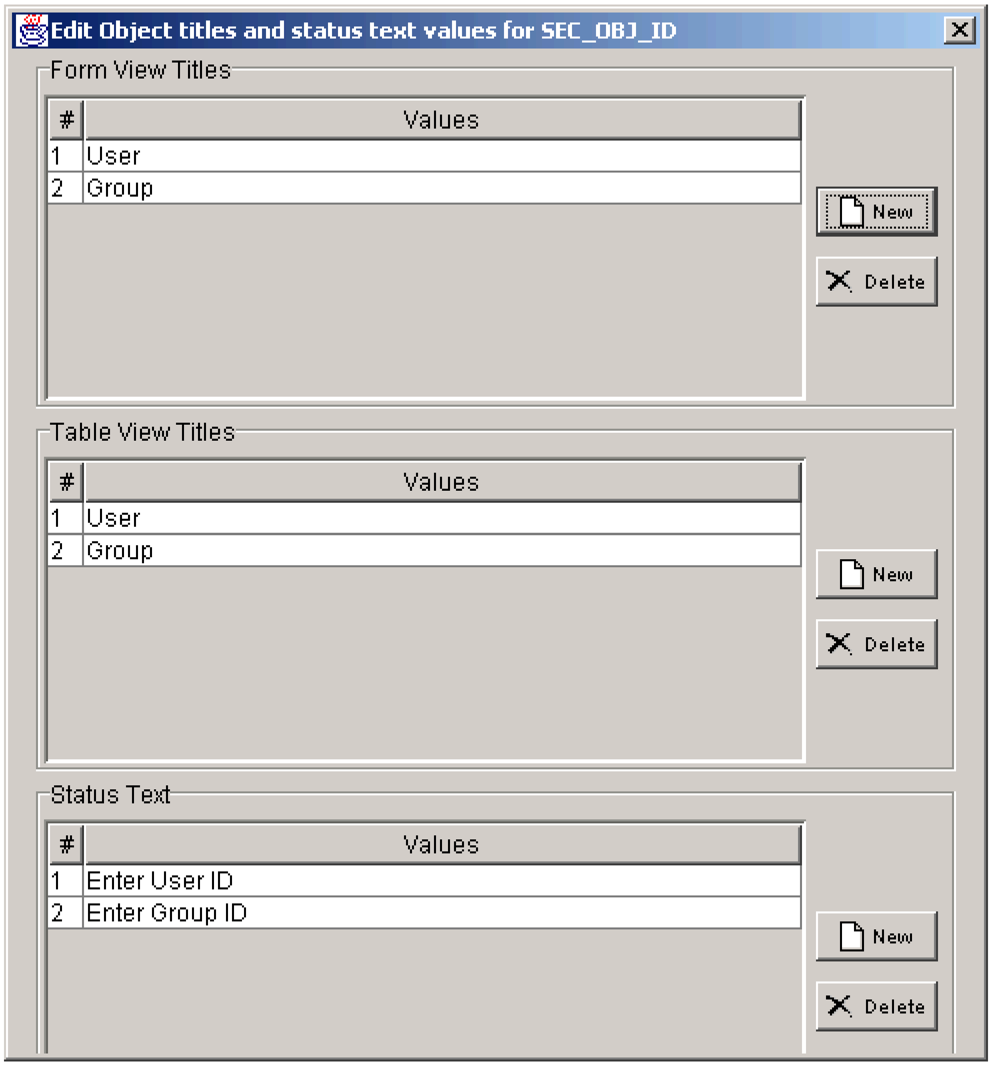
```
(TYPE=="G"?2:1)
```

RS and Object Ids: SYM_USERID_LKP

NAME
SEC_OBJ_ID
TYPE

The formula above returns a 2 if the TYPE is 'G', else it returns a 1.

In the labels, there must be 2 values for each element. Therefore, the label will be set to Group if the TYPE is G. Else, it will be set to 'User'.


Edit Object titles and status text values for SEC_OBJ_ID

Form View Titles

#	Values
1	User
2	Group

Table View Titles

#	Values
1	User
2	Group

Status Text

#	Values
1	Enter User ID
2	Enter Group ID

Formula Evaluation Timing

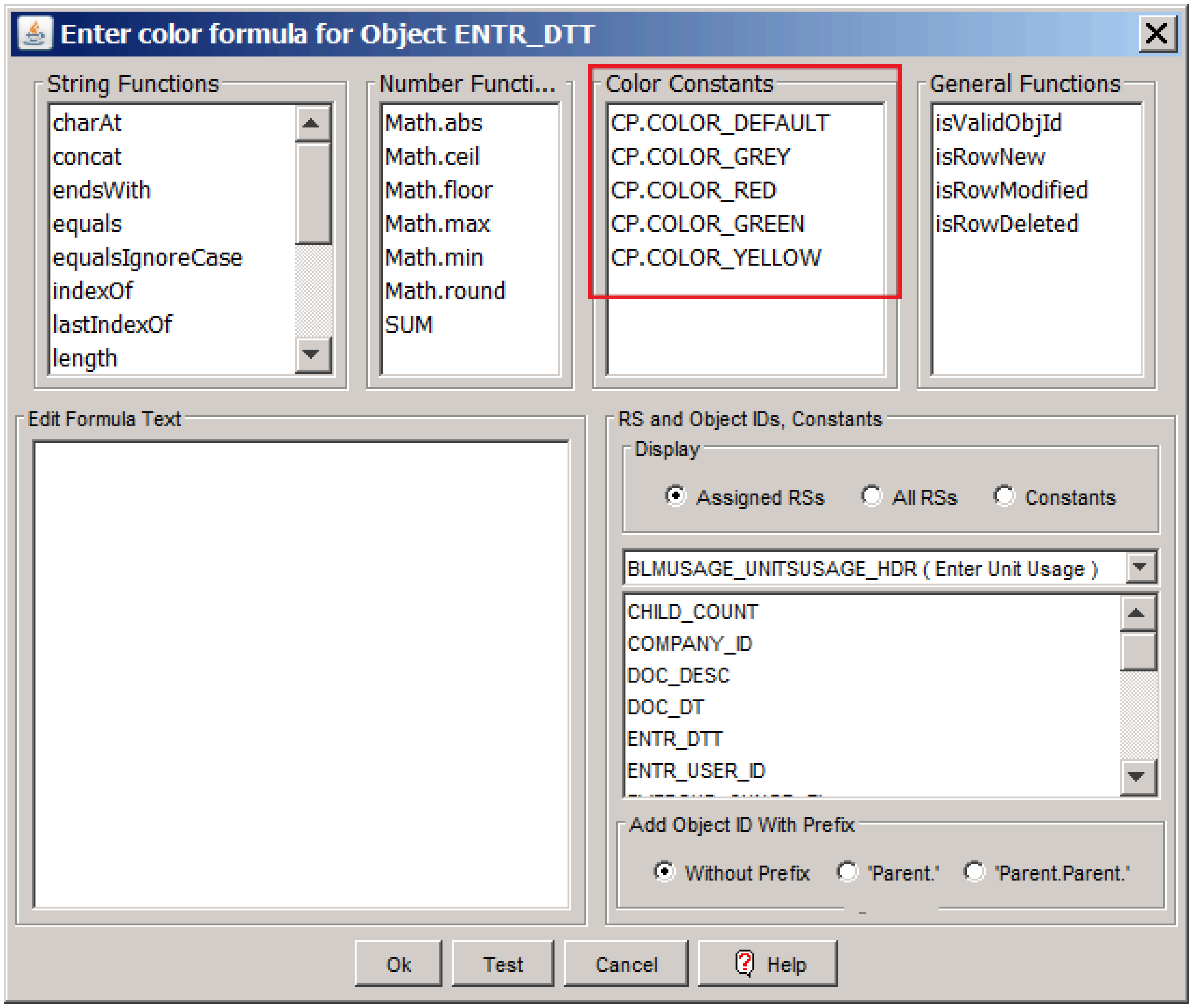
Evaluation of a label formula depends on the validation frequency currently in effect.

If the user has logged in with Field mode, a label formula is evaluated when focus changes on the field(s) referred to in the formula.

If the user has logged in with other modes, a label formula is evaluated *only when focus changes on the record containing the field(s)* referred to in the formula.

Color Formula

Click to enter Formula for Color of the column in Table view. Formula should return one of predefined color constants that you can select in the Formulas Builder dialog box.



Visible Formula

This opens a dialog box where you can enter a Visibility Formula similar to a Visibility Formula for Objects on the RS Desc tab, but it is for Presentation Only elements like labels, group boxes, and so on that are not present on the RS Desc tab.

Enabled Formula

This opens a dialog box where you can enter an Enable Formula similar to an Enable Formula for Objects on the RS Desc tab, but it is for Buttons (as they are not present on the RS Desc tab).

Form Designer

Click **Form Designer** to enter the Form Designer screen (see next section).

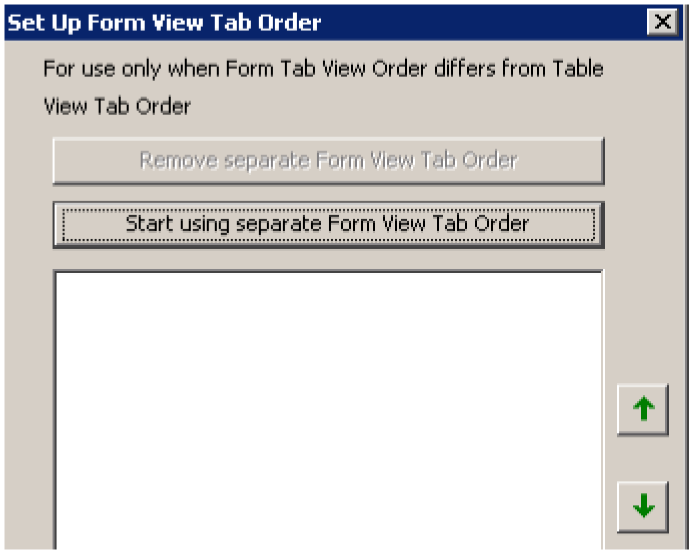
Move Labels

Click this to align all labels top coordinate to 2 pixels below the corresponding text fields (Design standard).

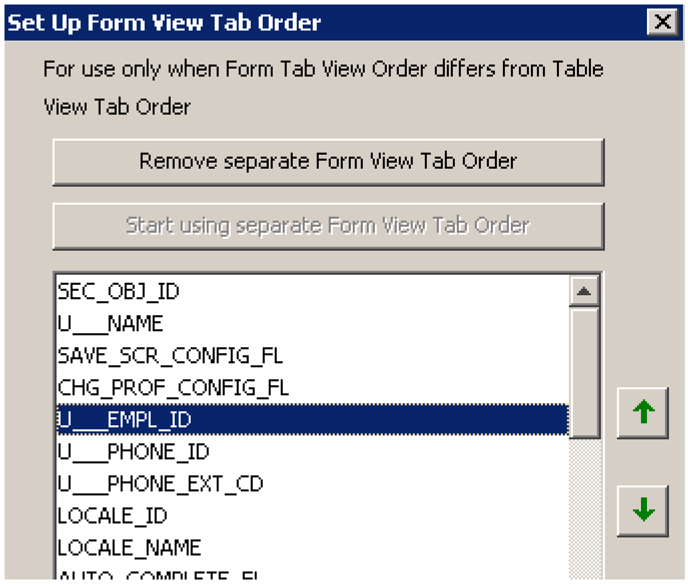
Note: This only moves the labels for those objects where the top pixel position for objects and labels are within 4 pixels.

Form Tab Order

If the Form View Tab Order differs from the Table View Tab Order, click this button to display the Set Up Form View Tab Order dialog box.


Set Up Form View Tab Order
For use only when Form Tab View Order differs from Table View Tab Order
Remove separate Form View Tab Order
Start using separate Form View Tab Order

If the order is the same, this dialog will be blank. Click **Start using separate Form View Tab Order**. As soon as you click this, all controls will come in Table order and you will be able to alter the order for Form View.


Set Up Form View Tab Order
For use only when Form Tab View Order differs from Table View Tab Order
Remove separate Form View Tab Order
Start using separate Form View Tab Order

SEC_OBJ_ID
U___NAME
SAVE_SCR_CONFIG_FL
CHG_PROF_CONFIG_FL
U___EMPL_ID
U___PHONE_ID
U___PHONE_EXT_CD
LOCALE_ID
LOCALE_NAME

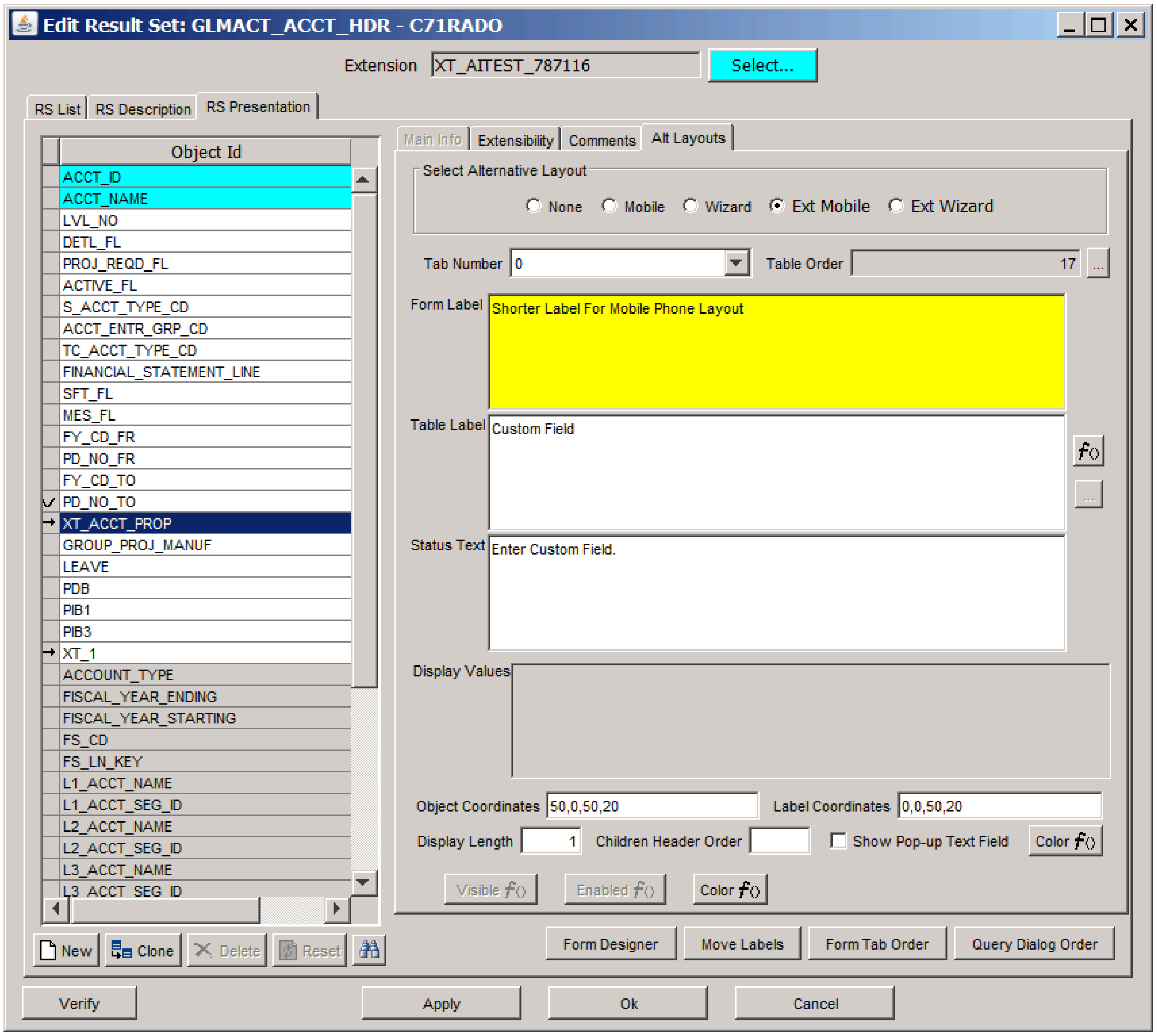
To change the Form view tab order, highlight a row and use **Alt+Up Arrow** or **Alt+Down Arrow** to move them.

To reset order to table order, click **Remove separate Form View Tab Order**.

Alternative Layouts Tab

The Alt Layouts tab displays both regular and custom information about Alternative layouts for a given Result Set if such information is entered. Alternative layouts are used to display Result Set differently if certain criteria are met. For example, you can display a different label for a given Object, or display Objects in a different order or be placed differently on the HTML page.

Currently the only alternative layout that is supported by the Framework is Mobile Layout. So, if the Result Set is being opened from the browser that is on the Mobile phone and Alternative Mobile layout data is entered, this **Mobile** Layout data will be used to create the HTML presentation to be displayed on the mobile phone. Alternative Layout **Wizard** is reserved for future use.


Edit Result Set: GLMACT_ACCT_HDR - C71RADO

Extension: XT_AITEST_787116 — Select...

RS List | RS Description | RS Presentation

Main Info | Extensibility | Comments | Alt Layouts

Object Id
ACCT_ID
ACCT_NAME
LVL_NO
DETL_FL
PROJ_REQD_FL
ACTIVE_FL
S_ACCT_TYPE_CD
ACCT_ENTR_GRP_CD
TC_ACCT_TYPE_CD
FINANCIAL_STATEMENT_LINE
SFT_FL
MES_FL
FY_CD_FR
PD_NO_FR
FY_CD_TO
✓ PD_NO_TO
→ XT_ACCT_PROP
GROUP_PROJ_MANUF
LEAVE
PDB
PIB1
PIB3
→ XT_1
ACCOUNT_TYPE
FISCAL_YEAR_ENDING
FISCAL_YEAR_STARTING
FS_CD
FS_LN_KEY
L1_ACCT_NAME
L1_ACCT_SEG_ID
L2_ACCT_NAME
L2_ACCT_SEG_ID
L3_ACCT_NAME
L3_ACCT_SEG_ID

Select Alternative Layout: None Mobile Wizard [✓ Ext Mobile] Ext Wizard

Tab Number: 0 Table Order: 17

Form Label: Shorter Label For Mobile Phone Layout

Table Label: Custom Field

Status Text: Enter Custom Field.

Display Values:

Object Coordinates: 50,0,50,20 Label Coordinates: 0,0,50,20

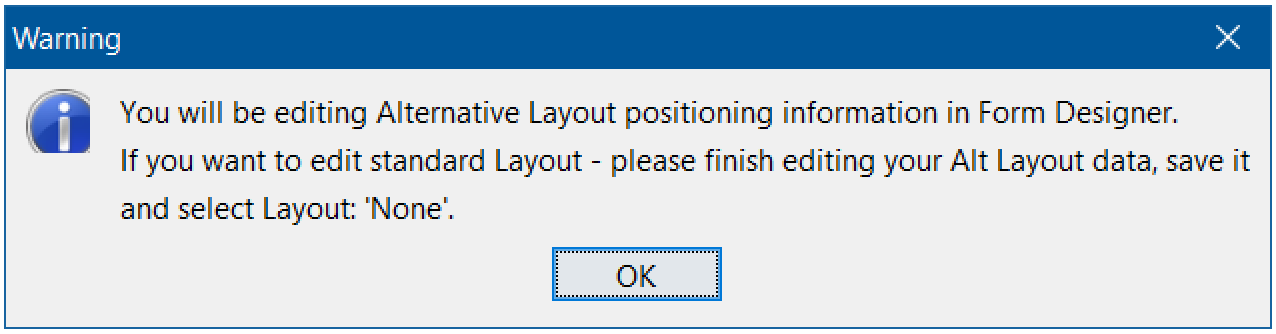
Display Length: 1 Children Header Order: Show Pop-up Text Field

At the top of this Alt Layouts tab, you need to select the Layout to review/edit.

- If you select **None**, this tab is empty.
- If you select **Mobile** or **Wizard** for the standard Result Sets, you can review standard alternative metadata if such metadata exists, but all fields on this tab are disabled.

To enter/customize Alternative Layout metadata for the selected RS, select **Ext Mobile** or **Ext Wizard**. After you select the layout, you will be able to select desired objects from the table on the left side of the screen and enter new metadata. The meaning of the fields directly corresponds to the fields on the Extensibility tab.

You can also use Form Designer to position fields in the Form Designer using drag and drop functionality in the opened browser. The tool will produce an informational message to let you know you are editing the field positioning information for the Alternative Layout:


Warning

You will be editing Alternative Layout positioning information in Form Designer.
If you want to edit standard Layout - please finish editing your Alt Layout data, save it and select Layout: 'None'.

This section describes how you can add brand new Result Sets (subtasks) and assign them to applications in Costpoint. Unlike extending a standard RS, new RS allows you complete control over the logic executed when users open the RS in the application.

Follow these steps when adding a new RS:

1. Click **Project » Unit » Add new RS** and follow the process for creating a new RS.

2. Link (assign) the new RS as a subtask, register it as a new Lookup (in other RS), or register it as a new Report Data Source RS.

Keep in mind that a new Extensibility RS (like any other New Extensibility Object) is available/accessible to and from other Extensibility Objects only in the same Ext Unit.

This new RS will be available when the extensibility package is deployed to the end system.

Create a New RS

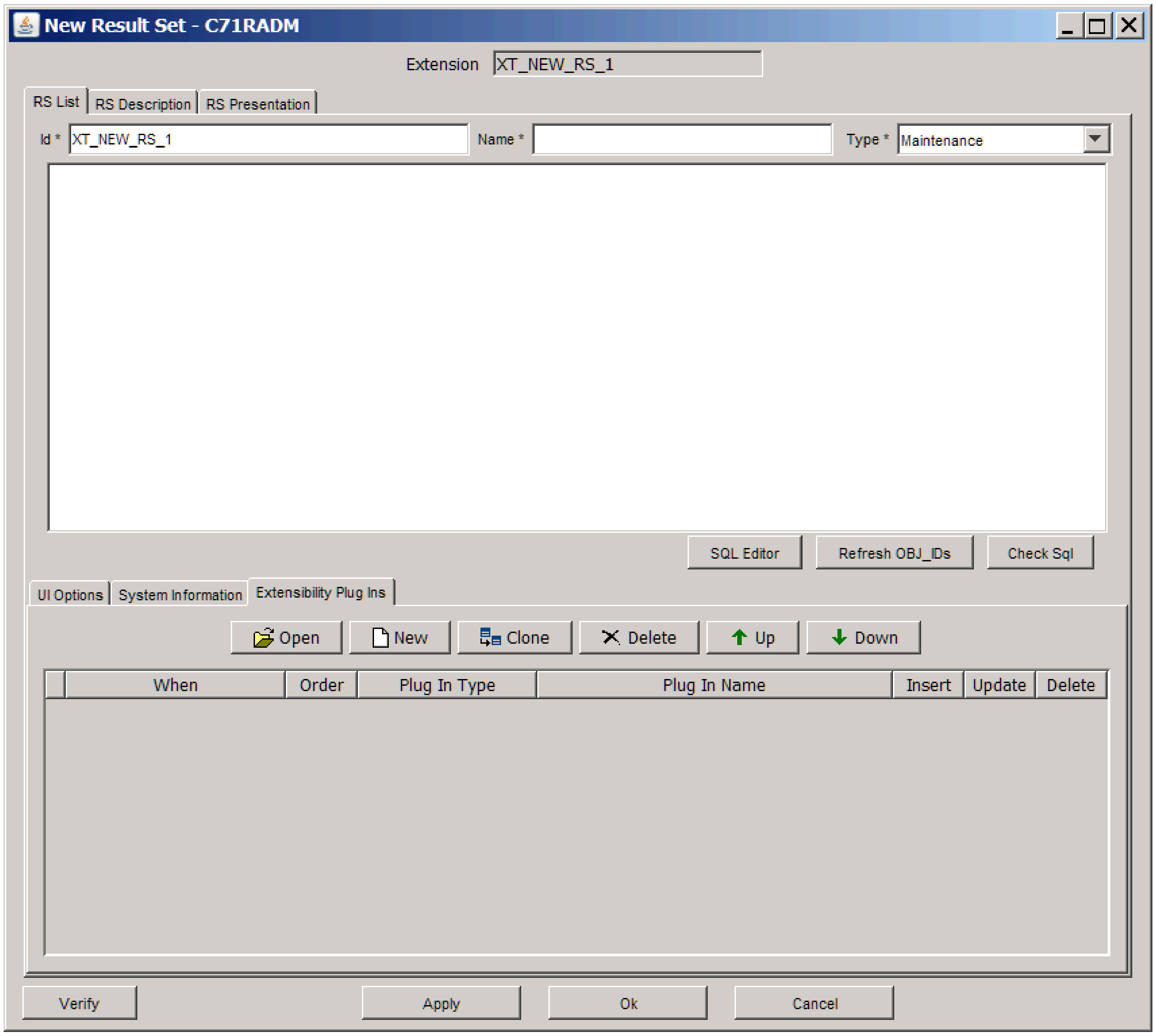
To create a new RS:

1. On the **Extensions** menu, click **Object » New » RS**.

   Alternately, click the **New** icon, and click **RS**.

2. On the **Select Extension Project for new RS** dialog box, select the project, and click **Next**.

3. On the **Select Extension Unit for new RS** dialog box, select the unit, and click **Next**.



Extension: XT_NEW_RS_1

Id *: XT_NEW_RS_1
Name *:
Type *: Maintenance

	When	Order	Plug In Type	Plug In Name	Insert	Update	Delete

Basic Steps When Creating a New Result Set

1. Enter the RS ID and the name of the result set.

2. Select the Type of RS

3. Enter the Data Source name if your result set is SQL-based, and SELECT is not being executed against the standard Costpoint DATA(DELTEK) database. If table(s) are in the DATA database, Data Source can be left empty.

4. Add the SQL select statement, and click **Check** to check the SQL statement against the database.

5. Deltek highly recommends adding the ORDER BY clause to your SELECT to have logical and/or physical Primary Keys in that ORDER BY so end users see the results consistently on the screen when data is being populated from the database.

6. Set the title of the result set.

7. Set the default view code (Form or Table format).

8. Refresh the OBJ_IDs to create rows in the RS Description and RS Presentation tabs. For each column selected in the SQL statement, this step will create a corresponding entry in the RS Description and RS Presentation tabs.

9. Review and edit the entries in RS Description tab.

   Attention: Refer to the Editing Result Set section for detailed instructions on each field.

   You can add additional RS Objects that are not part of the SELECT statement by clicking the **New** button on the RS Description Tab and entering their properties.

10. Review and edit entries in RS Presentation tab. Click **Form Designer** to design the form view's RS layout.

    You can add additional presentation-only RS Objects (like Group Boxes, Tabs, Lines, and so on) by clicking the **New** button on the RS Presentation Tab and entering their properties.

    You can also add RS plug-ins if required on any of the RS plugin events.

    Attention: Refer to Appendix A for more information about Plug-ins.

11. Click **OK** to save the Result Set or click **Apply** to save and continue.

Result Set Types

The system supports the following result set types:

- **Maintenance**: This is the most commonly used result set type and is used to allow end users to query, edit, and save data.

- **Lookup**: All lookup result sets must follow the standard of having all fields visible and must be available for query. The View mode is Table View only.
- **Reporting Data**: This result set type is used for data printed on a report. This result set does not generally require a user interface.
- **Reporting Parameters**: The table is always W_FUNC_PARM_CATLG. The tool will automatically generate the SELECT SQL statement from this table. The default attributes for the columns (RS_DESC and RS_PRESENTATION) are also generated.
- **Processing Parameters**: The table is always W_FUNC_PARM_CATLG. The tool will automatically generate the SELECT SQL statement from this table. The default attributes for the columns (RS_DESC and RS_PRESENTATION) are also generated.
- **Filter**: This is always one row and is editable. It is used in submitting query criteria for child result sets (that is, those acts as header in an inquiry application). The **Execute** button will act as query for the child result sets after the user has entered the query criteria in the Filter result set.

  > **Note:** This RS Type does not allow entering a SELECT statement.

  For the Filter result set, all objects will automatically be set to **Persist if Disabled**.
- **Single Row Parameters**: This RS Type does not allow entering a SELECT statement.
- **Single Row** (One row for each row set, i.e. for each parent row): Automatic UI features will apply to make sure there is only one row in the rowset for this result set.
  It's a requirement that Single Row Result Sets follow the following rules:
  - Must be open with parent (if it has one)
  - Must be auto-load
  - May not have any columns to be included in query so that sub-query is not possible.
  - The developer must add a server validation method to ensure that only one row exists since the UI can only enforce this rule for rows available at the client. (Rows scrolled off the screen, rows not seen due to sub-query, or so on cannot be accounted for by the system at the UI level).
- **NCR** (Non Contiguous Range): For reporting or processing parameter screen where user can enter non-contiguous range for selection criteria. This is basically a table view result set with predefined columns such as Range Code, From Value and To Value.
- **Dash Part**: This RS type should be used when developing simple table listing dashparts. Such table listings are done based on standard RS object functionality. Keep in mind that Dash Part RS should have the RS Open Java plug-in defined that implements the **com.deltek.enterprise.system.applicationinterface.AfterCreateRSInterface** interface, creates the **com.deltek.enterprise.system.applicationinterface.ExtraActionConfig** instance, and invokes appropriate methods on that instance to configure the dashpart functionality that links columns to other apps.

  > **Attention:** For more information, refer to the *Deltek Costpoint 8.2 Extensibility Designer Report Guide.*

> **Attention:** For more details on setting new RS properties, refer to the Extend a Result Set section.

If you want to use Standard Save for new Custom Result Set, your table should have these three following columns that the Framework needs to support concurrency:

```
MODIFIED_BY - varchar(20) NOT NULL,
TIME_STAMP - datetime NOT NULL,
ROWVERSION - int NULL
```

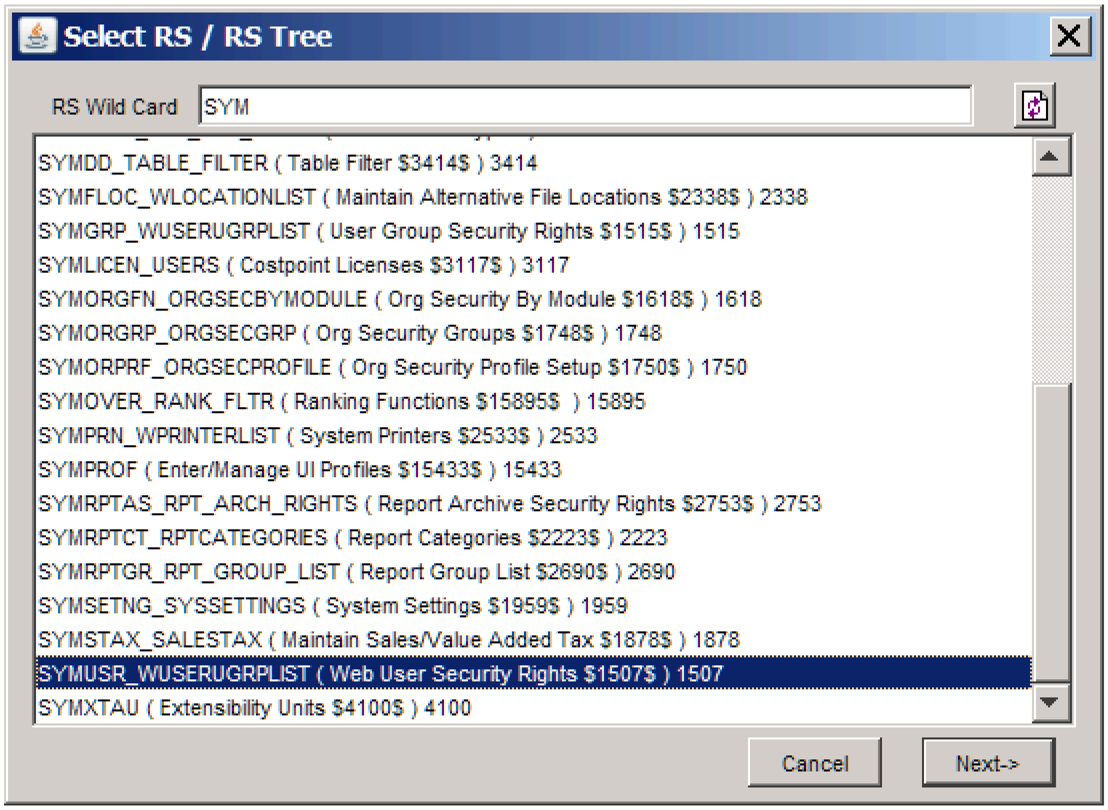
All Deltek-provided tables already have those 3 columns.

Add New RS as a Subtask to RS Tree

After you create a new Extensibility RS, it can be used as a new subtask. This section describes how you can add an already created Result Set (New Custom or Existing) and assign it to applications in Costpoint.

To add new RS as a subtask to the RS Tree:

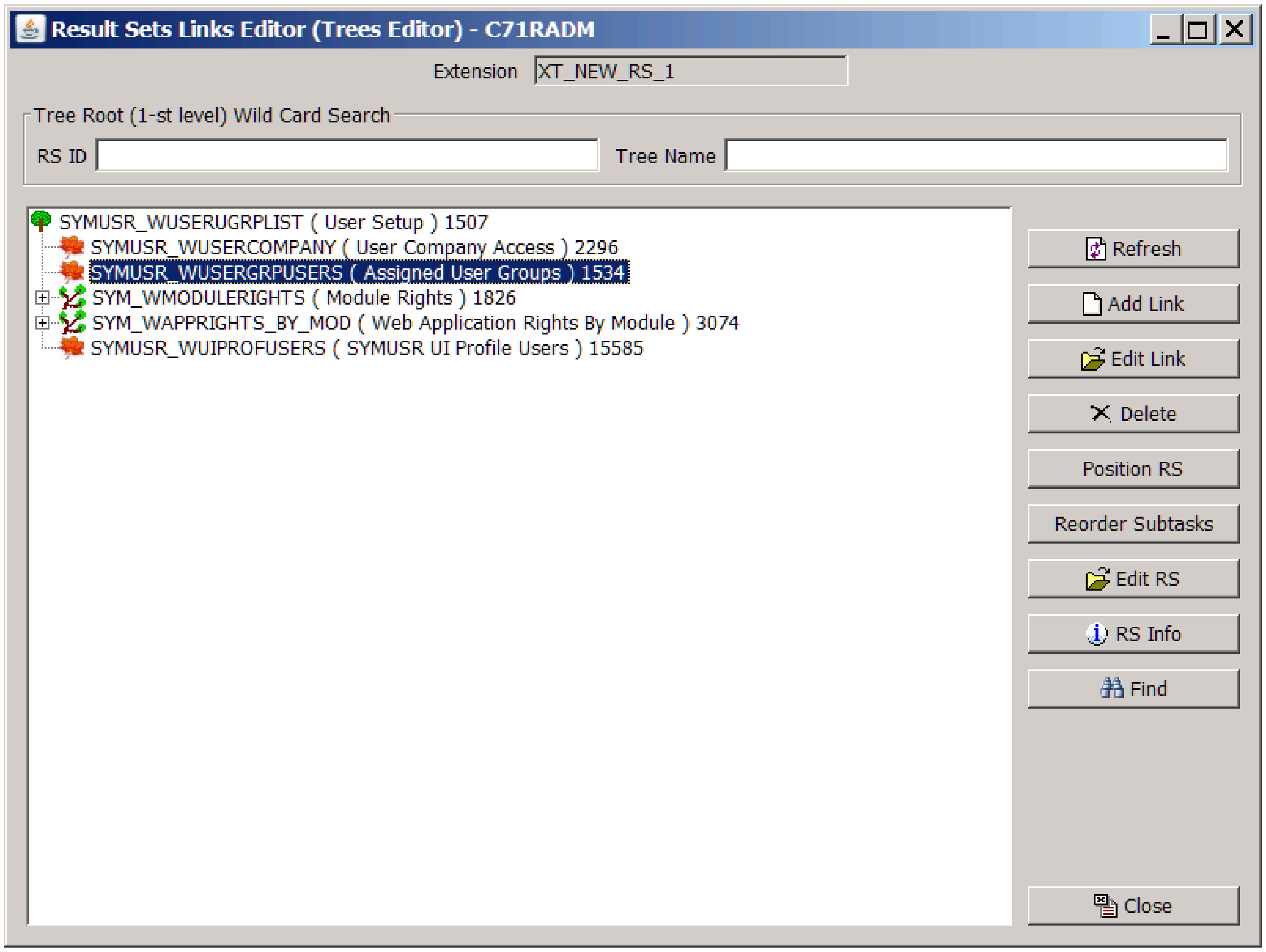
1. Click **Extensions» Extend » Extend RS Tree » Select Ext Project » Select Ext Unit » Select RS Tree to Extend.**



SYM

SYMDD_TABLE_FILTER (Table Filter 3414) 3414
SYMFLOC_WLOCATIONLIST (Maintain Alternative File Locations 2338) 2338
SYMGRP_WUSERUGRPLIST (User Group Security Rights 1515) 1515
SYMLICEN_USERS (Costpoint Licenses 3117) 3117
SYMORGFN_ORGSECBYMODULE (Org Security By Module 1618) 1618
SYMORGRP_ORGSECGRP (Org Security Groups 1748) 1748
SYMORPRF_ORGSECPROFILE (Org Security Profile Setup 1750) 1750
SYMOVER_RANK_FLTR (Ranking Functions 15895) 15895
SYMPRN_WPRINTERLIST (System Printers 2533) 2533
SYMPROF (Enter/Manage UI Profiles 15433) 15433
SYMRPTAS_RPT_ARCH_RIGHTS (Report Archive Security Rights 2753) 2753
SYMRPTCT_RPTCATEGORIES (Report Categories 2223) 2223
SYMRPTGR_RPT_GROUP_LIST (Report Group List 2690) 2690
SYMSETNG_SYSSETTINGS (System Settings 1959) 1959
SYMSTAX_SALESTAX (Maintain Sales/Value Added Tax 1878) 1878
SYMUSR_WUSERUGRPLIST (Web User Security Rights 1507) 1507
SYMXTAU (Extensibility Units 4100) 4100

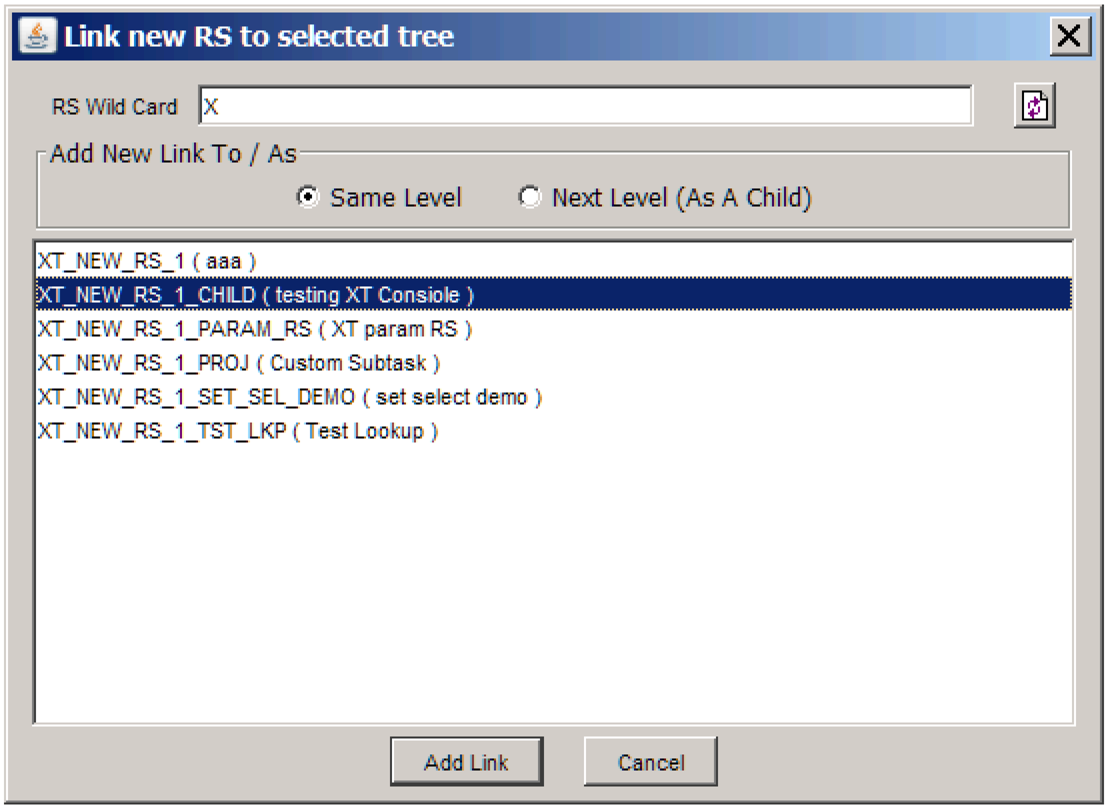
You will be presented with a View of RS Tree that you've selected to extend:



Extension: XT_NEW_RS_1

```
SYMUSR_WUSERUGRPLIST ( User Setup ) 1507
    SYMUSR_WUSERCOMPANY ( User Company Access ) 2296
    SYMUSR_WUSERGRPUSERS ( Assigned User Groups ) 1534
  + SYM_WMODULERIGHTS ( Module Rights ) 1826
  + SYM_WAPPRIGHTS_BY_MOD ( Web Application Rights By Module ) 3074
    SYMUSR_WUIPROFUSERS ( SYMUSR UI Profile Users ) 15585
```

2. Select a level in the RS Tree structure in which you want to add a new subtask, and click **Add Link**:



RS Wild Card: X

Add New Link To / As: [✓ Same Level] Next Level (As A Child)

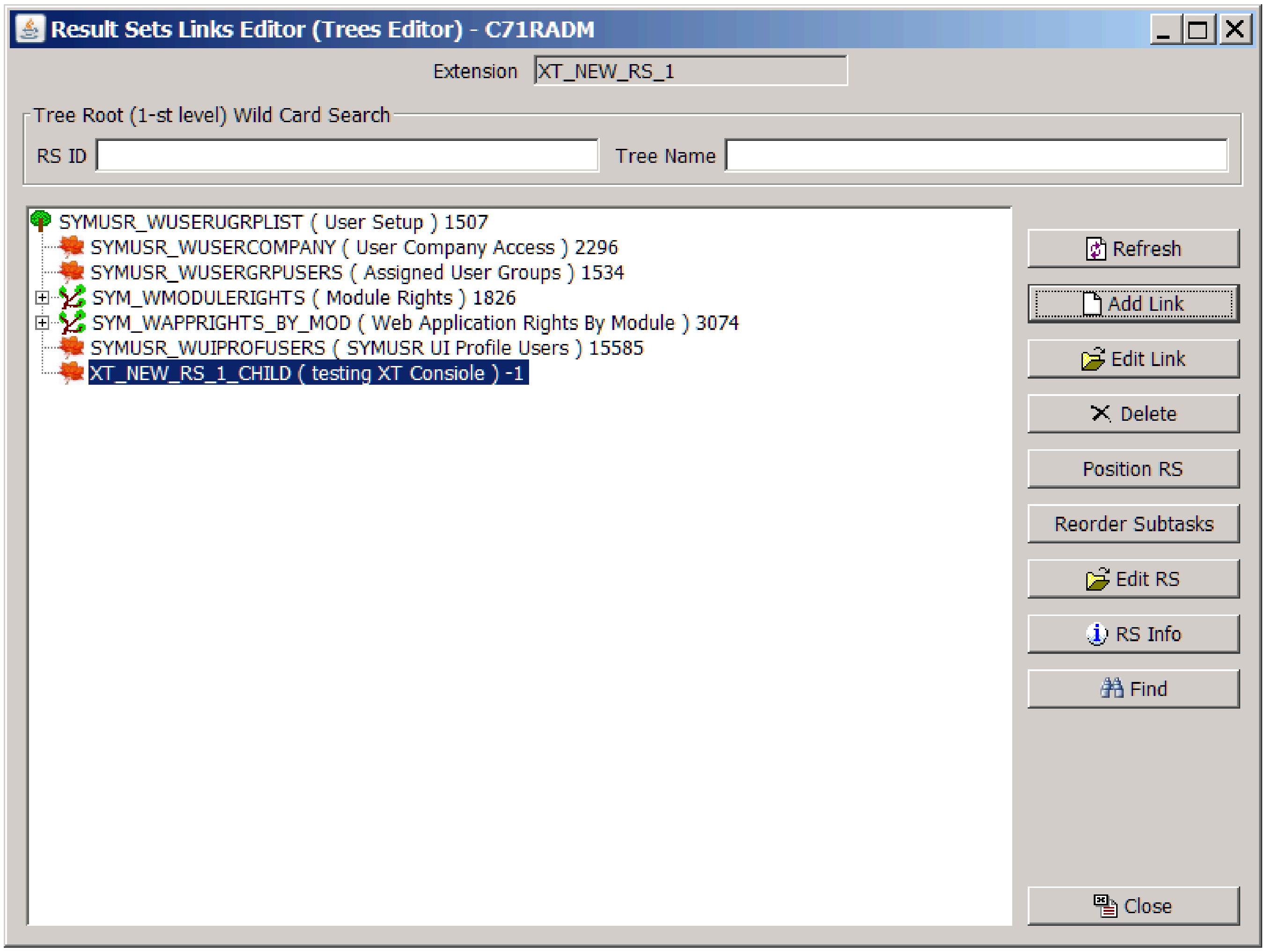
XT_NEW_RS_1 (aaa)
XT_NEW_RS_1_CHILD (testing XT Console)
XT_NEW_RS_1_PARAM_RS (XT param RS)
XT_NEW_RS_1_PROJ (Custom Subtask)
XT_NEW_RS_1_SET_SEL_DEMO (set select demo)
XT_NEW_RS_1_TST_LKP (Test Lookup)

3. Select the RS you want to add as a subtask, and select the level of the **RS Tree** in which it will be added:

   - **Same Level**: Select this option to add the RS as a sibling of the RS selected on the Tree screen.
   - **Next level**: Select this option to add the RS as a child to the selected RS.

   Typically, data in the child RS has a many-to-one relationship to the record in parent RS, but there is usually no such kind of relationship between the data rows displayed in sibling subtasks.

After adding the new subtask, it displays in the structure of the customized RS Tree:


Result Sets Links Editor (Trees Editor) - C71RADM

Extension XT_NEW_RS_1

Tree Root (1-st level) Wild Card Search

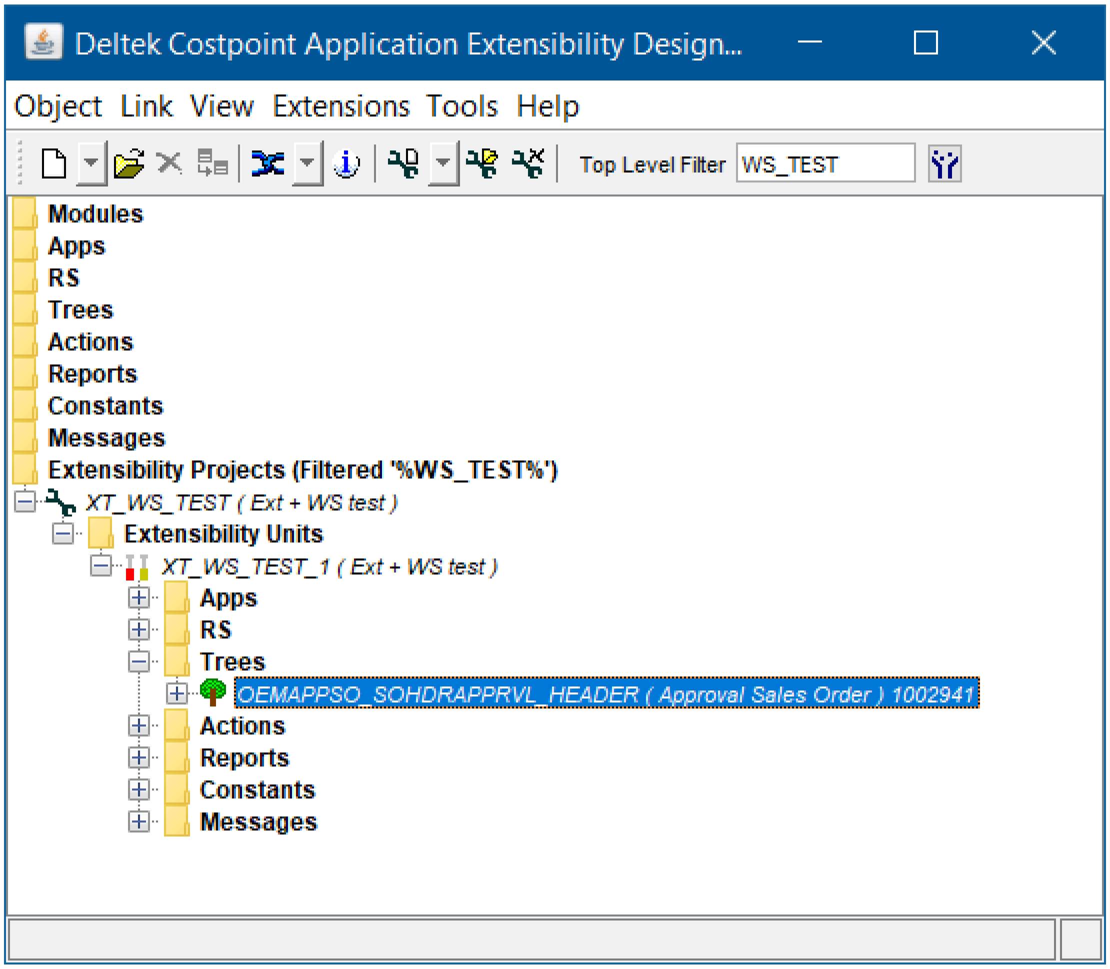
RS ID

Tree Name

```
SYMUSR_WUSERUGRPLIST ( User Setup ) 1507
    SYMUSR_WUSERCOMPANY ( User Company Access ) 2296
    SYMUSR_WUSERGRPUSERS ( Assigned User Groups ) 1534
 ⊞  SYM_WMODULERIGHTS ( Module Rights ) 1826
 ⊞  SYM_WAPPRIGHTS_BY_MOD ( Web Application Rights By Module ) 3074
    SYMUSR_WUIPROFUSERS ( SYMUSR UI Profile Users ) 15585
    XT_NEW_RS_1_CHILD ( testing XT Consiole ) -1
```

Modifying RS Trees

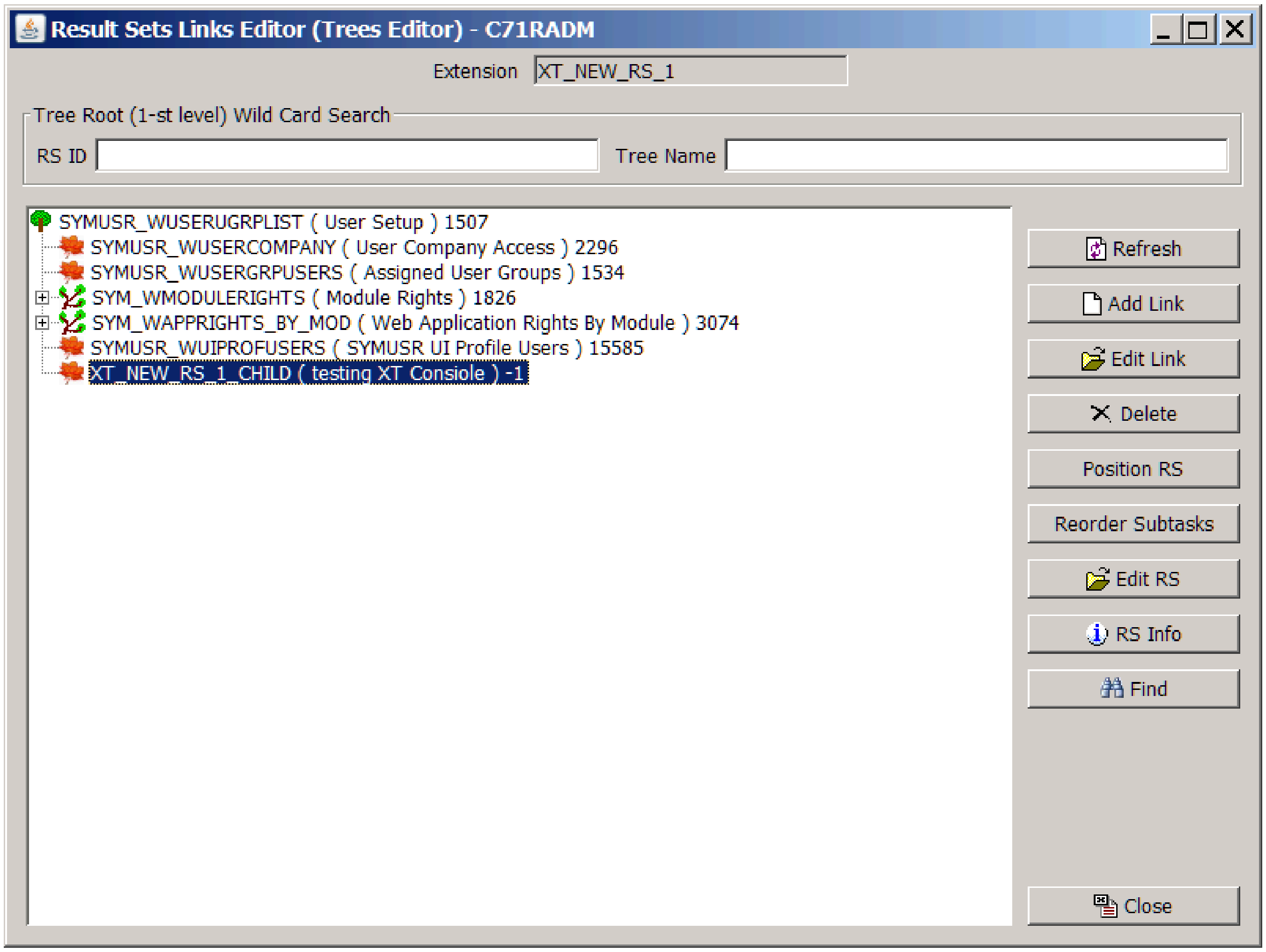
The list of customized RS Trees for a given Extensibility Units is displayed under the **Trees** subfolder of the selected Extensibility Unit:


Deltek Costpoint Application Extensibility Design...
Object Link View Extensions Tools Help
Top Level Filter WS_TEST
Modules
Apps
RS
Trees
Actions
Reports
Constants
Messages
Extensibility Projects (Filtered '%WS_TEST%')
XT_WS_TEST (Ext + WS test)
Extensibility Units
XT_WS_TEST_1 (Ext + WS test)
Apps
RS
Trees
OEMAPPSO_SOHDRAPPRVL_HEADER (Approval Sales Order) 1002941
Actions
Reports
Constants
Messages

Click **Open** to open the selected RS Tree extension. A view of the customized RS Tree displays. Each record has following structure:

The **RS_ID** followed by the **RS Name** in parenthesis followed by the **Child Number**.

The Child Number gives the subtask a unique identifier in the system since a result set can be reused in multiple RS Trees. Positive Child Numbers indicate that the subtask was added by a standard application developer, and negative Child Numbers indicate that it was added for customization in a given Extensibility Unit.


Result Sets Links Editor (Trees Editor) - C71RADM

Extension: XT_NEW_RS_1

Tree Root (1-st level) Wild Card Search

RS ID:

Tree Name:

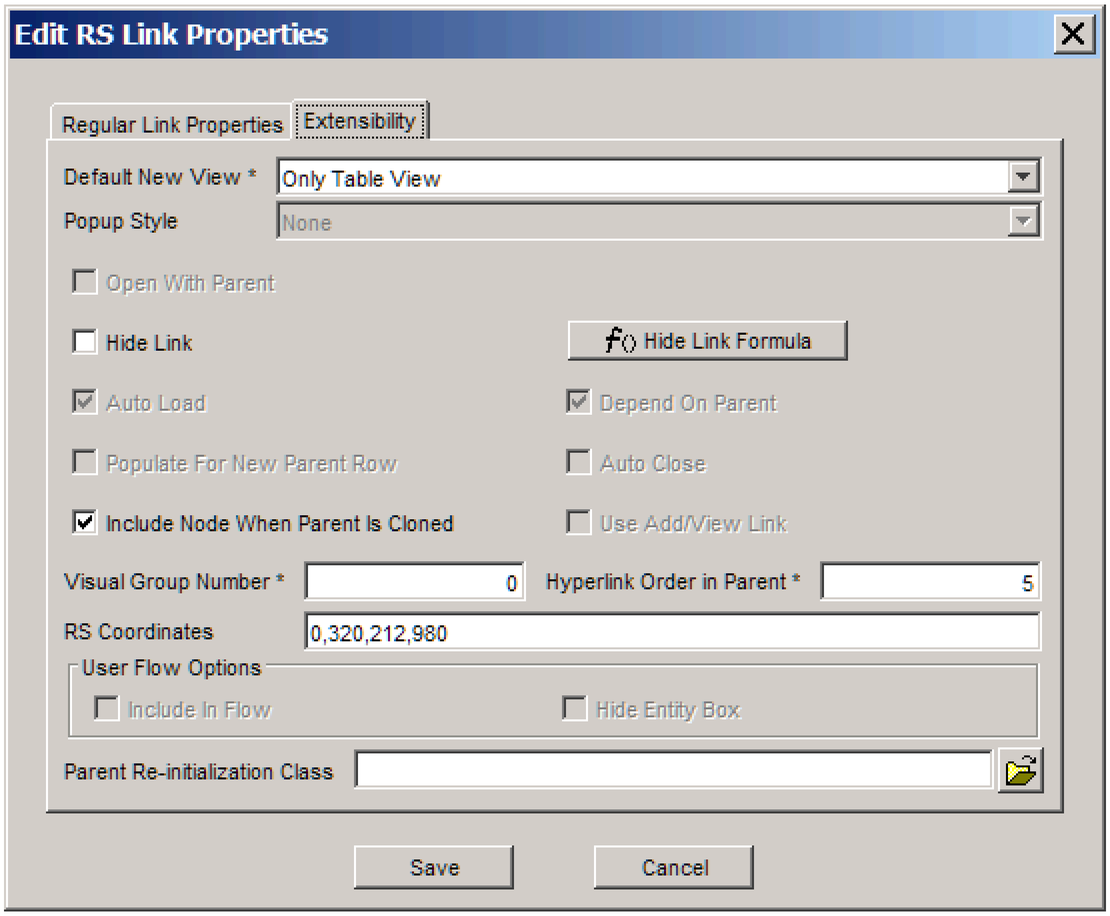
```
SYMUSR_WUSERUGRPLIST ( User Setup ) 1507
    SYMUSR_WUSERCOMPANY ( User Company Access ) 2296
    SYMUSR_WUSERGRPUSERS ( Assigned User Groups ) 1534
  ⊞ SYM_WMODULERIGHTS ( Module Rights ) 1826
  ⊞ SYM_WAPPRIGHTS_BY_MOD ( Web Application Rights By Module ) 3074
    SYMUSR_WUIPROFUSERS ( SYMUSR UI Profile Users ) 15585
    XT_NEW_RS_1_CHILD ( testing XT Console ) -1
```

In the RS Tree (RS Links) Editor, you can:

- Add new subtasks by clicking the **Add Link** button.
- View or Edit Link properties by clicking the **Edit Link** button.
- Delete (remove) subtasks from the RS Tree by clicking the **Delete** button. Keep in mind that you can delete only subtasks that were added by the same Extensibility Unit and you cannot remove standard subtasks that were added by a regular application developer. Instead, if a subtask is irrelevant in your organization, you can use the personalization functionality to change the layout of the application.
- Change the position and the size of the subtasks Tree when they are open by clicking the **Position RS** button.
- Reorder subtask Links as their links are presented to the end users by clicking the **Reorder Subtasks** button.
- View/Edit the RS structure by clicking the **Edit RS** button.
- View the RS usage information by clicking **RS Info** button
- Search for a specific subtask by clicking the **Find** button.

Modifying RS Link Properties for the Selected Subtask

To view or edit RS Link properties, select desired subtask and click the **Edit Link** button. The Edit RS Link Properties dialog box displays, showing link's properties.



Edit RS Link Properties

Regular Link Properties | Extensibility

Default New View *: Only Table View
Popup Style: None

- [ ] Open With Parent
- [ ] Hide Link
- [✓] Auto Load
- [✓] Depend On Parent
- [ ] Populate For New Parent Row
- [ ] Auto Close
- [✓] Include Node When Parent Is Cloned
- [ ] Use Add/View Link

Visual Group Number *: 0
Hyperlink Order in Parent *: 5
RS Coordinates: 0,320,212,980

User Flow Options
- [ ] Include In Flow
- [ ] Hide Entity Box

Parent Re-initialization Class:

If this is a custom Subtask added to the current Extensibility Unit, you will be able to change all the properties available in the dialog box. If the link was created by the standard application developer, the dialog box will have two tabs: the Regular Link Properties tab and the Extensibility tab. The Regular Link Properties tab displays the link properties in read-only mode, and you can edit a subset of the properties on the **Extensibility** Tab. Most of the properties in the dialog box have self-explanatory names. The following properties require more detailed explanations:

- **Depend On Parent**: If selected, the data will be re-selected when the parent context row changes. If a child row changes, the parent row is also updated when saved regardless of parent row changes or **Always Validated**.
- **Auto Close**: If selected, this subtask closes if the user sets focuses on any result set other than this subtask, child result set of this subtask, or lookup opened from this subtask. In other words, the result set will remain open for as long as the user navigates within the boundaries of the sub tree starting with this result set. You must use this feature with **Populate For New Parent Row** to ensure that you cannot change the parent data until this subtask is closed.

You can have nested Auto Close (that is, a child result set may also be marked as **Auto Close**). In this case, the behavior would apply for the sub tree similar to any Auto Close sub tree. Auto Close does not work with **Visual Group** except only for the result set at the top of the visual group. That is, if you have a visual group, only the top of the visual group should be marked Auto Close. Clicking away from the visual group will close the whole group.

> **Note:** If you use **Popup Style**, **Auto Close** will automatically be selected.

- **Visual Group #**: Enter a non-zero group number to indicate that this result set belongs to a group with other results having the same visual group number. All result sets having the same Visual Group number will be opened and closed at the same time as it is one screen.
- **Parent Re-Initialization Class**: If the parent result set shows totals from a child result set, these totals may be incorrect whenever the user performs a sub query on the child result set. If you have such totals, you need to implement a class to re-initialize the totals in the parent result set. The class must implement the ReInitParentRSInterface interface and the method void reInitializeParentRS(ResultSetInterface rs) throws DEException. The system will pass the RSI of the child result set and the application will refresh the total on the parent result set from there.

Not only can you add subtasks, but (if needed) you can create a whole new custom RS (subtask) Tree. It can consist of both standard Result Sets and newly created custom Result Sets.

To create a new RS Tree:

1. On the **Extensions** menu, click **Object** » **New** » **RS Tree**.

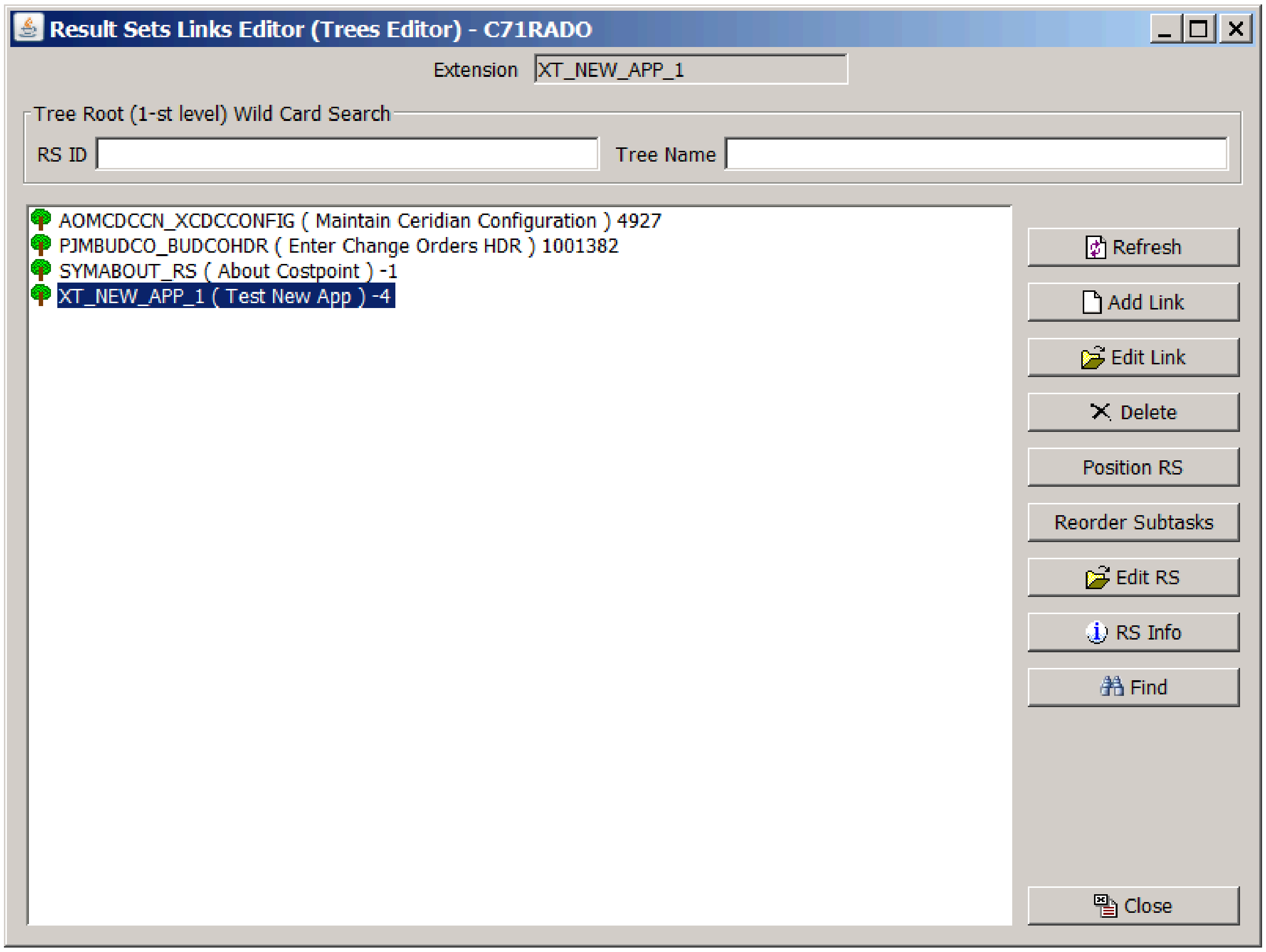
   Alternately, click the **New** icon, and click **RS Tree**.

2. On the **Select Extension Project for new RS Tree** dialog box, select the project, and click **Next**.

3. On the **Select Extension Unit for new RS Tree** dialog box, select the unit, and click **Next**.

4. Select the RS that will become the header for a given RS Tree.

   A new RS Tree will be created and **Result Sets Links Editor (Links Editor)** will be opened that displays all the extended or newly created custom RS Trees in the selected Extensibility Unit:



Extension: XT_NEW_APP_1

Tree Root (1-st level) Wild Card Search

RS ID

Tree Name

AOMCDCCN_XCDCCONFIG (Maintain Ceridian Configuration) 4927
PJMBUDCO_BUDCOHDR (Enter Change Orders HDR) 1001382
SYMABOUT_RS (About Costpoint) -1
XT_NEW_APP_1 (Test New App) -4

5. Select the appropriate RS Tree record(s), and click **Add Link** to add additional subtasks to this RS Tree.

6. Double-click each record in **Result Sets Links Editor (Links Editor)** to see its children subtasks.

This section describes how you can customize existing Actions (processes) in Costpoint. For existing Costpoint Actions you can customize Action Title and Status text seen on the UI. You can add custom logic (plug-in) to be executed before or after system invokes regular Action logic.

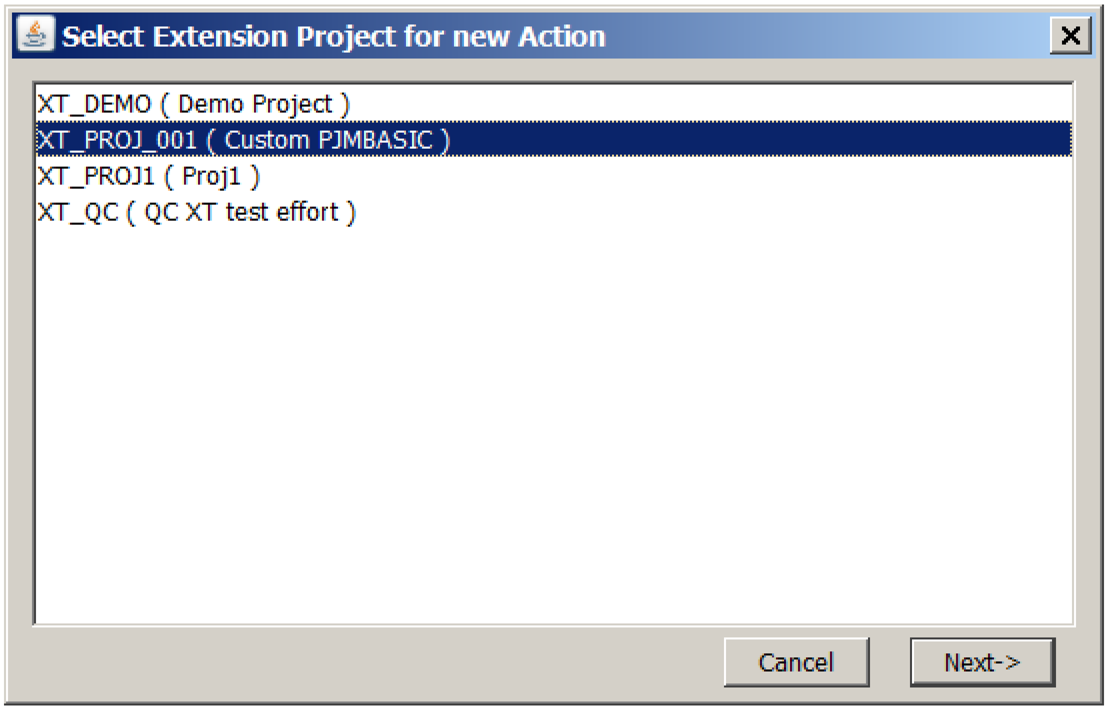
In addition to extending existing regular action, you can also add new action to a result set. See Adding new Actions/Processes.

Create New Extension for Action

To create a new extension for an action:

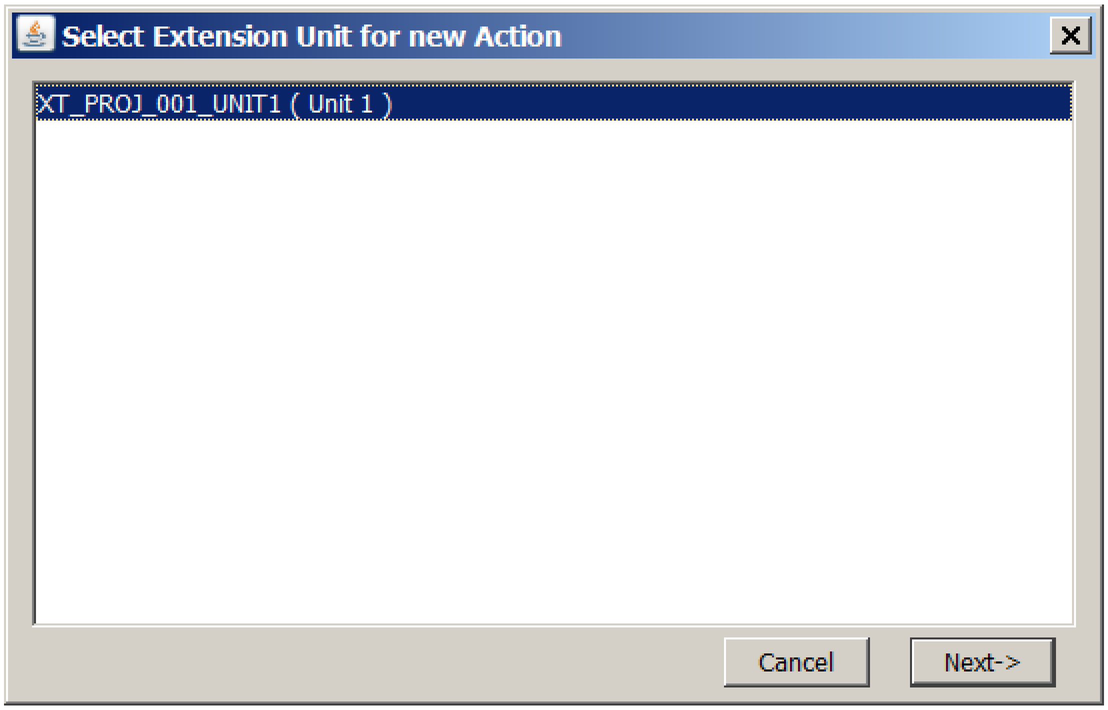
1. Click **Extensions » Extend » Action** or click the **New Extension** icon ( ) and then click **Action**. The **Select**

Extension Project for new Action dialog box displays.


Select Extension Project for new Action

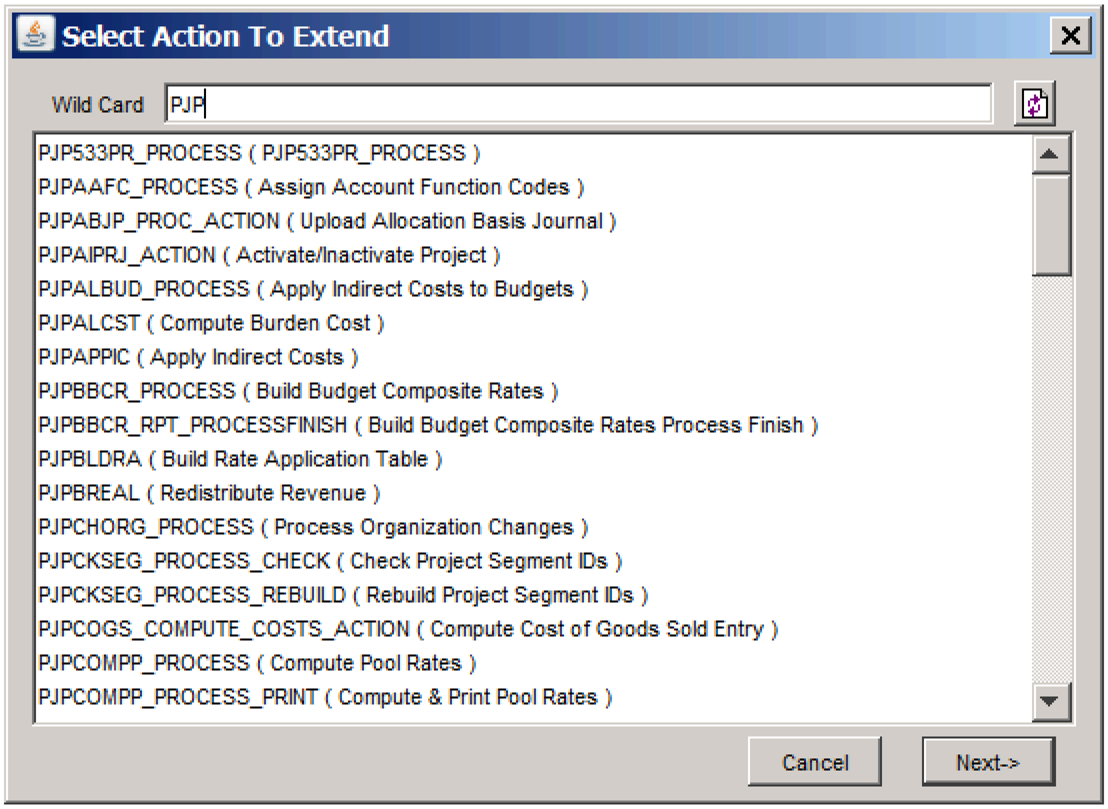
XT_DEMO (Demo Project)
XT_PROJ_001 (Custom PJMBASIC)
XT_PROJ1 (Proj1)
XT_QC (QC XT test effort)

2. Select the project and then click **Next**. The **Select Extension Unit for new Action** dialog box displays.


Select Extension Unit for new Action

XT_PROJ_001_UNIT1 (Unit 1)

3. Select the unit and then click **Next**. The **Select Action to Extend** dialog box displays.

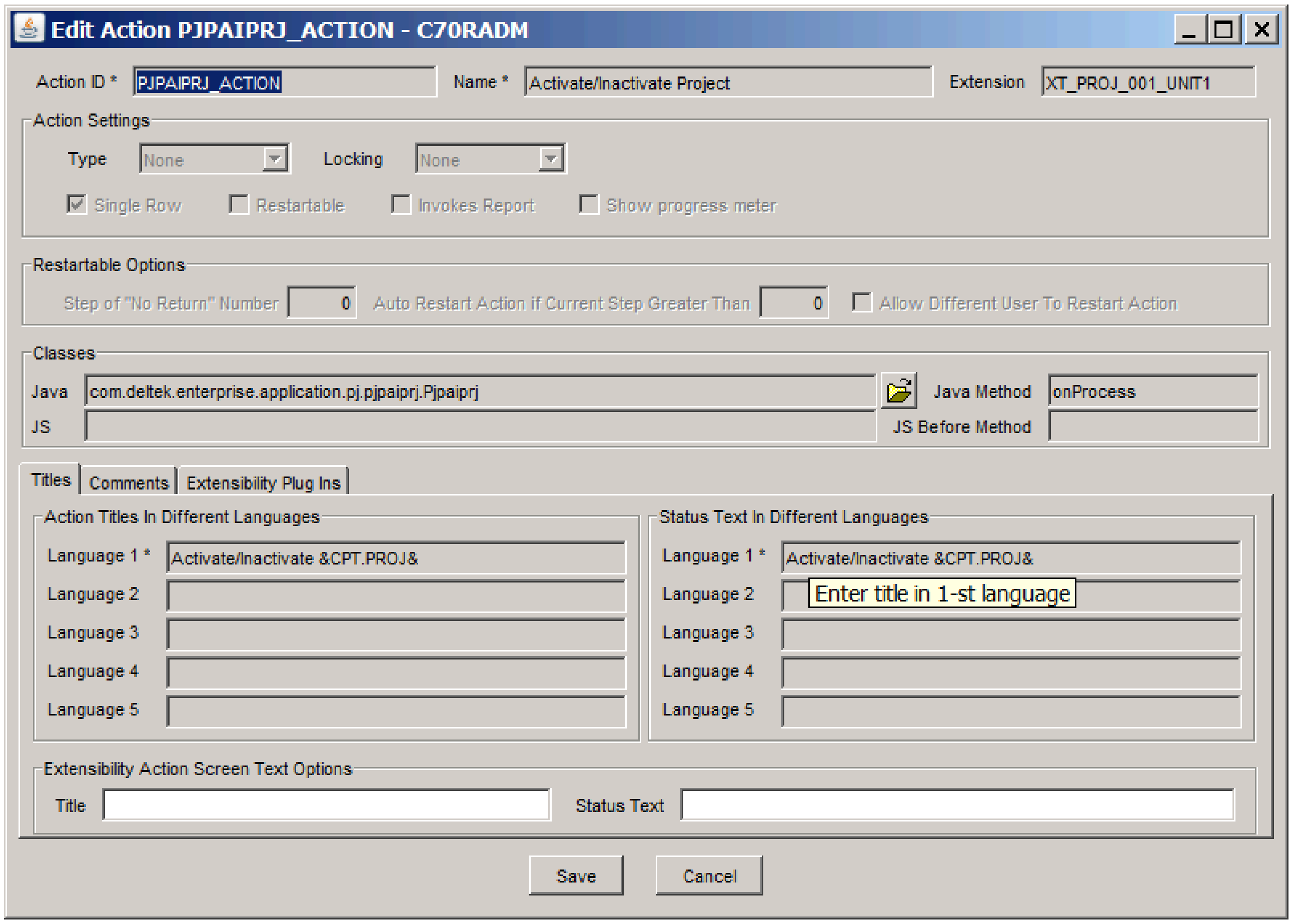

Select Action To Extend

Wild Card PJP

PJP533PR_PROCESS (PJP533PR_PROCESS)
PJPAAFC_PROCESS (Assign Account Function Codes)
PJPABJP_PROC_ACTION (Upload Allocation Basis Journal)
PJPAIPRJ_ACTION (Activate/Inactivate Project)
PJPALBUD_PROCESS (Apply Indirect Costs to Budgets)
PJPALCST (Compute Burden Cost)
PJPAPPIC (Apply Indirect Costs)
PJPBBCR_PROCESS (Build Budget Composite Rates)
PJPBBCR_RPT_PROCESSFINISH (Build Budget Composite Rates Process Finish)
PJPBLDRA (Build Rate Application Table)
PJPBREAL (Redistribute Revenue)
PJPCHORG_PROCESS (Process Organization Changes)
PJPCKSEG_PROCESS_CHECK (Check Project Segment IDs)
PJPCKSEG_PROCESS_REBUILD (Rebuild Project Segment IDs)
PJPCOGS_COMPUTE_COSTS_ACTION (Compute Cost of Goods Sold Entry)
PJPCOMPP_PROCESS (Compute Pool Rates)
PJPCOMPP_PROCESS_PRINT (Compute & Print Pool Rates)

4. Use the **Wild Card** field to narrow the list and select an action.

5. Click **Next**. The **Edit Action** dialog box displays. This dialog box contains full metadata information regarding an action you want to extend. At the bottom, you can override standard **Action Title** or **Status Text** if you need to customize them.


Edit Action PJPAIPRJ_ACTION - C70RADM

Action ID *	PJPAIPRJ_ACTION	Name *	Activate/Inactivate Project	Extension	XT_PROJ_001_UNIT1

Action Settings

Type: None Locking: None

[✓] Single Row [ ] Restartable [ ] Invokes Report [ ] Show progress meter

Restartable Options

Step of "No Return" Number: 0 Auto Restart Action if Current Step Greater Than: 0 [ ] Allow Different User To Restart Action

Classes

Java	com.deltek.enterprise.application.pj.pjpaiprj.Pjpaiprj	Java Method	onProcess
JS		JS Before Method	

Titles | Comments | Extensibility Plug Ins

Action Titles In Different Languages		Status Text In Different Languages	
Language 1 *	Activate/Inactivate &CPT.PROJ&	Language 1 *	Activate/Inactivate &CPT.PROJ&
Language 2		Language 2	Enter title in 1-st language
Language 3		Language 3	
Language 4		Language 4	
Language 5		Language 5	

Extensibility Action Screen Text Options

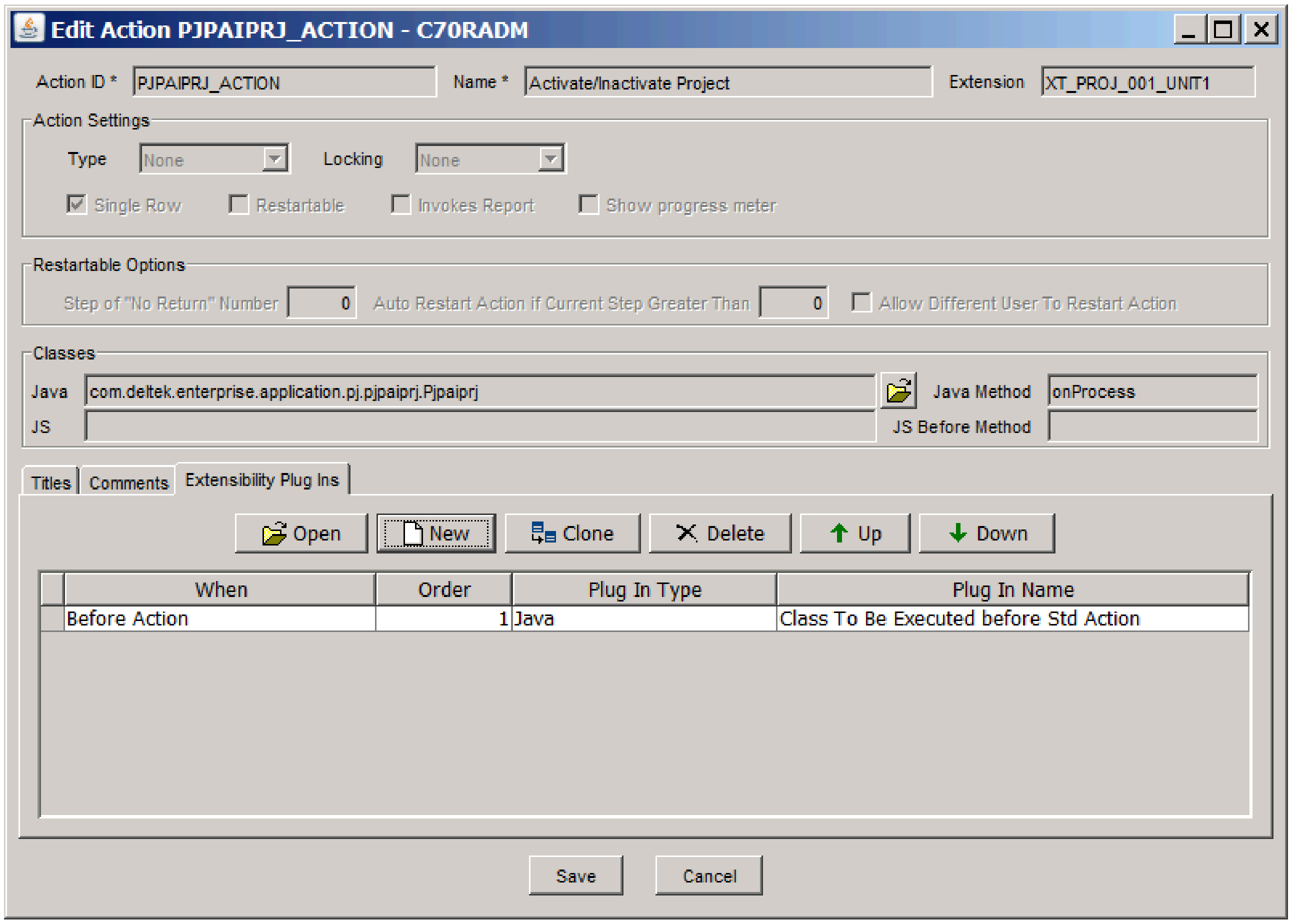
Title: Status Text:

Extensibility Plug-Ins for Actions

For an existing action you can create one or more extensibility plug-ins that will be executed before or after standard action.

To create extensibility plug-in for an action:

1. On the Extensibility Plug Ins tab, click **New**.



Edit Action PJPAIPRJ_ACTION - C70RADM

Action ID *	PJPAIPRJ_ACTION	Name *	Activate/Inactivate Project	Extension	XT_PROJ_001_UNIT1

Action Settings

Type: None — Locking: None

[✓ Single Row] [ ] Restartable [ ] Invokes Report [ ] Show progress meter

Restartable Options

Step of "No Return" Number: 0 — Auto Restart Action if Current Step Greater Than: 0 — [ ] Allow Different User To Restart Action

Classes

Java: com.deltek.enterprise.application.pj.pjpaiprj.Pjpaiprj — Java Method: onProcess

JS: — JS Before Method:

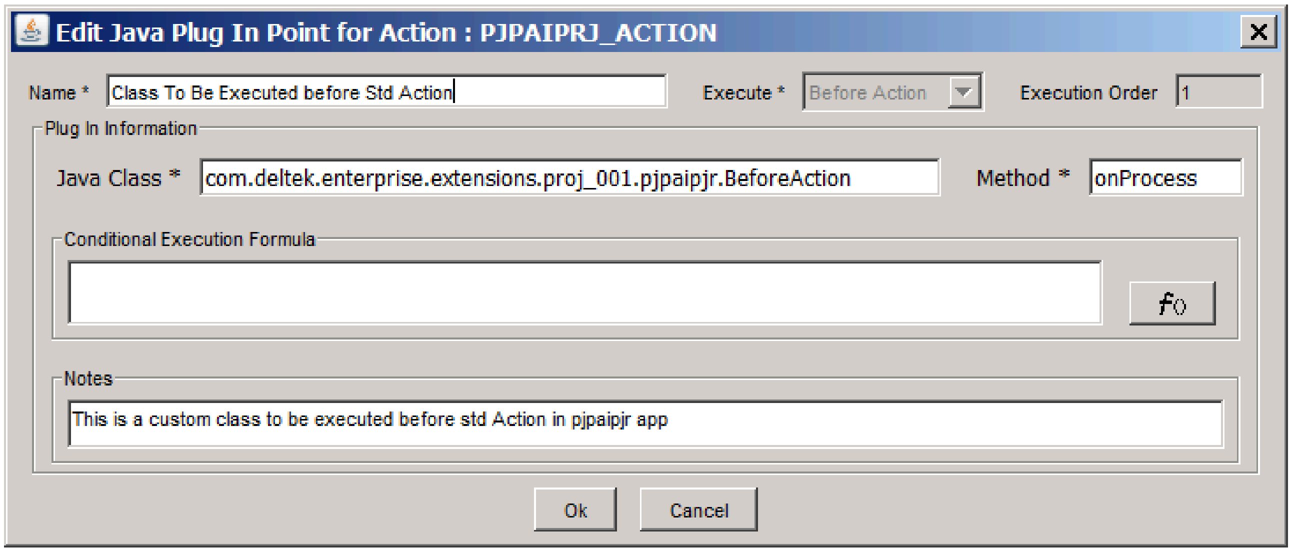
Titles | Comments | Extensibility Plug Ins

Open | New | Clone | Delete | Up | Down

When	Order	Plug In Type	Plug In Name
Before Action	1	Java	Class To Be Executed before Std Action

Save | Cancel

2. For Action plug-in, you can implement it with **Java Class** or **Stored Procedure**. You can also add email notification in the plug-in. For each plug-in, you can specify to have it executed before or after the standard action is executed.



Edit Java Plug In Point for Action : PJPAIPRJ_ACTION

Name *	Class To Be Executed before Std Action	Execute *	Before Action	Execution Order	1

Plug In Information

Java Class *: com.deltek.enterprise.extensions.proj_001.pjpaipjr.BeforeAction — Method *: onProcess

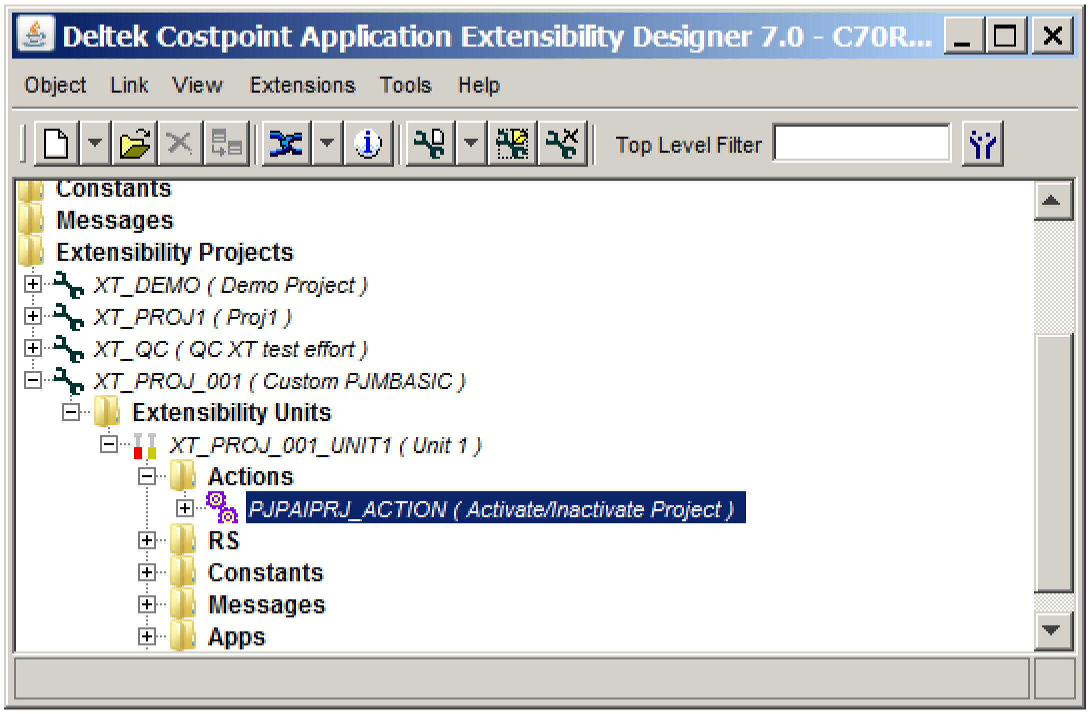
Conditional Execution Formula

Notes: This is a custom class to be executed before std Action in pjpaipjr app

Ok | Cancel

Attention: For more on Java plug-ins see Appendix A - Plug Ins to complete next step of creating plug-in.

3. Click **Ok** to save the extensions and close the **Edit Action** dialog box. The new extension for an action should now display under the **Actions** folder in the selected **Extensibility Unit**.

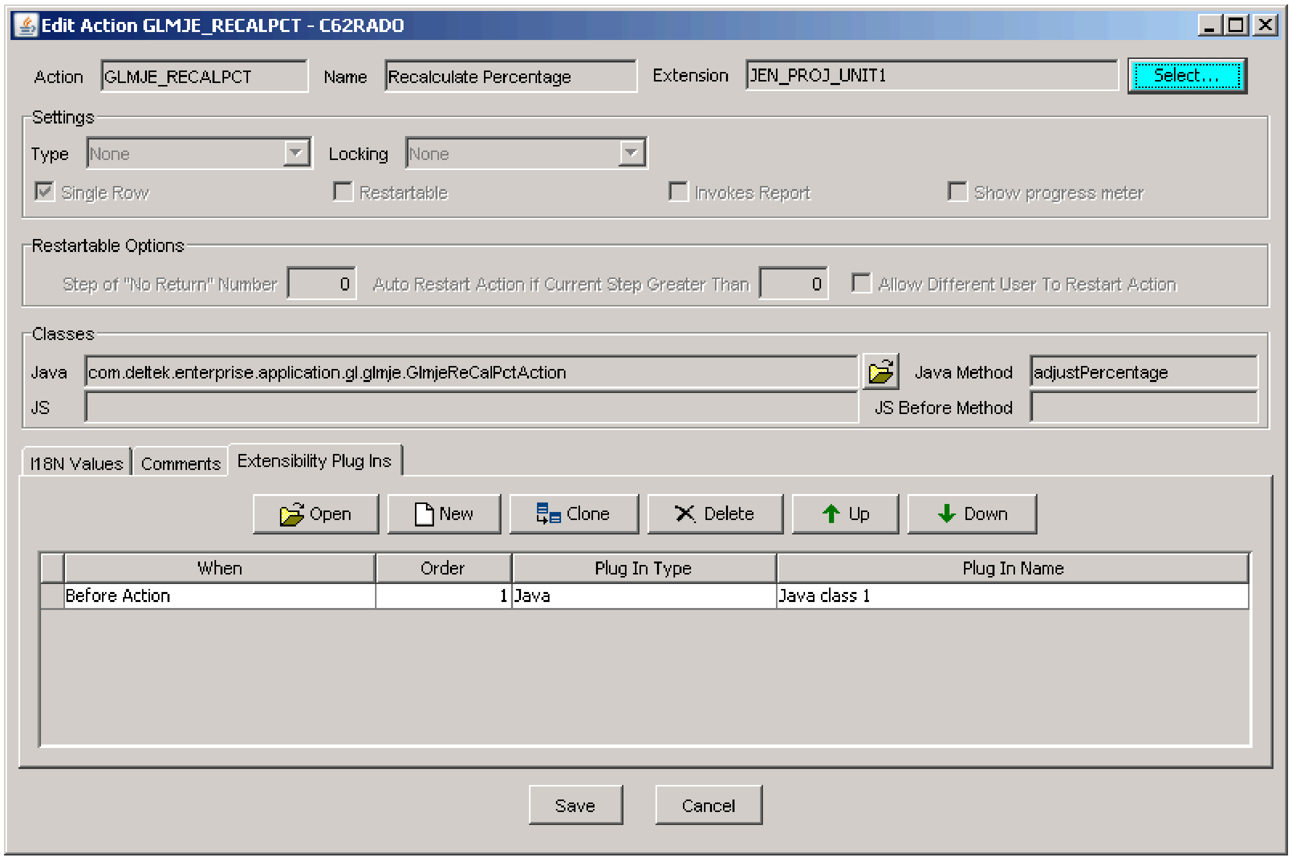

Deltek Costpoint Application Extensibility Designer 7.0 - C70R...
Object Link View Extensions Tools Help
Top Level Filter
Constants
Messages
Extensibility Projects
XT_DEMO (Demo Project)
XT_PROJ1 (Proj1)
XT_QC (QC XT test effort)
XT_PROJ_001 (Custom PJMBASIC)
Extensibility Units
XT_PROJ_001_UNIT1 (Unit 1)
Actions
PJPAIPRJ_ACTION (Activate/Inactivate Project)
RS
Constants
Messages
Apps

Edit an Existing Extended Action

To edit an extended action:

1. Expand the **Extensibility Projects** folder, the project to modify, the **Extensibility Units** folder, the unit to modify, and the **Actions** folder.

2. Select the action record to modify and then click **Extensions » Open Extension** or click the **Open Extension** icon. The **Edit Action** dialog box displays.


Edit Action GLMJE_RECALPCT - C62RADO

Action	GLMJE_RECALPCT	Name	Recalculate Percentage
Extension	JEN_PROJ_UNIT1		

Settings

Type	None
Locking	None

☑ Single Row ☐ Restartable ☐ Invokes Report ☐ Show progress meter

Restartable Options

Step of "No Return" Number: 0 Auto Restart Action if Current Step Greater Than: 0 ☐ Allow Different User To Restart Action

Classes

Java	com.deltek.enterprise.application.gl.glmje.GlmjeReCalPctAction
Java Method	adjustPercentage
JS	
JS Before Method	

When	Order	Plug In Type	Plug In Name
Before Action	1	Java	Java class 1

Delete an Existing Extended Action

To delete an extended action:

1. Expand the **Extensibility Projects** folder, the project to modify, the **Extensibility Units** folder, the unit to modify, and the **Actions** folder.

2. Select the Action to modify and then click **Extensions » Delete Extension** or click the **Delete Extension** icon.

This section describes how you can add brand new Actions (processes) and assign them to applications in Costpoint. Unlike extending a standard action, new action allows you complete control over the logic executed when user invokes the action in the application.

The following are the steps in adding new action:

1. Click **Project » Unit » Add new action.**

2. Register the action plug-in with java classes or stored procedure to be executed. The java class or stored

procedure logic are to be implemented outside of this Designer.

3. Link (assign) the action to an existing result set in an application

4. This new action will be available when the extensibility package is deployed to the end system.

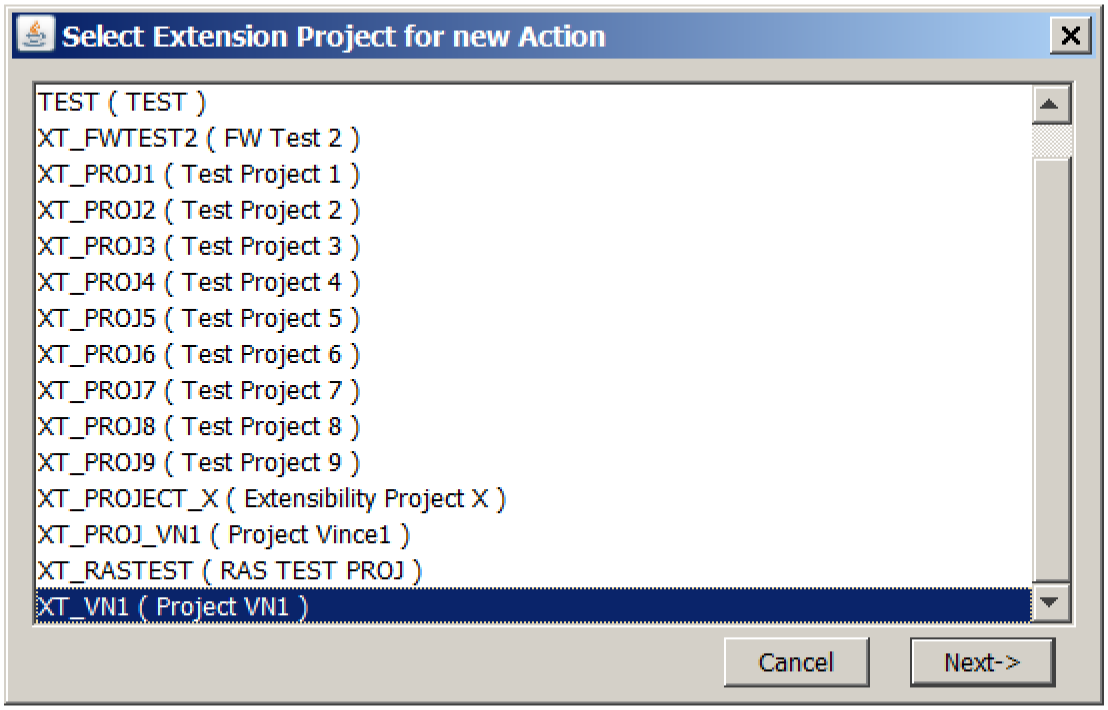
Create a New Action

To create a new action:

1. On the **Extensions** menu, click **Object » New » Action**.

   Alternately, click the **New** icon, and click **Action**.

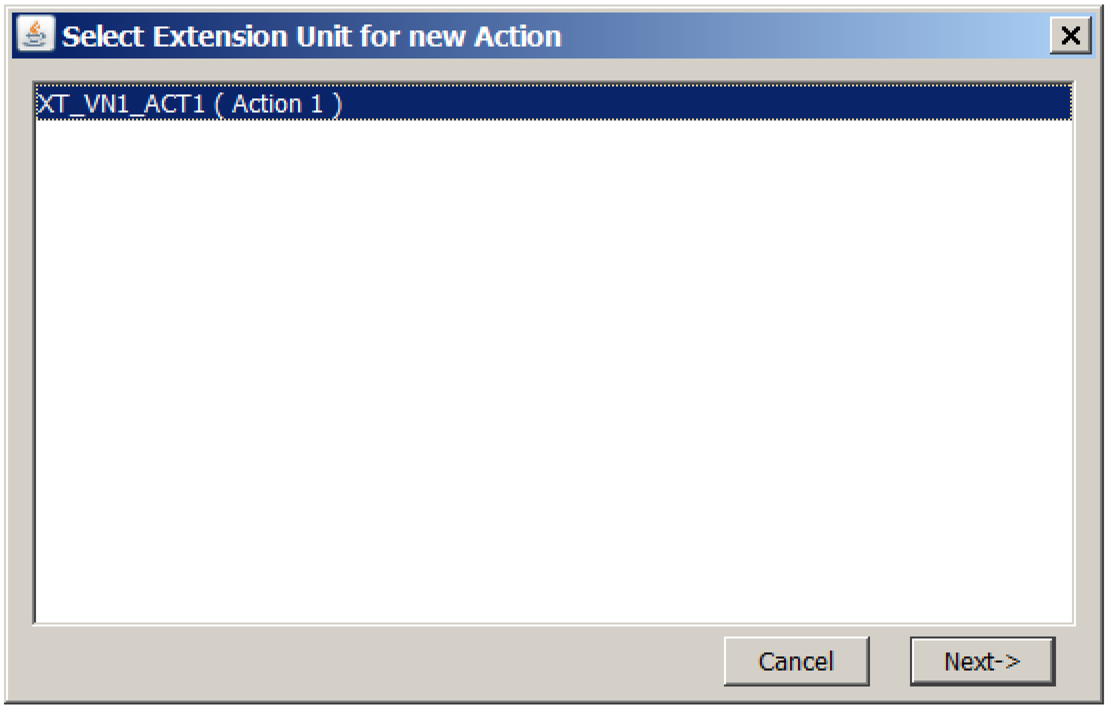
2. On the Select Extension Project for new Action dialog box, select the project and click **Next**.



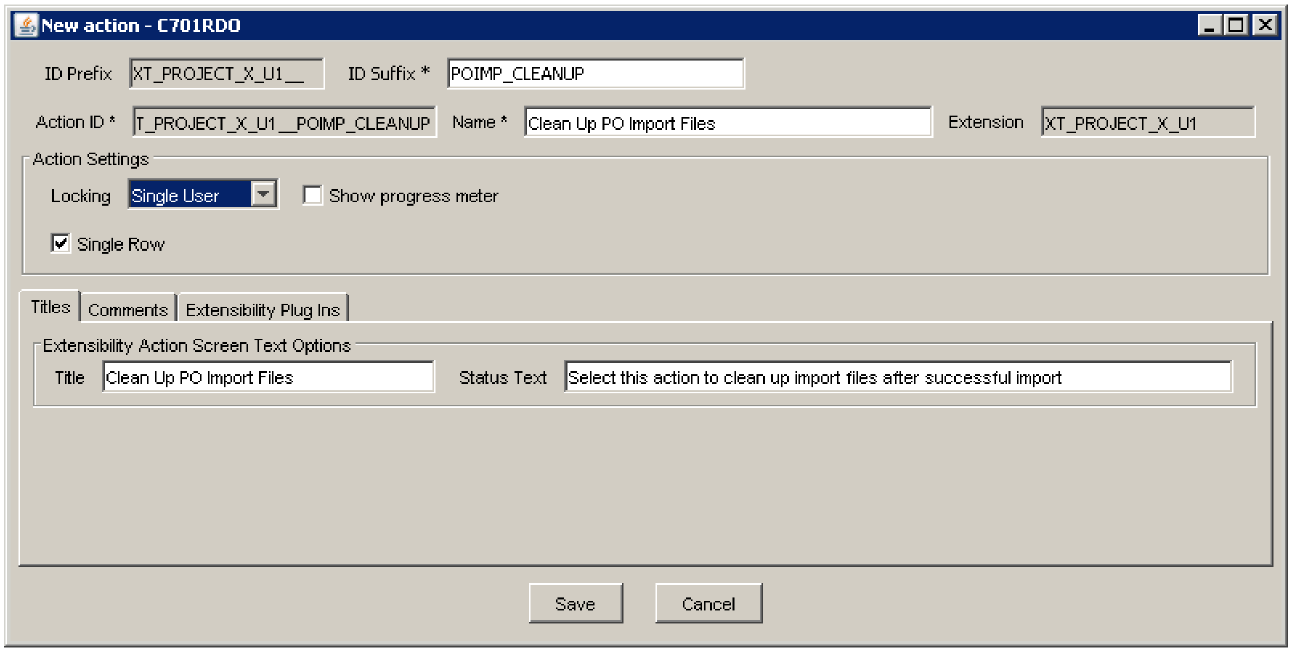
Select Extension Project for new Action

TEST (TEST)
XT_FWTEST2 (FW Test 2)
XT_PROJ1 (Test Project 1)
XT_PROJ2 (Test Project 2)
XT_PROJ3 (Test Project 3)
XT_PROJ4 (Test Project 4)
XT_PROJ5 (Test Project 5)
XT_PROJ6 (Test Project 6)
XT_PROJ7 (Test Project 7)
XT_PROJ8 (Test Project 8)
XT_PROJ9 (Test Project 9)
XT_PROJECT_X (Extensibility Project X)
XT_PROJ_VN1 (Project Vince1)
XT_RASTEST (RAS TEST PROJ)
XT_VN1 (Project VN1)

3. On the Select Extension Unit for new Action dialog box, select the unit and click **Next**.



4. On the New action dialog box, complete the following actions:



ID Prefix XT_PROJECT_X_U1__ ID Suffix * POIMP_CLEANUP

Action ID * T_PROJECT_X_U1__POIMP_CLEANUP Name * Clean Up PO Import Files Extension XT_PROJECT_X_U1

Action Settings

Locking Single User Show progress meter

[✓] Single Row

Extensibility Action Screen Text Options

Title Clean Up PO Import Files Status Text Select this action to clean up import files after successful import

- **Action ID**: Complete the Action ID. The ID automatically starts with the Extensibility Unit ID and an underscore (_).
- **Name**: Enter a descriptive name in this field.
- **Locking**: Select the locking schema from this drop-down list. This must be implemented correspondingly in the plug in java classes or stored procedure.

- Single User: This action can be run only in one instance at a time in the whole system. This is the most restrictive. If this is set, no special code needs to be implemented in the plug in logic. System will allow this action to run only one at a time.
- User: This action can be run only in one instance for each user but a different user can run it concurrently. This is less restrictive since a different user can run this action concurrently. The logic to allow concurrent processes must be implemented accordingly to ensure data integrity.
- Company: This action can be run only in one instance for each different company within Costpoint.
- Restart: This action can be restarted by the same user if it was running in one workstation and the same user tries to restart it in a different workstation.
- None: No locking. This action can be run concurrently by any user on any company. Usually this involves no updates to data, such as running a report using data directly from a view or base table.

(See reference document for more explanation).

- Show progress meter: Select this to show the progress dialog box on the UI while the action is running. The progress meter is controlled inside the java class plug-in. You also need to select this checkbox if your Action is not short running (like a quick calculation on the screen or some kind of defaulting) and you want Framework to keep tracking the execution of such Actions in the Job Status Report app.
- Single Row: Generally selected for report or process application. Leave blank if the action is on a maintenance or transaction application to aid in data entry. When this is blank, the plug in code must loop through rows selected by the user in the UI. When selected, the system will perform the loop. The plug-in code is implemented as though only one row is selected.
- Title and Status Text: Enter the title and status text to be displayed on the UI for this action.
- Comments: Enter additional comments for your notes regarding this action.
- Extensibility Plug-Ins: Add one or more plug-ins to be executed when this action is selected in Costpoint. Plug-Ins can be implemented with Java Classes or Stored Procedure. You can also add email notification in the plug-in.

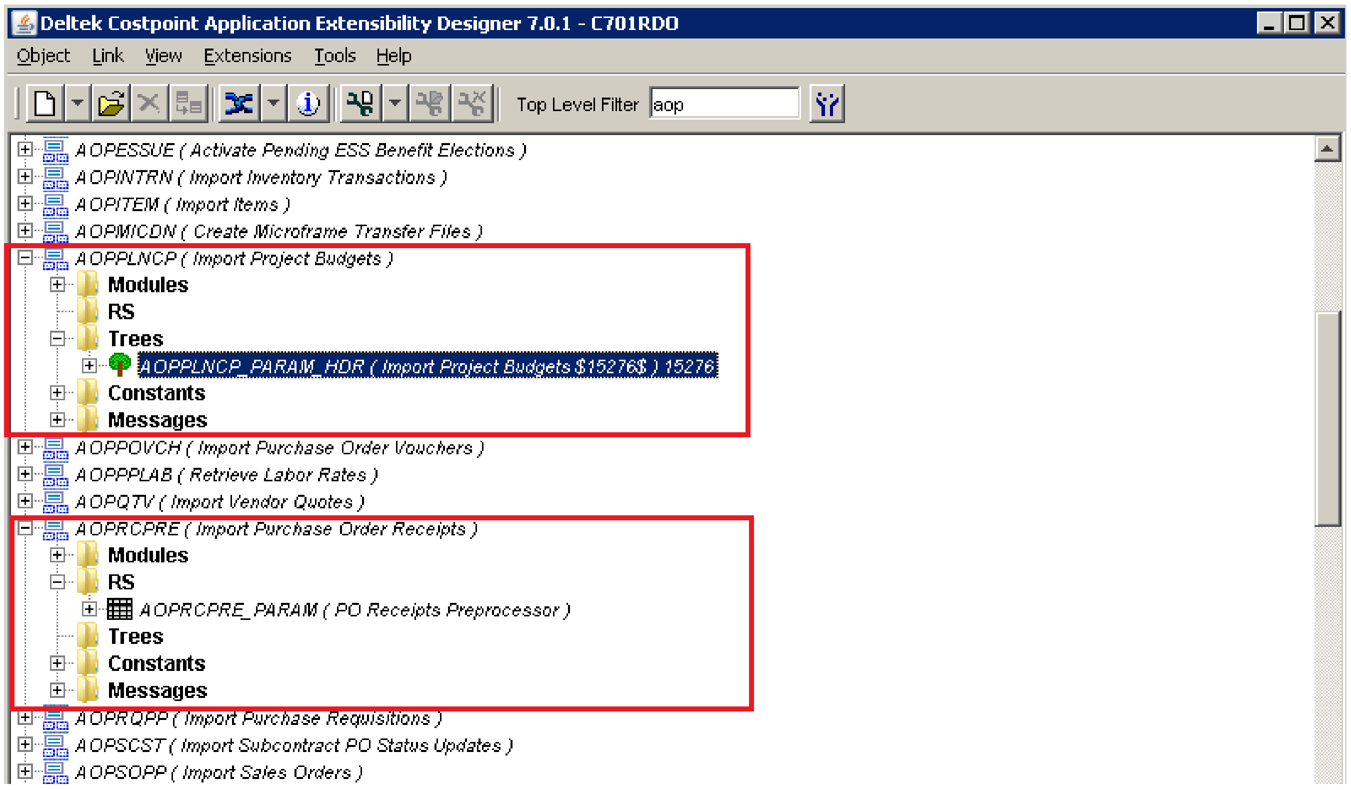
Attention: For more on Java plug-ins, see Appendix A - Plug-Ins.

Assign Action to Result Set/Tree

After an action is created, it needs to be assigned to a result set that belongs to the application. If the application contains more than one result set (a tree), then assign it to the tree. If the application contains only one result set, then assign it to the result set.

To find out if an application contains a single result set or a tree, browse the application at the top level. Expand the RS or the Trees at the next level. Only one of them should exist. That is, an application can be assigned to either a result set or a tree at the top level (not both). Note the name of the result set or the tree. If it is a tree, also note the tree number. When assigning the action to the tree, the tree number must be exact.

In the example below, the app Import Project Budgets is assigned to a Tree whereas the app Import Purchase Order Voucher is assigned to a single result set.


Deltek Costpoint Application Extensibility Designer 7.0.1 - C701RDO
Object Link View Extensions Tools Help
Top Level Filter aop

AOPESSUE (Activate Pending ESS Benefit Elections)
AOPINTRN (Import Inventory Transactions)
AOPITEM (Import Items)
AOPMICDN (Create Microframe Transfer Files)
AOPPLNCP (Import Project Budgets)
Modules
RS
Trees
AOPPLNCP_PARAM_HDR (Import Project Budgets 15276) 15276
Constants
Messages
AOPPOVCH (Import Purchase Order Vouchers)
AOPPPLAB (Retrieve Labor Rates)
AOPQTV (Import Vendor Quotes)
AOPRCPRE (Import Purchase Order Receipts)
Modules
RS
AOPRCPRE_PARAM (PO Receipts Preprocessor)
Trees
Constants
Messages
AOPRQPP (Import Purchase Requisitions)
AOPSCST (Import Subcontract PO Status Updates)
AOPSOPP (Import Sales Orders)

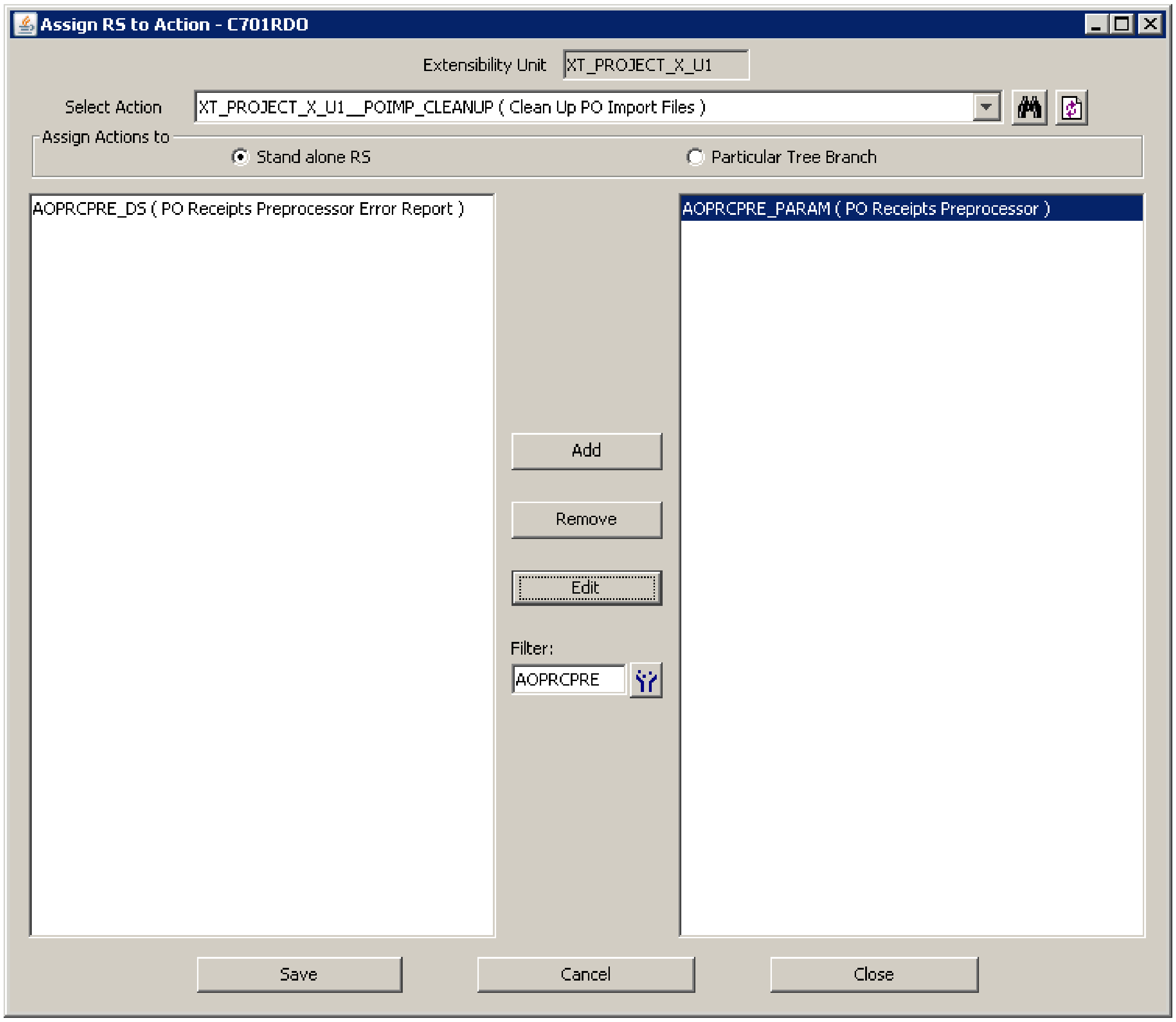
To assign a new action to an RS or a tree:

1. Select the action under the Project » Unit. Then from the menu bar, click **Link » Assign » RS** (or **Tree**).

2. The Assign RS to Action dialog comes up. Select the Stand Alone RS or Tree Branch as determined earlier. The left side will show all the RS or Trees for all applications.

3. Use Filter field in the middle to filter out the name.

4. Select the RS or Tree and click **Add**.



Extensibility Unit: XT_PROJECT_X_U1

Select Action: XT_PROJECT_X_U1__POIMP_CLEANUP (Clean Up PO Import Files)

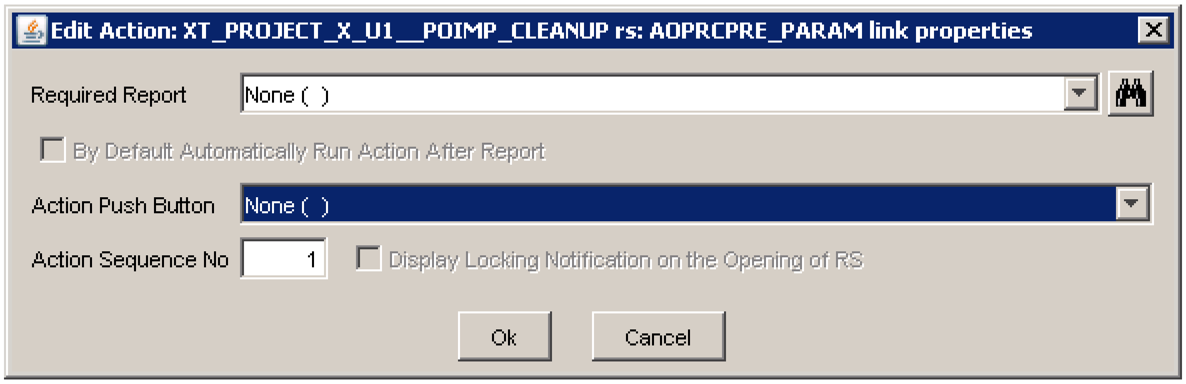
Assign Actions to: [✓ Stand alone RS] Particular Tree Branch

AOPRCPRE_DS (PO Receipts Preprocessor Error Report)

AOPRCPRE_PARAM (PO Receipts Preprocessor)

Filter: AOPRCPRE

5. Click **Edit**. The Edit link properties dialog comes up. Generally, you can leave all existing settings in here as default.



Edit Action: XT_PROJECT_X_U1__POIMP_CLEANUP rs: AOPRCPRE_PARAM link properties
Ok
Cancel

Required Report: None ( )

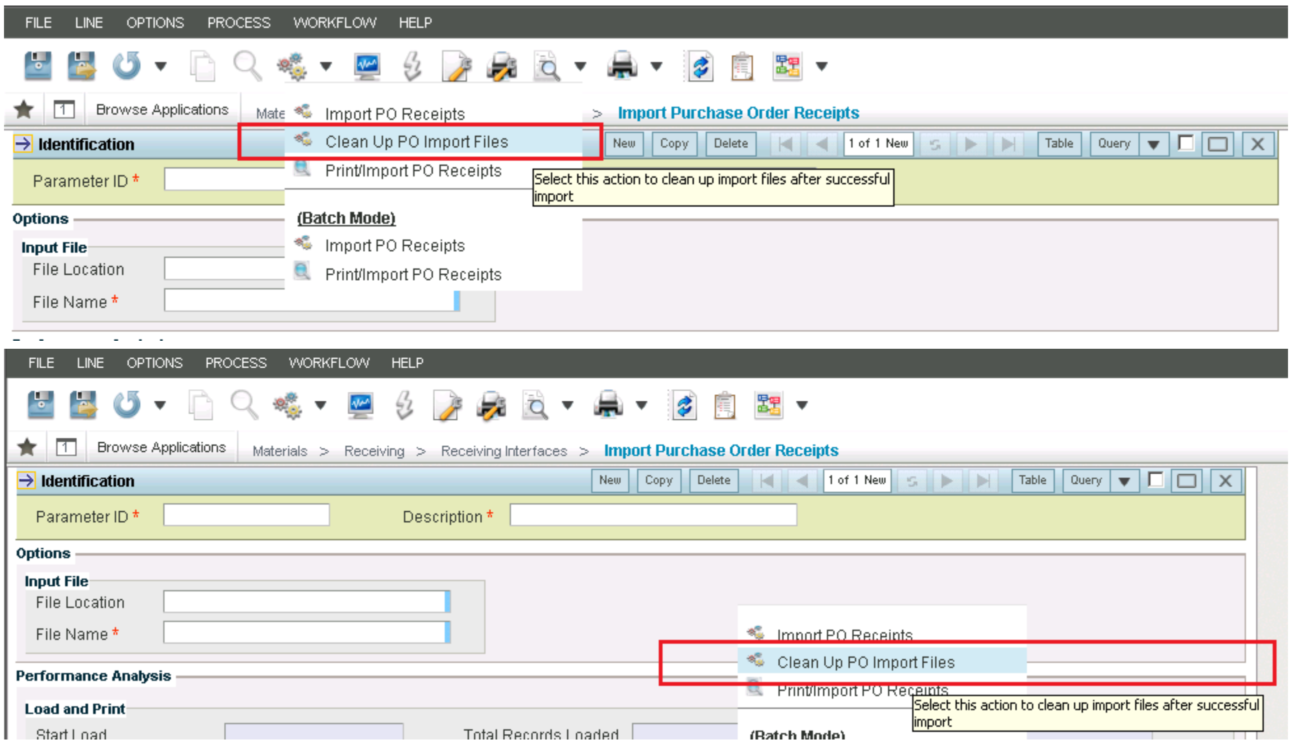
By Default Automatically Run Action After Report

Action Push Button: None ( )

Action Sequence No: 1

Display Locking Notification on the Opening of RS

6. If you have extended the linked RS to add a button for this action, you can click the Action Push Button drop-down list and select the button. Otherwise, actions are always seen in Costpoint UI from the top-level action icon drop down or via right click on the background of the result set.



Select this action to clean up import files after successful import

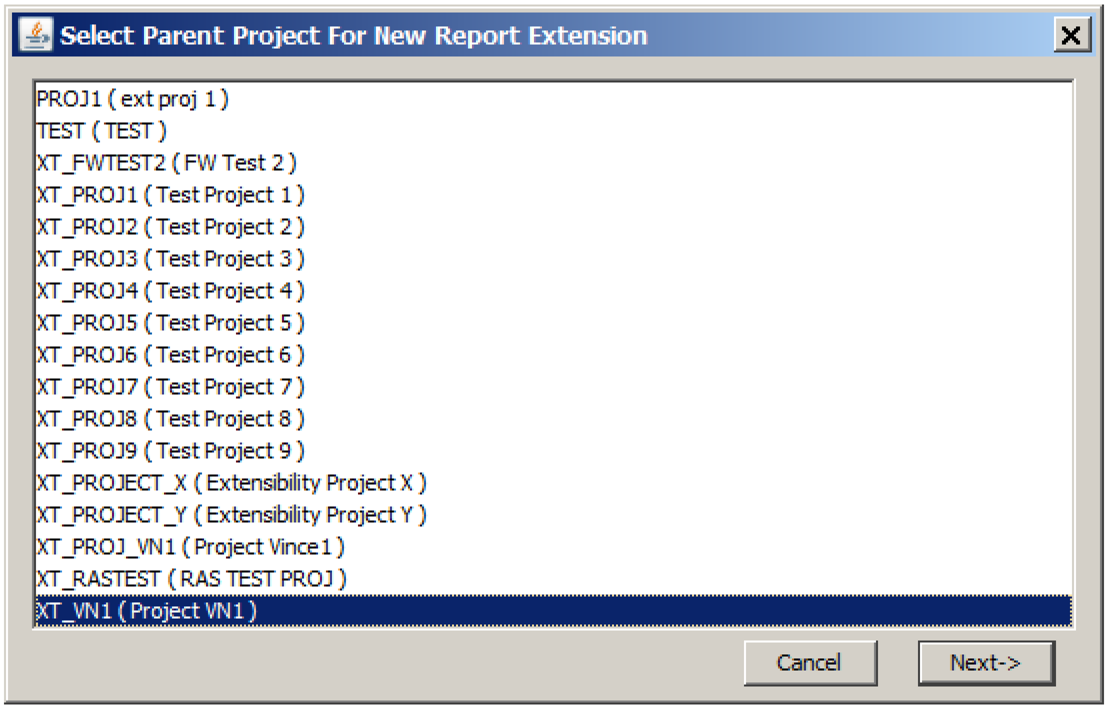
This section describes how you can create the report extension in the Designer. To customize a report, you also need to customize the report template, which is designed in Eclipse IDE for BIRT.

Attention: Please see the Extensibility Designer Report Guide for more information.

Create New Extension for Report

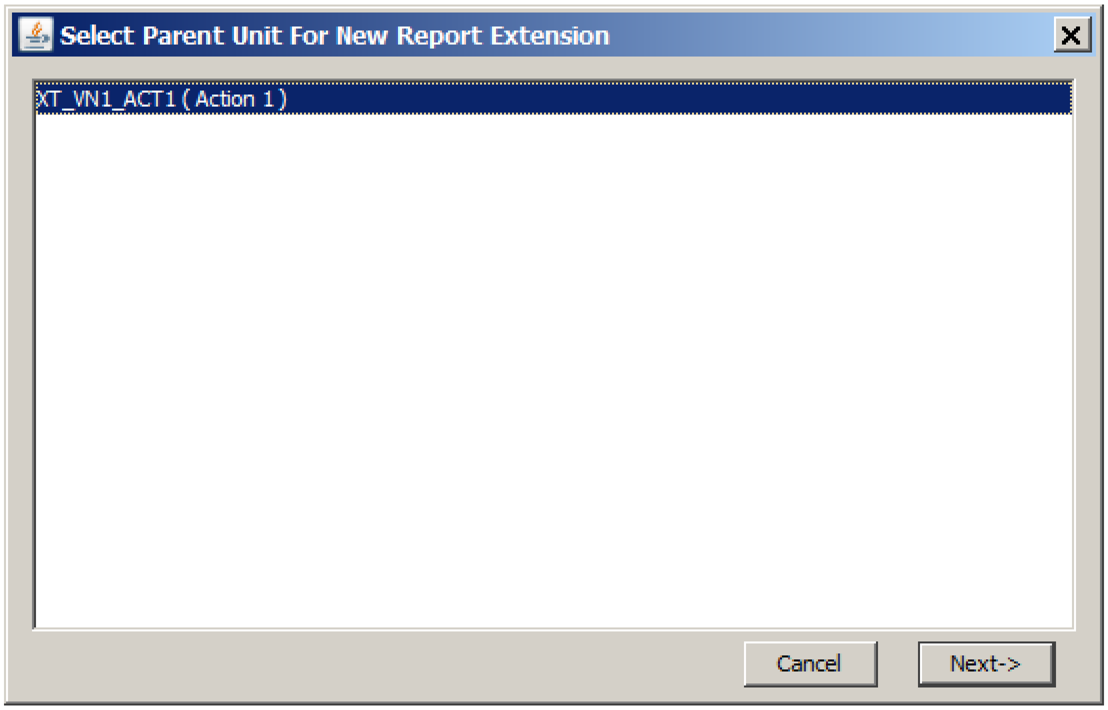
To create a new extension for an action:

1. Click **Extensions » Extend » Report**, or click the **New Extension** icon ( ) and then click **Report**. The **Select Parent Project for New Report Extension** dialog box displays.


Select Parent Project For New Report Extension

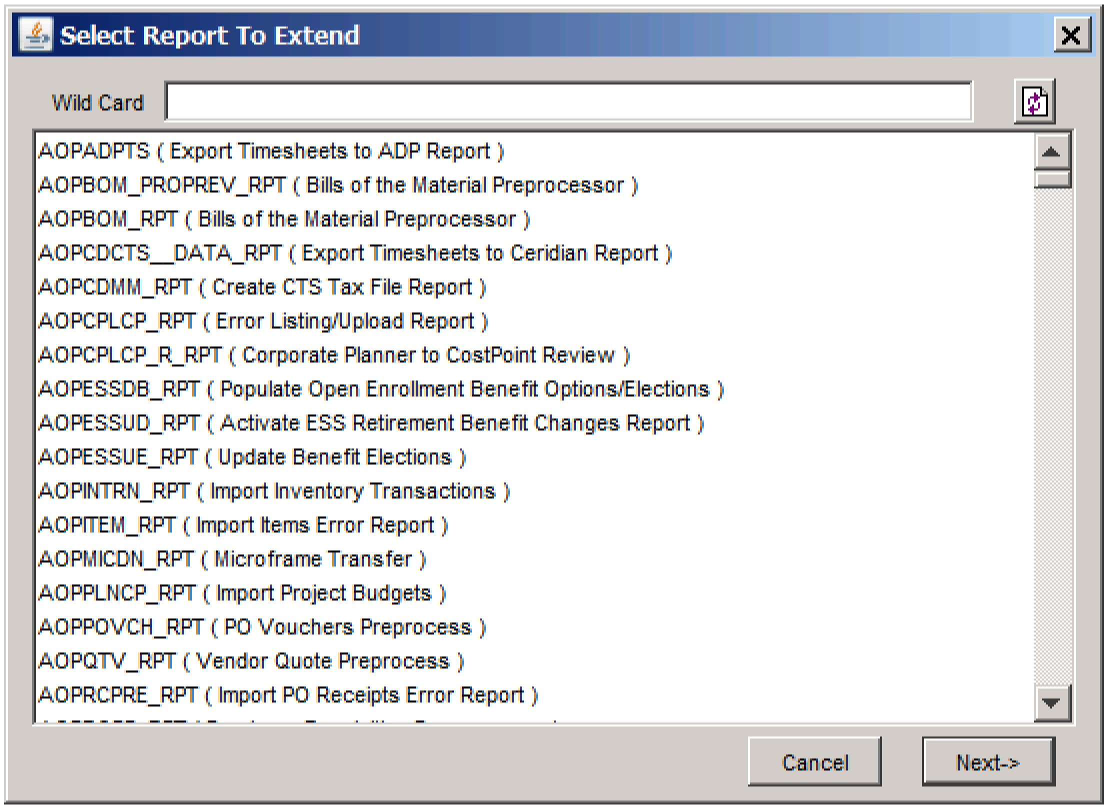
PROJ1 (ext proj 1)
TEST (TEST)
XT_FWTEST2 (FW Test 2)
XT_PROJ1 (Test Project 1)
XT_PROJ2 (Test Project 2)
XT_PROJ3 (Test Project 3)
XT_PROJ4 (Test Project 4)
XT_PROJ5 (Test Project 5)
XT_PROJ6 (Test Project 6)
XT_PROJ7 (Test Project 7)
XT_PROJ8 (Test Project 8)
XT_PROJ9 (Test Project 9)
XT_PROJECT_X (Extensibility Project X)
XT_PROJECT_Y (Extensibility Project Y)
XT_PROJ_VN1 (Project Vince1)
XT_RASTEST (RAS TEST PROJ)
XT_VN1 (Project VN1)

2. Select the project, and click **Next**. The **Select Parent Unit for New Report Extension** dialog box displays.


Select Parent Unit For New Report Extension

XT_VN1_ACT1 (Action 1)

3. Select the unit, and click **Next**. The **Select Report to Extend** dialog box displays.


Select Report To Extend

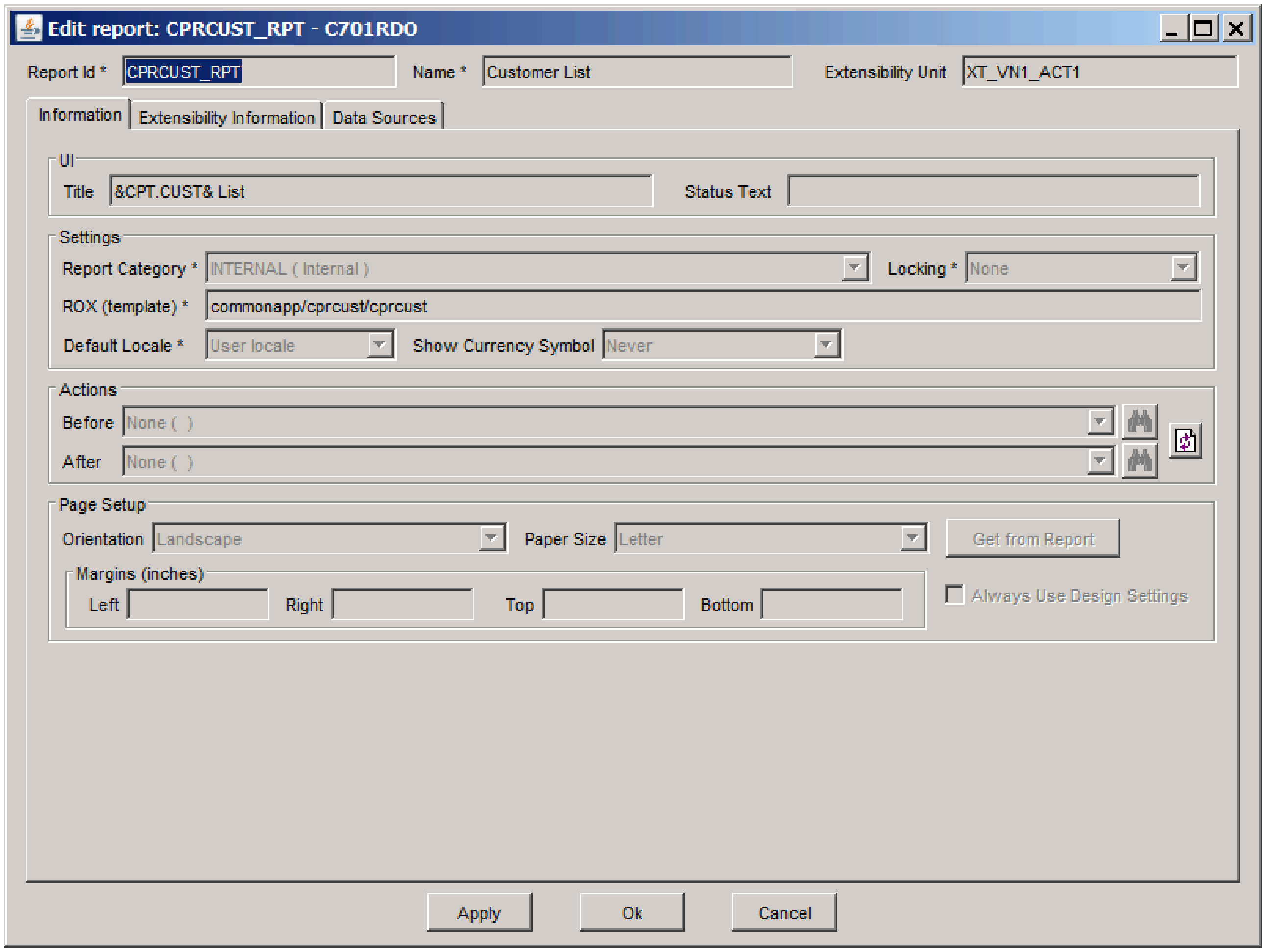
Wild Card

AOPADPTS (Export Timesheets to ADP Report)
AOPBOM_PROPREV_RPT (Bills of the Material Preprocessor)
AOPBOM_RPT (Bills of the Material Preprocessor)
AOPCDCTS__DATA_RPT (Export Timesheets to Ceridian Report)
AOPCDMM_RPT (Create CTS Tax File Report)
AOPCPLCP_RPT (Error Listing/Upload Report)
AOPCPLCP_R_RPT (Corporate Planner to CostPoint Review)
AOPESSDB_RPT (Populate Open Enrollment Benefit Options/Elections)
AOPESSUD_RPT (Activate ESS Retirement Benefit Changes Report)
AOPESSUE_RPT (Update Benefit Elections)
AOPINTRN_RPT (Import Inventory Transactions)
AOPITEM_RPT (Import Items Error Report)
AOPMICDN_RPT (Microframe Transfer)
AOPPLNCP_RPT (Import Project Budgets)
AOPPOVCH_RPT (PO Vouchers Preprocess)
AOPQTV_RPT (Vendor Quote Preprocess)
AOPRCPRE_RPT (Import PO Receipts Error Report)

4. Use the **Wild Card** field to narrow the list and select an action.

5. Click **Next**.

6. On the **Edit Report** dialog box, click the Information tab. This displays the original Costpoint metadata.


Edit report: CPRCUST_RPT - C701RDO

Report Id * CPRCUST_RPT Name * Customer List Extensibility Unit XT_VN1_ACT1

Information | Extensibility Information | Data Sources

UI
Title: &CPT.CUST& List
Status Text:

Settings
Report Category *: INTERNAL (Internal)
Locking *: None
ROX (template) *: commonapp/cprcust/cprcust
Default Locale *: User locale
Show Currency Symbol: Never

Actions
Before: None ( )
After: None ( )

Page Setup
Orientation: Landscape
Paper Size: Letter
Get from Report
Margins (inches): Left, Right, Top, Bottom
Always Use Design Settings

Apply | Ok | Cancel

7. Click the Extensibility Information tab. This tab contains the metadata regarding a report you want to extend. You can change the title and status text shown on the report parameter screen.

   If you are going to change the report content and layout itself (which is usually the case since you are extending the report), you also specify the new template name with the path.

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

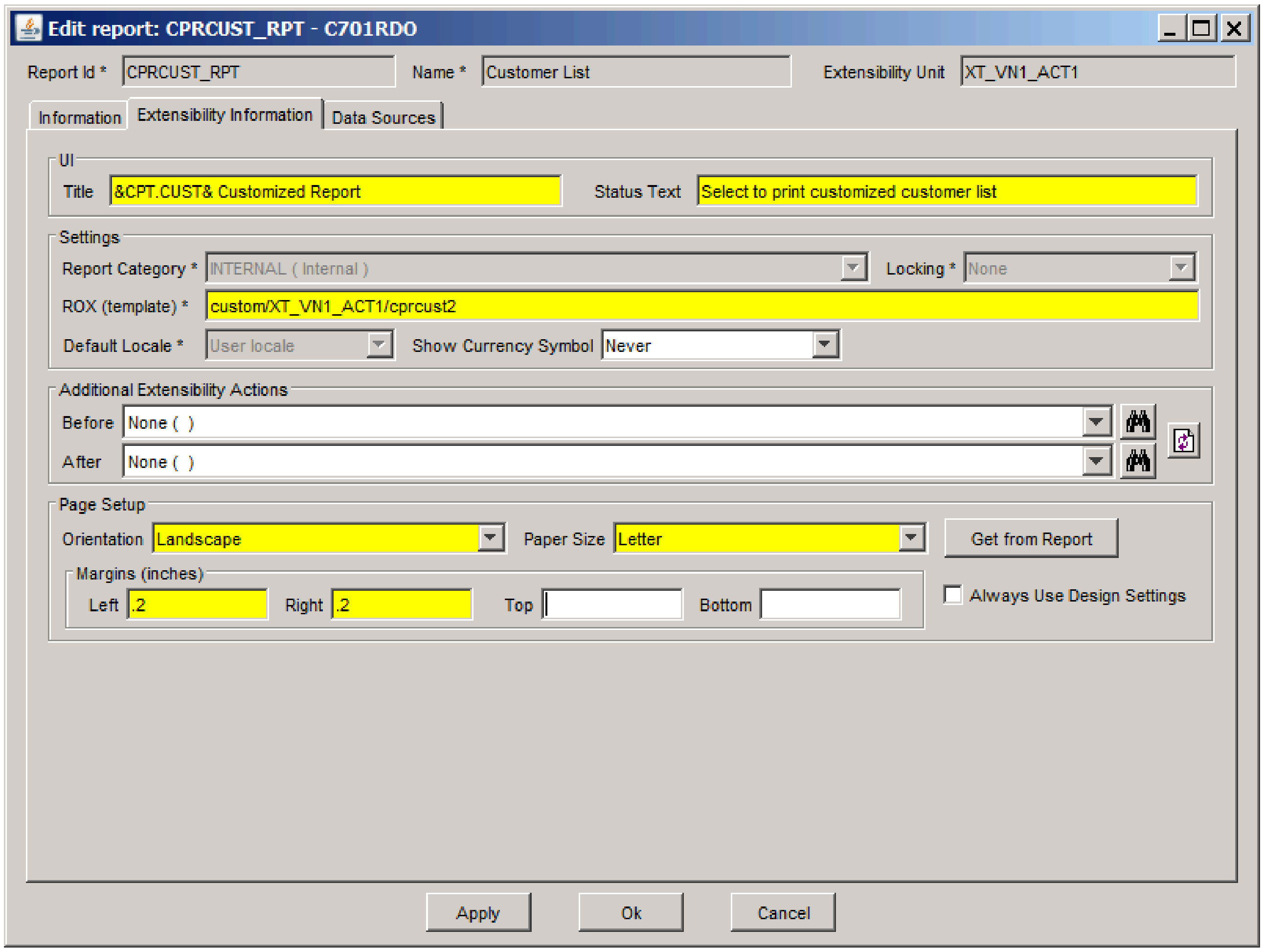
   The path must start with the folder named custom, then a subfolder with the Extensibility Project Id, and then the report template filename (without extension).

   In the example below, the path is custom\XT_VN1_ACT1 and the report template file name is cprcust2.

> **Attention:** Refer to the Costpoint Extensibility Designer Report Guide on how to copy an existing report template and modify it using Eclipse IDE for BIRT.

At run time, the server will expect the report template to be at the report root folder (which is C:\deltek\costpoint\82\applications\birt\report\bin) and the subfolder custom\XT_VN1_ACT1.

You can place the report template files in subfolders under the project folder. If you do so, you will need to create the same subfolder structure and place the report template files in corresponding folders before packaging the Extensibility project using the Extensibility Project Packager. For example, assume that you have entered custom/XT_VN1_ACT1/proj/mypjrbasic as ROX file name. In this case, you would need to create a subfolder \rpt/proj under your main project folder and place the mypjrbasic.rptdesign file in this folder before packaging your project.


Edit report: CPRCUST_RPT - C701RDO

Report Id * CPRCUST_RPT
Name * Customer List
Extensibility Unit XT_VN1_ACT1

UI
Title &CPT.CUST& Customized Report
Status Text Select to print customized customer list

Settings
Report Category * INTERNAL (Internal)
Locking * None
ROX (template) * custom/XT_VN1_ACT1/cprcust2
Default Locale * User locale
Show Currency Symbol Never

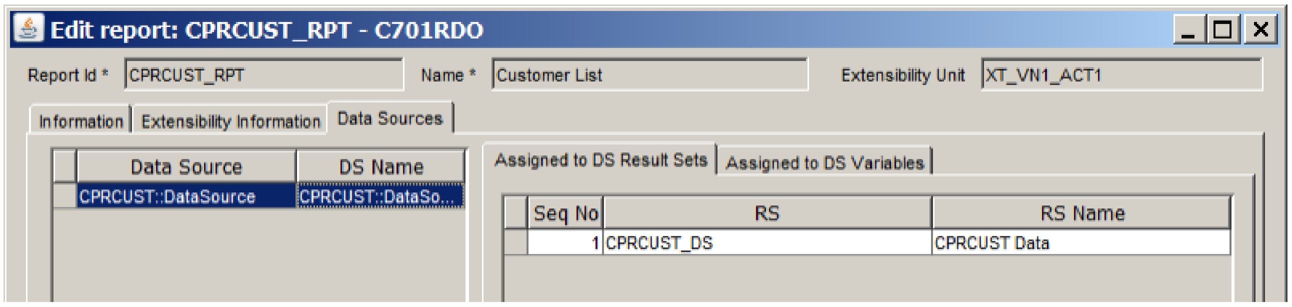
Additional Extensibility Actions
Before None ( )
After None ( )

Page Setup
Orientation Landscape
Paper Size Letter
Margins (inches)
Left .2
Right .2
Top
Bottom
Always Use Design Settings

- **Additional Extensibility Actions**: You can also select to run **Before** and/or **After** actions if you need to run some special process before or after the report is generated. These are available for new actions only (not extended actions) that you have created for this Extensibility Unit. They will be run before the report is formed (Before Action) and/or after the report is completed (After Action). These actions are executed in addition to the standard actions that are set in the Information tab for this report.

- **Page Setup**: You can also change the default page setup such as **Orientation**, **Paper Size**, and/or **Margins**. These are defaults only and will show up as defaults in the Page Setup option dialog box when you print the report. These settings should match how the report template is designed. At run time, you can override these defaults unless you select the **Always Use Design Settings** check box.
  If you already modified the report template and have changed these settings in the template, you can click the **Get from Report** button, and then select the modified report. The settings will be retrieved from the template and inserted into the appropriate fields.

8. Click the Data Sources tab to add new data fields for the report (if you are adding additional data fields to the report).


Edit report: CPRCUST_RPT - C701RDO
Information
Extensibility Information
Data Sources
Assigned to DS Result Sets
Assigned to DS Variables

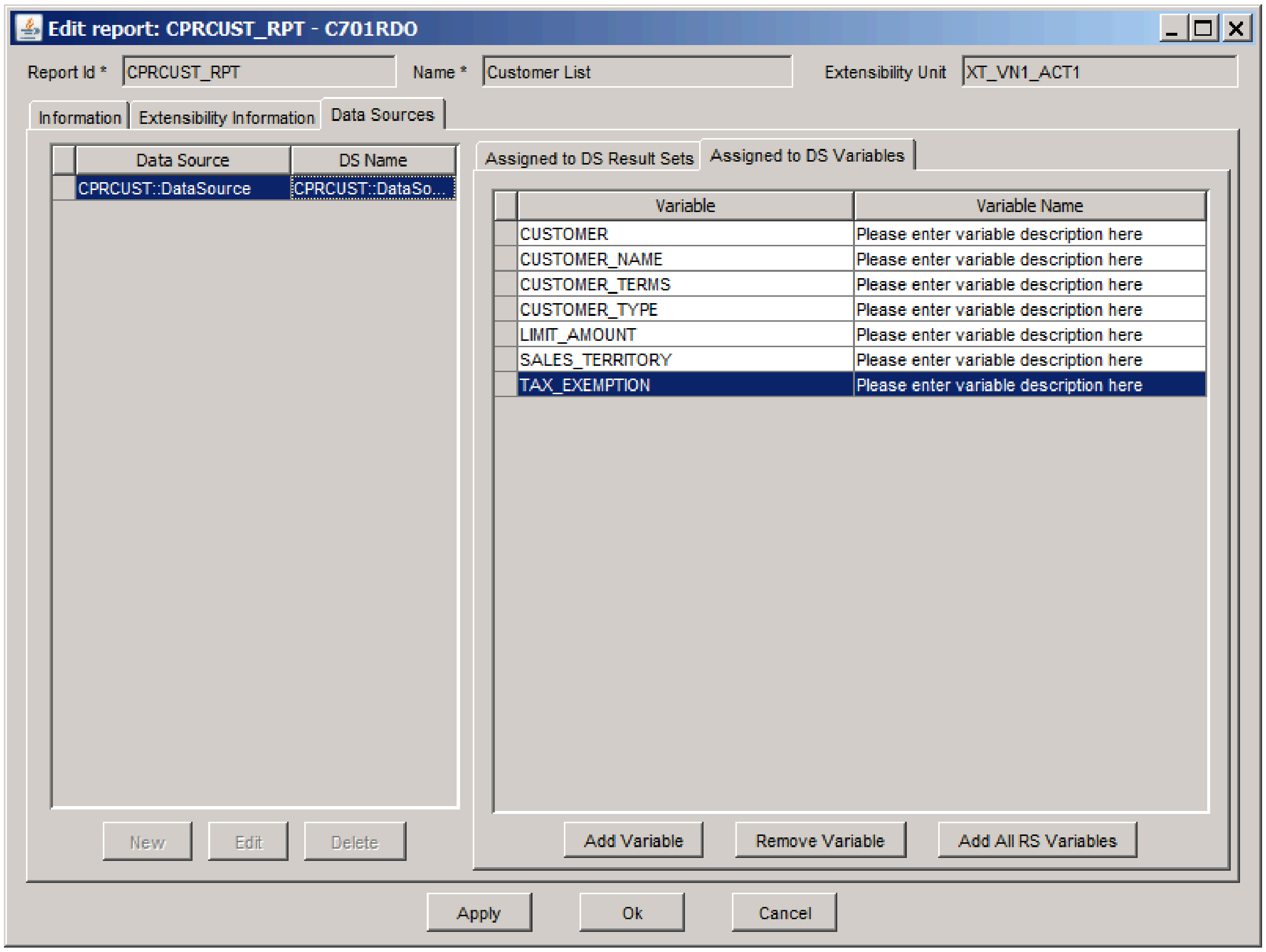
Report Id * CPRCUST_RPT
Name * Customer List
Extensibility Unit XT_VN1_ACT1

Data Source	DS Name
CPRCUST::DataSource	CPRCUST::DataSo...

Seq No	RS	RS Name
1	CPRCUST_DS	CPRCUST Data

On the right hand side under the Assigned to DS Result Sets tab, you will see the RS ID of the data source result set where the data for the report comes from. In the above example, the result set ID is CPRCUST_DS.

Click the Assigned to DS Variables tab to see the data fields from this data source result set that have been selected for the report.


Edit report: CPRCUST_RPT - C701RDO

Report Id * CPRCUST_RPT Name * Customer List Extensibility Unit XT_VN1_ACT1

Data Source	DS Name
CPRCUST::DataSource	CPRCUST::DataSo...

Variable	Variable Name
CUSTOMER	Please enter variable description here
CUSTOMER_NAME	Please enter variable description here
CUSTOMER_TERMS	Please enter variable description here
CUSTOMER_TYPE	Please enter variable description here
LIMIT_AMOUNT	Please enter variable description here
SALES_TERRITORY	Please enter variable description here
TAX_EXEMPTION	Please enter variable description here

You can use the **Add Variable** button to add additional fields from the data source result set. If you have already added new fields to the data source result set (by extending the data source result set), you can select and add them here. If not, you need to do that first, and then come back to add the variables.

At this point, you have completed extending the Report Definition. Refer to other sections of this document on how to extend a result set (to add additional fields) for a data source result set, or how to add an action if you need to include additional logic.

You also need to modify the report template to add the additional fields or to modify the layout, style, or formatting as desired.

Attention: For more information on how to modify the report template to add the additional fields or to modify the layout, style, or formatting, refer to the Costpoint 8.2 Extensibility Designer Report Guide.

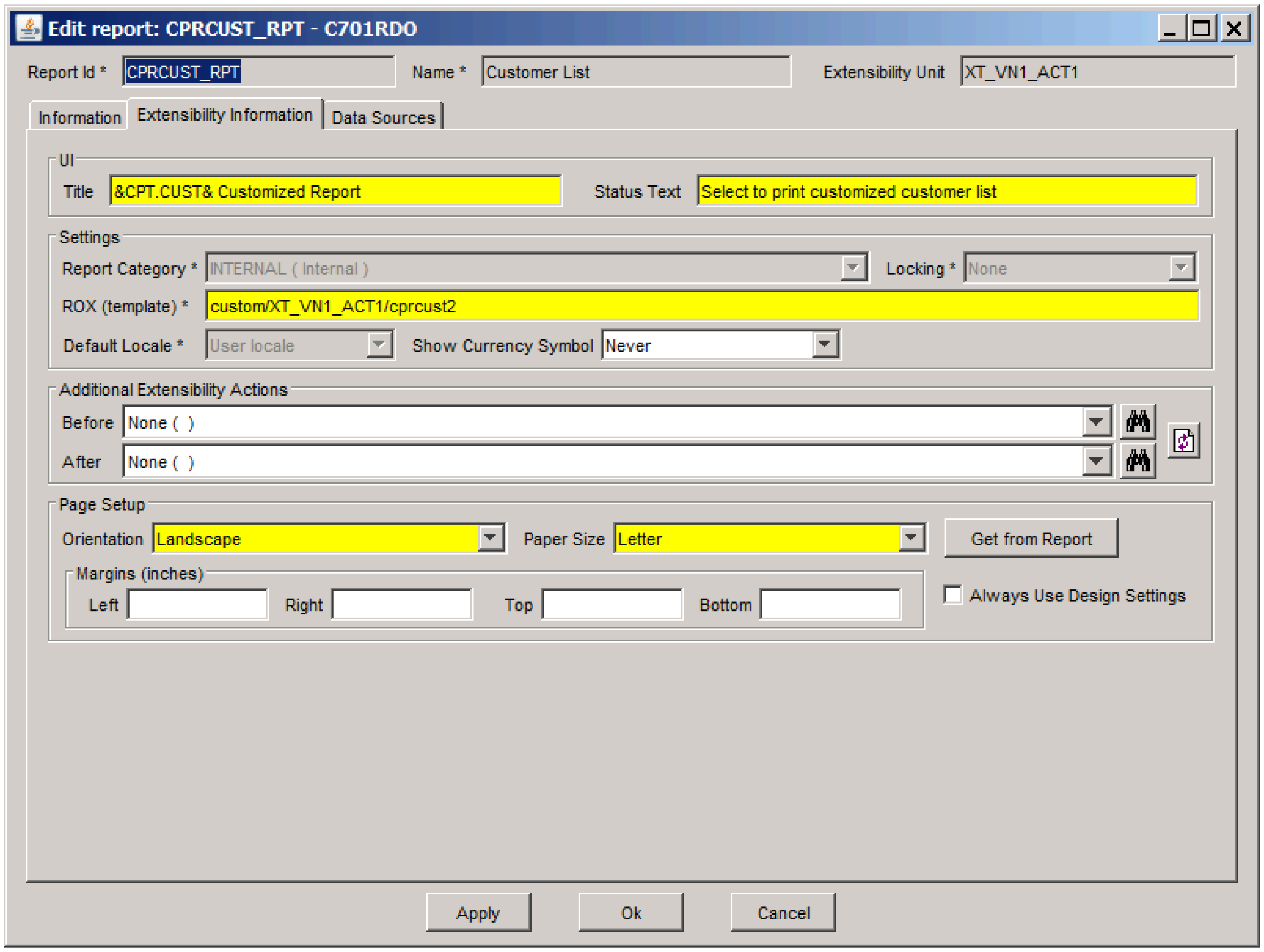
Deltek

Edit an Existing Report

To edit an extended report:

1. Expand the **Extensibility Projects** folder, the project to modify, the **Extensibility Units** folder, the unit to modify, and the **Reports** folder.

2. Select the report to modify, and then click **Extensions » Open Extension** or click the **Open Extension** icon. The **Edit report** dialog box displays.


Edit report: CPRCUST_RPT - C701RDO

Report Id * CPRCUST_RPT
Name * Customer List
Extensibility Unit XT_VN1_ACT1

UI
Title &CPT.CUST& Customized Report
Status Text Select to print customized customer list

Settings
Report Category * INTERNAL (Internal)
Locking * None
ROX (template) * custom/XT_VN1_ACT1/cprcust2
Default Locale * User locale
Show Currency Symbol Never

Additional Extensibility Actions
Before None ( )
After None ( )

Page Setup
Orientation Landscape
Paper Size Letter
Margins (inches)
Left
Right
Top
Bottom
Always Use Design Settings

Delete an Extension of the Report

To delete an extended report:

1. Expand the **Extensibility Projects** folder, the project to modify, the **Extensibility Units** folder, the unit to modify, and the **Reports** folder.

2. Select the Report to modify, and then click **Extensions » Delete Extension** or click the **Delete Extension** icon.

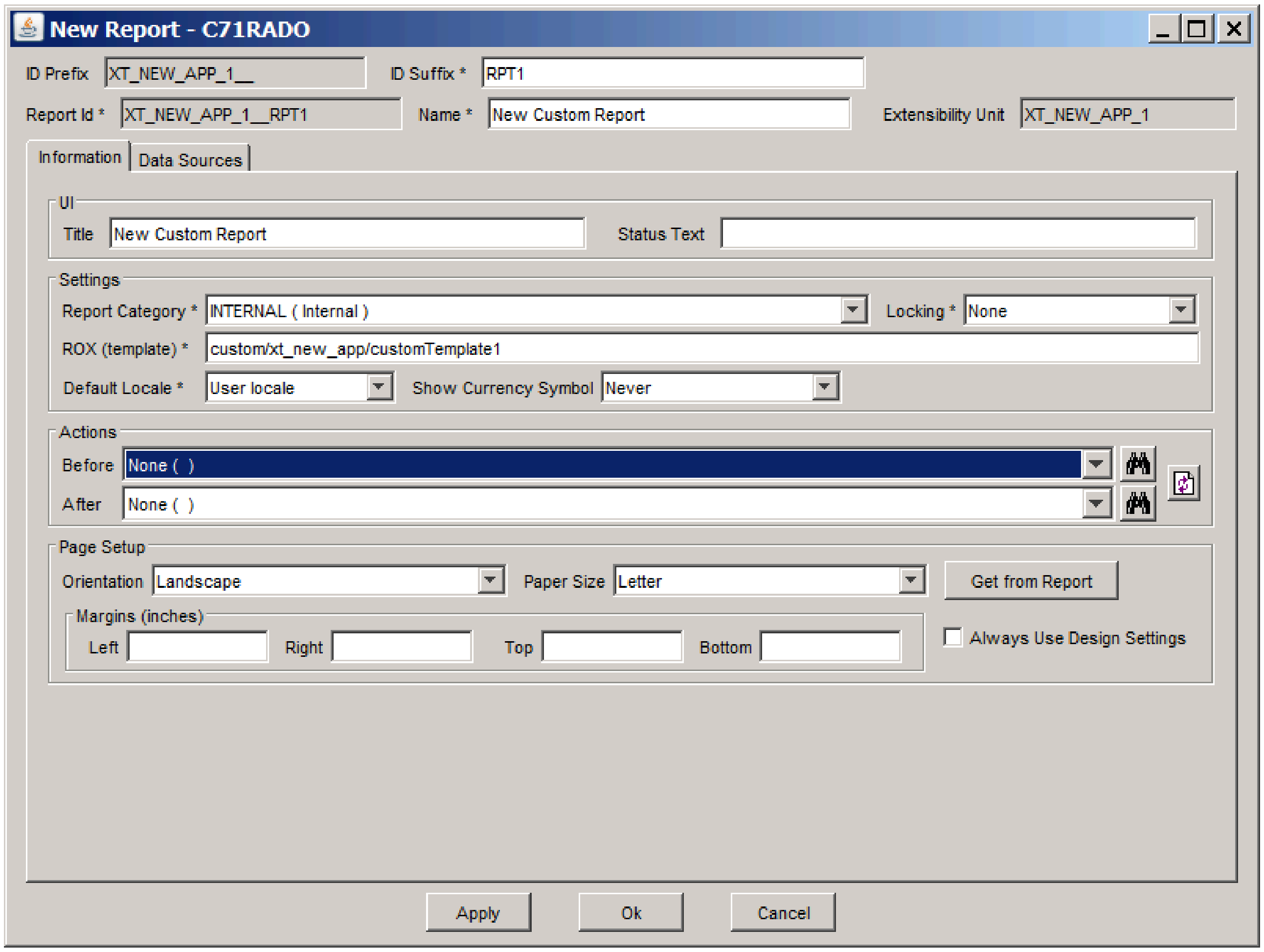
To create a new Report:

1. On the **Extensions** menu, click **Object » New » Report**.

   Alternately, click the **New** icon, and click **Report**.

2. On the Select Extension Project for new Report dialog box, select the project, and click **Next**.

3. On the Select Extension Unit for new Report dialog box, select the unit, and click **Next**.



New Report - C71RADO

ID Prefix	XT_NEW_APP_1__	ID Suffix *	RPT1		
Report Id *	XT_NEW_APP_1__RPT1	Name *	New Custom Report	Extensibility Unit	XT_NEW_APP_1

Information | Data Sources

UI
- Title: New Custom Report
- Status Text:

Settings
- Report Category *: INTERNAL (Internal)
- Locking *: None
- ROX (template) *: custom/xt_new_app/customTemplate1
- Default Locale *: User locale
- Show Currency Symbol: Never

Actions
- Before: None ( )
- After: None ( )

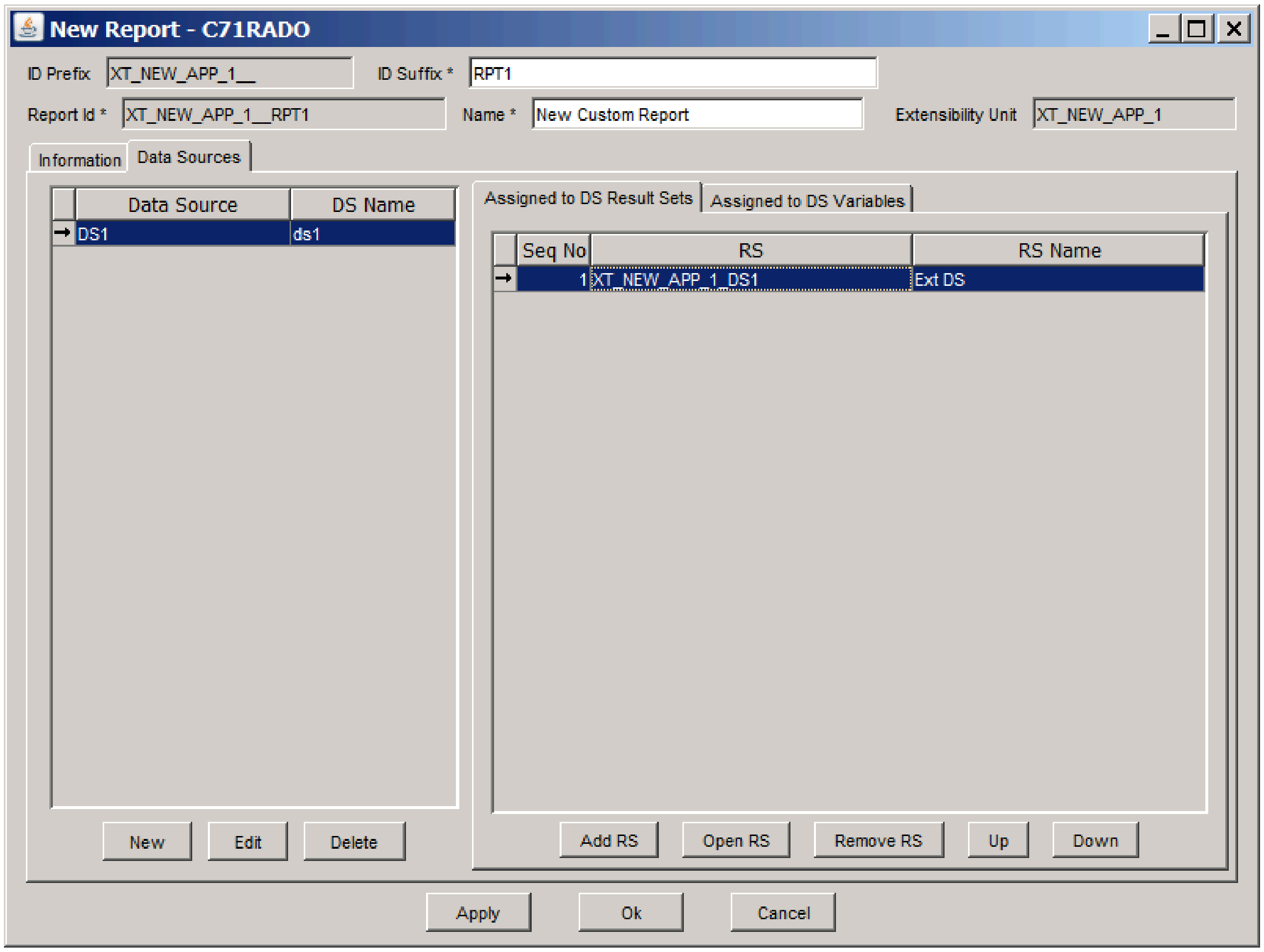
Page Setup
- Orientation: Landscape
- Paper Size: Letter
- Margins (inches): Left, Right, Top, Bottom
- Always Use Design Settings

1. Enter the **Report ID Suffix** (as all new report IDs in the given Unit starts with Unit ID) and the name of the Report.

2. Select the **Report Category**. Report Archiving policies may depend on the Report category in your organization.

3. Enter the **ROX (template)** name for the new Report.

4. On the **Data Sources** tab, create (depending on your needs) one or more Data Sources.



New Report - C71RADO

ID Prefix: XT_NEW_APP_1__
ID Suffix *: RPT1
Report Id *: XT_NEW_APP_1__RPT1
Name *: New Custom Report
Extensibility Unit: XT_NEW_APP_1

Data Source	DS Name
DS1	ds1

Seq No	RS	RS Name
1	XT_NEW_APP_1_DS1	Ext DS

5. To each **Data Source**, link the DS Result Set(s) that will be feeding data into the report.

6. On the Assigned to DS Variables tab for each Data Source, enter all the variables that will be used in the report template and are coming from assigned Result Sets.

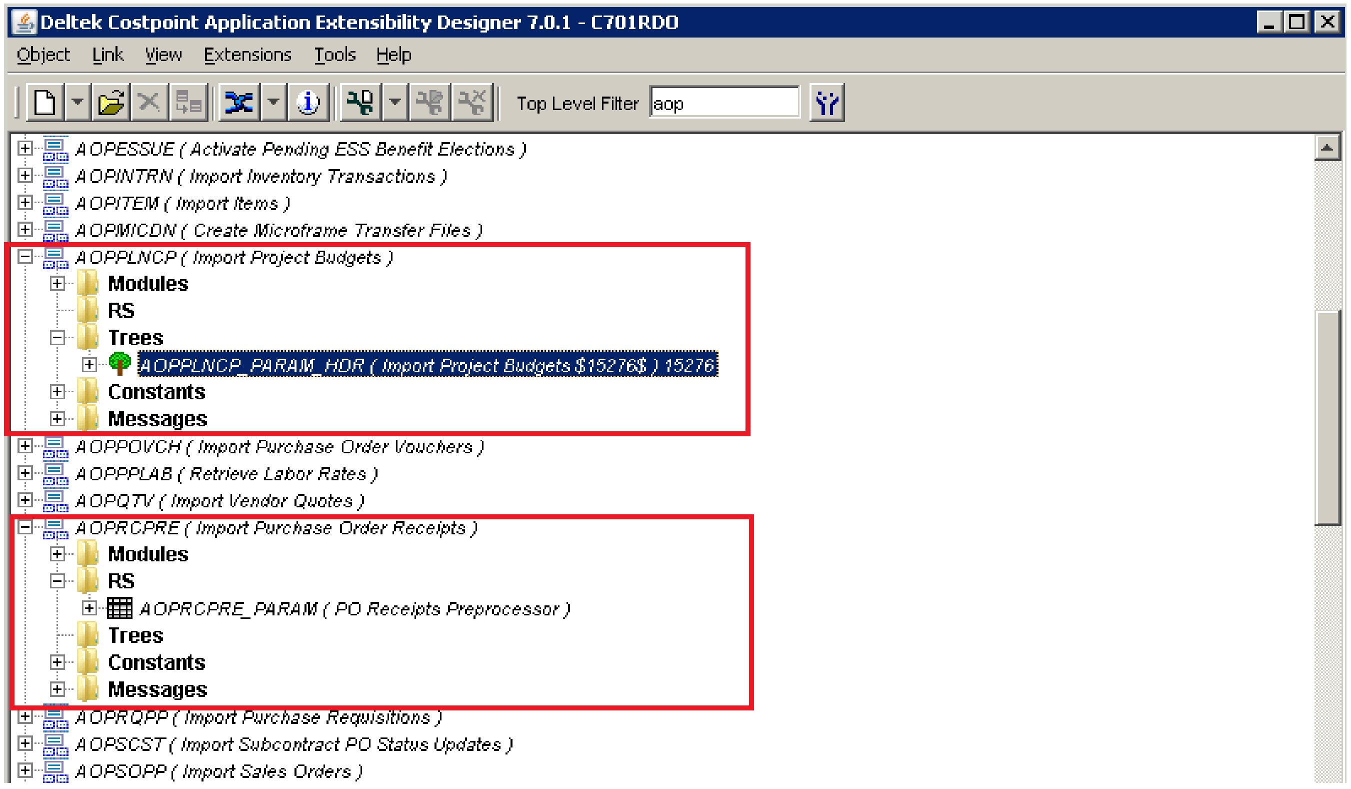
> **Attention:** For more information on creating new custom reports or extending standard reports, please see the Deltek Costpoint 8.2 Extensibility Designer Report Guide.

Assign Report to Result Set/Tree

After creating a report, it needs to be assigned to a result set that belongs to the application. If the application contains more than one result set (a tree), then assign it to the tree. If the application contains only one result set, then assign it to the result set.

To find out if an application contains a single result set or a tree, browse the application at the top level. Expand the RS or the Trees at the next level. Only one of them should exist. That is, an application can be assigned to either a result set or a tree at the top level, not both. Note the name of the result set or the tree. If it is a tree, also note the tree number. When assigning the action to the tree, the tree number must be exact.

In the following example, the Import Project Budgets application is assigned to a Tree whereas the Import Purchase Order Voucher application is assigned to a single result set.


Deltek Costpoint Application Extensibility Designer 7.0.1 - C701RDO
Object Link View Extensions Tools Help
Top Level Filter aop

```
AOPESSUE ( Activate Pending ESS Benefit Elections )
AOPINTRN ( Import Inventory Transactions )
AOPITEM ( Import Items )
AOPMICDN ( Create Microframe Transfer Files )
AOPPLNCP ( Import Project Budgets )
    Modules
    RS
    Trees
        AOPPLNCP_PARAM_HDR ( Import Project Budgets $15276$ ) 15276
    Constants
    Messages
AOPPOVCH ( Import Purchase Order Vouchers )
AOPPPLAB ( Retrieve Labor Rates )
AOPQTV ( Import Vendor Quotes )
AOPRCPRE ( Import Purchase Order Receipts )
    Modules
    RS
        AOPRCPRE_PARAM ( PO Receipts Preprocessor )
    Trees
    Constants
    Messages
AOPRQPP ( Import Purchase Requisitions )
AOPSCST ( Import Subcontract PO Status Updates )
AOPSOPP ( Import Sales Orders )
```

To assign a new report to a result set or a tree:

1. Select the report under the **Project » Unit**.

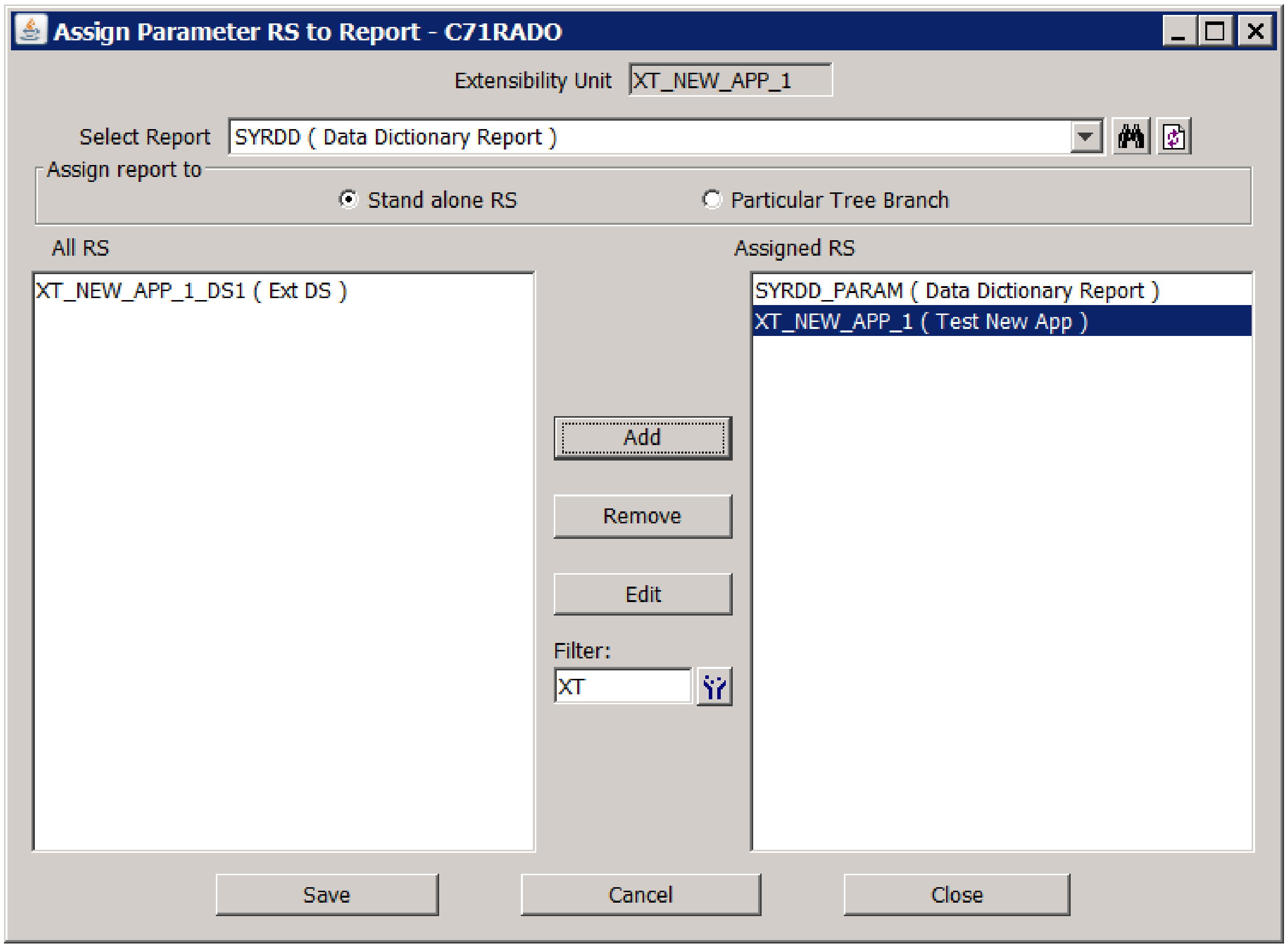
2. From the menu bar, click **Link » Assign » RS** (or **Tree**).

3. On the Assign RS to Report dialog box, select the stand-alone result set or tree branch as determined earlier.

   The left side of the screen displays all the result sets or trees for all applications.

4. Use the **Filter** field in the middle to filter out the name.

5. Select the result set or tree, and click **Add**.



Assign Parameter RS to Report - C71RADO

Extensibility Unit: XT_NEW_APP_1

Select Report: SYRDD (Data Dictionary Report)

Assign report to: [✓ Stand alone RS] Particular Tree Branch

All RS:
XT_NEW_APP_1_DS1 (Ext DS)

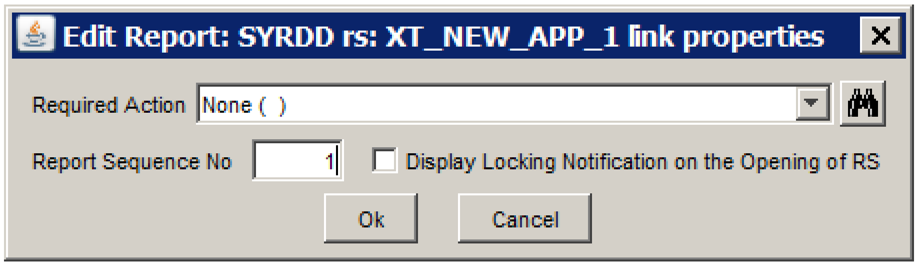
Assigned RS:
SYRDD_PARAM (Data Dictionary Report)
XT_NEW_APP_1 (Test New App)

Filter: XT

6. Click **Edit**.

7. On the Edit Report link properties dialog box, use Report Sequence No to change the order in which the assigned reports display when invoked by end users.

   Generally, you can leave all existing settings in here as default.

   Required Action: None ( )

   Report Sequence No: 1

   ☐ Display Locking Notification on the Opening of RS

This section describes how you can extend existing standard Costpoint applications or create new custom ones.

Extending Standard Applications

When extending existing applications, the only thing you can customize in the Application Object itself is the Application Title that end users see in the product menu.

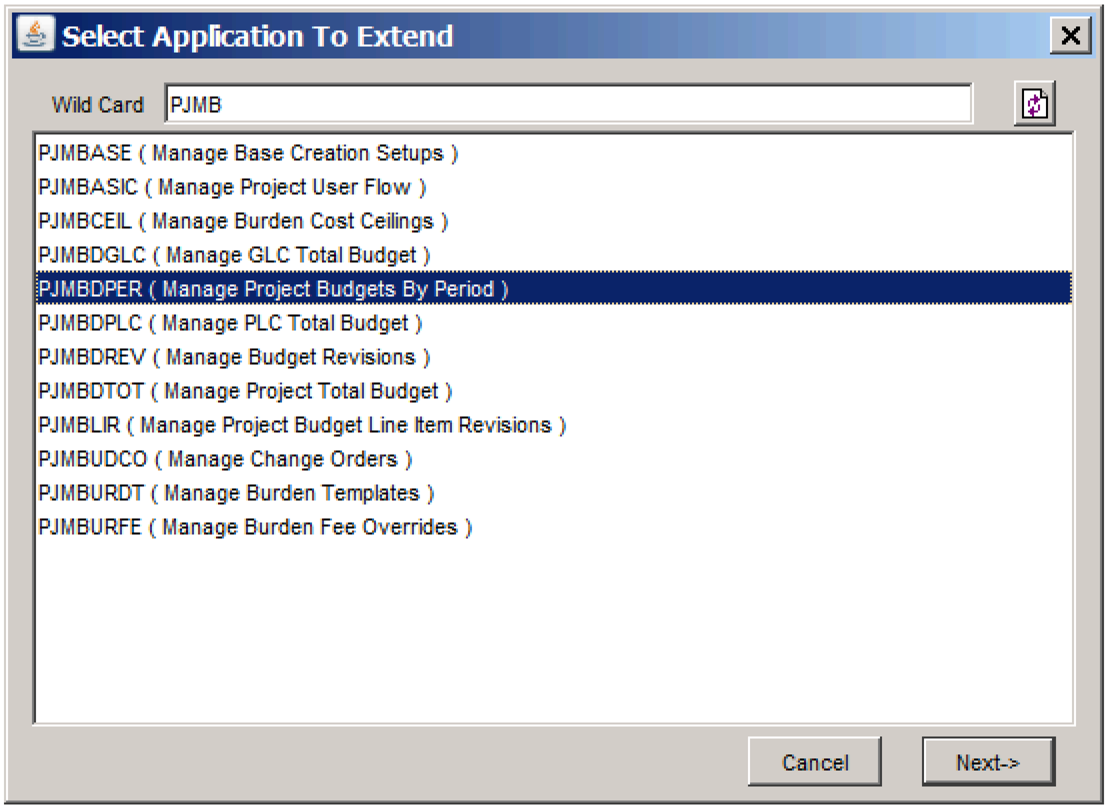
To extend an Application:

1. On the **Extensions** menu, click **Object » Extend » App**.

2. On the Select Extension Project for App dialog box, select the project, and click **Next**.

3. On the Select Extension Unit for App dialog box, select the unit, and click **Next**

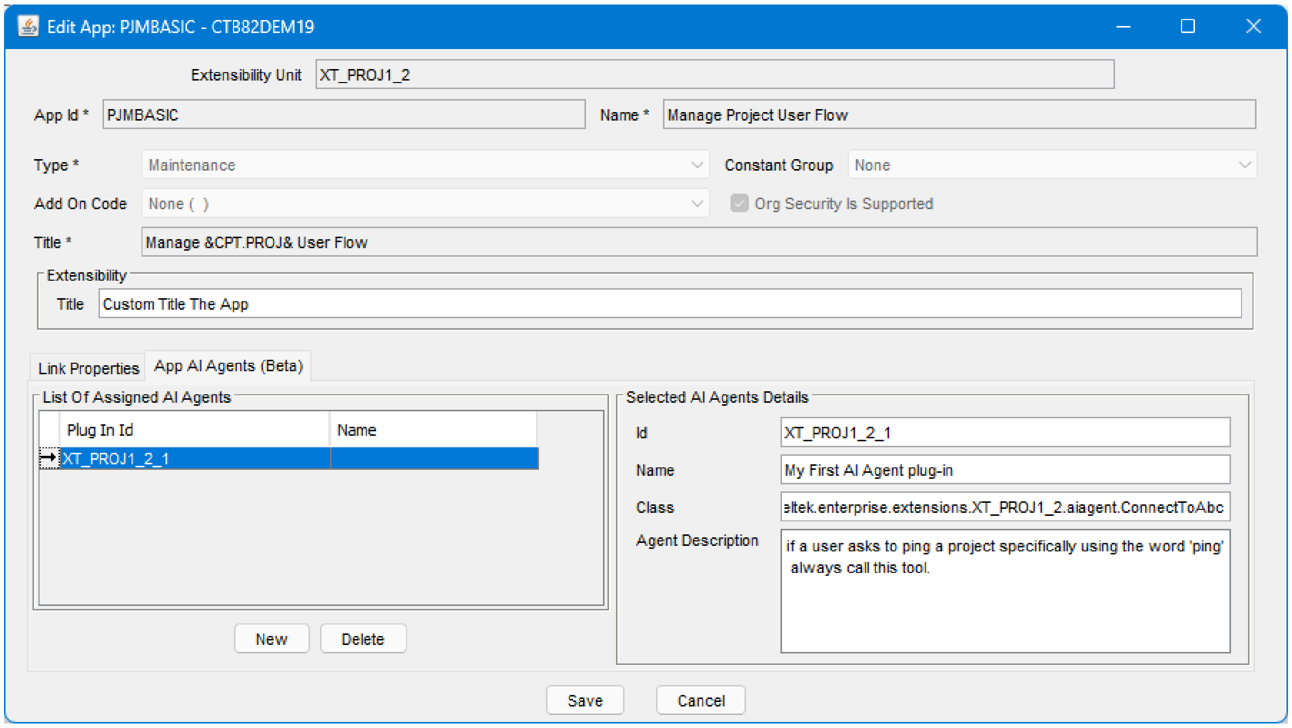
4. On Select Application To Extend dialog box, select the Application you want to extend, and click **Next**.


Select Application To Extend

Wild Card PJMB

PJMBASE (Manage Base Creation Setups)
PJMBASIC (Manage Project User Flow)
PJMBCEIL (Manage Burden Cost Ceilings)
PJMBDGLC (Manage GLC Total Budget)
PJMBDPER (Manage Project Budgets By Period)
PJMBDPLC (Manage PLC Total Budget)
PJMBDREV (Manage Budget Revisions)
PJMBDTOT (Manage Project Total Budget)
PJMBLIR (Manage Project Budget Line Item Revisions)
PJMBUDCO (Manage Change Orders)
PJMBURDT (Manage Burden Templates)
PJMBURFE (Manage Burden Fee Overrides)

5. On the Edit App dialog box, enter your custom app title, and click **Save**.


Edit App: PJMBASIC - CTB82DEM19

Extensibility Unit XT_PROJ1_2
App Id * PJMBASIC
Name * Manage Project User Flow
Type * Maintenance
Constant Group None
Add On Code None ( )
Org Security Is Supported
Title * Manage &CPT.PROJ& User Flow

Extensibility
Title Custom Title The App

List Of Assigned AI Agents

Plug In Id	Name
XT_PROJ1_2_1	

Selected AI Agents Details

Id XT_PROJ1_2_1
Name My First AI Agent plug-in
Class eltek.enterprise.extensions.XT_PROJ1_2.aiagent.ConnectToAbc
Agent Description if a user asks to ping a project specifically using the word 'ping' always call this tool.

6. In Costpoint 2026.2 and higher versions, you can register custom App AI Agents. For more information, refer to the Costpoint 8.2 Extensibility Designer Coding Guide.

Create a New Custom Application

In order to create a new application, you have to add a brand-new Application, create a single new result set or a new custom result set tree, and assign it to the new application. Unlike extending a standard application, a new application allows you complete control over the logic executed when a user invokes it.

To add a new application:

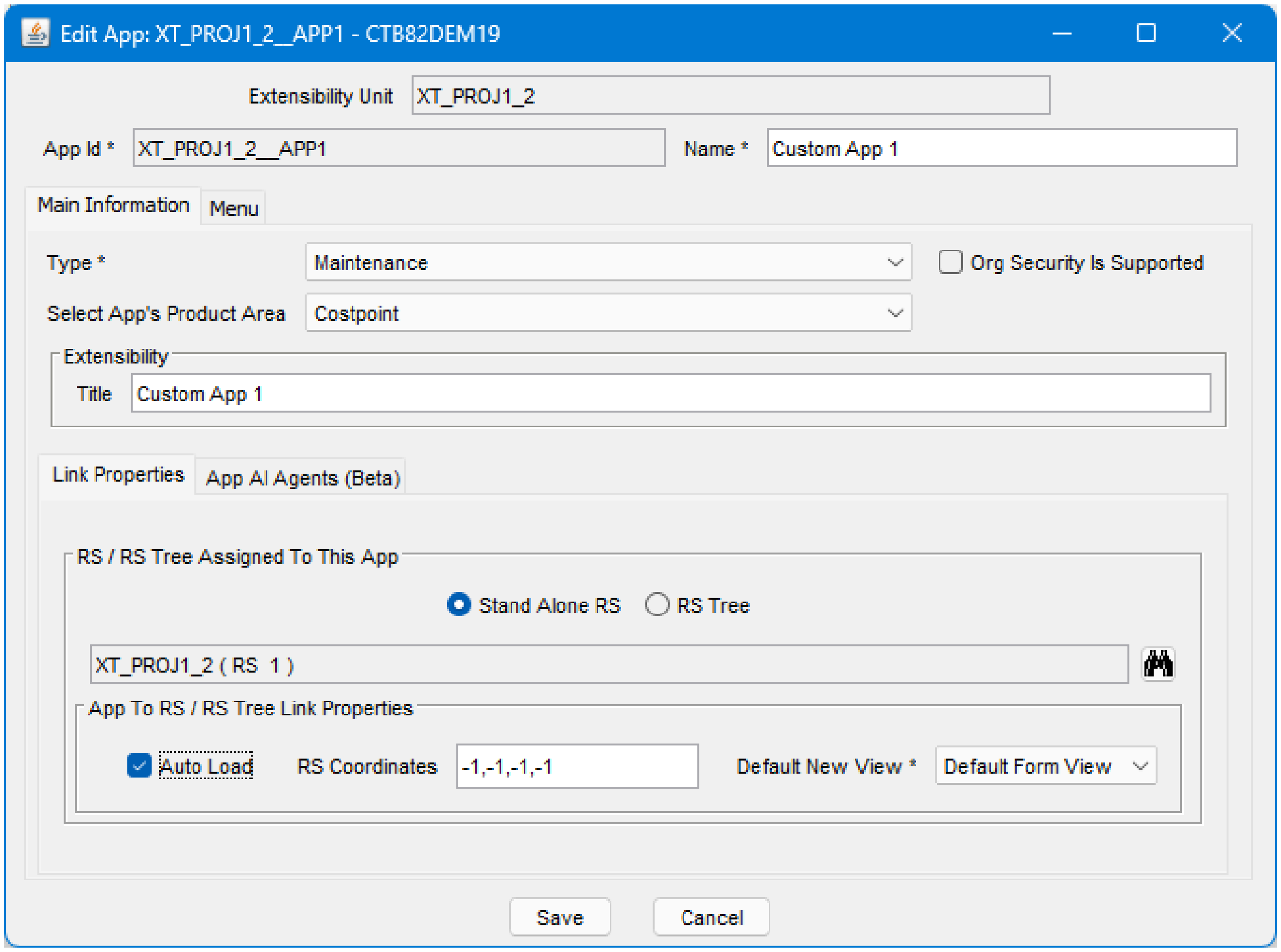
1. On the **Extensions** menu, click **Object » New » App**.

2. On the Select Extension Project for App dialog box, select the project, and click **Next**.

3. On the Select Extension Unit for App dialog box, select the unit, and click **Next**

4. On the New App dialog box, take the following actions:

- Enter the **App ID Suffix** (as all new application IDs in the given Unit start with Unit ID) and the name of the App.
- Select the appropriate **Application Type**. For example, if the purpose of this application is to enter new data, select **Maintenance**; if the application is designed to run a process, select **Processing**; and so on.
- Enter the application **Title.**
- Check if Org Security will be supported in this app or not. Selecting the **Org Security is Supported** check box will ensure that the new application will become available in SYMORGFN, where system administrators can turn on and off Org Security for applications


Edit App: XT_PROJ1_2__APP1 - CTB82DEM19

Extensibility Unit: XT_PROJ1_2

App Id *: XT_PROJ1_2__APP1

Name *: Custom App 1

Main Information | Menu

Type *: Maintenance

Org Security Is Supported: ☐

Select App's Product Area: Costpoint

Extensibility

Title: Custom App 1

Link Properties | App AI Agents (Beta)

RS / RS Tree Assigned To This App

[✓ Stand Alone RS] RS Tree

XT_PROJ1_2 (RS 1)

App To RS / RS Tree Link Properties

Auto Load: ✓

RS Coordinates: -1,-1,-1,-1

Default New View *: Default Form View

5. Select an RS or RS Tree to assign to the app.

6. Select an option in the **App To RS / RS Tree Link Properties** group box.

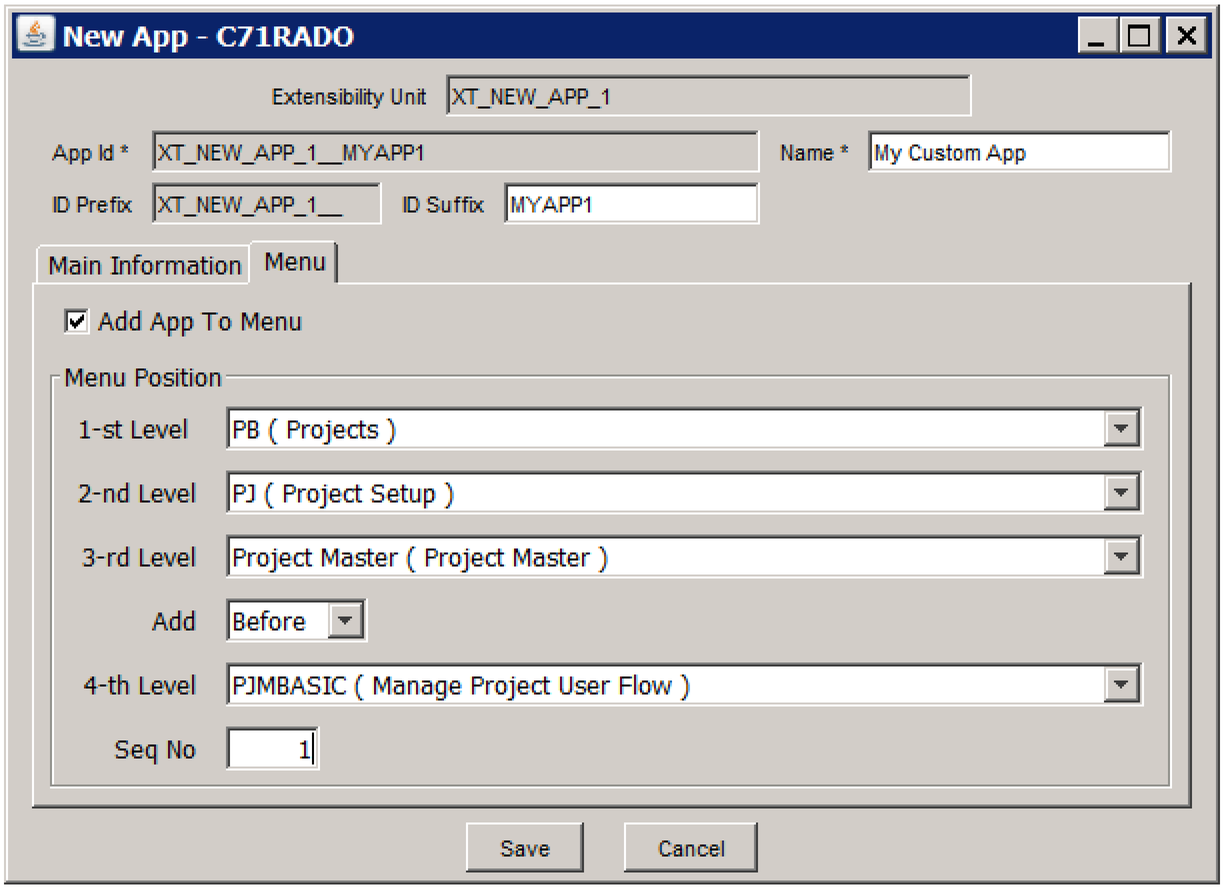
   You can specify if you want data to automatically load when the app opens, enter RS coordinates (for top level RS), and select a Default New View for the top level RS.

7. Click the Menu tab.

8. Select the position of the new application relative to the standard Costpoint applications.

   Call this the anchor app.

   If you are planning to add several custom applications before and/or after the same standard anchor app, you will need to specify the appropriate **Seq No**, since custom applications will be sorted according to the number.


New App - C71RADO

Extensibility Unit: XT_NEW_APP_1

App Id *: XT_NEW_APP_1__MYAPP1

Name *: My Custom App

ID Prefix: XT_NEW_APP_1__

ID Suffix: MYAPP1

☑ Add App To Menu

Menu Position

1-st Level	PB (Projects)
2-nd Level	PJ (Project Setup)
3-rd Level	Project Master (Project Master)
Add	Before
4-th Level	PJMBASIC (Manage Project User Flow)
Seq No	1

Note: For a custom application to display in the run-time menu, end users need to have at least read-only rights to both the standard anchor application and your custom application.

There is new functionality starting with Costpoint 8.2 that allows a developer to overwrite the standard app location to display the regular app higher in the menu on levels 2 or 3. If you have selected a regular application as an anchor for your custom application, a note displays in the tool next to **Seq No** field, and the Framework will move your custom app to the specified level to follow anchor app.

9. Click **Save**.

10. In Costpoint 2026.2 and higher versions, you can add custom AI agents to both new and existing apps. You register AI agents in the App AI Agents Tab. For more information, refer to the Costpoint 8.2 Extensibility Designer Coding Guide.

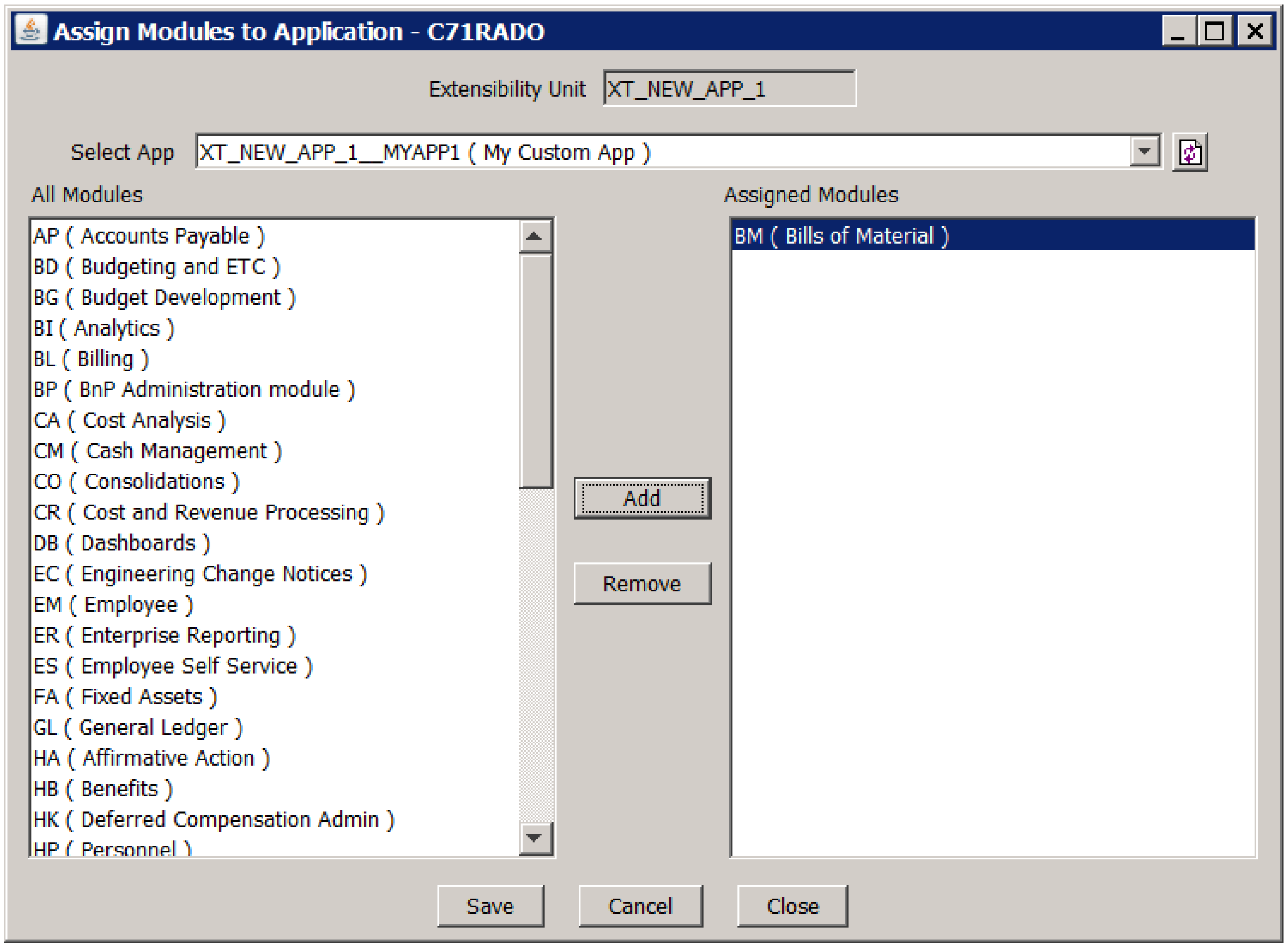
Assign an Application to Modules

Every new custom application should be assigned to at least one module so that module rights will be extended to this application.

To assign an application to a module:

1. Select the application in the list of custom applications in a given Extensibility Unit, and click **Menu** » **Link** » **Assign/Unassign** » **Modules**.

2. On the Assign Modules to Application dialog box, select the desired module(s) in the list of **All Modules**, and click **Add**.



Assign Modules to Application - C71RADO

Extensibility Unit: XT_NEW_APP_1

Select App: XT_NEW_APP_1__MYAPP1 (My Custom App)

All Modules:
AP (Accounts Payable)
BD (Budgeting and ETC)
BG (Budget Development)
BI (Analytics)
BL (Billing)
BP (BnP Administration module)
CA (Cost Analysis)
CM (Cash Management)
CO (Consolidations)
CR (Cost and Revenue Processing)
DB (Dashboards)
EC (Engineering Change Notices)
EM (Employee)
ER (Enterprise Reporting)
ES (Employee Self Service)
FA (Fixed Assets)
GL (General Ledger)
HA (Affirmative Action)
HB (Benefits)
HK (Deferred Compensation Admin)
HP (Personnel)

Assigned Modules:
BM (Bills of Material)

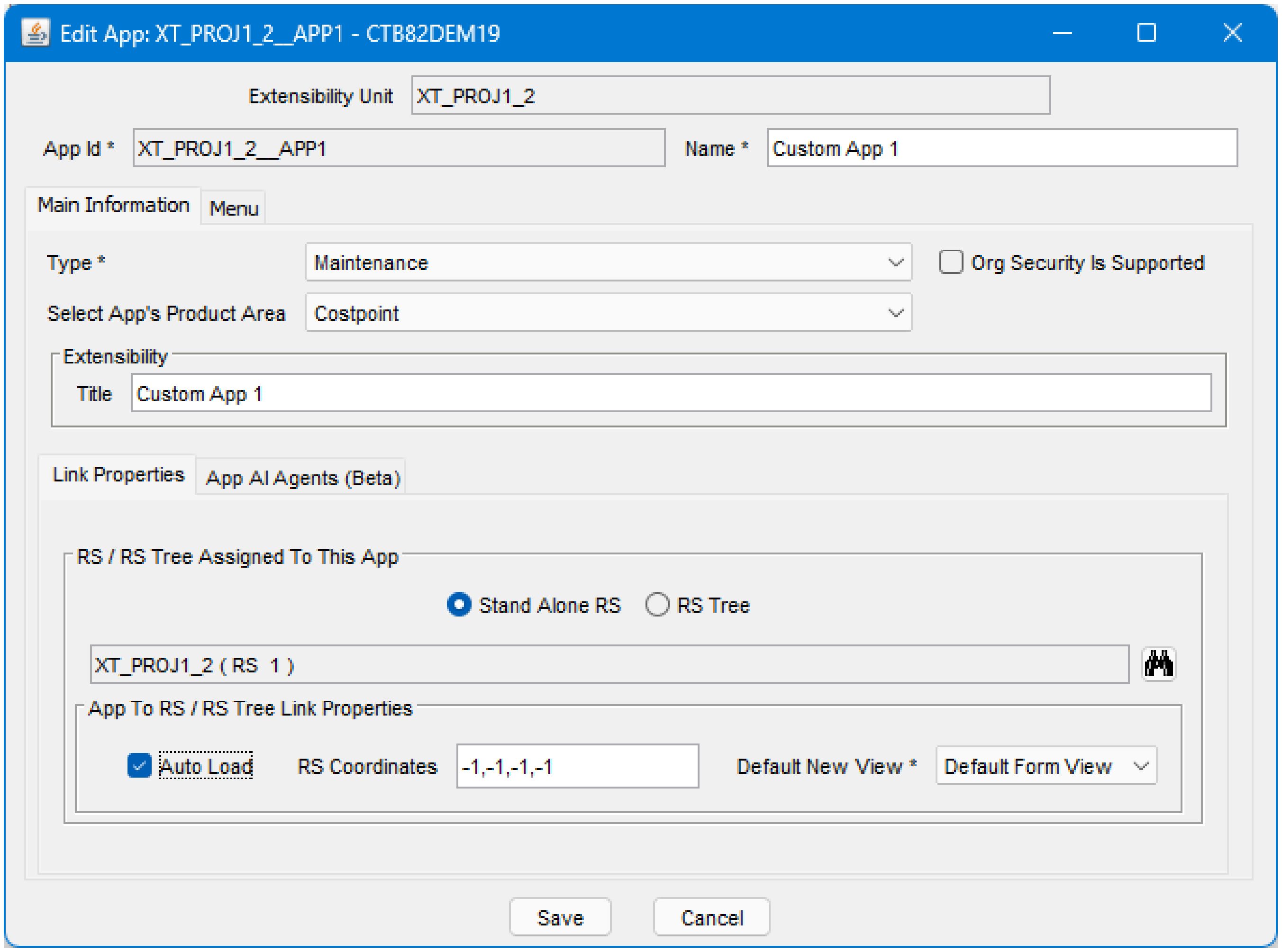
3. When done, click **Save**.

   To change the assignment of an application, follow the same steps to open the Assign Modules to Application dialog box, and click **Add** / **Remove** to achieve the desired assignments. Then save your changes.

Assign a Stand-Alone RS or RS Tree to a Custom Application

After creating a new application, you need to create other metadata objects (Result Set(s), RS Tree, Actions, Reports, Messages, and Constants) as needed. Then you assign them to each other and to the application.

You use the Assign RS to Application dialog box to assign a stand-alone RS or RS Tree to a custom application in the App Editor.



Edit App: XT_PROJ1_2__APP1 - CTB82DEM19

Extensibility Unit: XT_PROJ1_2

App Id *: XT_PROJ1_2__APP1
Name *: Custom App 1

Main Information | Menu

Type *: Maintenance
[ ] Org Security Is Supported
Select App's Product Area: Costpoint

Extensibility
Title: Custom App 1

Link Properties | App AI Agents (Beta)

RS / RS Tree Assigned To This App
[✓ Stand Alone RS] RS Tree
XT_PROJ1_2 (RS 1)

App To RS / RS Tree Link Properties
[✓] Auto Load
RS Coordinates: -1,-1,-1,-1
Default New View *: Default Form View

Note that you can link only one object from both lists since Costpoint supports only one top level result set per application.

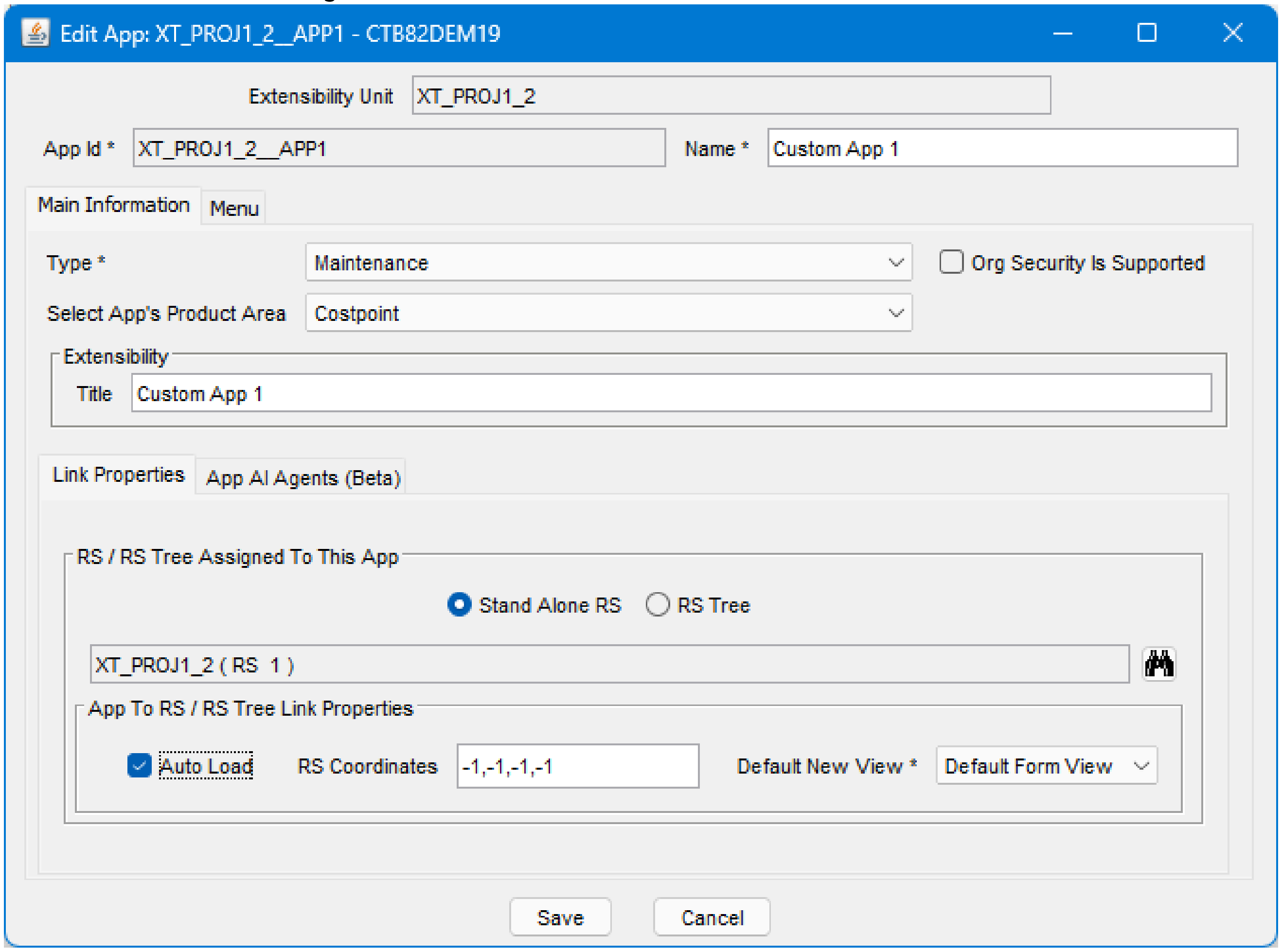
To assign a stand-alone result set or result set tree to the application:

1. In the Application Editor, do the following:
   - Select an option in the **RS / RS Tree Assigned To This App** group box.
   - Use the Lookup to select a custom RS or custom RS Tree for the application.
2. Click **Save**.

After assigning an RS and/or RS Tree, you can configure the linking properties between the application and the result set.

To configure the application and RS/RS Tree link properties:

- Select or clear **Auto Load** to indicate whether or not data is automatically loaded upon opening the application
- Enter **RS Coordinates**. The default is -1,-1,-1,-1. This commands the system to use the RS coordinates as it was laid out in the RS Form Designer.


Edit App: XT_PROJ1_2__APP1 - CTB82DEM19

Extensibility Unit: XT_PROJ1_2
App Id *: XT_PROJ1_2__APP1
Name *: Custom App 1

Type *: Maintenance
[ ] Org Security Is Supported
Select App's Product Area: Costpoint

Extensibility
Title: Custom App 1

RS / RS Tree Assigned To This App
[✓ Stand Alone RS] RS Tree
XT_PROJ1_2 (RS 1)

App To RS / RS Tree Link Properties
[✓] Auto Load
RS Coordinates: -1,-1,-1,-1
Default New View *: Default Form View

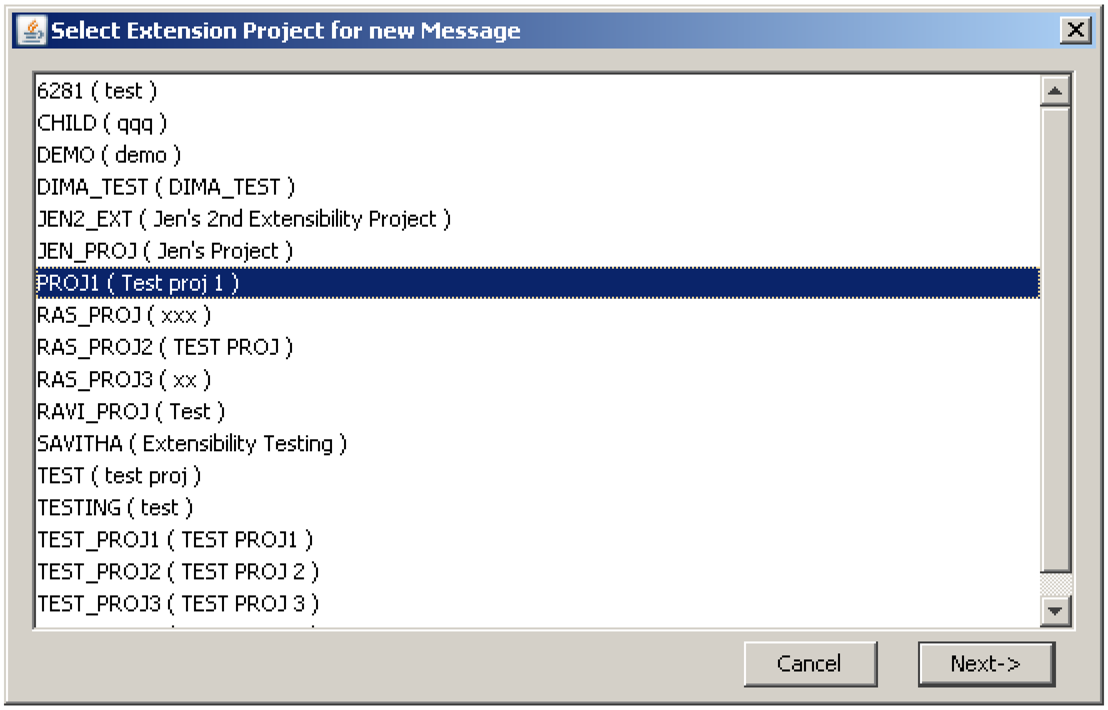
Message objects are text used by the system or application whenever there is a need to provide an information message, warning or validation error to the end users. As an extensibility developer, you can use any standard Costpoint messages or you can add new messages to the system to provide a new message to the end users. Messages can then be assigned in the Extensibility Designer to assign to an existing object validation or referred to in the Java code that will be implemented by you.

Adding a New Extensibility Message

To create a new Extensibility message:

1. Open the Deltek Costpoint Extensibility Designer.

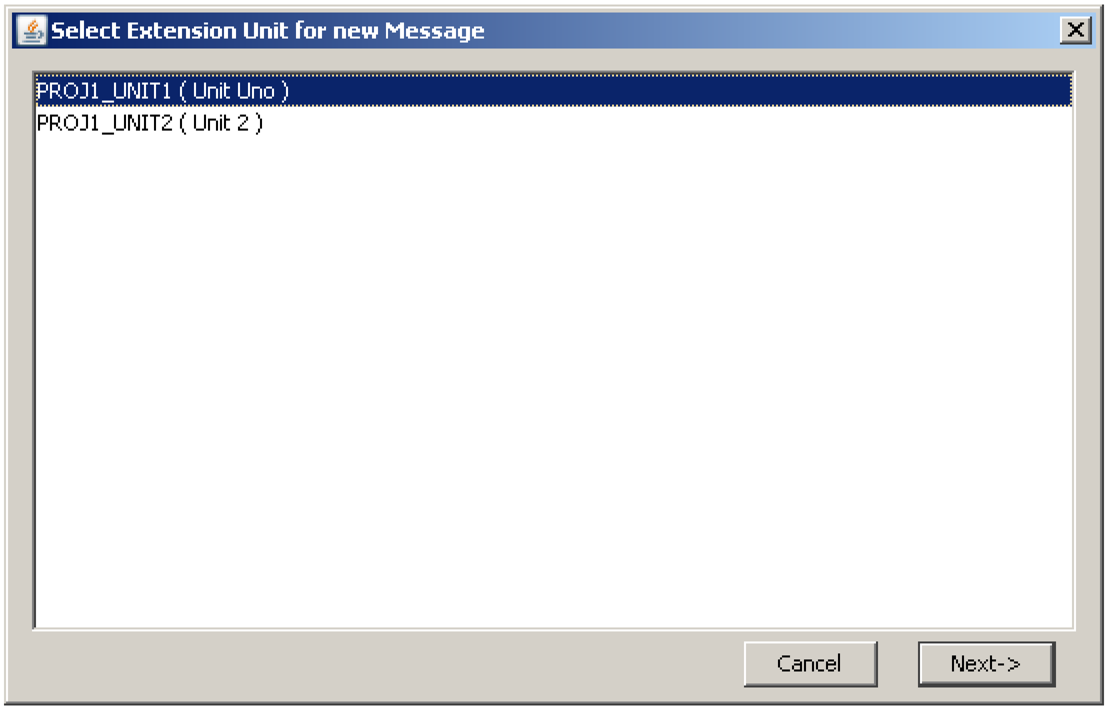
2. On the Object menu, click **New » Message**. The Select Extension Project for new Message dialog box displays.



Select Extension Project for new Message

6281 (test)
CHILD (qqq)
DEMO (demo)
DIMA_TEST (DIMA_TEST)
JEN2_EXT (Jen's 2nd Extensibility Project)
JEN_PROJ (Jen's Project)
PROJ1 (Test proj 1)
RAS_PROJ (xxx)
RAS_PROJ2 (TEST PROJ)
RAS_PROJ3 (xx)
RAVI_PROJ (Test)
SAVITHA (Extensibility Testing)
TEST (test proj)
TESTING (test)
TEST_PROJ1 (TEST PROJ1)
TEST_PROJ2 (TEST PROJ 2)
TEST_PROJ3 (TEST PROJ 3)

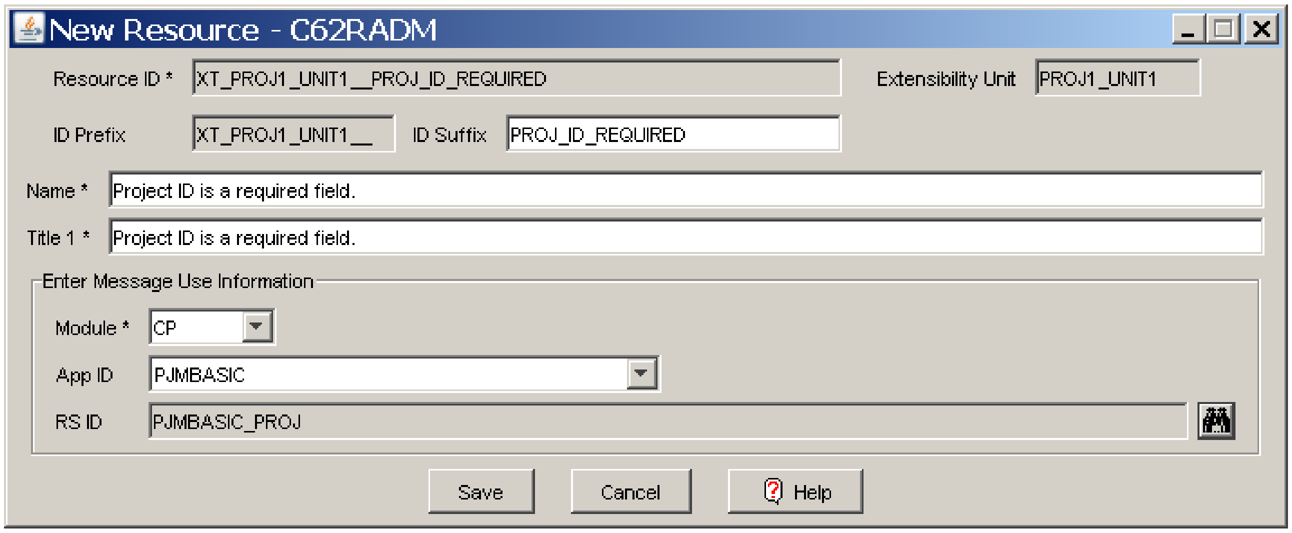
3. Click **Next**. The Select Extension Unit for new Message dialog box displays.



Select Extension Unit for new Message

PROJ1_UNIT1 (Unit Uno)
PROJ1_UNIT2 (Unit 2)

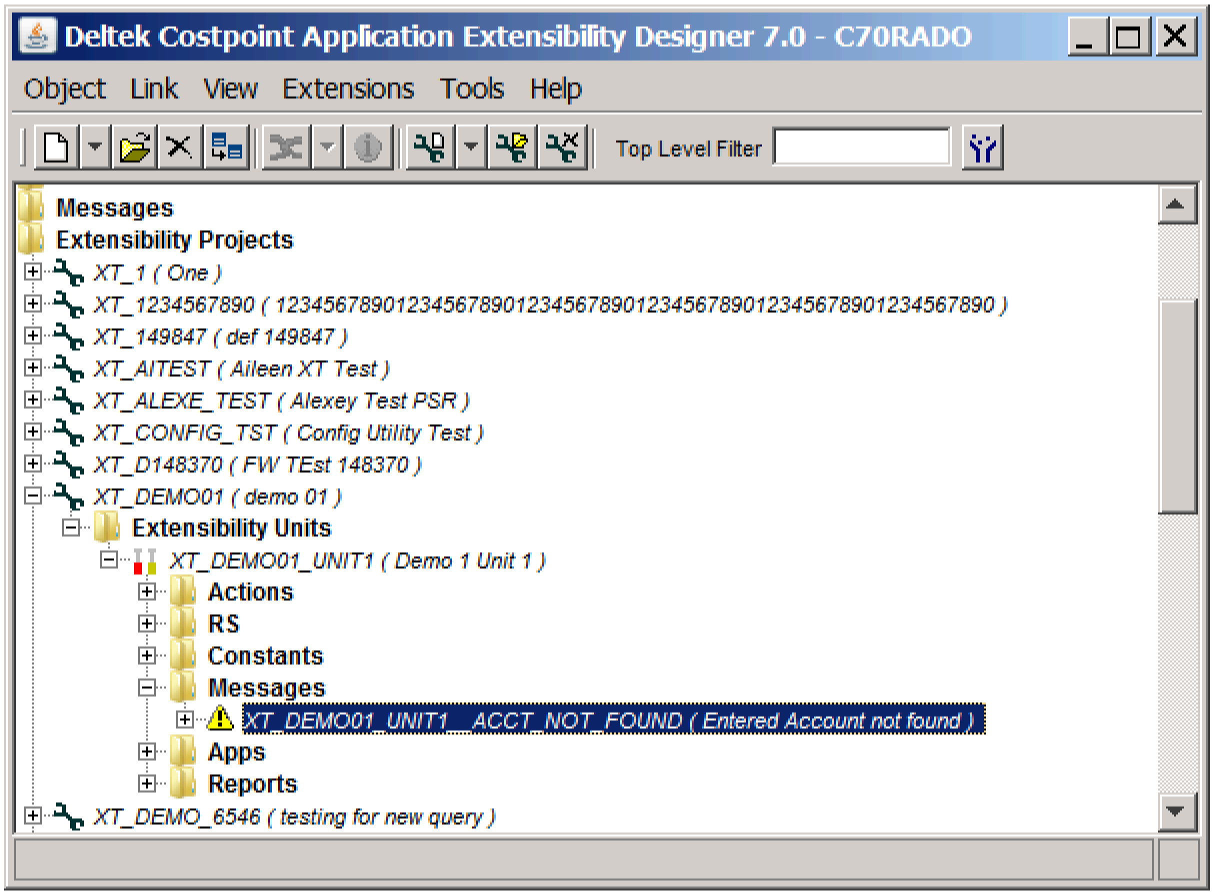
4. Click **Next**. The **New Resource** dialog box displays.


New Resource - C62RADM
Resource ID * XT_PROJ1_UNIT1__PROJ_ID_REQUIRED
Extensibility Unit PROJ1_UNIT1
ID Prefix XT_PROJ1_UNIT1__
ID Suffix PROJ_ID_REQUIRED
Name * Project ID is a required field.
Title 1 * Project ID is a required field.
Enter Message Use Information
Module * CP
App ID PJMBASIC
RS ID PJMBASIC_PROJ
Save
Cancel
Help

5. Complete the appropriate fields for your message.

   - **Resource ID**: This field automatically displays the following:
     - **T_PROJECT_ID_UNIT_ID__SUFFIX where PROJECT_ID**: This is the project identifier that was selected in step 2.
     - **UNIT_ID**: This is the unit identifier that are selected in step 3.
     - **SUFFIX**: This is the value entered in the **ID Suffix** field.
   - **Extensibility Unit**: This field automatically displays the extensibility unit being used.
   - **ID Prefix**: This field automatically displays **XT_PROJECT_ID_UNIT_ID__**, where **PROJECT_ID** is the project identifier that was selected in step 2 and **UNIT_ID** is the unit identifier that are selected in step 3.
   - **ID Suffix**: Enter a value to assign an identifier to the new message.
   - **Name**: Enter a description of the new message. This is a required field. Name can be different from Title 1 (which is a text of the message itself) and is used to help searching for particular message.
   - **Title 1**: Enter the text of the message to be displayed. This will automatically be populated with the text that is entered in the **Name** field, but it can be edited to be different text. This is a required field. You can use special syntax inside the message. For more details click on **Help** button.
   - **Module**: Select the module from this drop-down list for which the message will be used. This is a required field.
   - **App ID**: This drop-down list is populated based on the value that is selected in the Module drop-down list. Select the appropriate application identifier for the new message. This is an optional field.
   - **RS ID**: Click the binoculars icon to select the appropriate result set for the new message. This is an optional field.

6. Click **Save**. The **Message ID** and **Name** will display in the **Messages** folder within the appropriate **Extensibility Unit**.


Deltek Costpoint Application Extensibility Designer 7.0 - C70RADO
Object Link View Extensions Tools Help
Top Level Filter

Messages
Extensibility Projects
XT_1 (One)
XT_1234567890 (123456789012345678901234567890123456789012345678901234567890)
XT_149847 (def 149847)
XT_AITEST (Aileen XT Test)
XT_ALEXE_TEST (Alexey Test PSR)
XT_CONFIG_TST (Config Utility Test)
XT_D148370 (FW TEst 148370)
XT_DEMO01 (demo 01)
Extensibility Units
XT_DEMO01_UNIT1 (Demo 1 Unit 1)
Actions
RS
Constants
Messages
XT_DEMO01_UNIT1__ACCT_NOT_FOUND (Entered Account not found)
Apps
Reports
XT_DEMO_6546 (testing for new query)

If you are planning to use your new message in Java here is an example of how this invocation should look like:

```
rsI.addLineMessage("XT_DEMO01_UNIT1__ACCT_NOT_FOUND",rsI.FATAL);
```

whereas first parameter you provide your message Id, and as a second parameter you provide message severity.

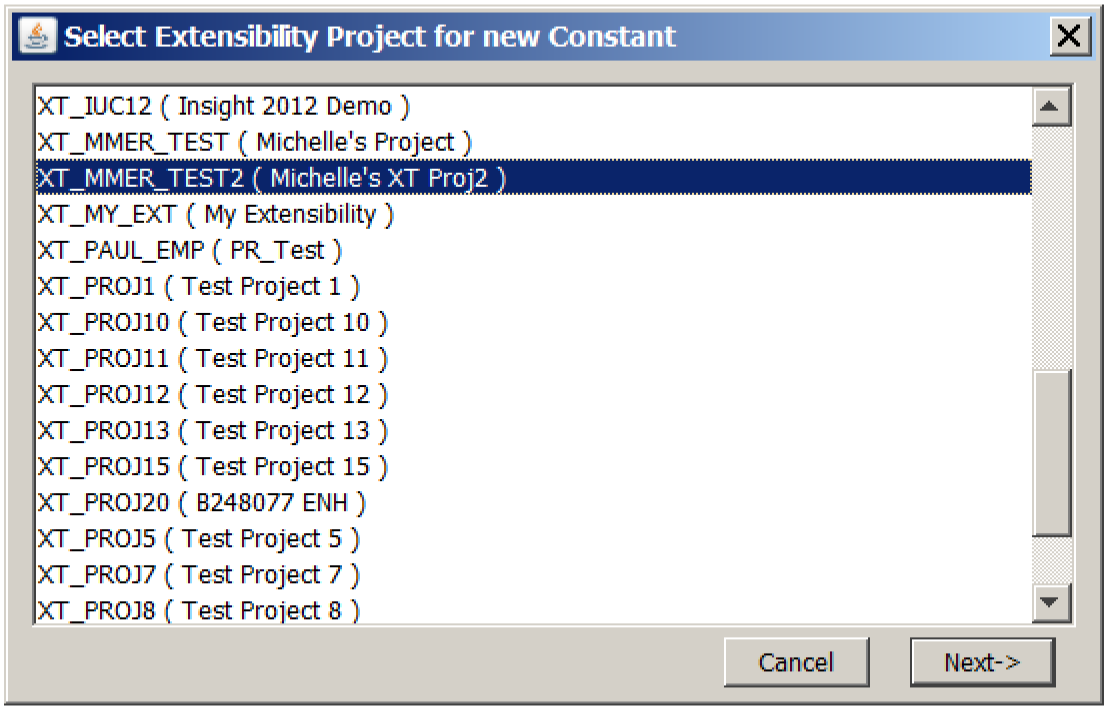
Constants are entities set up in the design database. Typically, they represent values from settings tables. Once defined they are loaded /calculated only once in the application server when your application is opened. Application constants are available at the client browser as well as at the server. Once defined, they are automatically available in the SQL statement (both in RS SQL in design tool and in java class). They are also available for use in field default, field formula (for example, value, editable, visible), label/status text or label/status text formula and application JavaScript.

Adding a New Extensibility Constant

To create a new Extensibility constant:

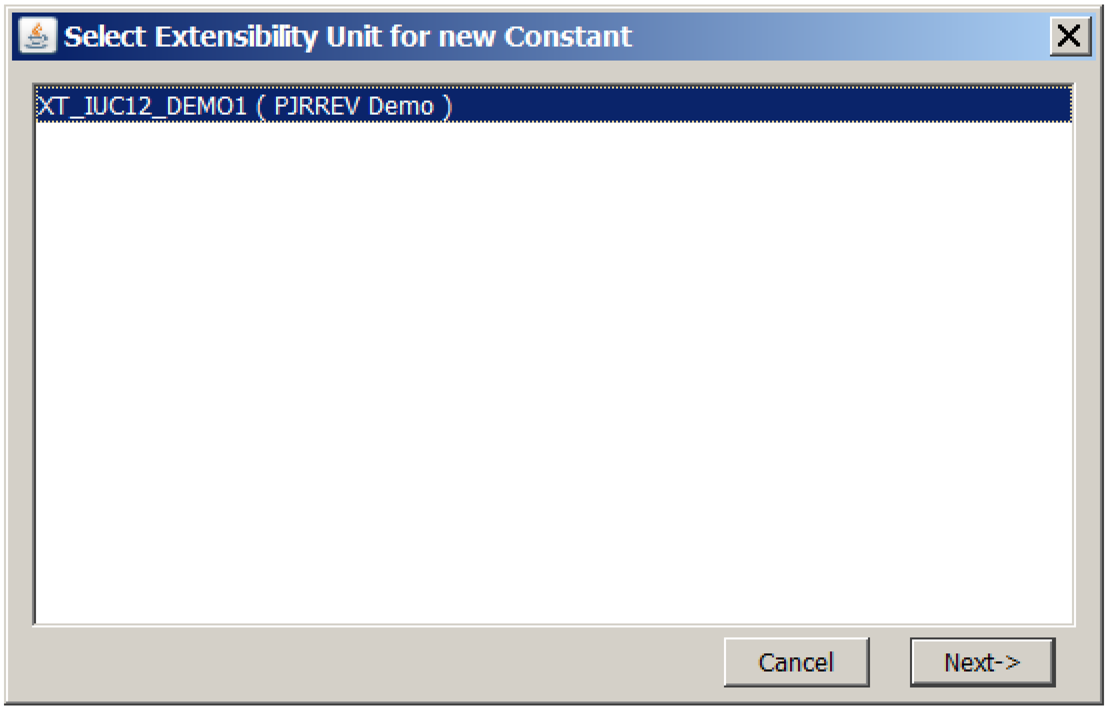
1. Open the Deltek Costpoint Extensibility Designer.

2. On the **Object** menu, click **New » Constant**. The Select Extension Project for new Constant dialog box appears.



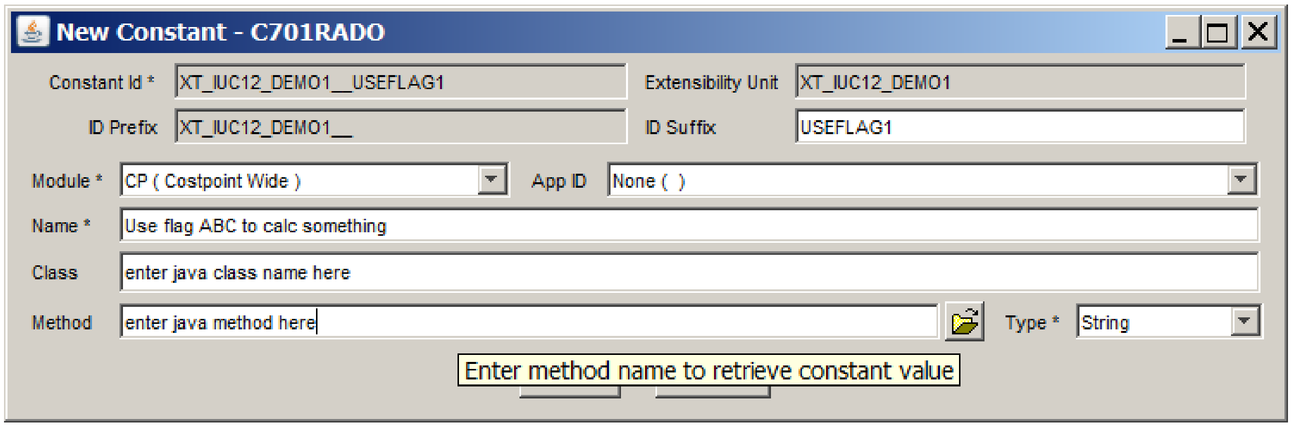
XT_IUC12 (Insight 2012 Demo)
XT_MMER_TEST (Michelle's Project)
XT_MMER_TEST2 (Michelle's XT Proj2)
XT_MY_EXT (My Extensibility)
XT_PAUL_EMP (PR_Test)
XT_PROJ1 (Test Project 1)
XT_PROJ10 (Test Project 10)
XT_PROJ11 (Test Project 11)
XT_PROJ12 (Test Project 12)
XT_PROJ13 (Test Project 13)
XT_PROJ15 (Test Project 15)
XT_PROJ20 (B248077 ENH)
XT_PROJ5 (Test Project 5)
XT_PROJ7 (Test Project 7)
XT_PROJ8 (Test Project 8)

3. Click **Next**. The Select Extension Unit for new Constant dialog box appears.



XT_IUC12_DEMO1 (PJRREV Demo)

4. Click **Next**. The **New Constant** dialog box appears.



New Constant - C701RADO

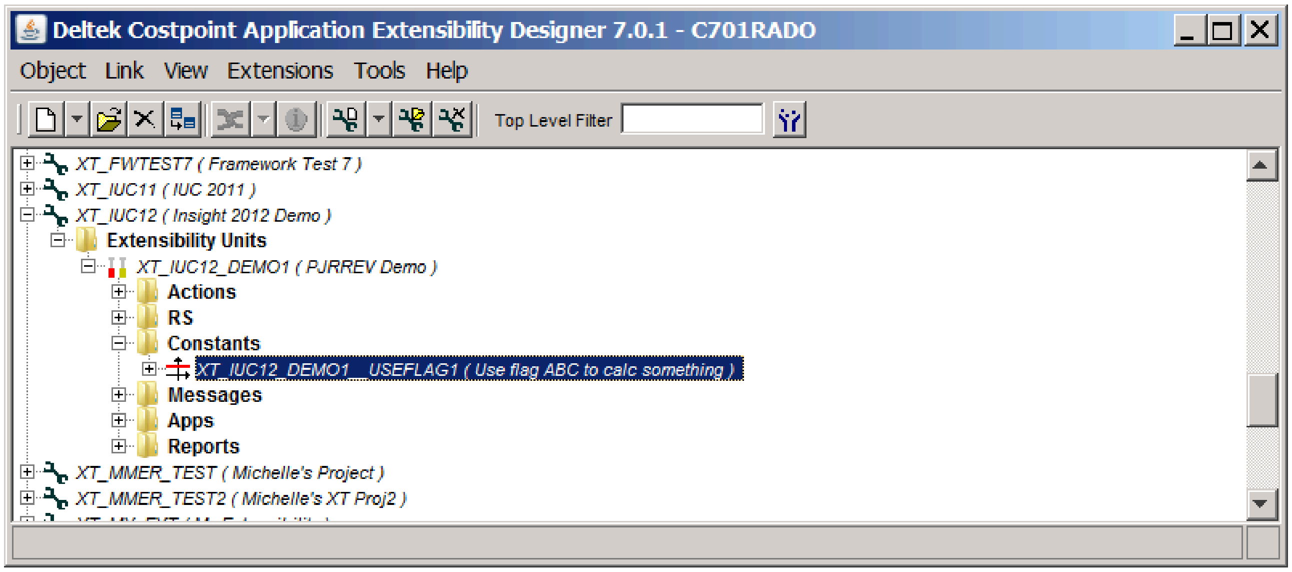
Constant Id *	XT_IUC12_DEMO1__USEFLAG1	Extensibility Unit	XT_IUC12_DEMO1
ID Prefix	XT_IUC12_DEMO1__	ID Suffix	USEFLAG1
Module *	CP (Costpoint Wide)	App ID	None ( )
Name *	Use flag ABC to calc something		
Class	enter java class name here		
Method	enter java method here	Type *	String

Enter method name to retrieve constant value

5. Complete the appropriate fields for your constant.

   - **Constant ID**: This field automatically displays XT_*PROJECT_ID_UNIT_ID__SUFFIX*, where PROJECT_ID is the project identifier that was selected in step 2, UNIT_ID is the unit identifier that are selected in step 3, and SUFFIX is the value entered in the **ID Suffix** field. This is a required field.
   - **Extensibility Unit**: This field automatically displays the extensibility unit being used.
   - **ID Prefix**: This field automatically displays XT_*PROJECT_ID_UNIT_ID__* where *PROJECT_ID* is the project identifier that was selected in step 2, and *UNIT_ID* is the unit identifier that are selected in step 3.
   - **ID Suffix**: Enter a value to assign an identifier to the new constant.
   - **Name**: Enter a description of the new constant. This is a required field. It is used to help search for particular constant.
   - **Module**: Select the module from this drop-down list for which the constant will be used. This is a required field.
   - **App ID**: This drop-down list is populated based on the value that is selected in the Module drop-down list. Select the appropriate application identifier for the new constant. This is an optional field.
   - **Class**: Enter the Java class name that should be loaded to calculate the value of the constant.
   - **Method**: Enter the Java method name that should be executed to calculate the value of the constant.
   - **Type**: Select what type of Object this constant returns. This is a required field.

6. Click **Save**. The **Constant ID** and **Name** will display in the **Constants** folder within the appropriate **Extensibility Unit**.


Deltek Costpoint Application Extensibility Designer 7.0.1 - C701RADO
Object Link View Extensions Tools Help
Top Level Filter
XT_FWTEST7 (Framework Test 7)
XT_IUC11 (IUC 2011)
XT_IUC12 (Insight 2012 Demo)
Extensibility Units
XT_IUC12_DEMO1 (PJRREV Demo)
Actions
RS
Constants
XT_IUC12_DEMO1__USEFLAG1 (Use flag ABC to calc something)
Messages
Apps
Reports
XT_MMER_TEST (Michelle's Project)
XT_MMER_TEST2 (Michelle's XT Proj2)

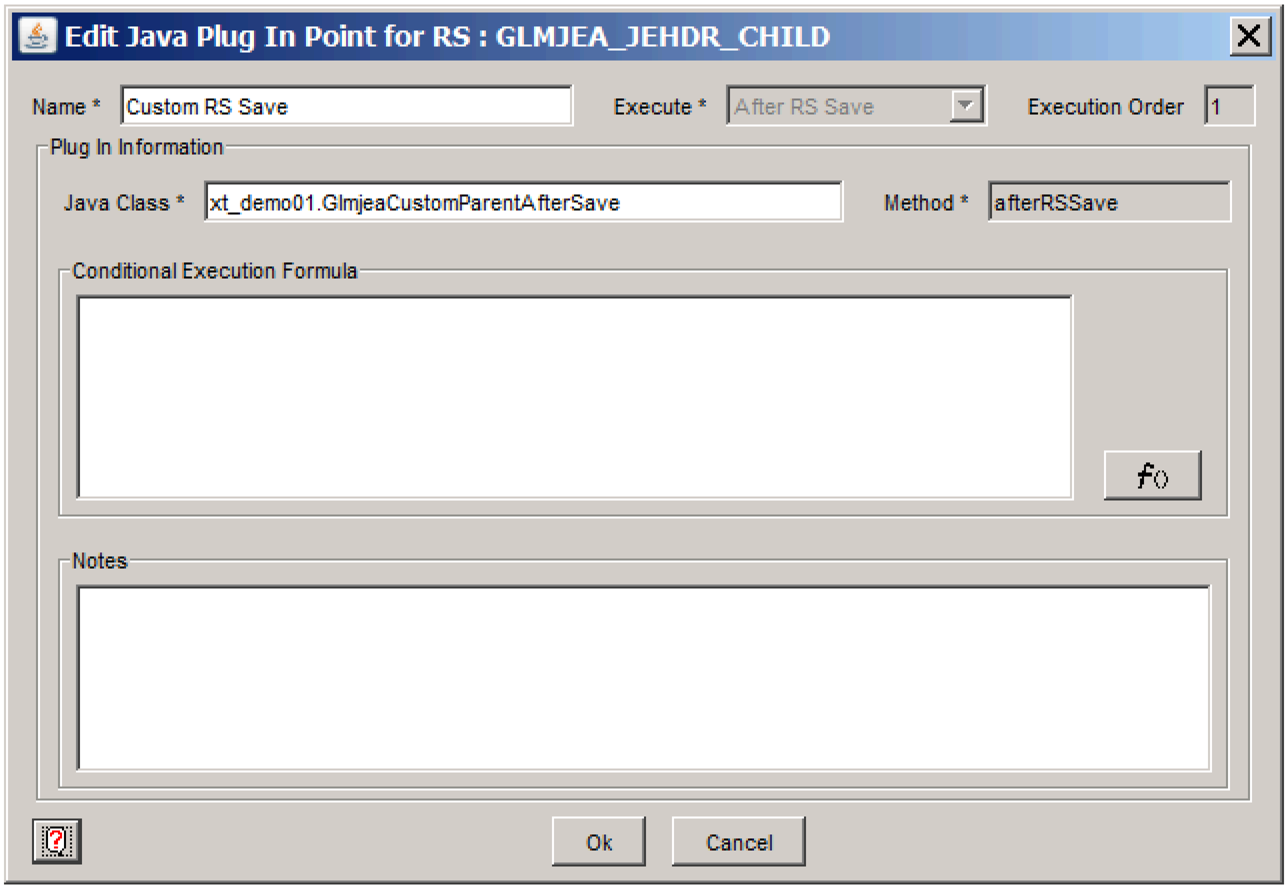
Plug-in points in Costpoint Architecture allow developers to execute needed business logic at a certain moment in application lifecycle. Plug-ins are available in Result Sets and Actions. Depending on Object Type, different plug-in events are available. There are three major types of plug-ins: **Java Class** plug-in, **Database Stored Procedure** plug-in or **E-mail** plug-in. Not all three types are available for each plug-in event. Let us review each plug-in Type.

Java Class Plug In

Java class is the most generic and most powerful plug-in type. It has the fullest array of capabilities to implement business logic. To create it, a developer needs to write a Java class (that should implement certain Java interface) and register it with the system to be invoked at a specified moment in application lifecycle.

Application plug-in classes invoked by the framework (such as object/cell/line validations, actions, and so on) are stateless objects. Between method calls, class variables can be of any state since the framework does not clear its cache. Therefore, class variables should be initialized inside the method when the method is called as a plug in entrance.

Tip: Click the **Help** button to see the name of the Java Interface you class should implement.


Edit Java Plug In Point for RS : GLMJEA_JEHDR_CHILD

Name * Custom RS Save

Execute * After RS Save

Execution Order 1

Plug In Information

Java Class * xt_demo01.GlmjeaCustomParentAfterSave

Method * afterRSSave

Conditional Execution Formula

Notes

Form Fields

Name

Enter the name of the **Java Plug In**. This is a required field.

Execute

Select the Application lifecycle event for which this plug in should executed. This cannot be modified subsequently so if later on you need to change this setting, you must delete and recreate the plug-in.

Execution Order

After the plug in is created, the **Execution Order** text box will contain a number that indicates the order in which the plug in executes in relation to other plug-ins of the same event. You can reorder your plug-in points in the main table that lists all your plug-ins for the Object by using green **Up** and **Down** arrows.

Java Class

Enter the name of the **Java Class**. This is a required field.

Note: All extensibility classes should be under com.deltek.enterprise.extensions. package.

Method

Enter the name of the **Method**. For most events, the method name is predefined based on Java Interface. This is a required field.

Conditional Execution Formula

Enter the conditional execution formula or use the **F()** button to open the **Formula Editor** dialog box to generate a conditional execution formula.

Attention: See Appendix B for more information about formulas.

Notes

Enter some notes that are useful for remembering the purpose of the plug in.

Database Stored Procedure Plug In[]{.indexref entry="Database Stored Procedure Plug In"}

If you are more familiar with Database Stored Procedures (SPs), you can use Database Stored Procedure as a Plug-In.

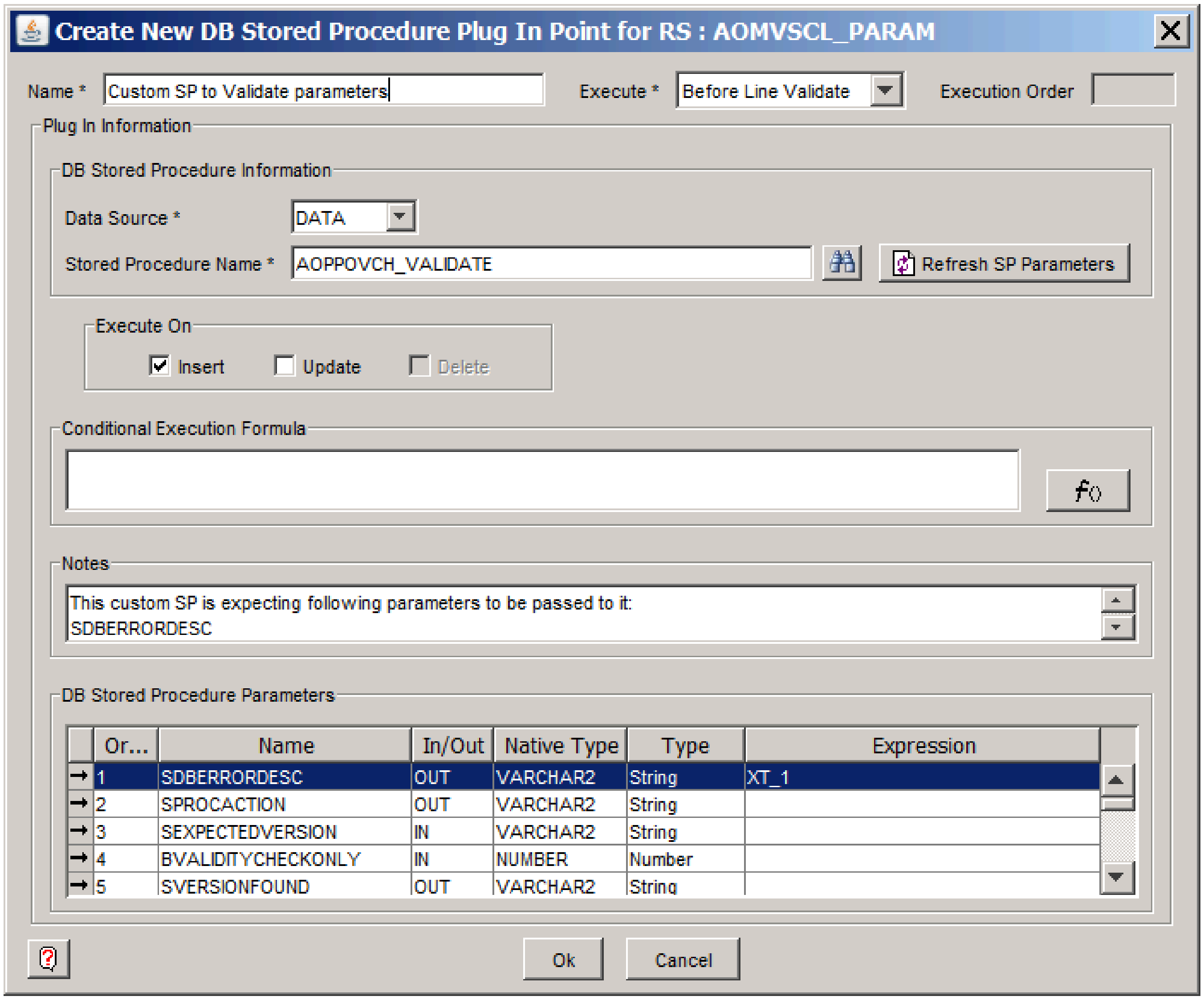
Tip: You can use Database SP not just to perform update in the database but also as a validation plug-in. The system will analyze the return of your SP to see if the validation succeeded or not.

Please write and apply your Database SPs first. Database Stored Procedures Plug-In dialog is used to register an **existing** SP with the system.

Stored Procedure plug-ins cannot be used in actions that are not Single Row actions. As end users may select any number of rows (from 0 to maximum rows in the table), it will be impossible to pass to the Stored Procedure plug-ins an undetermined upfront number of parameters, so Stored Procedure (and email) plug-ins can only be registered with Single Row actions.

Stored Procedure plug-ins allow you to use the two special predefined output parameters of SP:

VALIDATE_SEVERITY and VALIDATE_MSG_ID. During run-time after the plug in is executed -- the return of those two parameters will be analyzed by the system. VALIDATE_SEVERITY indicates if the plug-in succeeded or failed. VALIDATE_MSG_ID contain the ID of the message that will be displayed to the end user.


Create New DB Stored Procedure Plug In Point for RS : AOMVSCL_PARAM

Name *: Custom SP to Validate parameters
Execute *: Before Line Validate
Execution Order:

Plug In Information

DB Stored Procedure Information
Data Source *: DATA
Stored Procedure Name *: AOPPOVCH_VALIDATE

Execute On: [✓ Insert] Update Delete

Conditional Execution Formula

Notes: This custom SP is expecting following parameters to be passed to it:
SDBERRORDESC

DB Stored Procedure Parameters

Or...	Name	In/Out	Native Type	Type	Expression
1	SDBERRORDESC	OUT	VARCHAR2	String	XT_1
2	SPROCACTION	OUT	VARCHAR2	String	
3	SEXPECTEDVERSION	IN	VARCHAR2	String	
4	BVALIDITYCHECKONLY	IN	NUMBER	Number	
5	SVERSIONFOUND	OUT	VARCHAR2	String	

Form Fields

Name

Enter the name of the **Database Stored Procedure Plug-In**. This is a required field.

Execute

Select the sequence in which this plug in should execute. This cannot be changed while modifying the plug in.

Execution Order

After the plug in is created, this can be modified. The **Execution Order** text box will then contain a number that indicates the order in which the plug in executes in relation to other plug ins.

Execution On

For certain types of RS plug-ins, you can specify when to execute current plug-in by selecting one or several **Execute On** flags: **Insert, Update, Delete**. If flag particular flag is not selected (say **Delete**), the plug-in point will not be invoked for deleted rows.

Data Source

Select the data source (database) in which SP should be invoked from the drop-down list.

Stored Procedure Name

Enter the name of the stored procedure or use the **Lookup** button to look up existing database stored procedures. This is a required field.

Conditional Execution Formula

Enter the conditional execution formula or use the **F()** button to open the **Formula Editor** dialog box to generate a conditional execution formula. See Appendix B for more information about formulas.

Notes

Enter some notes that are useful for remembering the purpose of the plug in.

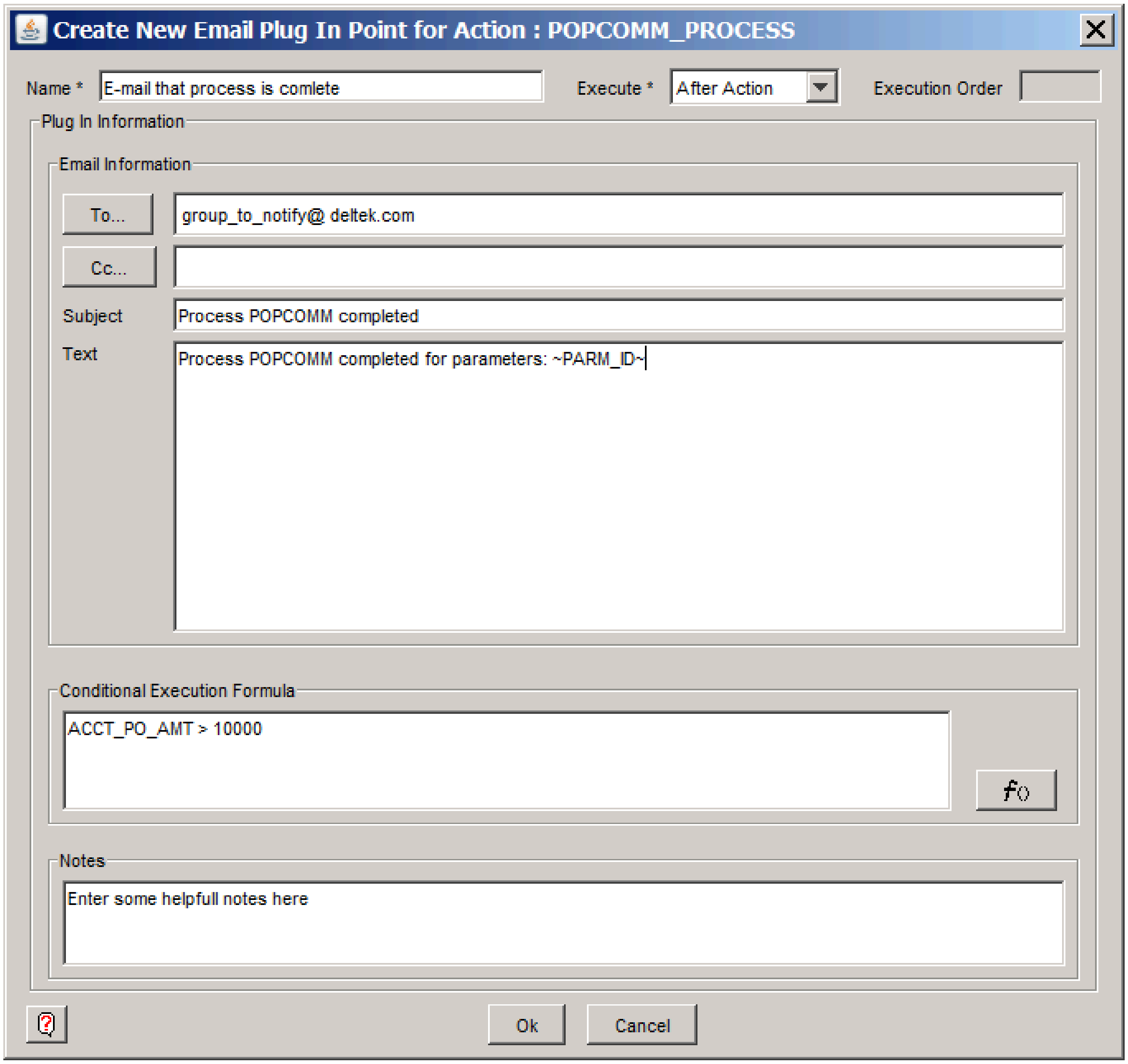
DB Stored Procedure Parameters

Stored procedure should be applied to the database before this step. Click **Refresh SP Parameters** to display existing stored procedure parameters. They will be read from the Database. Modify the type of parameter by selecting **String**, **Number**, **Integer**, or **Date/Time** from the **Type** drop-down list if needed. In addition, new expressions can be associated with individual parameters by double-clicking in the blank area in the same row of the parameter in the **Expression** column. You can specify fields (or formulas) for 'In' type parameters of your SP and the system will pass those to your Database SP during the execution. For 'OUT' parameters enter Object names from current RS and the system will place values into those Objects when SP returns them.

Email Plug In

Email plug-in can be used for sending email after Costpoint record is saved (inserted, updated or deleted) or

Action (Process) is executed for any application.


Create New Email Plug In Point for Action : POPCOMM_PROCESS

Name * E-mail that process is comlete

Execute * After Action

Execution Order

Plug In Information

Email Information

To... group_to_notify@ deltek.com

Cc...

Subject Process POPCOMM completed

Text Process POPCOMM completed for parameters: ~PARM_ID~

Conditional Execution Formula

ACCT_PO_AMT > 10000

Notes

Enter some helpfull notes here

Form Fields

Name

Enter the name of the **Email Plug In**. This is a required field.

Execute

Select the sequence in which this plug in should execute. This setting cannot be modified subsequently after the plug in has been created.

Execution Order

This setting can be modified after the plug in is created. The **Execution Order** text box will then contain a number that indicates the order in which the plug in executes in relation to other plug-ins.

To...

Enter an email address or click on the **To...** button to select a stored contact. You can send specify regular e-mail addresses, Employee ID (Employee record in Costpoint should have valid email entered), User ID (User record in Costpoint should have valid e-mail entered) or all Users in User Group (All User records in Costpoint should have valid e-mail entered).

CC...

Enter an email address or click on the **CC...** button to select a stored contact.

Subject

Enter the subject of the email to go out. Use **~XXX~**, where **XXX** is the OBJ_ID or Parent.OBJ_ID or Parent.Parent.OBJ_ID to replace it with the object's value. Use **&CP.XXX&**, where **XXX** is the **Constant ID** to replace it with the constant's value.

Text

Enter the body of the email. Use **~XXX~** where **XXX** is the OBJ_ID or Parent.OBJ_ID or Parent.Parent.OBJ_ID to replace it with the object's value. Use **&CP.XXX&**, where **XXX** is the **Constant ID** to replace it with the constant's value.

Execute On

Click the **Insert**, **Update**, and/or the **Delete** checkboxes to indicate when the email should be sent if this plug-in is for **After RS Save** event. At least one of these options must be selected.

Conditional Execution Formula

Enter the conditional execution formula or use the **F()** button to open the **Formula Editor** dialog box to generate a conditional execution formula.

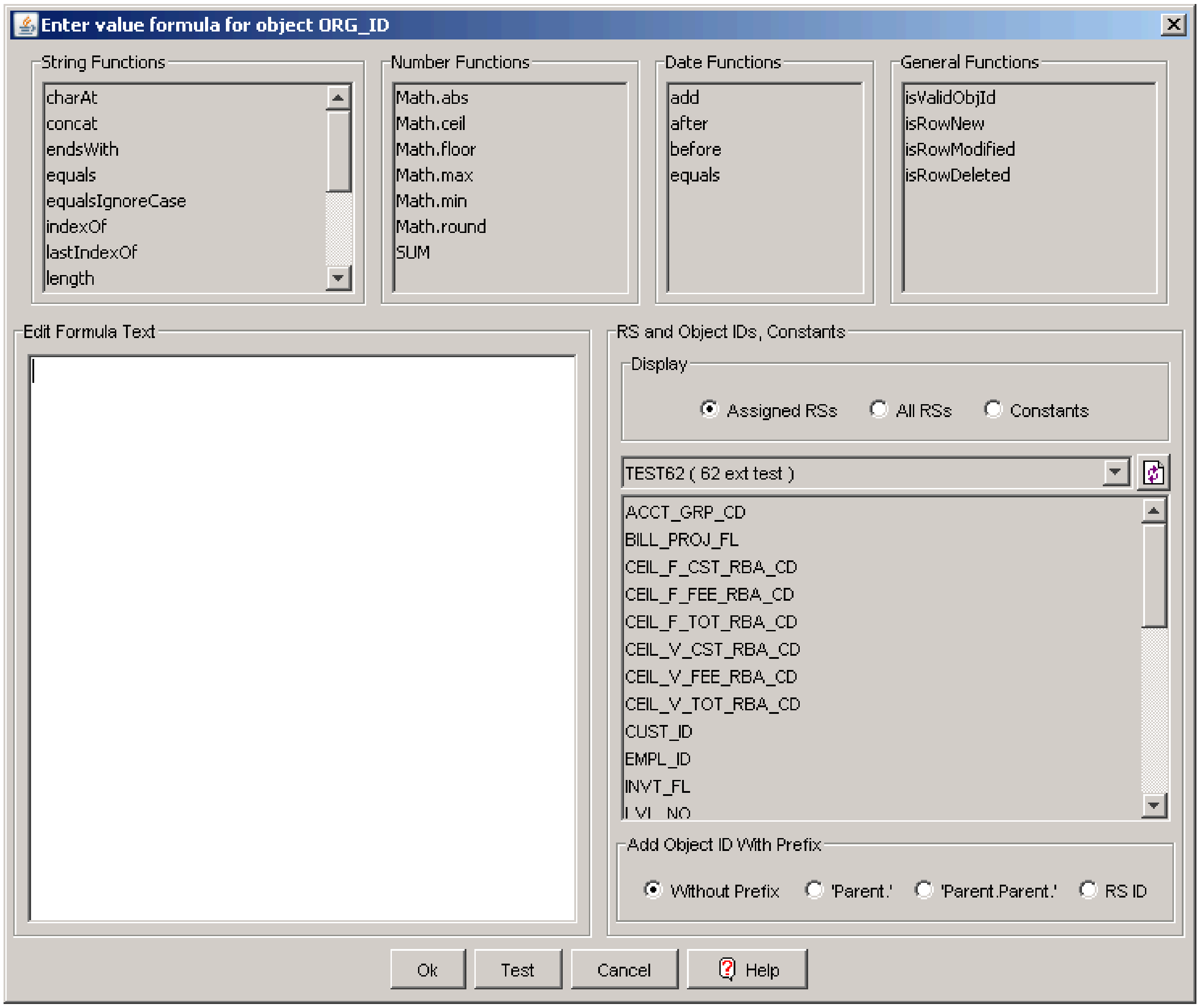
Note: See Appendix B - Formula Dialog Box.

Notes

Enter some notes that are useful for remembering the purpose of the plug in.

Formulas are used in the Extensibility Designer (in lieu of programming code) to return a value to be used in the context it is used. For example, when a visibility formula for a field is true, the field is visible or false the field is hidden. It can be used in to disable or enable a field. It can be used to execute or skip particular plug-in. It can be used to put value into a field (based on values of other fields: For example, amount of sales tax = total * sales tax rate.

Your formula can use constant values, values of fields in the row from the parent RS row and in special cases like TOTAL values in child RS .



Enter value formula for object ORG_ID

String Functions: charAt, concat, endsWith, equals, equalsIgnoreCase, indexOf, lastIndexOf, length

Number Functions: Math.abs, Math.ceil, Math.floor, Math.max, Math.min, Math.round, SUM

Date Functions: add, after, before, equals

General Functions: isValidObjId, isRowNew, isRowModified, isRowDeleted

Edit Formula Text

RS and Object IDs, Constants

Display: [✓ Assigned RSs] All RSs Constants

TEST62 (62 ext test)

ACCT_GRP_CD
BILL_PROJ_FL
CEIL_F_CST_RBA_CD
CEIL_F_FEE_RBA_CD
CEIL_F_TOT_RBA_CD
CEIL_V_CST_RBA_CD
CEIL_V_FEE_RBA_CD
CEIL_V_TOT_RBA_CD
CUST_ID
EMPL_ID
INVT_FL
LVL_NO

Add Object ID With Prefix: [✓ Without Prefix] 'Parent.' 'Parent.Parent.' RS ID

Note: Use #XXX# (where XXX is the **Object Id**) inside the formula for method isValidObjId(), which checks whether a given **Object Id** exists in a specific record set (current, parent, or child).

Note: The **Test** button will check the formula for valid syntax, but it does not guarantee successful execution of the formula.

How to use the Formula Dialog Box

General Information

- The formula is entered in the **Edit Formula Text** field.

- The formula is executed on the client browser. There is no database access, so all variables are from fields that are available on the client side.
- The formula is added for user convenience only (like running total or default value). If it executes a required calculation (business rule), the logic must also exist on the server side.
- A formula cannot be executed for server-only fields. Such logic must be added to the server-side application code.

Elements

- An **Object Id** can be used in the same result set by using the **Object Id** in the formula, or from a parent result set by using Parent.<Object Id> or Parent.Parent.<Object Id>.
- An **Object Id** can be used from a child result set by qualifying the ID with the record set ID of the child result set. A child object ID can only be used in the formula if it is in the SUM format (since the child result set can only be looked at by the parent in total).
- Application constants must be assigned to the application before they can be used. Constants can be referred to in the formula by pre-pending the constant with the keyword CP.

Expressions

- A formula must be a valid Java expression.
- Click the **Test** button to check for valid syntax.
- Arithmetic expressions such as +, -, *, and so on can be used.
- String functions such as length(), substr(), and so on are available.
- Conditional expressions such as (object1>0?object2:object3) are allowed. This formula translates to: If object1 is greater than zero, then the formula returns object2. Otherwise, it returns object3.
- SUM can be used for a child result set (SUM(<childRSID>.<childObjectID>)). Multiplication (*) and division (/) are not allowed with a SUM.

Events

- **Formula** is called when a value is changed in a related field and the user has moved focus away from the field.
- Unlike validation, **Formula** is always called when the field is changed. Validation is not called if the field is changed to blank.
- **Formula** in a child result set using "parent" is not called when a parent field is changed because only context rows are recalculated.
- **Formula** in a parent result set is called when rows in related children result sets are added, deleted, or changed.
- **Formula** is called by the system after client-side JavaScript and after server-side Java validation (if in fast or medium mode).
- **Formula** is not called when the result set is first populated. If the field is not part of the SELECT SQL, it must be populated by the App Populate Java Interface on data load.

Deltek

Order of Execution

- When a column is modified, all formulas that directly depend on this column will be recomputed with columns that belong to the same result set as the modified column being computed first.
- Avoid circular reference (that is, Field A depends on Field B and Field B depends on Field A). The presence of circular reference makes the recalculation order unpredictable.
- Columns from the parent result set linked to the modified column through SUM() is computed next.
- Then all formulas that depend on columns that were programmatically recomputed (current result set first and then parent result set).

Form Fields

String Functions

Name	Description
charAt	Returns the character at a specified position.
concat	Used to join two or more arrays.
endsWith	Tests if an input string ends with the specified suffix.
equals	Compares values for equality
equalsIgnoreCase	Returns whether the specified string objects are the same when case is ignored
indexOf	Returns the offset within a string where the specified substring first occurs.
lastIndexOf	Returns the offset within a string where the specified substring last occurs.
length	Returns the number of characters in a string.
replace	Returns a string where all occurrences of a given character are replaced with another character.
startsWith	Returns whether the given string starts with the specified prefix.
substring	Returns a string that is a substring of the specified string.
toLowerCase	Returns a string where all the letters in the specified string are turned into lowercase.
toUpperCase	Returns a string where all the letters in the specified string are turned to uppercase.
toString	Returns a string that represents the current object.
trim	Returns a string where all the leading and trailing white spaces are removed in the specified string.

Number Functions

Name	Description
Math.abs	Returns the absolute value of a specified number.
Math.ceil	Returns the smallest integer greater than or equal to the specified number.
Math.floor	Returns the largest integer less than or equal to the specified number.
Math.max	Returns the larger of two specified numbers.
Math.min	Returns the smaller of two numbers.
Math.round	Rounds a value to the nearest integer or specified number of decimal places.
SUM	Returns the sum of specified numbers.

Date Functions

Name	Description
add	Adds the value of the specified TimeSpan to the value of this instance
after	Tests if this date is after the specified date
before	Tests if this date is before the specified date
equals	Compares two dates for equality. The result is true if and only if the argument is not null and is a Date object that represents the same point in time, to the millisecond, as this object.

General Functions

Name	Description
isValidObjId	Reports whether the object ID is valid
isRowNew	Reports whether the row has been newly inserted
isRowModified	Reports whether the row has been modified
isRowDeleted	Reports whether the row has been deleted

Edit Formula Text

Enter any applicable formula text associated with the appropriate field in the RS Description tab. Use the provided functions to assist with building a formula.

RS and Object IDs, Constants Field Set

Display

- **Assigned RSs**: Display only the assigned result sets in the drop-down list below. An assigned RS is the result set that is currently being modified.
- **All RSs**: Display all result sets in the drop-down list below.
- **Constants**: Display a list of constants in the drop-down list below.

Add Object ID With Prefix

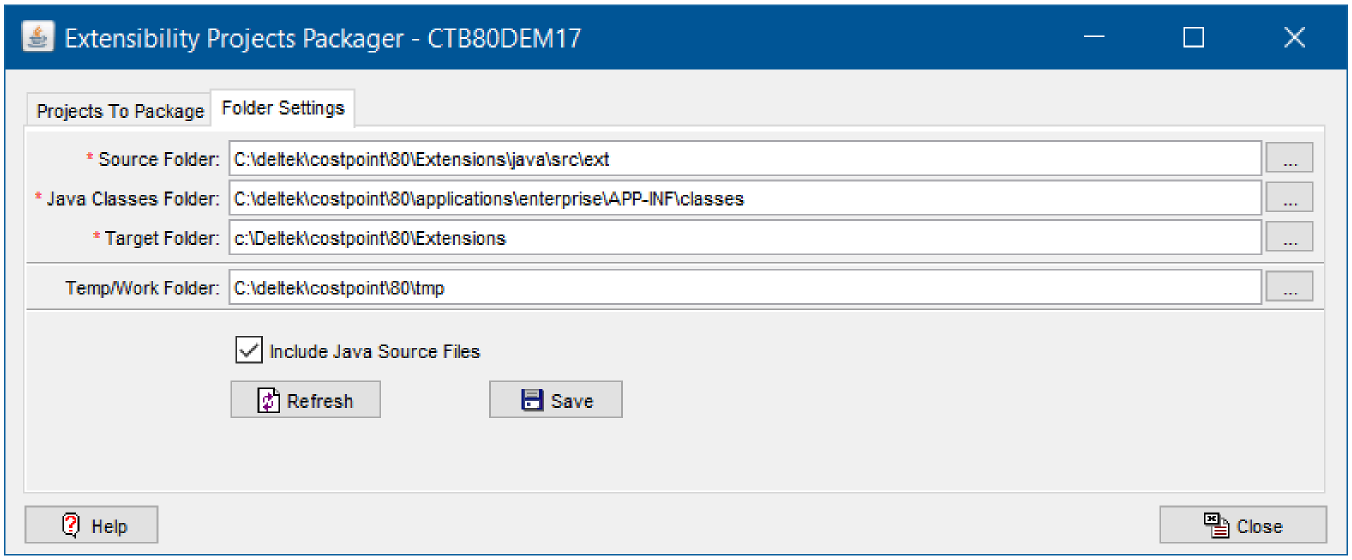
Select one of these options for prefixing to the selected **Object Id** for use in the **Edit Formula Text** field.

- **Without Prefix**: Add Object ID without RS prefix
- **'Parent'**: Add Object ID with 'Parent.' prefix (for parent result sets)
- **'Parent.Parent'**: Add Object ID with 'Parent.Parent.' prefix (for grandparent result sets)
- **RS ID**: Add Object ID with RS ID prefix (for child result sets)

The **Extension Packager** utility is used for assembling and packaging Extensibility Project files when the development stage is finished so that the Extensibility Project can be delivered to clients and deployed (using DBWizard) to testing and production systems. The output of this Utility is a zip file that contains all files needed to deploy the Extensibility project:

- **Extensibility Metadata scripts**: This is a database script with the name *Your_Extensibility_ProjectId*.extproj.SQL. It contains all the changes you have made to metadata while customizing a standard or creating new Custom Objects. This is a required piece.
- **Database change scripts** (to create or modify custom tables, views, indexes, and database stored procedures): This is applicable only if you use additional custom objects in the database. If you are using such custom objects in your Extensions, Deltek highly recommends (and even requires if you are sending your Extensibility Project to Deltek Support or if you are creating an Extension for the Deltek's Cloud customer) to include such DB scripts in your project as it ensures that the Project can be easily applied to the new system or new environment.
  Deltek also recommends that you write such scripts in a way that the script can be executed multiple times without errors and without unnecessary dropping and re-creating the same database objects. This should be done to ensure that the project can be reapplied to same system or environment multiple times. For examples of this syntax, review the database patches that Deltek releases within the hot-fix files. As with all database patches that Deltek releases, they are written in this way.
- **Java Classes**. This is applicable only if you implement plug-in logic in Java classes or have entered or modified Formulas. To open Extensibility packager, click **Extensibility Packager** on the **Tools** menu.

Folder Settings Tab



* Source Folder: C:\deltek\costpoint\80\Extensions\java\src\ext

* Java Classes Folder: C:\deltek\costpoint\80\applications\enterprise\APP-INF\classes

* Target Folder: c:\Deltek\costpoint\80\Extensions

Temp/Work Folder: C:\deltek\costpoint\80\tmp

[✓] Include Java Source Files

1. Click the ellipses button (...) to the right of the **Source Folder** text field or manually enter a folder name where is your source code is located. For example: c:\Deltek\costpoint\82\Extensions\java\src

2. Click the ellipses button to the right of the **Classes Folder** text field or manually enter a folder name where your java classes are located. For example: c:\Deltek\costpoint\81\applications\enterprise\APP-INF\classes.

   Note: All extensibility classes should be under com.deltek.enterprise.extensions. .

   You can also include Java source files into your Project file. And if you are creating Extension for the Deltek's Cloud -- you are required to include those Java source files so Cloud team can review them before deploying your Extension.

3. Click the ellipses button to the right of the **Target Folder** text field or manually enter a folder name where packaged files will be stored.

4. Click the ellipses button to the right of the **Temp Folder** text field or manually enter a Temporary (Working) folder name.

5. Click **Include Java Source Files** if you want to include Java source files (.java) inside the package.

   This option is recommended for everybody, so you wouldn't lose your source files. For Cloud customers, it is a requirement to have it checked.

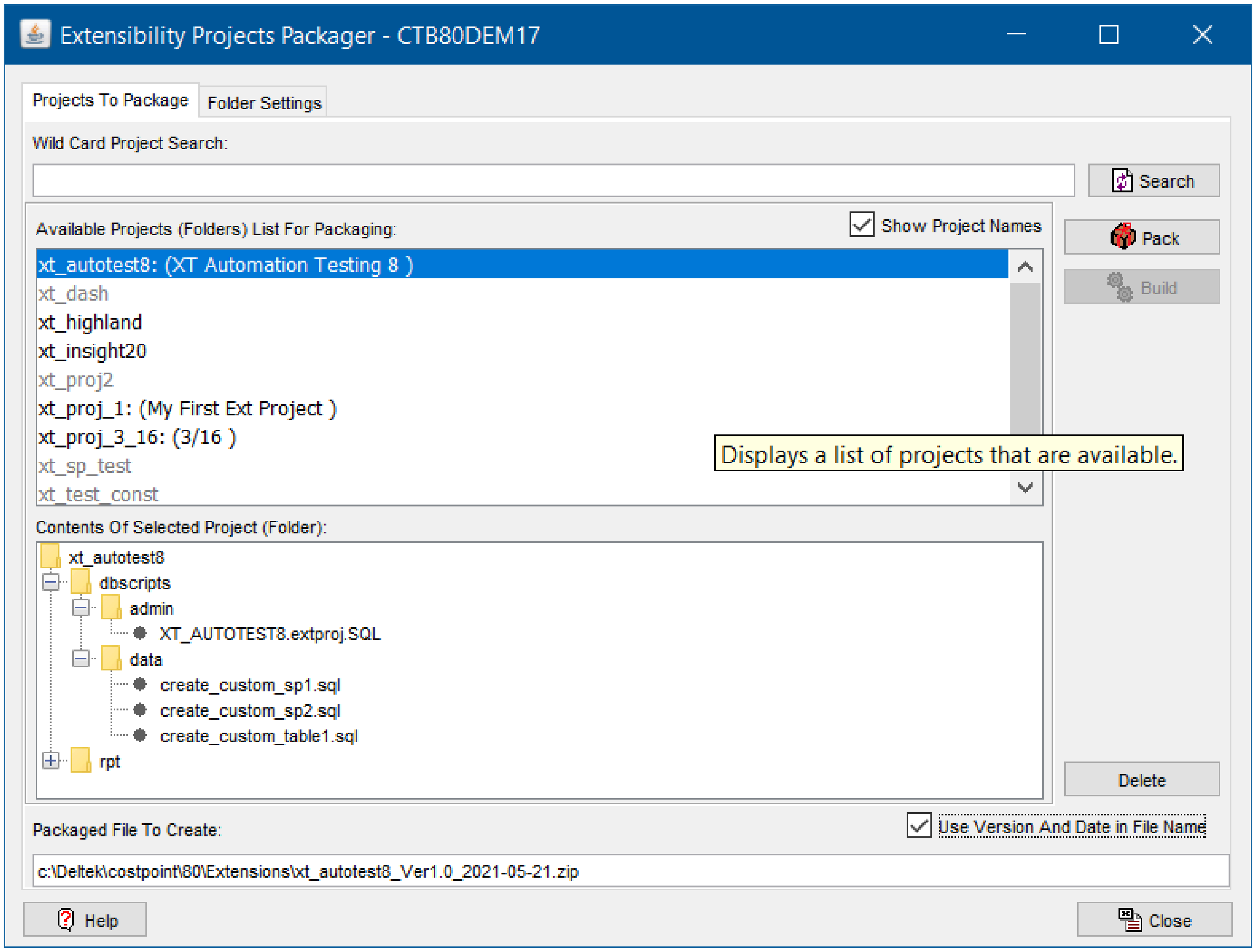
6. Click the **Save** button.

7. Click **Refresh** to refresh the project list on the **Projects** tab based on the new folder information.

> **Note:** The following folder structure must be in place for this tool to find the project:
>
> *SOURCE_FOLDER*\com\deltek\enterprise\extensions<em>PROJECT_ID\dbscripts\Admin\
>
> Where *SOURCE FOLDER* is defined on the **Settings** tab and *PROJECT_ID* is the packaged project that is saved in the given file as *PROJECT_ID*.extproj.SQL. The folder names are case-sensitive, and the project identifier must be in lower case.
>
> Also note that even though you can export Extensibility Units as scripts -- this option is provided for backup purposes and all unit scripts *.extunit.sql should be removed before packaging process from the Source Folder.

Projects Tab

This is a tab where you do actual Extensibility Project packaging.


Extensibility Projects Packager - CTB80DEM17

Wild Card Project Search:

Available Projects (Folders) List For Packaging: ✓ Show Project Names

xt_autotest8: (XT Automation Testing 8)
xt_dash
xt_highland
xt_insight20
xt_proj2
xt_proj_1: (My First Ext Project)
xt_proj_3_16: (3/16)
xt_sp_test
xt_test_const

Displays a list of projects that are available.

Contents Of Selected Project (Folder):

```
xt_autotest8
  dbscripts
    admin
      XT_AUTOTEST8.extproj.SQL
    data
      create_custom_sp1.sql
      create_custom_sp2.sql
      create_custom_table1.sql
  rpt
```

Packaged File To Create: ✓ Use Version And Date in File Name

c:\Deltek\costpoint\80\Extensions\xt_autotest8_Ver1.0_2021-05-21.zip

Attention: For more detailed instructions on which file types can be packaged and in what folder each file should go, click the **Help** button in the Extensibility Packager.

1. In **Available Projects (Folders) List For Packaging** you need to select an Extensibility Project you want to package.

2. Optionally select **Show Description** to see the Project ID's description.

3. Optionally use the **Wild Card Project Search** field to narrow down the list of Project IDs available.

4. **Contents Of Selected Project (Folder)** box, shows a list of files for selected Extensibility Project from the file system that are found under Extensibility Project Folder.

5. Select the Use Version And Date in File Name check box to include the Extensibility Project's Version and current date as part of the packaged project file name.

   This option is required if you are creating the Extensibility project for Deltek's cloud and recommended for everybody else to distinguish and track which versions of the project are being used/sent for deployment more easily.

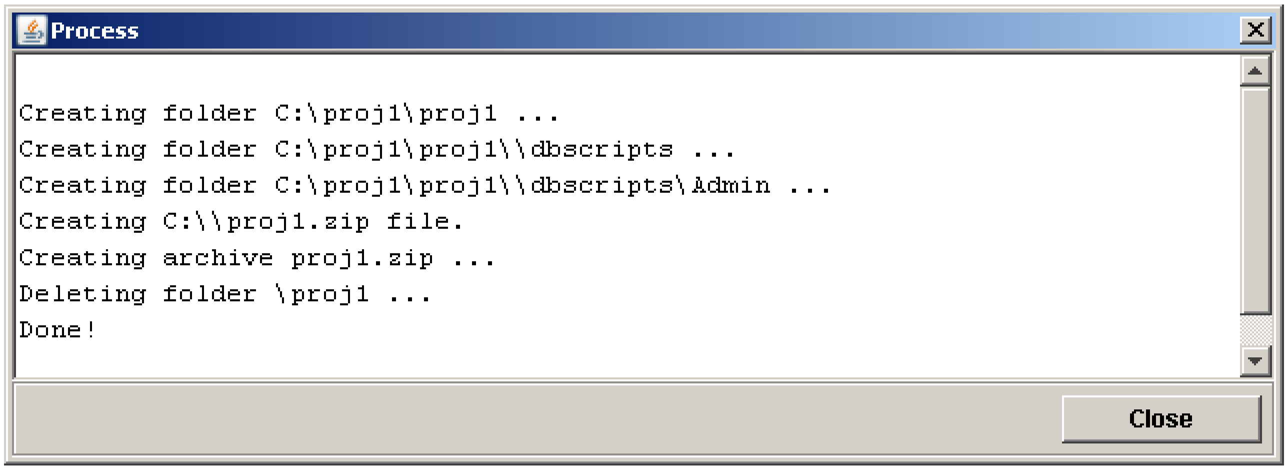
6. Packaged File To Create field displays the packaged zip file name that will be created by the packager.

7. Select Pack to start actual packaging process.

8. If Projected is greyed out, the selected folder does not have Extensibility Project scripts under the appropriate folder.

   Before packaging, please make sure you have all your latest work in project script (extproj.SQL) and no unit scripts (.extunit.SQL) are present in the folder you will be packing.

9. When the process screen says "Done!" select Close. Your packaged Extensibility project file is ready.

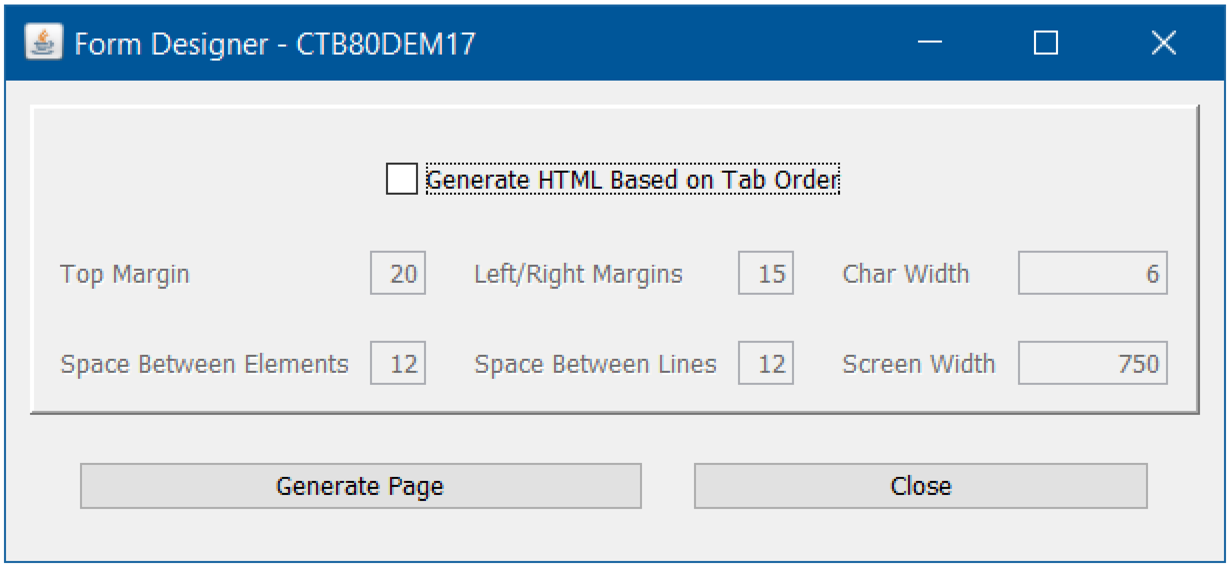


```
Creating folder C:\proj1\proj1 ...
Creating folder C:\proj1\proj1\\dbscripts ...
Creating folder C:\proj1\proj1\\dbscripts\Admin ...
Creating C:\\proj1.zip file.
Creating archive proj1.zip ...
Deleting folder \proj1 ...
Done!
```

Tip: Deltek recommends that you open packaged zip files with WinZip or other zip utilities to verify all the files are packaged inside.

Form Designer is used to arrange the position of all objects a result set when shown in the form view mode in Costpoint. If you are adding the new fields or new HTML controls (group boxes, tabs, and so on), use the Designer to arrange the layout.

Click the **Form Designer** button in the **Result Set's Presentation** tab and the Form Designer dialog displays.


Form Designer - CTB80DEM17

☐ Generate HTML Based on Tab Order

Top Margin	20	Left/Right Margins	15	Char Width	6
Space Between Elements	12	Space Between Lines	12	Screen Width	750

Form Fields

Generate HTML Based on Tab Order

Deltek recommends this option to be used only for the first time when the result set is created (layout has not been designed). When selected, the tool will spread all visible elements on the page with provided margins.

Do not use this option if you are only adding additional objects to the result set or want to adjust positioning of the fields.

Output File

Enter the name and location of the output file to be generated or use the **Browse** button to navigate to select a location and enter a file name.

Generate Page

Click this button to generate an HTML page for the result set and load browser to show the page in design mode. If there are no tabs in the form, the first page is the only page for the form.

See the **Designing Page Layout** section for details about editing the page layout.

Designing Page Layout

A browser window will be opened upon clicking Generate Page. Objects will be displayed in the same position as they are displayed at run time in form view mode.

You can start designing the page by moving or resizing objects on this window.

> **Note:** You cannot add objects in here. Objects can only be added back in the RS Desc tab or RS Presentation tab. In addition, certain object cannot be resized here since the length (size) is specified in the RS_Presentation tab.

Element Information

While designing page layout, move the cursor over each element to obtain additional information about the element. The status bar will display the coordinates of the cursor (left, top), the control type, Object Id, and the coordinates of the object (left, top).

The tool also shows yellow bubble help when the cursor is placed over an element. The information displayed is as follows: TabOrderSequence#ControlTypeCode#ObjectID.

Positioning Elements

- Position fields or labels by dragging the blue **M** square in the top left corner of the data field.
- Moving the data field will also move the associated label.
- Moving the labels will not move the associated data field, so position the labels relative to the fields first then move the fields.
- Elements that are newly added usually appear on the top left corner and may be stacking up on top of each other. Drag them out to make them visible individually.
- Elements can be moved in groups. Selecting multiple elements by pressing the **Shift** key, selecting all the elements to be moved, and moving the objects within the group. Deselect elements by pressing and holding the **Shift** key and clicking on objects to be removed from the group. Press the **Enter** key to remove all links.
- Elements can also be selected by drawing a rectangle surrounding the desired objects. Do this by moving the cursor to the top left corner of the intended rectangular area, press and hold the **Ctrl** key, click, and release. A rectangle will appear. Click and drag the R symbol at the bottom right corner to resize the area. Release the symbol to select all the elements within the rectangle.
- Elements can also be moved using the **Move Element To** popup menu.
- If elements are within an information box (entity, primary, or secondary), moving the box will also move the elements within it.

Resizing Elements

- Elements can be resized only if the control type appears with a red R in the bottom right hand corner on mouse over.
- Multi-line text fields are size with the last two elements of the coordinates. They express the number of rows and number of characters wide.

Positioning Tab Panel

- If there are tab controls, the tab panel will appear in the positioning screen.
- The tab panel can only be moved the first time the page is generated.
- The tab panel can be positioned to be at the top or middle of the screen.
- All controls on the first screen must be either above or below the tab panel.
- Controls on subsequent screens must be below the tab panel.

Popup Menu

Right-click on the form to display the popup menu for additional options and commands.

Save

Save new positions and other settings to the RS Presentation tab. This does not save it into the database. It just saves the new information to the presentation screen. To save to the database, go back to the RS Presentation tab. Notice that objects on the left panel have been flagged as changed. Click Save to save to the database.

Set Default TAB order

The position of the elements does not necessarily control the tab order. To use positioning to reset the tab order, click **Set Default TAB order**. This will reset the tab order on the **RS Presentation** tab.

Align Edges

Object edges can be aligned left, right, top, or bottom. Select elements while pressing the **Shift** key, right click for the popup menu, click **Align Edges**, and click **Left**, **Right**, **Top**, or **Bottom**. Remember that the label always moves with the input control.

Move Element To

Move a single element to an exact pixel location with this option. Before using this option, close the popup menu if it is open. Right-click on a movable element (one with a blue M on mouse over), and Click **Move Element To**. A small dialog box will display showing the left and top pixel position. Enter the new coordinate information and click **OK** to move the element.

Resize Element

Resize a single element to an exact pixel size with this option. Before using this option, close the popup menu if it is open. Right-click on a resizable element (one with a red R on mouse over) and click **Resize Element**. A small dialog box will display showing the width and height of the element in pixels. Enter the new height and width and click **OK** to resize the element.

Show/Hide Grid

Click **Show Grid** in the popup menu to show the grid in the form designer. Grid squares are four pixels square. If the grid is already showing, click **Hide Grid** in the popup menu to hide the grid in the form designer.

Primary Info Area

If the result set has either entity box or primary information box, the Primary Info Area menu item will be available. Use this option to programmatically arrange or resize the entity box and/or the primary information box.

- **Set Layout**: If the number of primary boxes is between two and three, the Set Layout menu item is available. For two boxes, select the **2 equal** or **2 unequal** layout. If there is also an entity box, **1 2 3** layout is also available. For three boxes, select **3 equal** or **3 unequal**. These choices specify the layout of the boxes and specify if the horizontal width of each box is equal or unequal to the others within the same group.
- **Set Height**: Select this option to set the height of entity boxes or primary information boxes. A dialog box will appear showing the number of rows to be set for these boxes.
  For entity boxes, enter the number of rows. The larger the number, the larger the boxes will be set.
  For primary boxes, enter the number of rows. The larger the number, the larger the boxes will be set. If there are group boxes inside any of these boxes, the height can be incremented minimally by entering the number of group boxes. For each group box, the height is incremented by five pixels.
- **Show/Hide Labels**: To make it easy to position fields, hide all the labels by selecting this option. Once all the fields are aligned, toggle this option to show the labels and align the labels with the fields.

Term	Definition
App	Application
CC	Carbon Copy
DB	Database
HTML	Hypertext Markup Language
ID	Identification
Org	Organization
RS	Result Set
SP	Stored Procedure
SSN	Social Security Number
SQL	Structured Query Language

Term	Definition
UI	User Interface

Term	Definition
Result Set	A result set is generally a screen included in an application. It usually matches data that can be represented in an SQL Select statement. Additional fields not included in the select statement can be added but the values must be populated programmatically outside of the SQL statement (generally Java Classes).
Segment Character	Segment characters are the special characters (that is, hyphen (-), period (.), and so on) that are used for segment formatting. For example, a social security number typically uses hyphens in its formatting to display the number in a standard format (XXX-XX-XXXX). Each segment has a prefixed length, defined by the system.