Sterling Selling and Fulfillment Foundation

Using The Rich Client Platform Extensibility Tool

Version 9.1

Sterling Selling and Fulfillment Foundation

Using The Rich Client Platform Extensibility Tool

Version 9.1

Note
Before using this information and the product it supports, read the information in “Notices” on page 91.

Contents

Chapter 1. Checklist for Customization Projects

Customization Projects

Projects to customize or extend Sterling Selling and Fulfillment Foundation vary with the type of changes that are needed. However, most projects involve an interconnected series of changes that are best carried out in a particular order. The checklist identifies the most common order of customization tasks and indicates which guide in the documentation set provides details about each stage.

The items identified for extension and/or modification in the documentation are Source Components (to the extent such item involves source code) and Sample Materials for purposes of the License Information file associated with this product.

Prepare Your Development Environment

Set up a development environment that mirrors your production environment, including whether you deploy your application on a WebLogic, WebSphere®, or JBoss application server. Doing so ensures that you can test your extensions in a real-time environment.

You install and deploy your application in your development environment following the same steps that you used to install and deploy it in your production environment. Refer to your system requirements and installation documentation for details.

You have an option to customize your application with Microsoft COM+. Using Microsoft COM+ has advantages such as increased security, better performance, increased manageability of server applications, and support for clients of mixed environments. If this is your choice, see the *Customization Basics Guide* about additional installation instructions.

Plan Your Customizations

Are you adding a new menu entry? Or customizing the sign-in screen or logo? Or customizing views or wizards? Or creating new themes or new screens? Each type of customization varies in scope and complexity.

For background, see the *Customization Basics Guide*, which summarizes the types of changes that you can make and provides important guidelines about file names, keywords, and other general conventions.

Extend the Database

For many customization projects, the first task is to extend the database so that it supports the other UI or API changes that you make later. For instructions, see the *Extending the Database Guide*, which includes information about the following topics:

- Important guidelines about what you can and cannot change in the database.

- Information about modifying APIs. If you modify database tables so that any APIs are impacted, you must extend the templates of those APIs or you cannot store or retrieve data from the database. This step is required if table modifications impact an API.
- How to generate audit references so that you improve record management by tracking records at the entity level. This step is optional.

Make Other Changes to APIs

Your application can call or invoke standard APIs or custom APIs. For background about APIs and the services architecture of service types, behavior, and security, see the *Customizing APIs Guide*. This guide includes information about the following types of changes:

- Invoke standard APIs for displaying data in the UI and for saving changes made in the UI to the database.
- Invoke customized APIs for executing your custom logic in the extended service definitions and pipeline configurations.
- APIs use input and output XML to store and retrieve data from the database. If you don't extend these API input and output XML files, you may not get the results you want in the UI when your business logic is executing.
- Every API input and output XML file has a DTD and XSD associated to it. Whenever you modify input and output XML, you must generate the corresponding DTD and XSD to ensure data integrity. If you don't generate the DTD and XSD for extended XMLs, you may get inconsistent data.

Customize the UI

IBM® applications support several UI frameworks. Depending on your application and the customizations you want to make, you may work in only one or in several of these frameworks. Each framework has its own process for customizing components such as menu items, logos, themes, and so on.

Depending on the framework you want, consult one of the following guides:

- *Customizing the Console JSP Interface Guide*
- *Customizing the Swing Interface Guide*
- *Customizing User Interfaces for Mobile Devices Guide*
- *Customizing the Rich Client Platform Guide* and *Using the RCP Extensibility Tool Guide*
- *Customizing the Web UI Framework Guide*

Depending on the framework you want, consult one of the following guides:

- *Customizing the Console JSP Interface Guide*
- *Customizing the Swing Interface Guide*
- *Customizing User Interfaces for Mobile Devices Guide*
- *Customizing the Rich Client Platform Guide* and *Using the RCP Extensibility Tool Guide*
- *Customizing the Web UI Framework Guide*

Extend Transactions

You can extend and enhance the standard functionality of your application by extending the Condition Builder and by integrating with external systems. For background about transaction types, security, dynamic variables, and extending the Condition Builder, see the *Extending Transactions Guide* and *Extending the Condition Builder Guide*. These guides includes information about the following types of changes:

- Extend the Condition Builder to define complex and dynamic conditions for executing your custom business logic and using a static set of attributes.
- Define variables to dynamically configure properties belonging to actions, agents, and services configurations.
- Set up transactional data security for controlling who has access to what data, how much they can see, and what they can do with it.
- Create custom time-triggered transactions. You can invoke and schedule custom time-triggered transactions in much the same manner as you invoke and schedule the time-triggered transactions supplied by your application.
- Coordinate your custom, time-triggered transactions with external transactions and run them either by raising an event, calling a user exit, or invoking a custom API or service.

Build and Deploy your Customizations or Extensions

After performing the customizations that you want, you must build and deploy your customizations or extensions.

1. Build and deploy your customizations or extensions in the test environment so you can verify them.
2. When you are ready, repeat the same process to build and deploy your customizations and extensions in your production environment.

For instructions about this process, see the *Customization Basics Guide* which includes information about the following topics:

- Building and deploying standard resources, database extensions, and other extensions (such as templates, user exits, and Java interfaces).
- Building and deploying enterprise-level extensions.

Chapter 2. Basics of Using the Rich Client Platform Extensibility Tool

Getting Started with the Rich Client Platform Extensibility Tool

The Rich Client Platform Extensibility Tool allows you to extend the Rich Client Platform UI by adding new controls, modifying existing controls, and so forth. The tool facilitates the adding of UI controls such as labels, text boxes, combo boxes, list boxes, and so forth. The Rich Client Platform Extensibility Tool provides the ability to change the properties of the existing Rich Client Platform-provided fields. You can also add new fields and specify the layout, bindings, and theme properties for these fields. The Rich Client Platform Extensibility Tool also allows you to synchronize the differences in theme entries, bundle entries, and templates.

Note: Some screens in the Rich Client Platform application cannot be extended using the Rich Client Platform Extensibility Tool. When you perform any operation on such screens using the Rich Client Platform Extensibility Tool, the following message displays:

```
Cannot extend an Inextensible Screen
```

Starting the Rich Client Platform Extensibility Tool

About this task

Before you start extending the screens, you need to start the Rich Client Platform Extensibility Tool.

To extend a screen:

Procedure

1. From the Rich Client Platform application's menu bar, select **File** → **Extend**. The Rich Client Platform Extensibility Tool opens.
2. Click  to load the extension file, if applicable.
3. Click  to start extending the screen.

Loading the Rich Client Platform Extension File

About this task

Before you extend or modify the existing forms using the Rich Client Platform Extensibility Tool, you must load *Plug-in id*_extn.yuix extension file in the Rich Client Platform Extensibility Tool. When you open the Rich Client Platform Extensibility Tool, all registered extension files automatically gets loaded in the tool, and displays in the bottom panel. If the extension files are unregistered, you must register them and then load the files prior to making changes to the form.

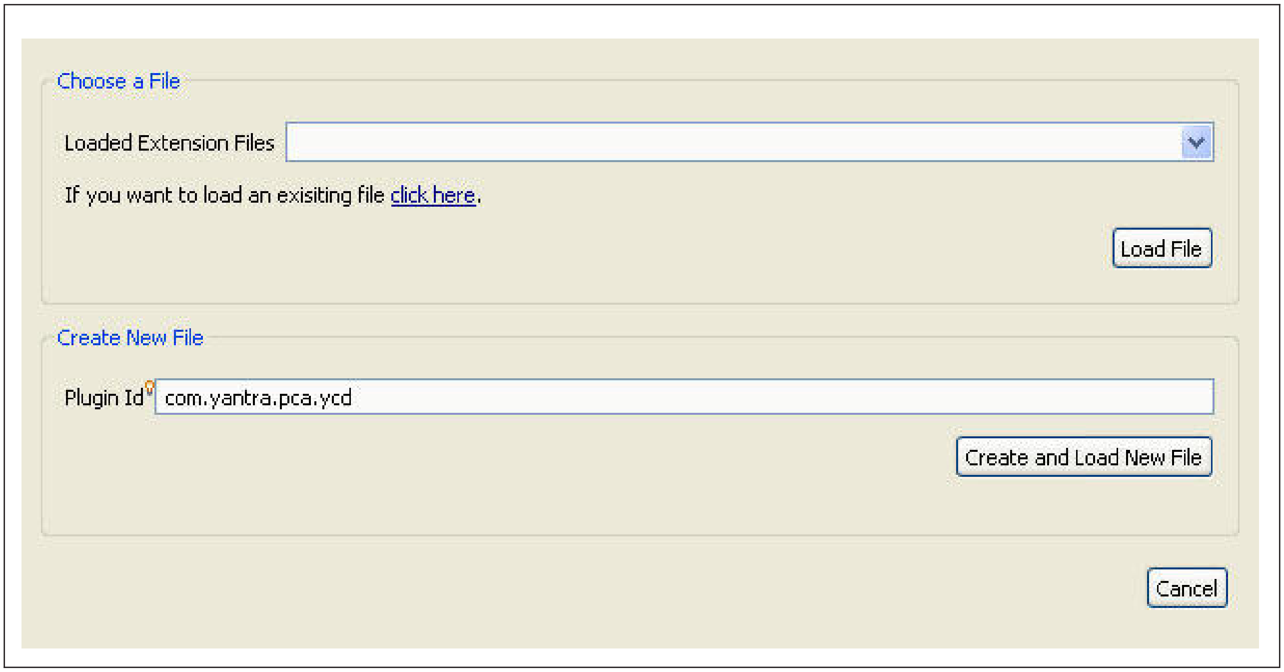
To load the extension files:

Procedure

1. From the menu bar, select **File** → **Extend**. The Rich Client Platform Extensibility Tool opens.

2. Click . The Open dialog displays.
3. Select the *Plug-in id*_extn.yuix extension file.

   If you do not load *Plug-in id*_extn.yuix extension file before extending or modifying forms, the Load/Create Extension File window displays. You can either load an existing extension file or create and load the new extension file.



Field	Description
Choose a File	
Loaded Extension Files	Select the extension file from the drop-down list, if applicable. If an extension file is not loaded in the tool, this combo box gets disabled.
click here	Click this hyperlink. The Choose File dialog displays. Select an extension file in which a new form element is automatically created. Click open. The extension file that you selected automatically gets loaded in the tool.
Load File	Click the Load File button to load the extension file that you selected in the Loaded Extension Files combo box.
Create New File	
Plugin Id	Enter the plugin identifier of the plug-in that registers this new extension file. Press **Ctrl+Space** to select the plug-in Id from the list of available plug-ins.
Create and Load New File	When you click this button, the Save File dialog displays. If you are creating a new extension file, enter the extension file name and click Save. The new extension file gets created and loaded automatically. Otherwise, select an existing extension file in which a new form element is automatically created. Click **Save**. The extension file gets automatically loaded in the tool.

Viewing Screen Information

About this task

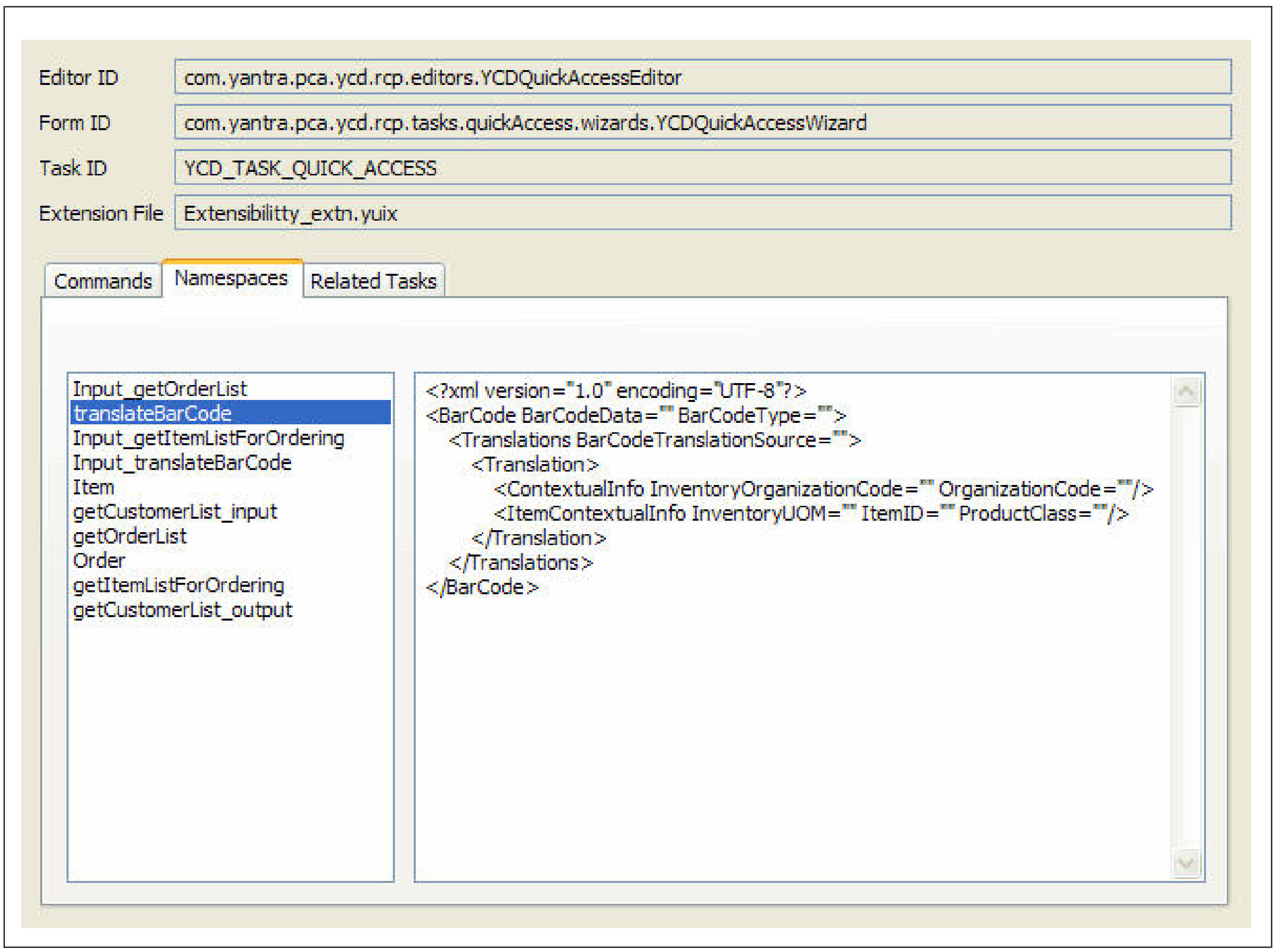
You can view the following information for a screen:

- Identifier of the editor contained by the screen.
- Identifier of the form is used to identify the screen.
- Identifiers of the task associated with an editor.
- *Plug-in_id*_extn.yuix extensions file associated to a particular screen.
- List of all the commands used by the screen to call APIs or services to get the required data for the screen.
- List of all the namespaces defined for the screen.

To view screen information:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click i Screen Info . The Screen Information window displays.



Field	Description
Editor ID	The identifier of the editor contained by the screen.
Form ID	The identifier of the form is used to identify the screen.
Task ID	The identifiers of the task associated with the editor.
Extension File	The *Plug-in_id*_extn.yuix extensions file associated with the screen.

3. Select the commands tab to view all the commands that are being used by the screen.
4. Select the Namespaces tab to view the list of namespaces that are used by the screen. You can also view the XML model of the output template associated with the a namespace in the right hand side panel.
5. Select the Related Tasks tab to view the list of category id's in which this current active task is interested in. You can also view the list of group id's along with their group sequence number associated with the related tasks of the current active task.

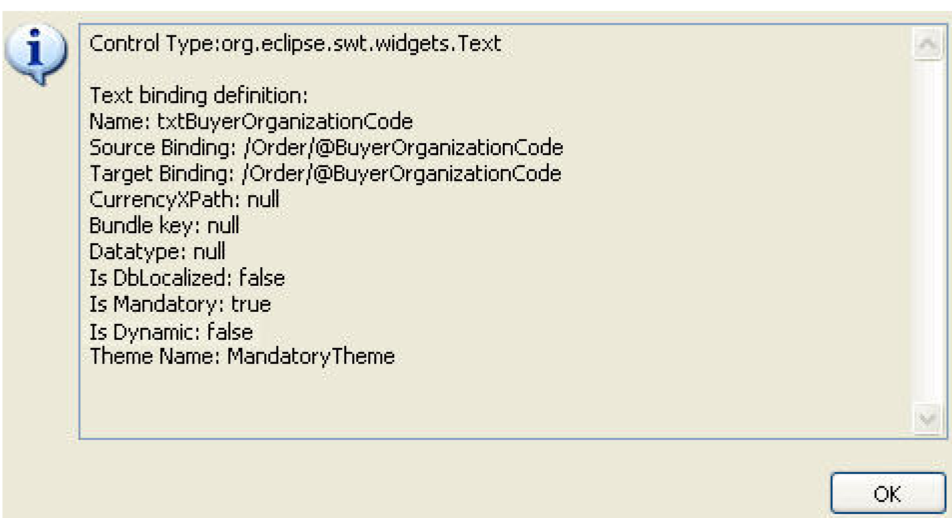
Viewing Control Information

About this task

To view control information:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click Control Info . The Control Information window displays. The following figure displays the sample control info screen for a text box control.



The system provides the following information for various controls:

- Control Type—Type of the control. For example, composite, group, button, and so forth.
- Name—Name of the control if applicable. For example, rdOpen, cmbPrice.

- Control Binding Definition—Information about the various bindings defined for a control. For example, source binding, target binding, checked binding.
- Theme Information—Theme applied to a control is displayed. For example, Theme Name: Mandatory Theme.

For more information about binding definitions for each control, see the Javadocs.

Building Rich Client Platform Extensions

About this task

Building the Rich Client Platform extensions is as follows:

Procedure

1. Start the Eclipse SDK.
2. From the menu bar, select **Window** → **Show View** → **Navigator**. The plug-in project is displayed in the Navigator view.
3. Right-click on the plug-in project that you want to build and deploy.
4. Select **Export...** from the pop-up menu.

   The Export window displays.
5. From the list of export destinations, under Plug-in Deployment, select **Deployable plug-ins and fragments**.
6. Click **Next**.
7. In the Destination tab, Choose **Archive file:**.
8. Click **Browse** and browse to the folder where you want to store the exported plug-in compressed file.
9. In the Options tab, make sure that the Package plug-ins as individual JAR archives box is checked.
10. Click **Finish**. The plug-in jar is generated and stored in the plugins folder in the compressed file specified in Step 8.

Deploying Rich Client Platform Extensions

About this task

After you build the Rich Client Platform extensions plugin jar, you must deploy this plug-in.

To deploy the Rich Client Platform extensions, copy the plugin jar that you built to the plugins directory of the *RCP_EXTN_FOLDER* folder and follow the steps as described in Sterling Selling and Fulfillment Foundation *Installation Guide*.

Chapter 3. Adding and Moving Controls and Table Columns

Adding a Label

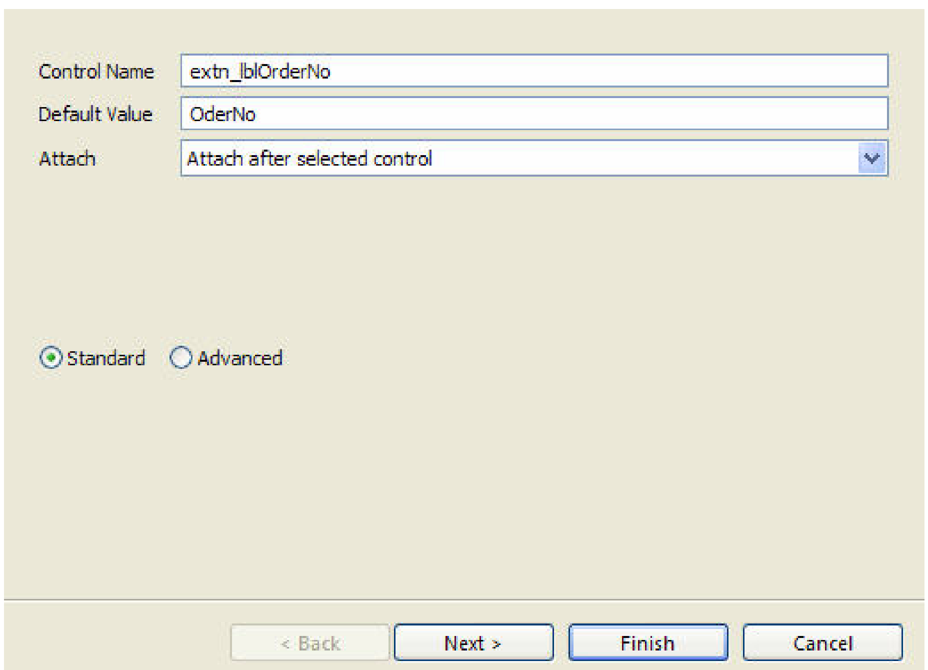
About this task

To add a label:

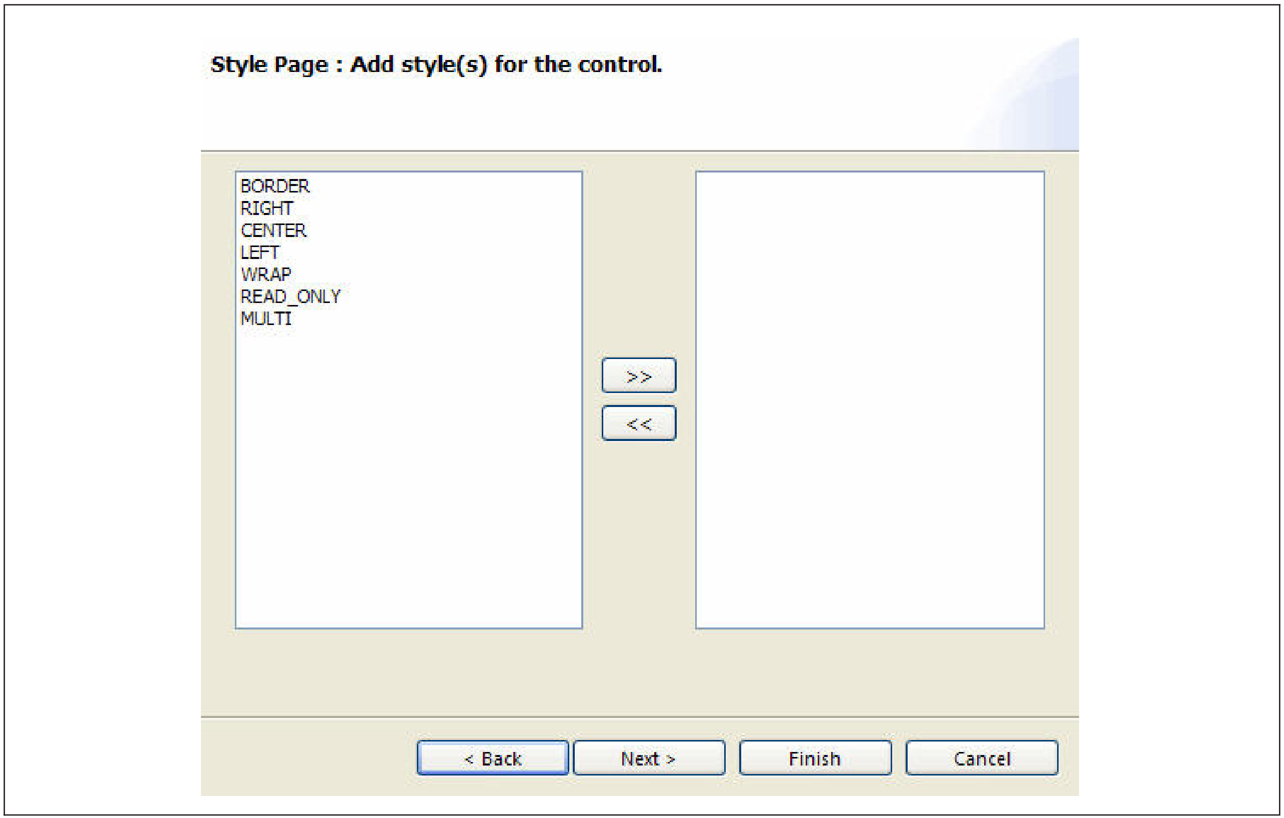
Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click Label .
3. To add a label, do one of the following:
   - Select the control where you want to add the label and click once.
   - Select the composite or group where you want to add the label and click once.

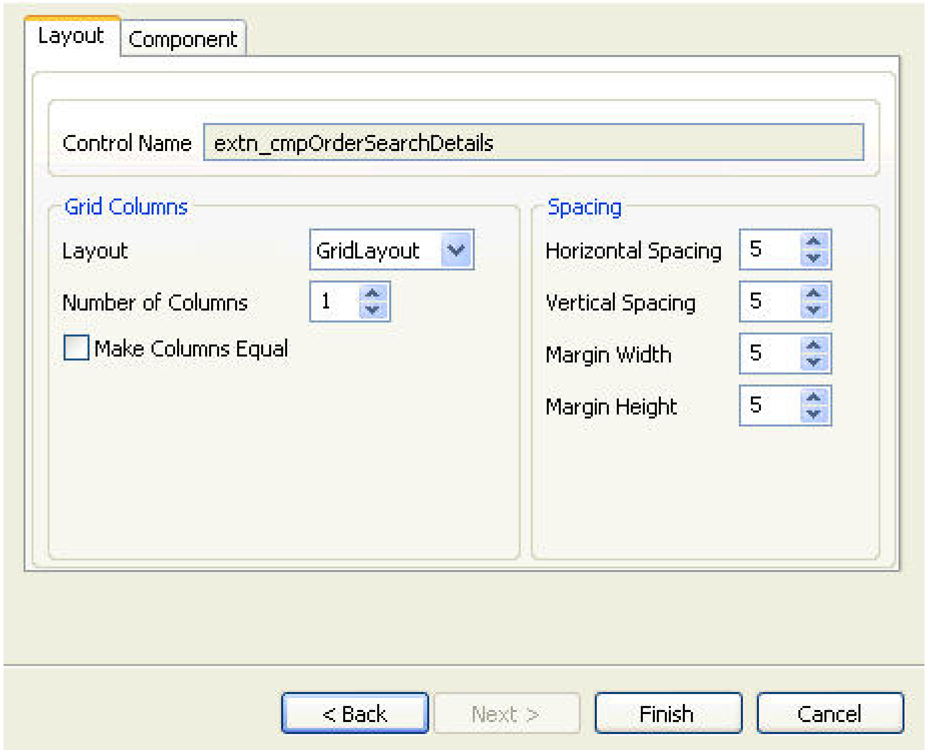
   The Add Label window displays.



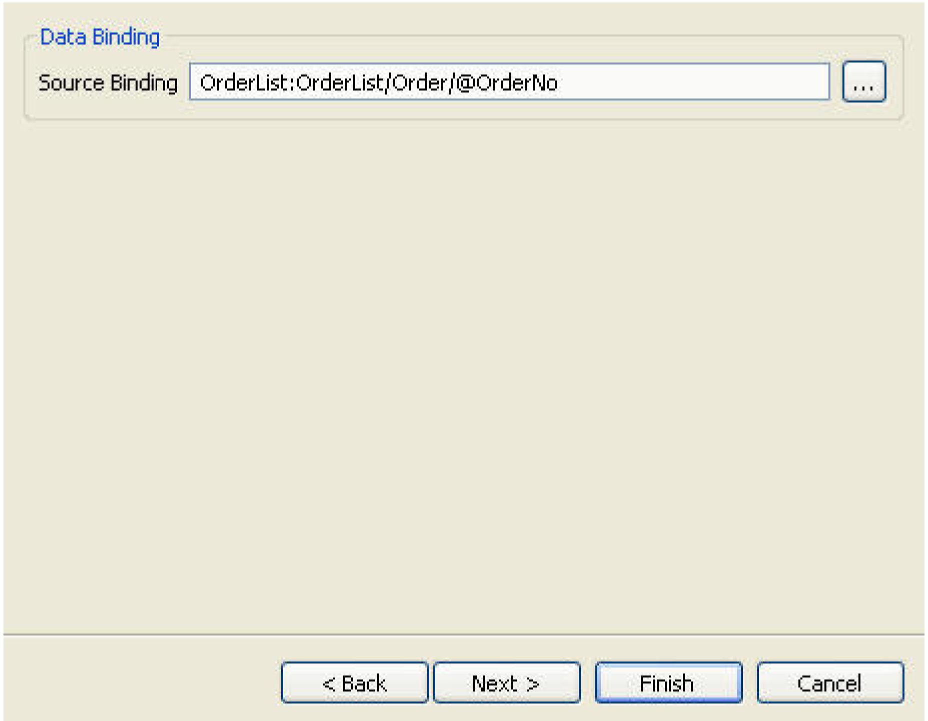
   For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."
4. Click **Finish** to add the new label with the default style(s), layout data, theme, and null binding attributes.
5. Click **Next** to specify the style options.



6. Click **Finish** to add the new label with the default layout data, theme, and null binding attributes.
7. Click **Next** to specify the data layout options.

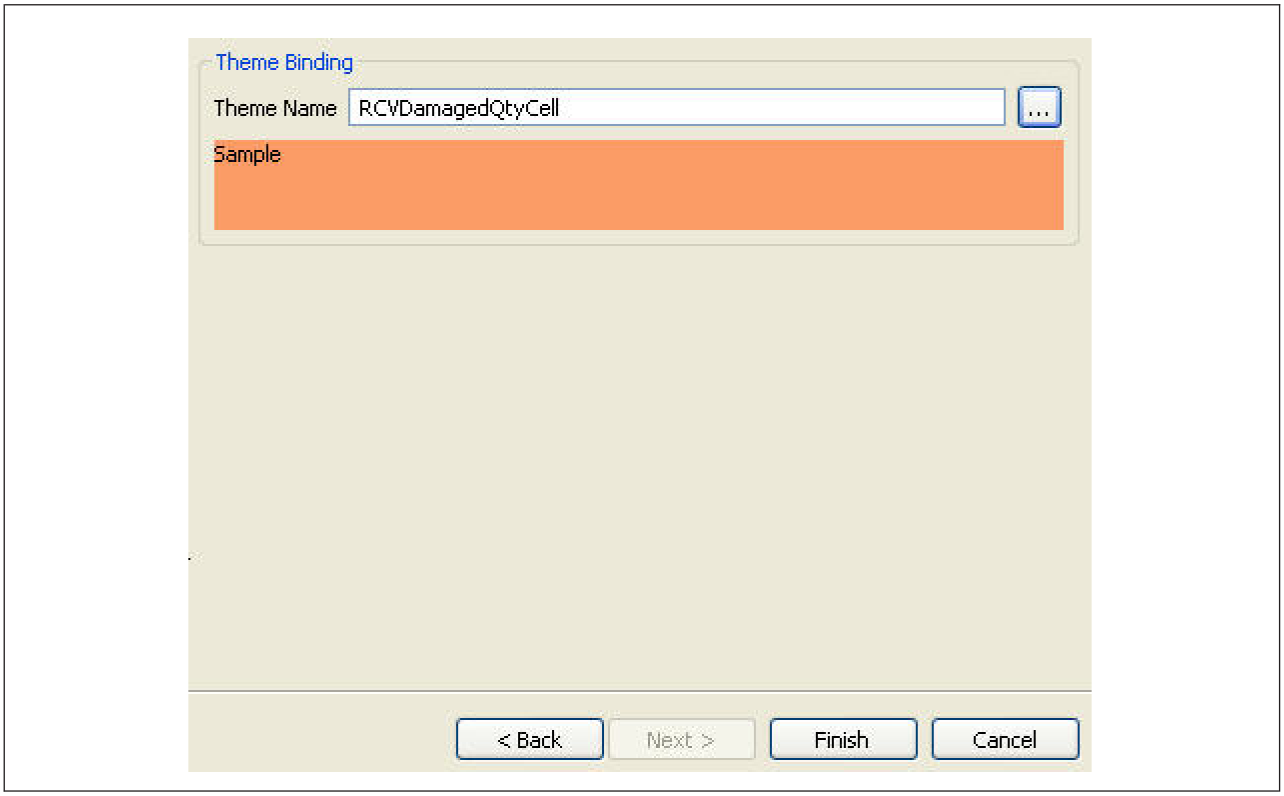


8. Click **Finish** to add the new label with the default layout data, theme and null binding attributes.
9. Click **Next** to specify the binding options.



These are the fields and descriptions for the Data Binding window:

10. Click **Finish** to add the new label with the default theme binding.
11. Click **Next** to specify the theme binding.



12. Click **Finish**.
13. Click to save the changes made to the extension file.
14. If you have specified the default value, source binding, or theme entry for the label, synchronize the resource files.

    For information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Adding a Button

About this task

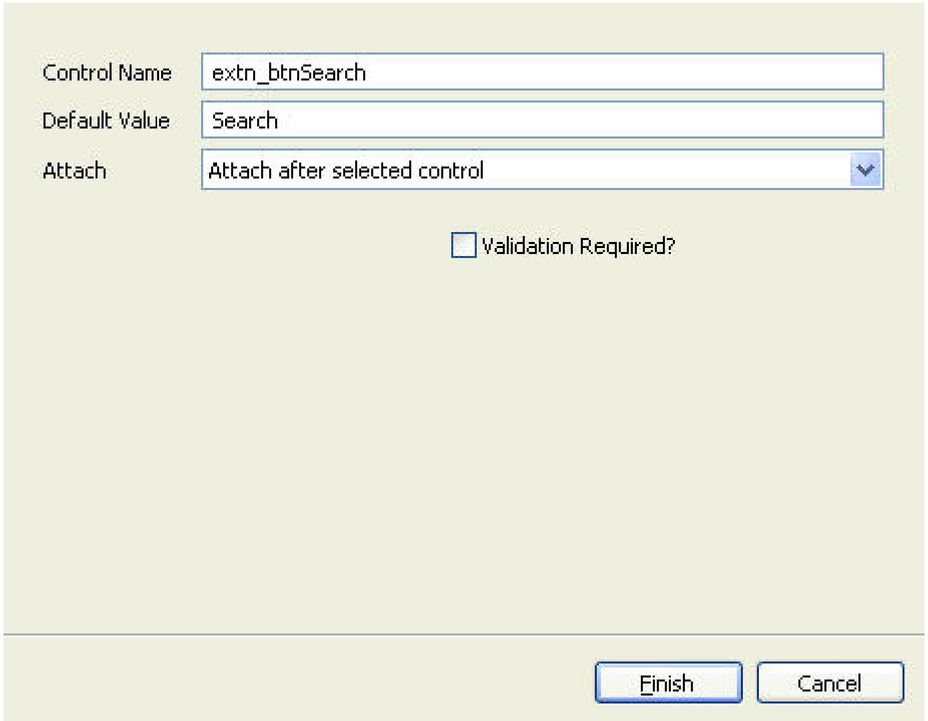
To add a button:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click Button .
3. Perform any of the following tasks to add a button:
   - Select the control where you want to add the button and click once.
   - Select the composite or group where you want to add the button and click once.

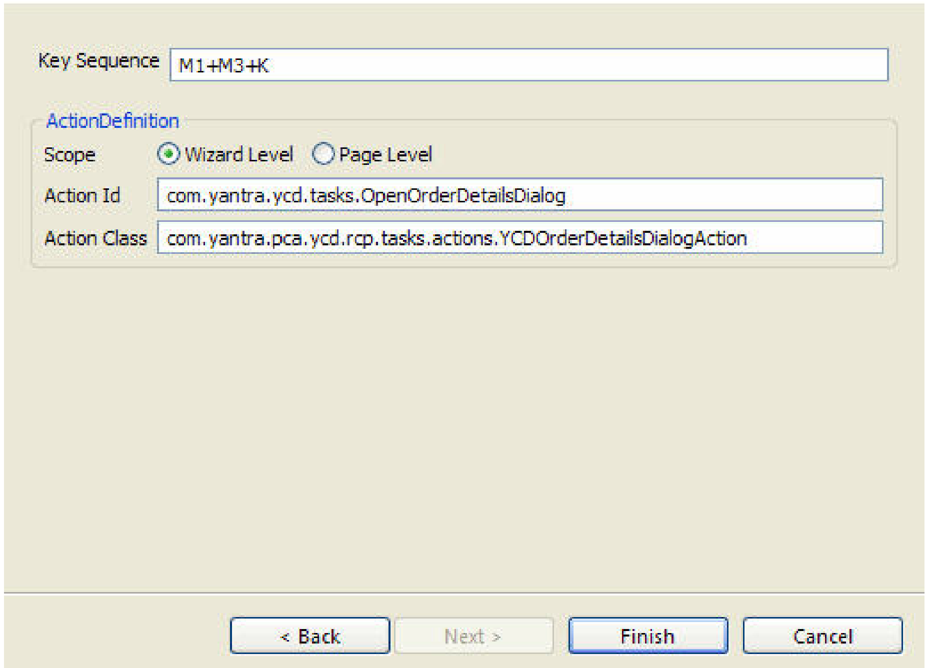
   The Add Button window displays.

   Note: You can place the button either in the row layout container or grid layout container. However, the row layout container does not have the layout data. Therefore, if you place the button in the row layout container, the Layout Setup Page is disabled.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

4. Click **Finish** to add the new button with the default style(s), layout data, theme, and empty hot key binding.
5. Click **Next** to specify the style options.
6. Click **Finish** to add the new button with the default layout data, theme, and empty hot key binding.
7. Click **Next** to specify the layout data options.
8. Click **Finish** to add the new button with the default theme and empty hot key binding.
9. Click **Next** to specify the theme binding.
10. Click **Finish** to add the new button with empty hot key binding.
11. Click **Next** to specify the hot key binding.



Note: If you enter and save the hot key binding, you cannot modify it.

12. Click **Finish**.

    If you have specified the hot key binding, the Restart Application pop-up window displays and prompts you to save the extension files and restart the application to view the changes.

    Note: Whenever you specify the hot key binding for a control, you must save the extension file and restart the application to view the changes made to the hot key binding. Ensure that you clear the configuration data before starting the application.

13. Click  to save the changes made to the extension file.

14. If you have specified the default value for the button, synchronize the resource files.

    For information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

15. (Optional) To implement the logic on the button click event:

    Note: When adding the new button, make sure that you check the Validation Required? box.

    a. Synchronize the extension behavior for the button.

    b. Start the Eclipse SDK.

    c. In the navigator view, expand the plug-in project that contains this screen.

    d. Expand the package and open the extension behavior class that you specified when synchronizing the extension behavior.

    e. In the validateButtonClick() method, add logic to provide the desired implementation for the button click event. On the link click event, you can display a new screen either in an editor or as a pop-up screen. For more information about adding new screens, see the Sterling Selling and Fulfillment Foundation: *Customizing Rich Client Platform Interface Guide*. In addition, if you want to invoke an action on the button click event, call the fireAction(String actionId) method of the YRCPlatformUI class in the validateButtonClick() method. The fireAction() method returns a boolean value: `true` or `false`. If the method returns a `true` value, it indicates that the invoked action was found and executed. For example:

```
YRCPlatformUI.fireAction(actionOpenMyView);
```

where `actionOpenMyView` is the identifier of the action that needs to be invoked. For more information about eclipse actions, see the Sterling Selling and Fulfillment Foundation: *Customizing Rich Client Platform Interface Guide*.

Adding a Checkbox

About this task

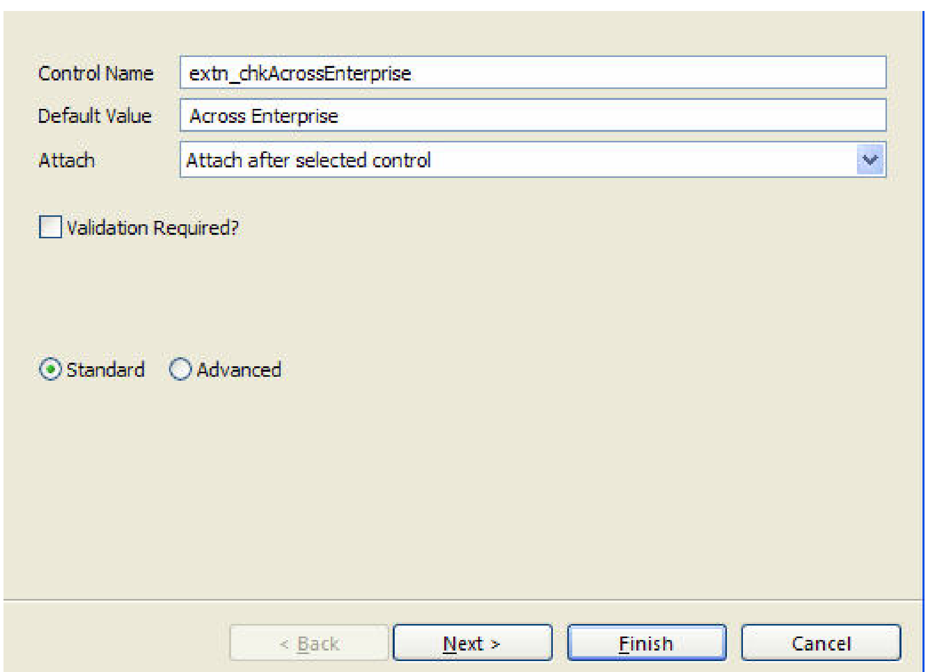
To add a check box:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Do any of the following to add a checkbox:
   - Select the control where you want to add the checkbox and click once.
   - Select the composite or group where you want to add the checkbox and click once.

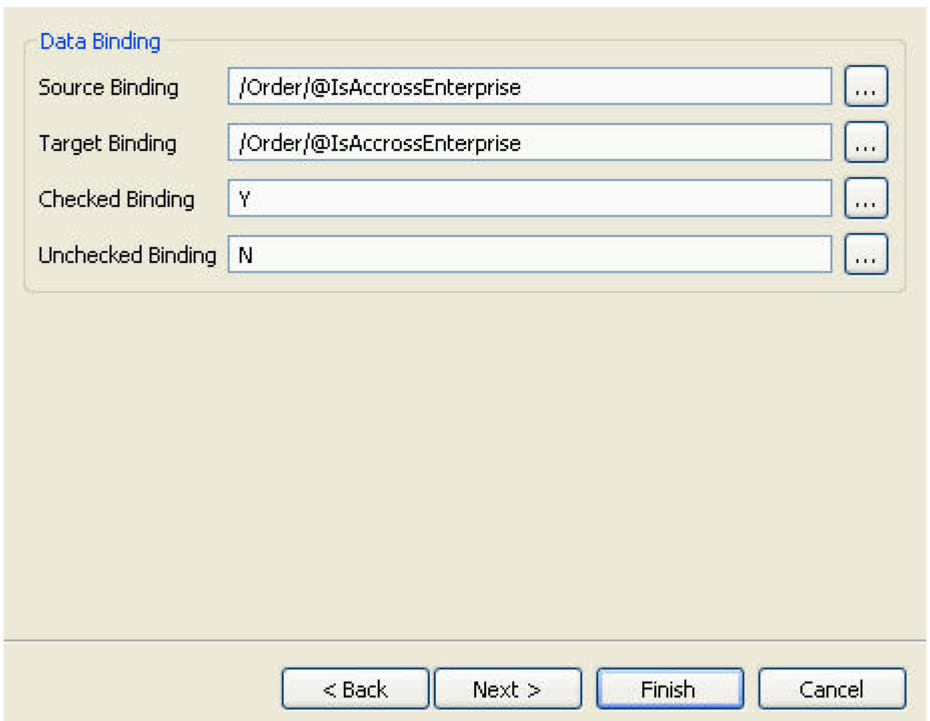
   The Add Checkbox window displays



Note: You can place the checkbox either in the row layout container or grid layout container. However, the row layout container does not have the layout data. Therefore, if you place the checkbox in the row layout container, the Layout Setup Page is disabled.

For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

3. Click **Finish** to add the new checkbox with the default style(s), layout data, theme, and null binding attributes.
4. Click **Next** to specify the style options.
5. Click **Finish** to add the new checkbox with the default layout data, theme, and null binding attributes.
6. Click **Next** to specify the layout data options.
7. Click **Finish** to add the new checkbox with the default theme and null binding attributes.
8. Click **Next** to specify the binding options.



9. Click **Finish** to add the new checkbox with the default theme binding.
10. Click **Next** to specify the theme binding.
11. Click **Finish**.
12. Click  to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the checkbox, synchronize the resource files.

    For information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Adding a Radio Button

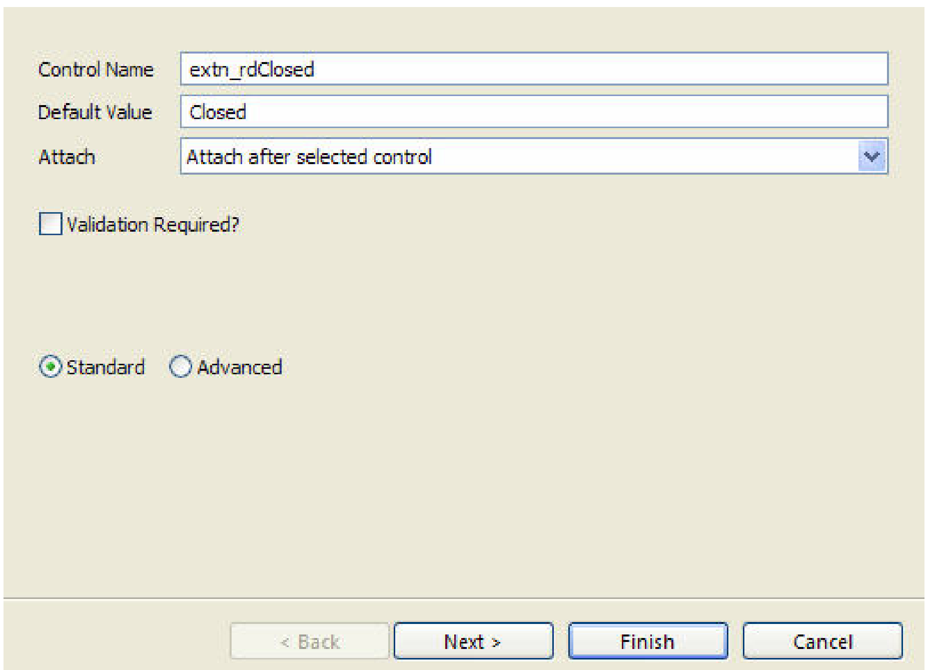
About this task

To add a radio button:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click Radio Button . Do any of the following to add a radio button:
   - Select the control where you want to add the radio button and click once.
   - Select the composite or group where you want to add the radio button and click once.

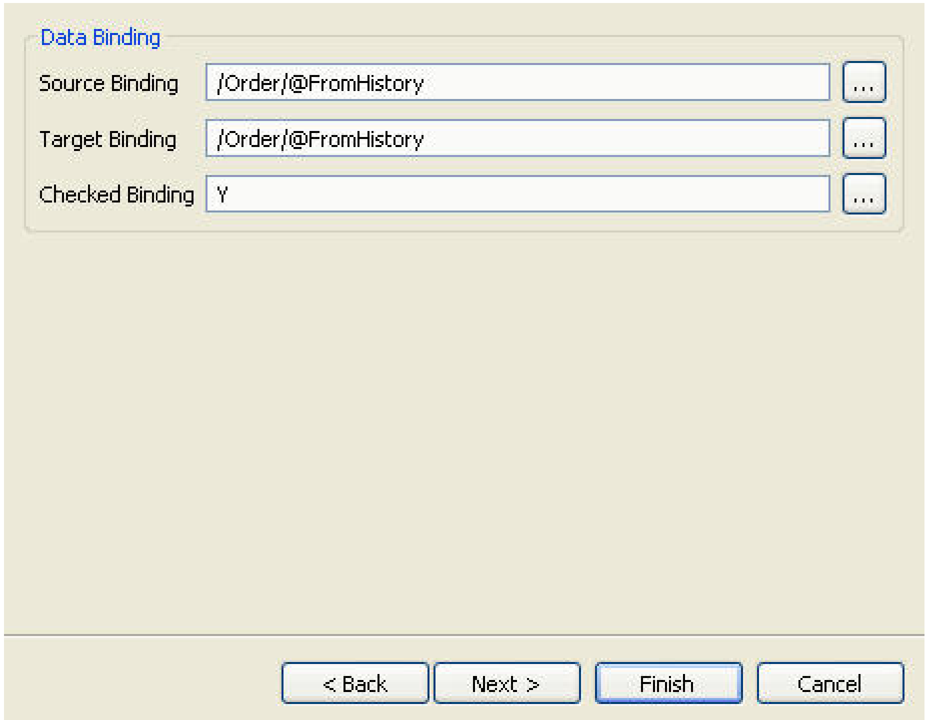
   The Add Radio Button window displays.



   Note: You can place the radio button either in the row layout container or grid layout container. However, the row layout container does not have the layout data. Therefore, if you place the radio button in the row layout container, the Layout Setup Page is disabled.

   For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."
3. Click **Finish** to add the new radio button with the default style(s), layout data, theme, and null binding attributes.
4. Click **Next** to specify the style options.
5. Click **Finish** to add the new radio button with the default layout data, theme, and null binding attributes.
6. Click **Next** to specify the layout data options.
7. Click **Finish** to add the new radio button with the default theme and null binding attributes.

8. Click **Next** to specify the binding options.



9. Click **Finish** to add the new radio button with the default theme binding.
10. Click **Next** to specify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the radio button, synchronize the resource files.

    For information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Adding a Text Box

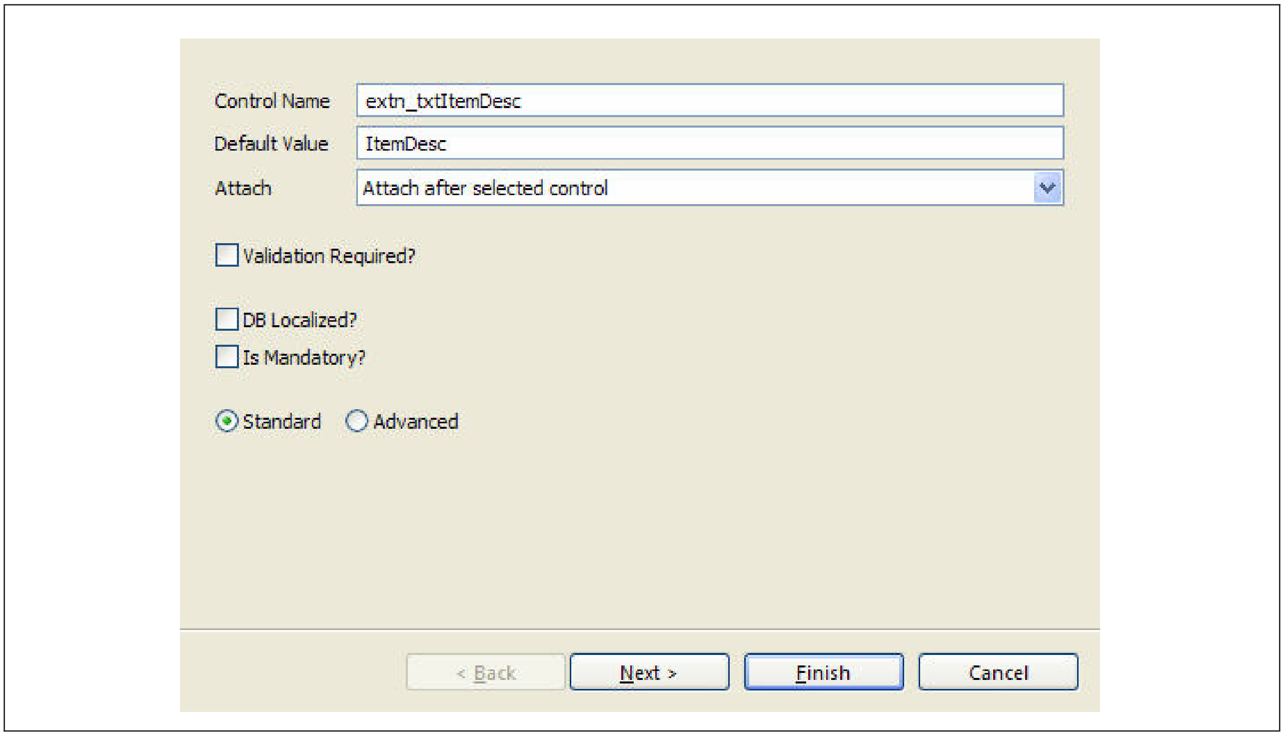
About this task

To add a text box:

Procedure

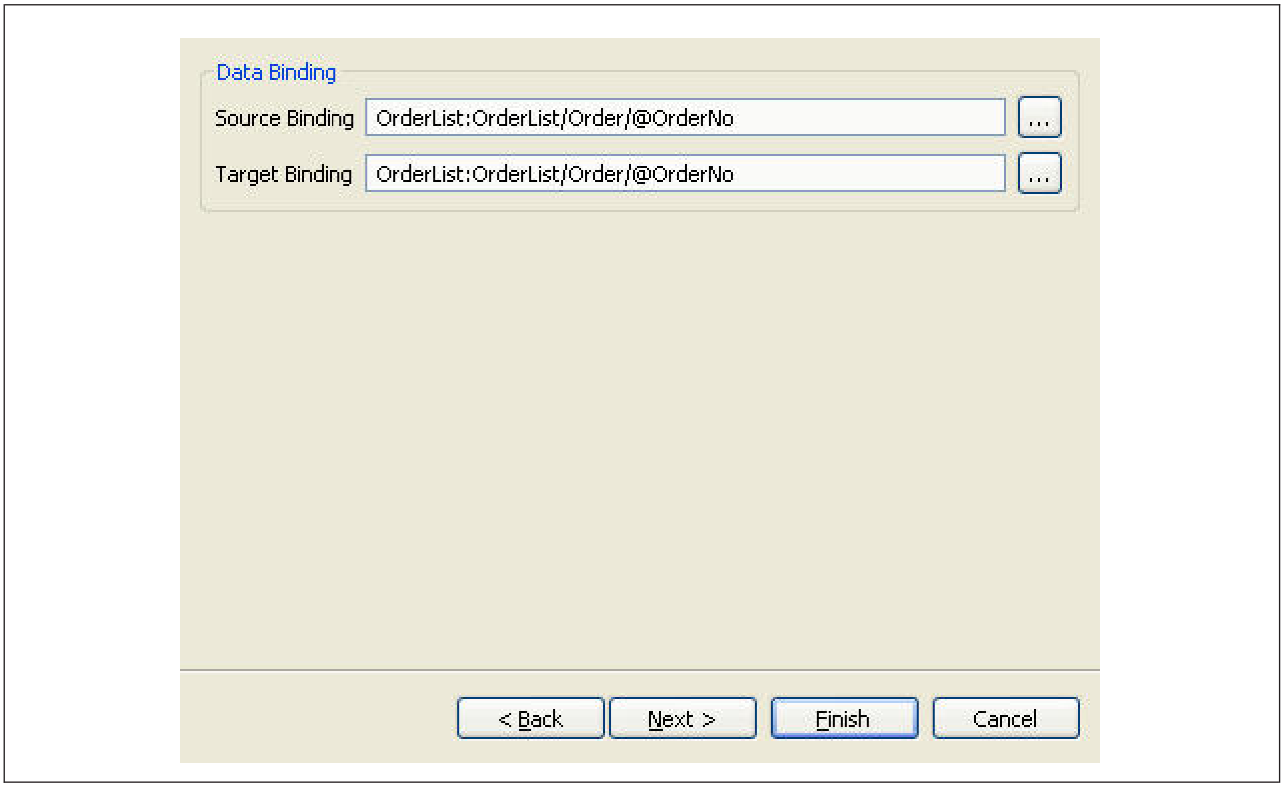
1. Start the Rich Client Platform Extensibility Tool.
2. Click Text Box . Do any of the following to add a text box:
   - Select the control where you want to add the text box and click once.
   - Select the composite or group where you want to add the text box and click once.

   The Add Text Box window displays.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

3. Click **Finish** to add the new text box with the default style(s), layout data, theme, and null binding attributes.
4. Click **Next** to specify the style options.
5. Click **Finish** to add the text box with the default layout data, theme, and null binding attributes.
6. Click **Next** to specify the layout data options.
7. Click **Finish** to add the text box with the default theme and null binding attributes.
8. Click **Next** to specify the binding options.



9. Click **Finish** to add the text box with the default theme binding.
10. Click **Next** to specify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the text box, synchronize the resource files.

    For information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Adding StyledText Component

About this task

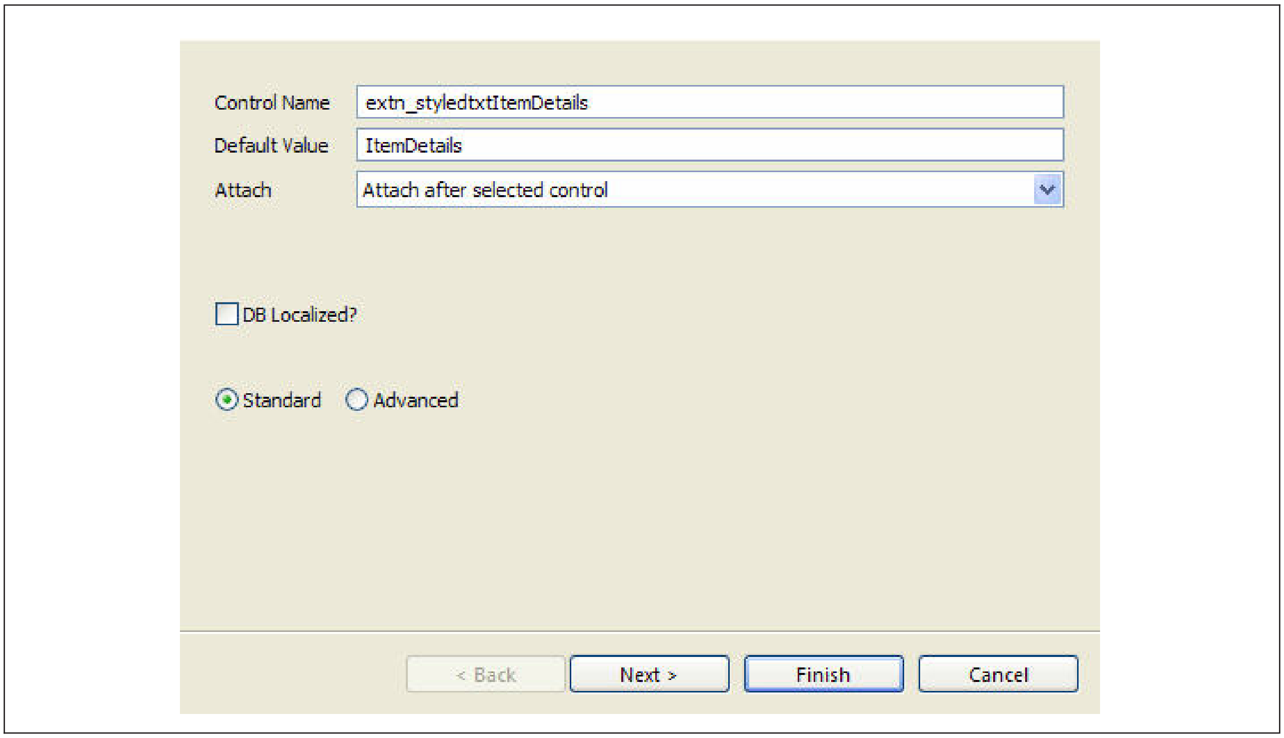
The StyledText component is a customized control that can be used to display and edit text with different colors and font styles. The StyledText components are powerful and multifaceted components suitable for high-end needs and offer more avenues for customization in comparison to other text components.

To add styled text:

Procedure

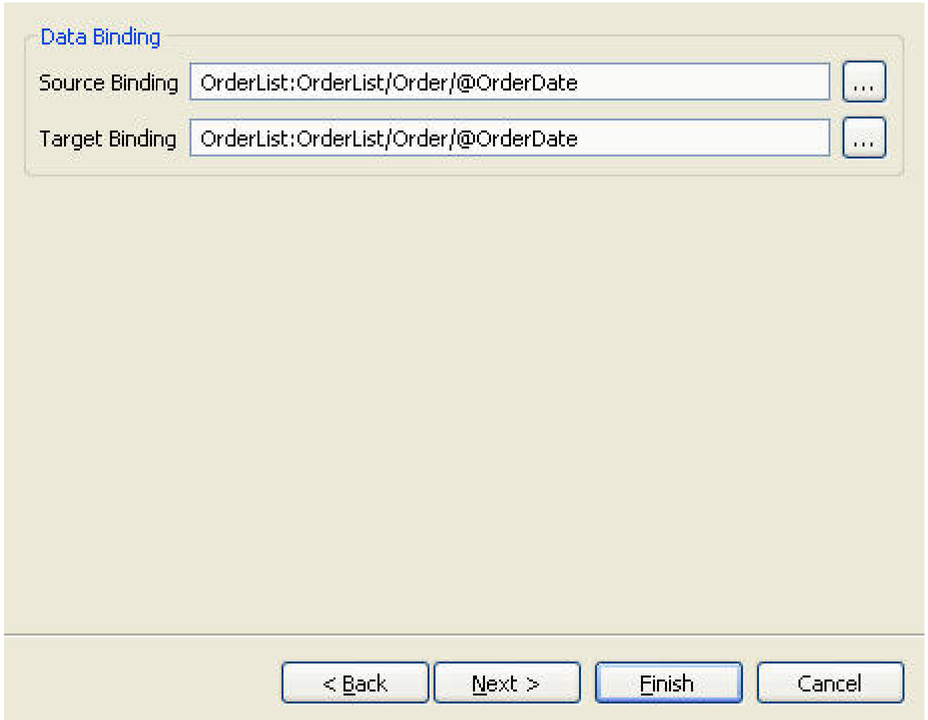
1. Start the Rich Client Platform Extensibility Tool.
2. Click StyledText . Do any of the following to add styled text:
   - Select the control where you want to add the styled text and click once.
   - Select the composite or group where you want to add the styled text and click once.

   The Add Styled Text window displays.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

3. Click **Finish** to add the new styled text with the default style(s), layout data, theme, and null binding attributes.
4. Click **Next** to specify the style options.
5. Click **Finish** to add the new styled text with the default layout data, theme, and null binding attributes.
6. Click **Next** to specify the layout data options.
7. Click **Finish** to add the styled text with the default theme and null binding attributes.
8. Click **Next** to specify the binding options.



9. Click **Finish** to add the styled text with the default theme binding.
10. Click **Next** to specify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the styled text, synchronize the resource files.

    For information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Adding a Combo Box

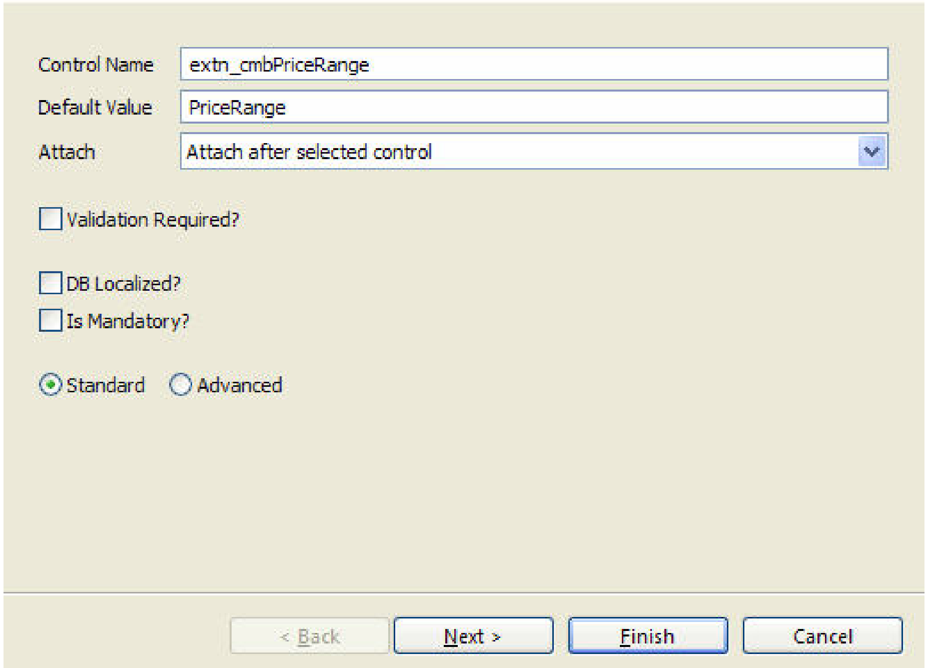
About this task

To add a combo box:

Procedure

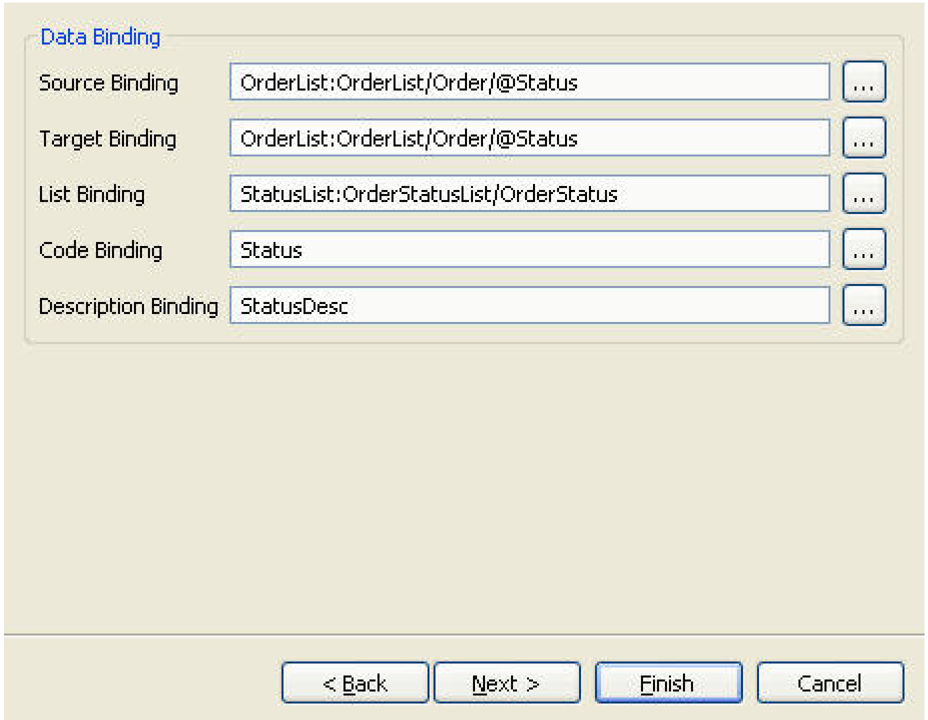
1. Start the Rich Client Platform Extensibility Tool.
2. Click Combo Box. Do any of the following to add a combo box:
   - Select the control where you want to add the combo box and click once.
   - Select the composite or group where you want to add the combo box and click once.

   The Add Combo Box window displays.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

3. Click **Finish** to add the new combo box with the default style(s), layout data, theme, and null binding attributes.
4. Click **Next** to specify the style options.
5. Click **Finish** to add the new combo box with the default layout data, theme, and null binding attributes.
6. Click **Next** to specify the layout data options.
7. Click **Finish** to add the new combo box with the default theme and null binding attributes.
8. Click **Next** to specify the binding options.



9. Click **Finish** to add the new combo box with the default theme binding.
10. Click **Next** to specify the theme binding.
11. Click **Finish**.
12. Click on save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the combo box, synchronize the resource files.

    For information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Adding a List Box

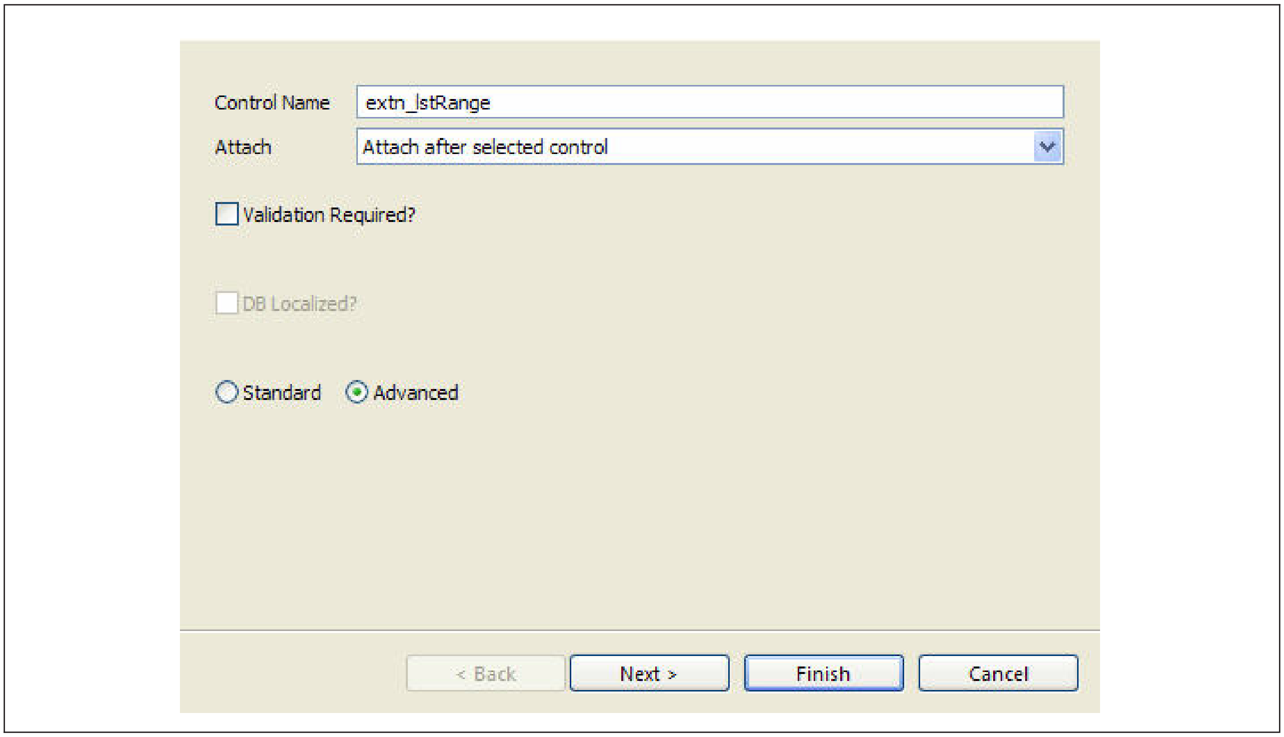
About this task

To add a list box:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click List Box . Do any of the following to add a list box:
    - Select the control where you want to add the list box and click once.
    - Select the composite or group where you want to add the list box and click once.

    The Add List Box window displays.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

3. Click **Finish** to add the new list box with the default style(s), layout data, theme, and null binding attributes.
4. Click **Next** to specify the style options.
5. Click **Finish** to add the list box with the default layout data, theme, and null binding attributes.
6. Click **Next** to specify the layout data options.
7. Click **Finish** to add the list box with the default theme and null binding attributes.
8. Click **Next** to specify the binding options.
9. Click **Finish** to add the list box with the default theme binding.
10. Click **Next** to specify the theme binding.
11. Click **Finish**.
12. Click  to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the list box, synchronize the resource files.

    For information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."
14. Make sure that you create the extension behavior for the screen on which you are adding this custom control. Also, you need to override the getBindingdata() method to return the custom control binding data object. This method is called by Rich Client Platform while creating the custom control.

Adding a Table Column

About this task

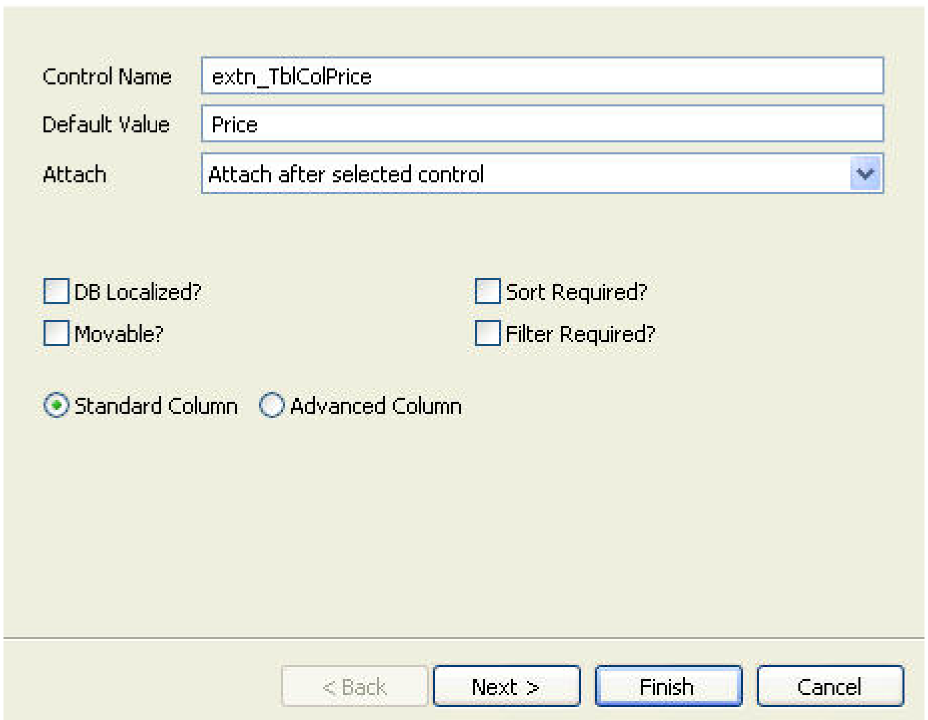
You can add either a standard column or advanced column to a table.

To add a standard or advanced column to a table:

Procedure

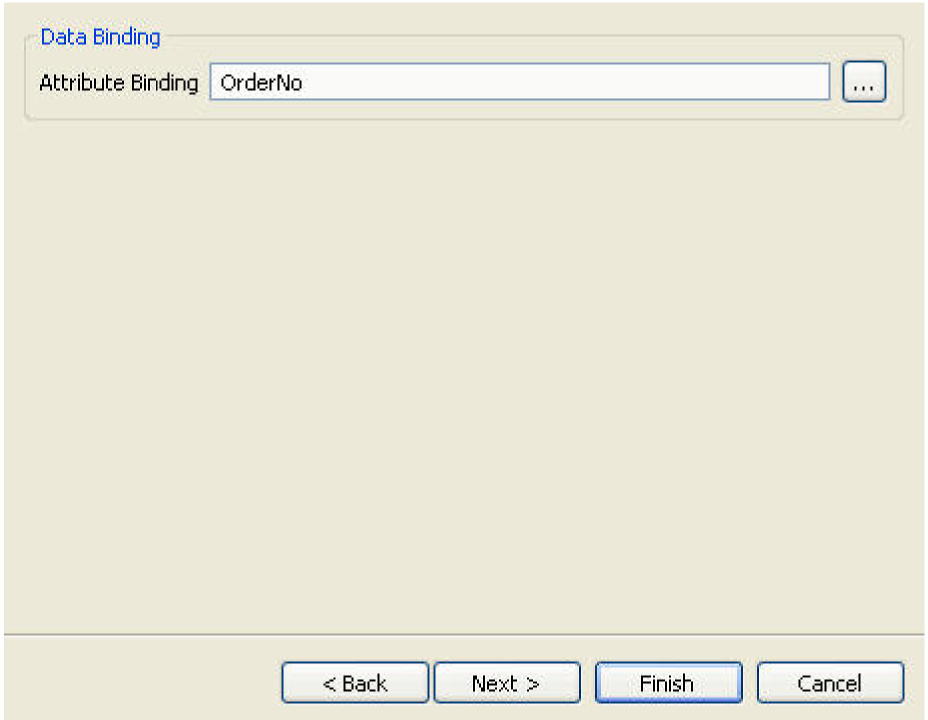
1. Start the Rich Client Platform Extensibility Tool.
2. Click Table Column . Select the column where you want to add the new standard or advanced column and click once.

   The Add Table Column window displays.

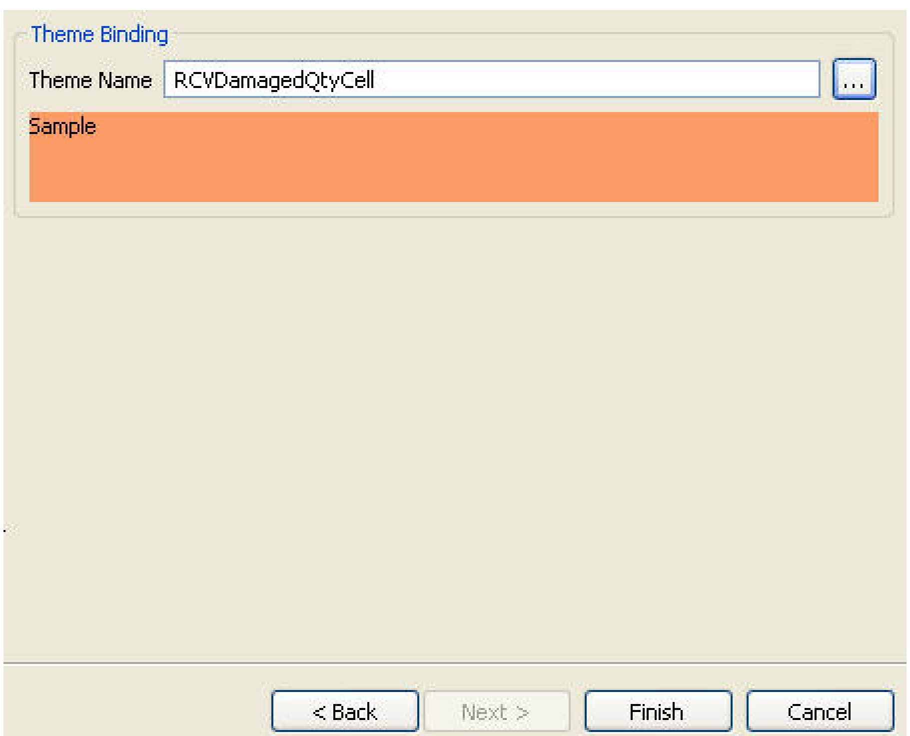


   For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."
3. Click **Finish** to add the new column with the default column layout data, theme, and null binding attributes.
4. Click **Next** to specify the column layout options.

5. Increase or decrease the width of the column, if necessary.
6. Click **Finish** to add the column with the default theme and null binding attributes.
7. Click **Next** to specify the binding options.



8. Click **Finish** to add the new column with the default theme binding.
9. Click **Next** to specify the theme binding.



10. Click **Finish**.
11. Click  to save the changes made to the extension file.
12. If you have specified the default value, bindings, or theme entry for the table column, synchronize the resource files.

    If you add an advanced column, create and synchronize the appropriate extension behavior for setting the bindings for the extended table.

    For information on creating and synchronizing extensions, see "Creating Extension Behavior" on page 54.

What to do next

After creating and synchronizing the extension behavior, set the bindings for the extended table and advanced column you added in the extension behavior class.

Adding a Link

About this task

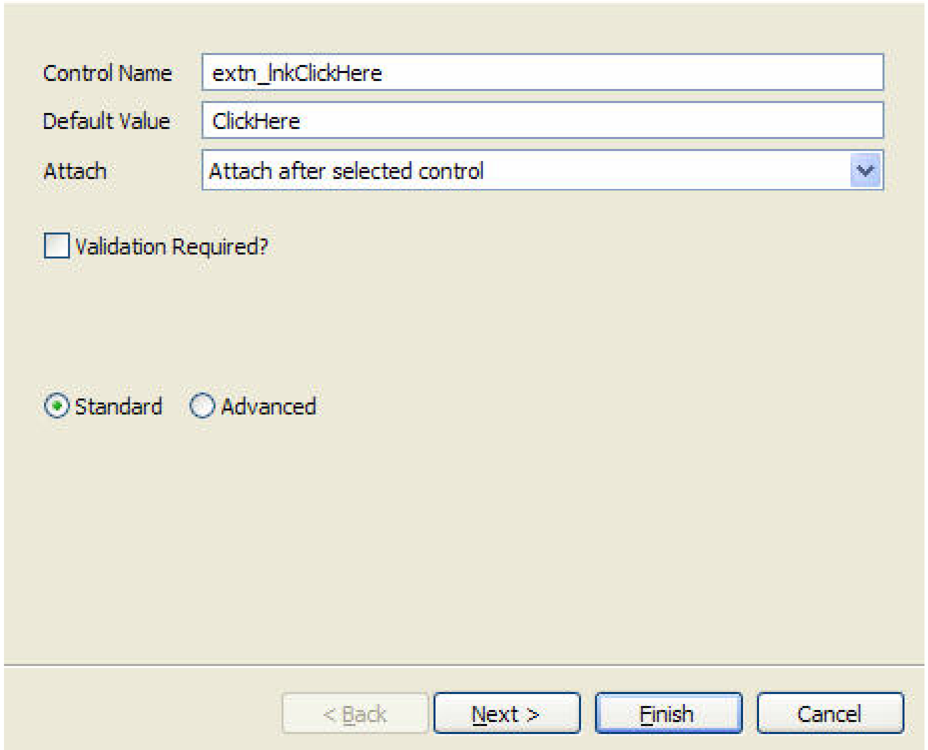
To add a link:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Do any of the following to add a link:
   - Select the control where you want to add the link and click once.
   - Select the composite or group where you want to add the link and click once.

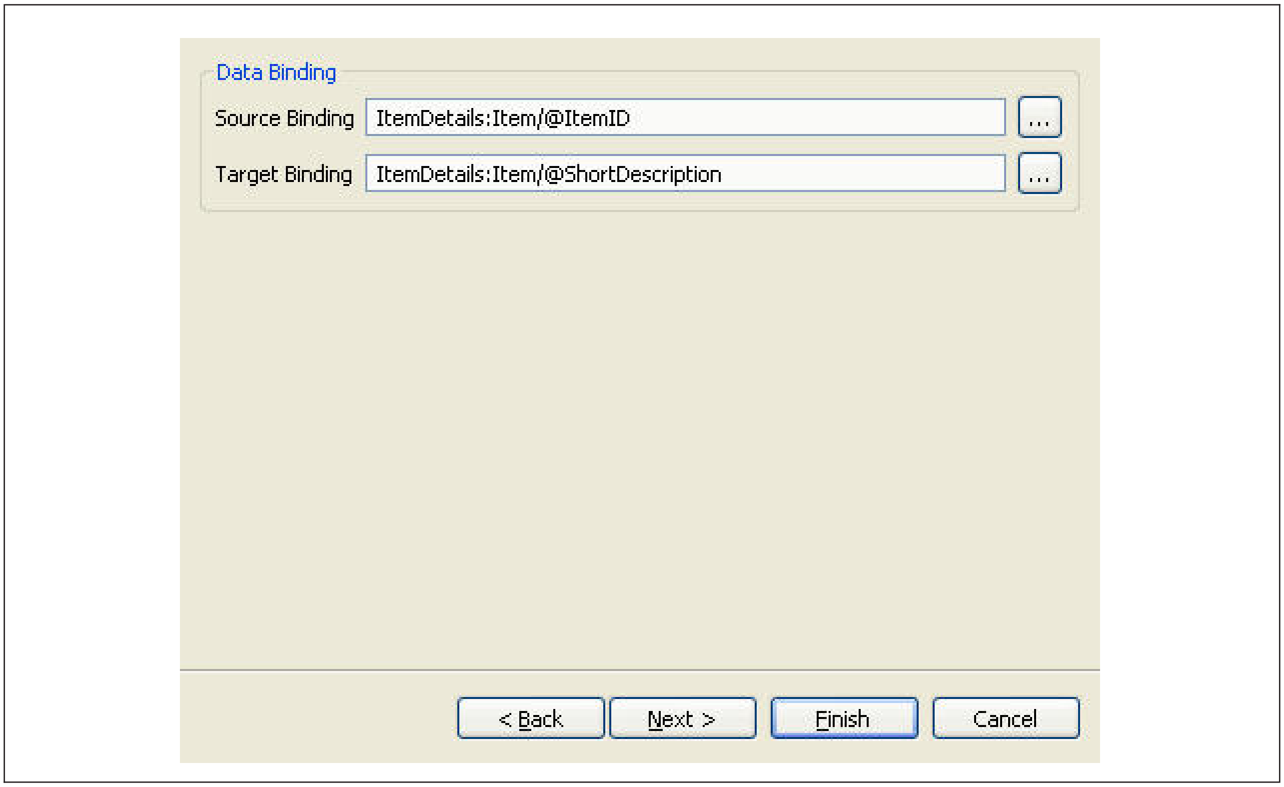
   The Add Link window displays.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

Note: You can place the link either in the row layout container or grid layout container. However, the row layout container does not have the layout data. Therefore, if you place the link in the row layout container, the Layout Setup Page is disabled.

3. Click **Finish** to add the new link with the default layout data, theme, null binding attributes, and empty hot key binding.
4. Click **Next** to specify the layout data options.
5. Click **Finish** to add the link with the default theme, null binding attributes, and empty hot key binding.
6. Click **Next** to specify the binding options.



7. Click **Finish** to add the link with the default theme binding and empty hot key binding.
8. Click **Next** to specify the theme binding.
9. Click **Finish** to add the link with the empty hot key binding.
10. Click **Next** to specify hot key binding.

    Note: If you are adding an advanced link, hot key binding page is disabled. You cannot specify the hot key binding for an advanced link using the Rich Client Platform Extensibility Tool.
11. Click **Finish**.

    If you have specified the hot key binding, the Restart Application pop-up window displays and prompts you to save the extension file and restart the application to view the changes.

    Note: Whenever you specify the hot key binding for a control, you must save the extension file and restart the application to view the changes of the hot key binding. Make sure that you clear the configuration data before restarting the application.
12. Click  to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the link, synchronize the resource files.

    For information on creating and synchronizing extensions, see "Creating Extension Behavior."
14. (Optional) To implement the logic on the link click event:

    Note: When adding the new link, make sure that you check the Validation Required? box.

a. Synchronize the extension behavior for the link

b. Start the Eclipse SDK.

c. In the navigator view, expand the plug-in project that contains this screen.

d. Expand the package and open the extension behavior class that you specified when synchronizing the extension behavior.

e. In the validateLinkClick() method, add logic to provide the desired implementation for the link click event. On the link click event, you can display a new screen either in an editor or as a pop-up screen. For more information about adding new screens, see theSterling Selling and Fulfillment Foundation *Customizing Rich Client Platform Interface Guide*. In addition, if you want to invoke an action on the link click event, call the fireAction(String actionId) method of the YRCPlatformUI class in the validateLinkClick() method. The fireAction() method returns a boolean value: `true` or `false`. If the method returns a `true` value, it indicates that the invoked action was found and executed. For example:

```
YRCPlatformUI.fireAction(actionOpenMyView);
```

where `actionOpenMyView` is the identifier of the action that needs to be invoked. For more information about eclipse actions, see the Sterling Selling and Fulfillment Foundation: *Customizing Rich Client Platform Interface Guide*

Adding a Composite

About this task

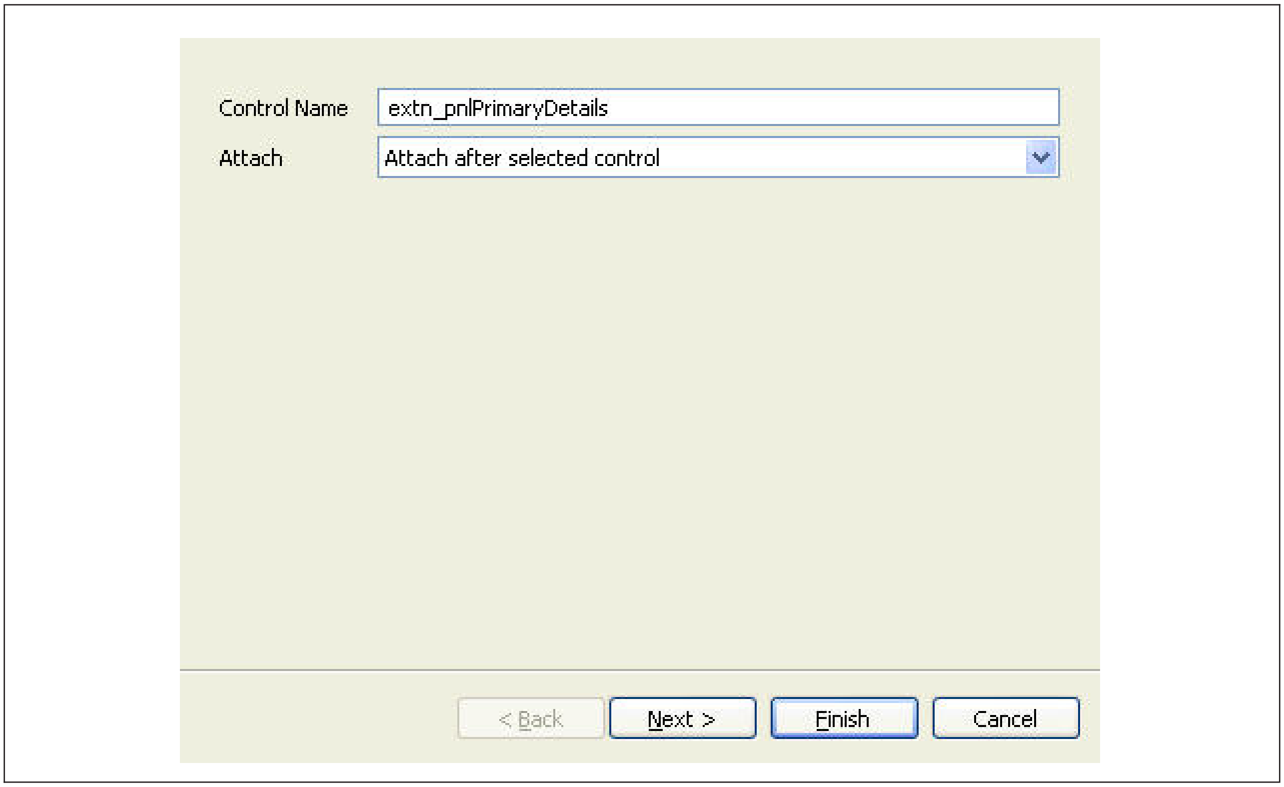
To add a composite:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Do any of the following to add a composite:
   - Select the control where you want to add the composite and click once.
   - Select the composite or group where you want to add the composite and click once.

   The Add Composite window displays.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

3. Click **Finish** to add the composite with the default layout data.
4. Click **Next** to specify the layout data options.
5. Enter information in the applicable fields and click **Finish**.
6. Click to save the changes made to the extension file.

Adding a Group

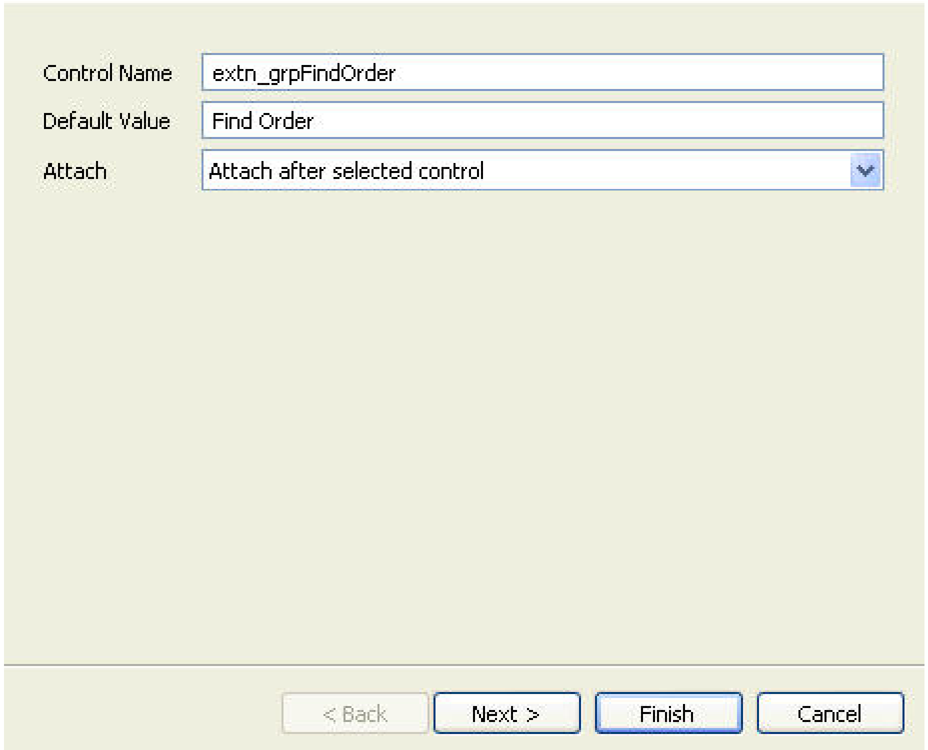
About this task

To add a group:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click Group . Do any of the following to add a group:
   - Select the control where you want to add the group and click once.
   - Select the composite or group where you want to add the group and click once.

   The Add Group window displays.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

3. Click **Finish** to add the group with the default layout data.
4. Click **Next** to specify the layout data options.
5. Click **Finish**.
6. Click to save the changes made to the extension file.
7. If you have specified the default value for the new group, synchronize the resource files.

   For more information on synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Adding an External Panel

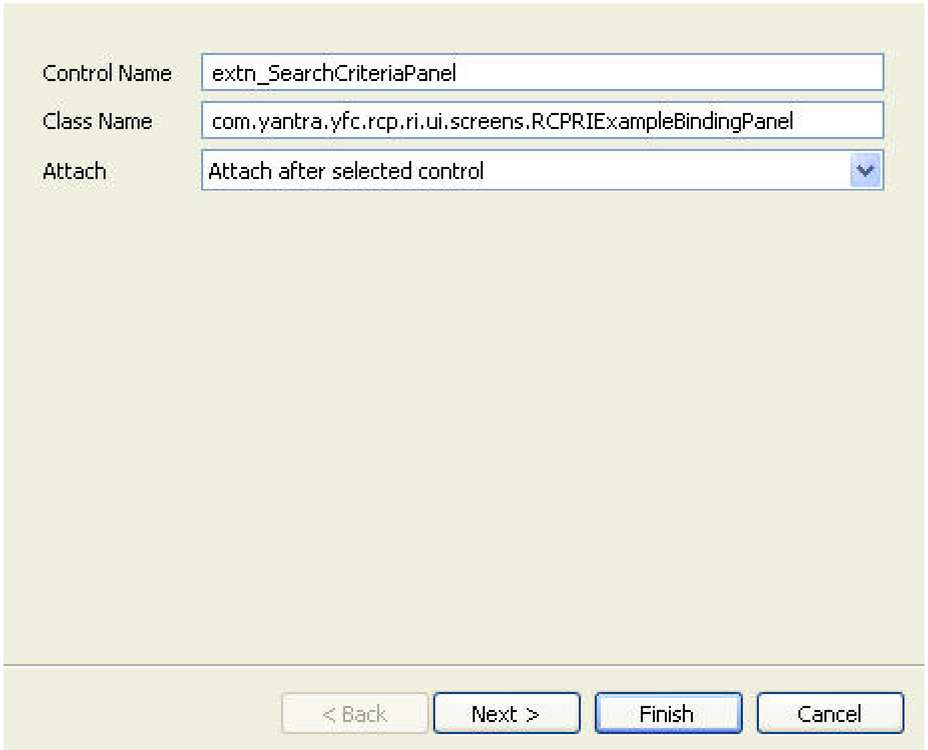
About this task

To add an external panel:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click External Panel .
3. Do any of the following:
   - Select the control where you want to add the external panel and click once.
   - Select the composite or group where you want to add the external panel and click once.

   The Add External Panel window displays.



For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."

4. Click **Finish** to add the group with the default layout data.
5. Click **Next** to specify the layout data options.
6. Click **Finish**.
7. Click to changes made to the extension file.
8. Open the external panel's class in the Java Editor.
9. In the class, create the constructor as following:

```
public RCPDRIBindingPanel1(Composite parent, int style,
YRCExtentionBehavior behavior) {
}
```

Note: Make sure that the constructor contains only the following arguments and in this specific order: Composite parent, int style, and YRCExtensionBehavior behavior.

Moving Controls and Table Columns

Using the move option you can move or rearrange controls on the screen.

You can move:

- Individual controls on the screen.
- A complete composite or group containing one or more controls

  All child controls of the composite or group move along with the group.
- Individual columns in the table.

Note: If a screen comprises two independent or reusable forms, the Rich Client Platform Extensibility Tool does not allow you to move a control from one form to another.

Moving a Control

About this task

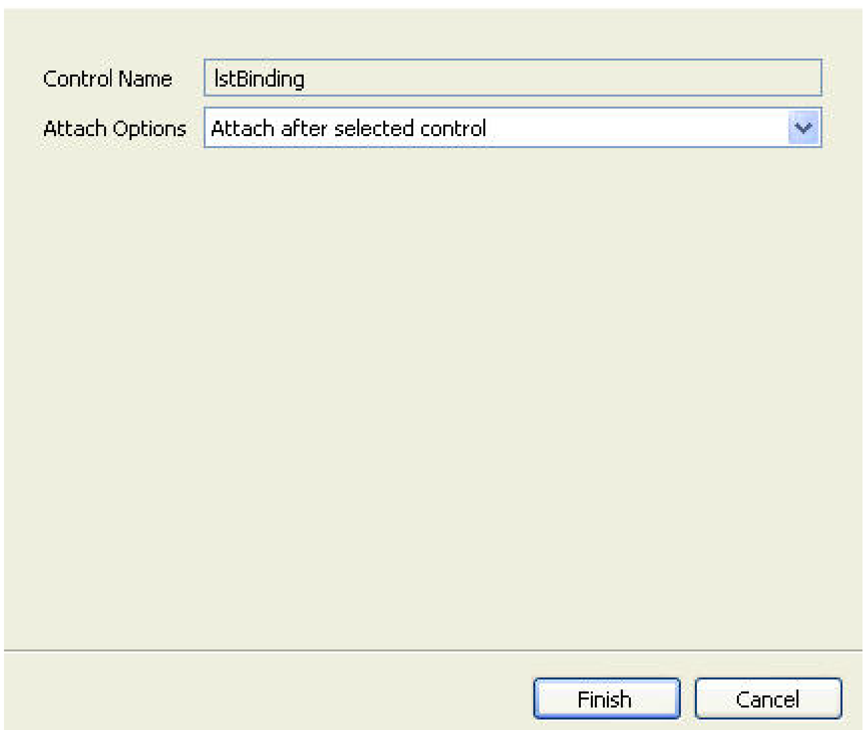
To move a control:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Select the control you want to move.

   Note: If you attempt to move controls for which a unique name is not set, an error message displays.
3. Click the target control where you want to move the selected control. The Move Field window displays.



   For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."
4. Click **Finish**.

Moving Table Columns

About this task

You can rearrange columns in a table. However, you cannot move a column from one table to another table.

Restriction: You cannot move a table column after or before a newly added column.

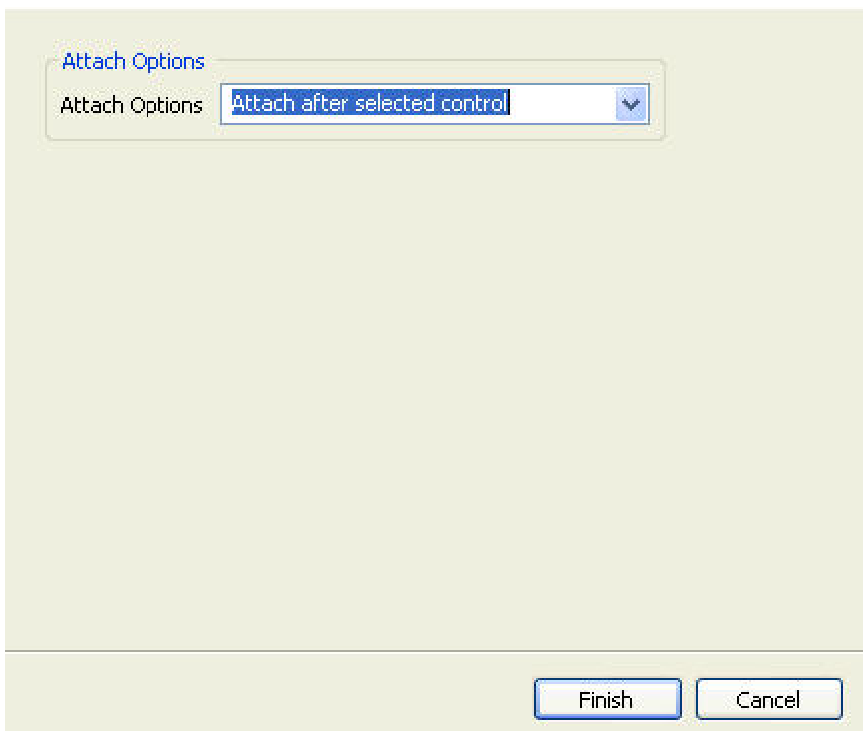
To move a table column:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Select the table column you want to move.

   Note: Set a unique name for all columns in a table. Otherwise, an error message displays.
3. Click the target table column where you want to move the selected table column. The Move Table Column window displays.



   For descriptions of the window fields referenced in this procedure, see "Control and Table Column Field Definitions."
4. Select the appropriate value from the drop-down list. Valid values are:
   - Attach after selected control
   - Attach before selected control
5. Click **Finish** and reopen the screen to view the changes made to the table.

Control and Table Column Field Definitions

This topic contains the names and descriptions of window fields that are completed or referenced while adding or moving controls and table columns.

Add Button Window Fields

Field	Description
Control Name	Mandatory field. Enter a unique control name for the new control. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Default Value	Mandatory field. Enter the default value to display on the screen.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". Note: If you are adding the new field to an empty composite or group, by default, the "Attach within selected control" value displays.
Validation Required?	Check this box to validate a new field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.

Add Control Window Fields

Field	Description
Control Name	Mandatory field. Enter a unique control name for the new control. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Default Value	Mandatory field. Enter the default value to display on the screen.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are adding the new field to an empty composite or group, by default, the "Attach within selected control" value displays.

Field	Description
Validation Required?	Check this box to validate a new field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
DB Localized?	Check this box to retrieve data for the new field from a localized database (DB) column, if applicable. **Note:** If you are adding an advanced control, this field is disabled.
Is Mandatory?	Check this box if you want to make this field mandatory. **Note:** If you are adding an advanced control, this field is disabled.
Standard	Choose this if you want to add a standard control.
Advanced	Choose this if you want to add an advanced control. With an advanced control you can set the custom binding data for the control. For more information about binding data, refer to the Javadocs.

Add External Panel Window Fields

Field	Description
Control Name	Mandatory field. Enter a unique control name for the external panel. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Class Name	Mandatory field. Enter the external panel's class name including the package name. For example, com.yantra.yfc.rcp.ri.ui.screens.RCPRIExampleBindingPanel. Here com.yantra.yfc.rcp.ri.ui.screens is the package name and RCPRIExampleBindingPanel is the external panel's class name.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control".

Add Label Window Fields

Field	Description
Control Name	Mandatory field. Enter a unique control name for the new label. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Default Value	Mandatory field. Enter the default value to display on the screen.

Field	Description
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are adding a label to an empty composite or group, by default, the "Attach within selected control" value displays.
Standard	Choose this if you want to add a standard label.
Advanced	Choose this if you want to add an advanced label. With an advanced label you can set the custom label binding data. For more information about binding data, refer to the Javadocs.

Add Table Column Window Fields

Field	Description
Control Name	Mandatory field. Enter a unique control name for the new column. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Default Value	Mandatory field. Enter the default value to display on the screen.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control".
DB Localized?	Check this box if you want to retrieve data for this column from a localized DB column. **Note:** If you are adding an advanced column, this field is disabled.
Movable?	Check this box to move the column, if applicable. **Note:** If you are adding an advanced column, this field is disabled.
Sort Required?	Check this box to sort data in a particular order, if applicable. **Note:** If you are adding an advanced column, this field is disabled.
Filter Required?	Check this box to filter data, if applicable. **Note:** If you are adding an advanced column, this field is disabled.
Standard Column	Choose this if you want to add a standard column.
Advanced Column	Choose this if you want to add an advanced column.

Data Binding Fields

The exact contents will vary according to the control you are adding.

Field	Description
Source Binding	Enter the XML path to populate the new field, if applicable. Click  to view the available XML paths. The Source Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify the source binding for an advanced control using the Rich Client Platform Extensibility Tool.
Target Binding	Enter the XML path to send data to the API from the new field, if applicable. You can specify multiple target bindings by using a semicolon. Click  to view the available XML paths. The Target Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify the target binding for an advanced control using the Rich Client Platform Extensibility Tool.
Checked Binding	Enter the XML path to specify the checked binding attribute, if applicable. Click  to view the available XML paths. The Checked Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify the checked binding for an advanced control using the Rich Client Platform Extensibility Tool.
Unchecked Binding	Enter the XML path to specify the unchecked binding attribute, if applicable. Click  to view the available XML paths. The Unchecked Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify the unchecked binding for an advanced control using the Rich Client Platform Extensibility Tool.
List Binding	Enter the XML path to get a list of items for the new field. Click  to view the available XML paths. The List Bindings Tree window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify the list binding for an advanced control using the Rich Client Platform Extensibility Tool.

Field	Description
Code Binding	Enter the XML path to send the value of an attribute to the API for the new field. Click ... to view the available XML paths. The Code Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify the code binding for an advanced control using the Rich Client Platform Extensibility Tool.
Description Binding	Enter the XML path to specify the value to display on the screen. Click ... to view the available XML paths. The Description Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify the description binding for an advanced control using the Rich Client Platform Extensibility Tool.
Attribute Binding	Enter the XML path to specify the binding attribute, if applicable. Click ... to view the available XML paths. The Attribute Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced column, this field is disabled. You cannot specify the attribute binding for an advanced column using the Rich Client Platform Extensibility Tool.

Data Layout Fields

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td colspan="2">Layout Tab

This tab is used to set the layout properties for the control you want to add. Based on the control, you can set the layout properties for the following:
• Parent Composite-If you want to add a control that is not a composite or group, in the layout tab you can set the layout properties of the parent control (composite or group) to which you want to add the new control. You can set the position and size of the control in the parent control. The positioning and sizing of a field depends on the layout of the parent control.
• Composite or Group-If you want to add a composite or group, in the layout tab you can set the layout properties of the composite or group, itself.

Note: If you add a control that is not a composite or group and the control name for its parent control (composite or group) is not set, the layout tab is disabled.</td></tr>
<tr><td>Control Name</td><td>Disabled field.

If you are adding a control which is not a composite or group, this field displays the name of the parent control (composite or group) to which you want to add the new control.

If you add a composite or group, this field displays the name of the composite or group, itself.</td></tr>
</table>

Field	Description
Grid Columns	
Layout	Select the layout for the control. The valid value is "GridLayout".
Number of Columns	Enter a number or click to increase or decrease the number of columns for the layout. **Note:** You can add one or more controls to the same row by changing the number of columns in the layout.
Make Columns Equal	Check this box if you want all columns in this layout to be of equal width.
Spacing	
Horizontal Spacing	If you increase or decrease the horizontal space, the horizontal space between the two neighboring columns also increases or decreases.
Vertical Spacing	If you increase or decrease the vertical space, the vertical space between the two neighboring columns also increases or decreases.
Margin Width	If you increase or decrease the margin width, the horizontal margin along the left and right edges of the layout also increases or decreases.
Margin Height	If you increase or decrease the margin height, the vertical margin along the top and bottom edges of the layout also increases or decreases.
Component Tab Set the properties of the layout data of the control.	
Pick Layout Like	To apply the same layout as another field in the same parent, select the field from the drop-down list. The drop-down list contains the control names that are siblings of the selected control.
Pick This	When you click this button all other fields are automatically populated based on the layout of the control you selected in the Pick Layout Like option.
Height	Enter any value between -1 and 9999 for the control.
Width	Enter any value between -1 and 9999 for the control.
Horizontal Span	If you increase the horizontal span by specifying any value between 1 and 25, the new control spans to the right of its current position.
Vertical Span	If you increase the vertical span by specifying any value between 1 and 25, the new control spans to the cell below its current position.
Grab Horizontal Space	Check this box if you want the new control to grab the extra horizontal space.
Grab Vertical Space	Check this box if you want the new control to grab the extra vertical space.

Field	Description
Horizontal Alignment	Select the appropriate horizontal alignment type for the new control. Valid values are 'BEGINNING', 'CENTER', 'END', or 'FILL'. • BEGINNING aligns the new control with the left side of the horizontal space. • CENTER centers the new control within the horizontal space. • END aligns the new control with the right side of the horizontal space. • FILL fills the excess horizontal space.
Vertical Alignment	Select the appropriate vertical alignment type for the new control. Valid values are 'BEGINNING', 'CENTER', 'END', or 'FILL'. • BEGINNING aligns the new control with the top of the vertical space. • CENTER centers the new control within the vertical space. • END aligns the new control with the bottom of the vertical space. • FILL fills the excess vertical space.
Horizontal Indent	Set the horizontal indentation that you want to have on the left side of the control.

Hot Key Binding Fields

Field	Description
Key Sequence	Enter a valid key sequence of the hot key for the control. For example, F7, M1+K, and so forth. • Use M1 to specify the **Ctrl** key. • Use M2 to specify the **Shift** key. • Use M3 to specify the **Alt** key. To specify a combination of keys use the "+" operator. For example, to specify the hot key for a control as **Ctrl+Alt+K**, enter the key sequence as `M1+M3+K`. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify hot key binding for an advanced control using the Rich Client Platform Extensibility Tool.
Action Definition	
Scope	Choose the scope or context for the hot key. For a wizard, the hot key is applicable for all the pages of the wizard. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify hot key binding for an advanced control using the Rich Client Platform Extensibility Tool.
Action Id	Enter the identifier of the action you want to invoke upon pressing the hot key. **Note:** The action identifier specified in this field must not exist in the plugin.xml file of your Rich Client Platform extension plug-in. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify hot key binding for an advanced control using the Rich Client Platform Extensibility Tool.

Field	Description
Action Class	Enter the fully qualified java classpath of the action class you want to invoke upon pressing the hot key. **Note:** If you are adding an advanced control, this field is disabled. You cannot specify hot key binding for an advanced control using the Rich Client Platform Extensibility Tool.

Style Fields

Field	Description
BORDER	Include this property to set the border for the control.
RIGHT	Include this propety to left-align the control.
CENTER	Include this property to center-align the control.
LEFT	Include this property to center-align the control.
WRAP	Include this property to wrap the text in the control.
READ_ONLY	Include this property to make the control read only.
MULTI	Include this property for using multi-selection in lists and multiple line on text fields.

Move Field Window Fields

Field	Description
Control Name	Disabled field. Displays the name of the selected field you want to move. Every field on the form must have a unique logical name used only for reference.
Attach Options	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control". If you are moving the selected field to an empty composite or group, the "Attach within selected control" value displays.

Theme Binding Fields

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td colspan="2">Note: You cannot define theme for the following RCP UI controls. This is an eclipse SWT limitation:
• Button—You can only change the font (size , name , and style) for a button. The background and foreground color cannot be changed as it is OS specific.
• Combo Box—You can only change the foreground , background and font of the contents of a combo box. The combo box color cannot be changed as it is OS specific.
• Table Column—Theme cannot be applied for a table column.</td></tr>
</table>

Field	Description
Theme Name	Enter the theme name you want to apply for the new field, if applicable. Click  to view the available themes. The Themes Tree pop-up window is displayed, which lists all the available themes and also provides a preview of each theme. Select the appropriate theme name you want to apply. You can also press Ctrl+Space and select the appropriate theme name you want to apply from the drop-down list. The current theme applied to the selected control is displayed below the theme name. If an image is specified as the theme, then the image is displayed. **Note:** If you are adding an advanced column, this field is disabled. You cannot specify the theme binding for an advanced column using the Rich Client Platform Extensibility Tool.

Chapter 4. Related Tasks, Extension Behavior, and Hot Keys

Adding Related Tasks

About this task

Using the Related Task option of the Rich Client Platform Extensibility Tool, you can add new related tasks for a task that is open in an editor. You must define the group and category for each related task. All the related tasks can belong to multiple categories, but are limited to one group.

In addition, you can move the existing related tasks from one group to another group by creating a new related task in the second group with same the action ID as given in the first group and then hiding the existing related task in the first group. For more information about hiding existing related tasks, see the Sterling Selling and Fulfillment Foundation: *Customizing Rich Client Platform Interface Guide*.

Prior to adding a new related task, you must be aware of the following information:

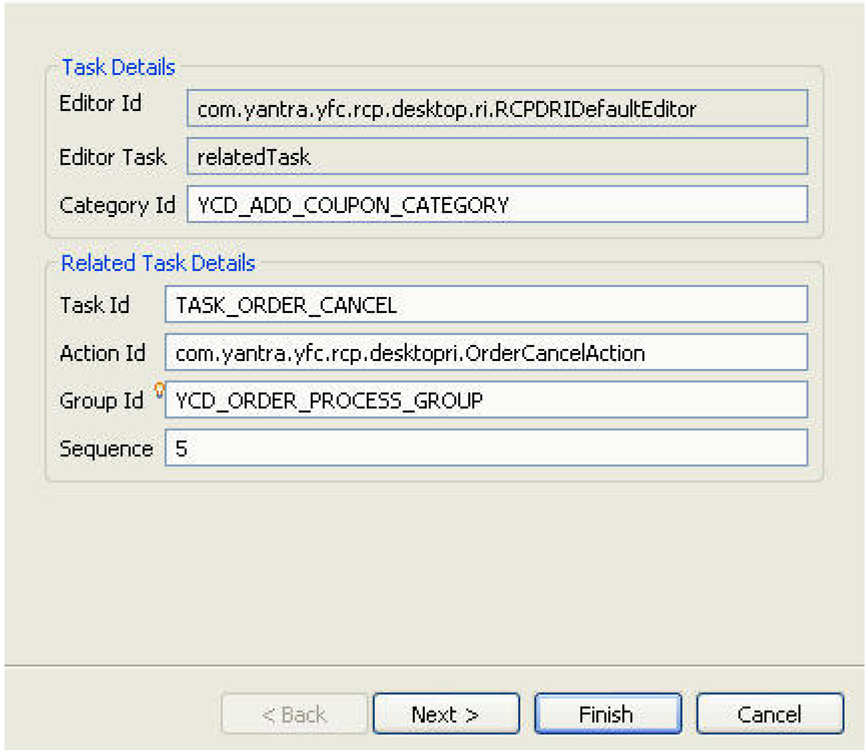
- Category Information—When a new related task is associated with the active task running in the current editor, you must know the categories in which the active task is included so that the new related task can be displayed under the Related Tasks view.
- Group Information—To display a new related task, you can either use an existing group or create a new group.

To add a new related task:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Navigate to the task you want to extend in the Rich Client Platform application.
3. Click 

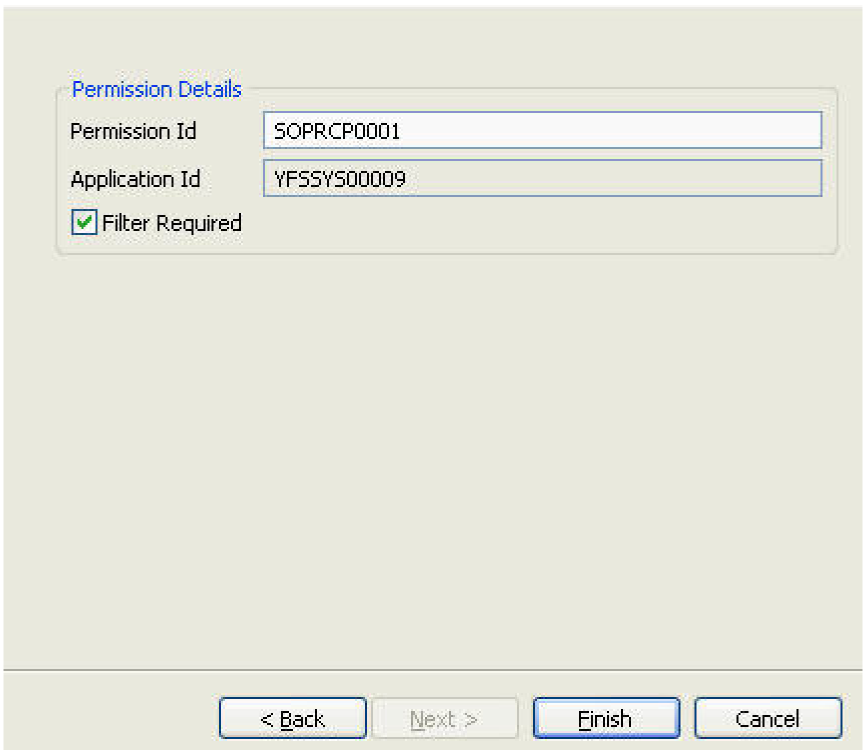
 and click once on the form. The Add Related Task window displays.



Field	Description
Task Details	
Editor Id	Disabled field. Displays the identifier of the currently opened editor on the screen.
Editor Task	Disabled field. Displays the identifier of the task that is currently open in the editor.
Category Id	Enter the identifier of the category in which the new related task is included. You can add a related task to an existing category or new category. You can also press Ctrl+Space to select the category identifier from the list of available categories.
Related Task Details	
Task Id	Enter a unique identifier for the new related task you want to add.
Action Id	Enter the identifier of the action you want to invoke upon clicking the related task. The action class of the actionId that you specified should extend the YRCRelatedTaskAction class. You can also press **Ctrl+Space** to select the action identifier from the list of available actions. For more information about actions, see the Sterling Selling and Fulfillment Foundation: *Customizing Rich Client Platform Interface Guide*

Field	Description
Group Id	Enter the identifier of the group in which you want to display the related task so that the new related task can be displayed in the Related Tasks view. You can add a related task in an existing group or a new group. You can also press **Ctrl+Space** to select the group identifier from the list of available groups.
Sequence	Enter the sequence number to display the new group in a particular order, if applicable. The groups are displayed in ascending order by their sequence number. **Note:** Note: This field is enabled only if you specify a new group identifier to display the related task.

4. Click **Next** to set permissions for the new related task.



Field	Description
Permission Id	Enter the unique resource identifier that you defined for this resource in the ConfiguratorApplications Manager. For more information about defining Application resources, see the : Configuration Guide .
Application Id	Disabled field. This field displays the unique identifier of the application.
Filter Required	Check this box if you want to filter the related task depending on the custom criteria.

5. Click **Finish** to view the newly added related task in the Related Task view.

Note: You must synchronize files after adding the new related task in order to add them to the plugin.xml file. The name of the related task and the group are based on bundle entries defined for the corresponding identifiers.

6. Synchronize the resource files for the changes you made to the new related task.

   For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Creating Extension Behavior

About this task

You must create the extension behavior whenever you are extending an existing wizard or screen:

- To perform custom validations for the newly added fields
- To call an API or Service
- To set the extension model to populate a screen or wizard page
- To set bindings for advanced controls you add through the Rich Client Platform Extensibility Tool.

If you are extending a screen which is not a wizard, the tool creates the normal extension behavior and if you are extending a screen which is a wizard, the tool creates the wizard extension behavior.

For more information about synchronizing extension behavior, see "Synchronizing New and Existing UI Resources."

To create extension behavior:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click Extension Behavior and click on the form. The `Create Extension Behavior?` message displays. The message box prompts you to confirm whether you want to create an extension behavior for the current form. The message box also displays the identifier of the current form.
3. Click **OK**.
4. Synchronize the extension behavior that you created to specify the implementation class, if applicable.

Configuring Hot Keys

About this task

Using the Rich Client Platform Extensibility Tool, you can configure the hot keys to perform a specific task or operation.

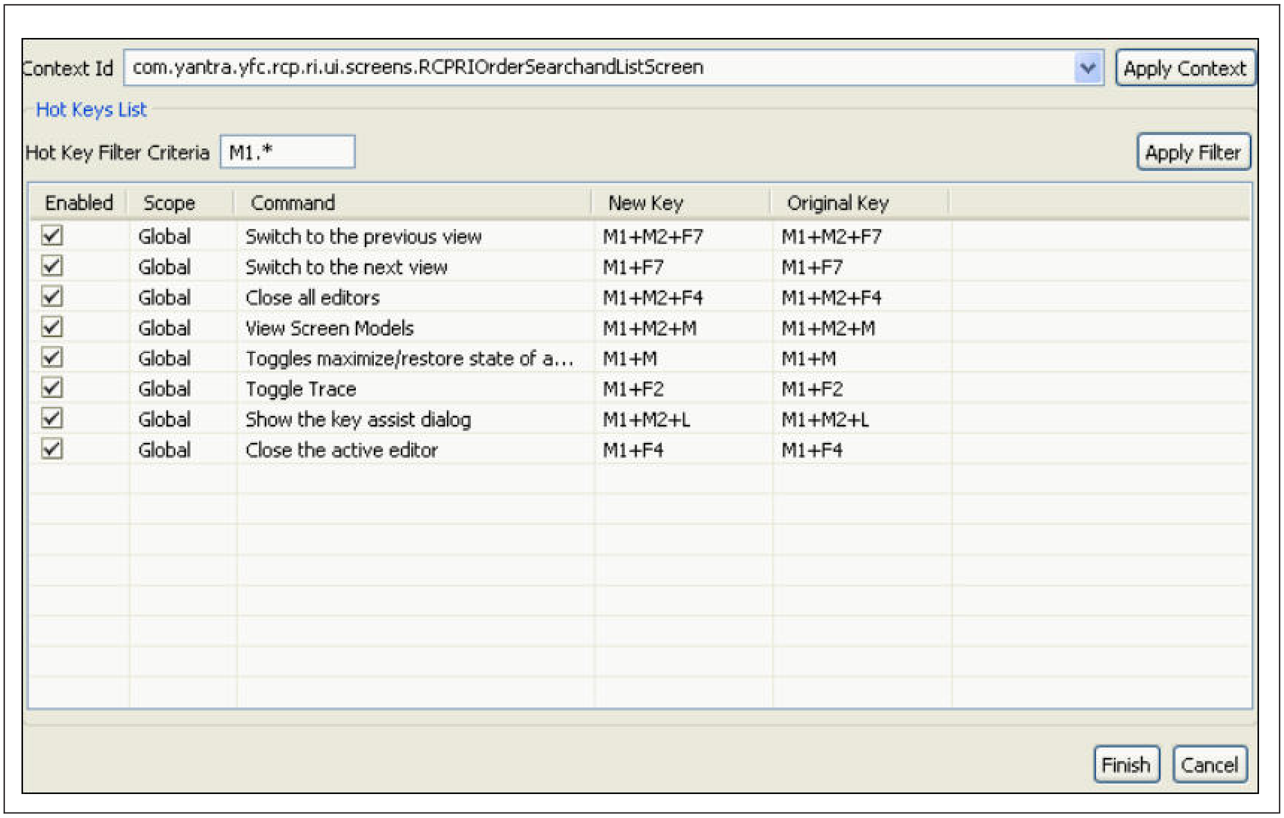
To configure the hot keys:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click Configure Hot Keys and click once on the screen.

   The Hot Key Configuration window displays.

By default, in the ContextId field, the context identifier of the current screen displays. A list of hot keys defined in the global context for the defaulted context identifier also displays.



Field	Description
Context Id	Select the context identifier for which you want to view a list of hot keys, if applicable.
Apply Context	Click this button to view all hot keys defined for the selected context identifier. A list of hot keys defined in the global context also displays.
Hot Keys List	
Hot Key Filter Criteria	Enter the custom criteria based on which you want to filter the hot keys displayed in the New Key column. You can use regular expressions for filtering the hot keys. For example, if you want to view hot keys that start with M1, specify `M1.*`.
Apply Filter	Click this button to view a list of hot keys as a result of your filter criteria.
Enabled	Uncheck the box corresponding to the hot key that you want to disable. **Note:** When you uncheck a box, the complete row is grayed out.
Scope	Displays the scope or context of a particular hot key. Valid values are: "Local" and "Global". • Local—indicates that the hot key is applicable for a specific screen in the application. • Global—indicates that the hot key is applicable for any screen in the application.

Field	Description
Command	Displays the name of the command defined for a hot key.
New Key	Displays the new key sequence defined for the hot key. You can modify the new key sequence, if applicable. **Note:** If you uncheck the box, you cannot modify the new key sequence.
Original Key	Displays the original key sequence defined for the hot key. You cannot modify the original key sequence.
Finish	Click this button to save the changes made to the hot key configurations in the plugin.xml file of the plug-in project.

Resolving Hot Key Conflicts

About this task

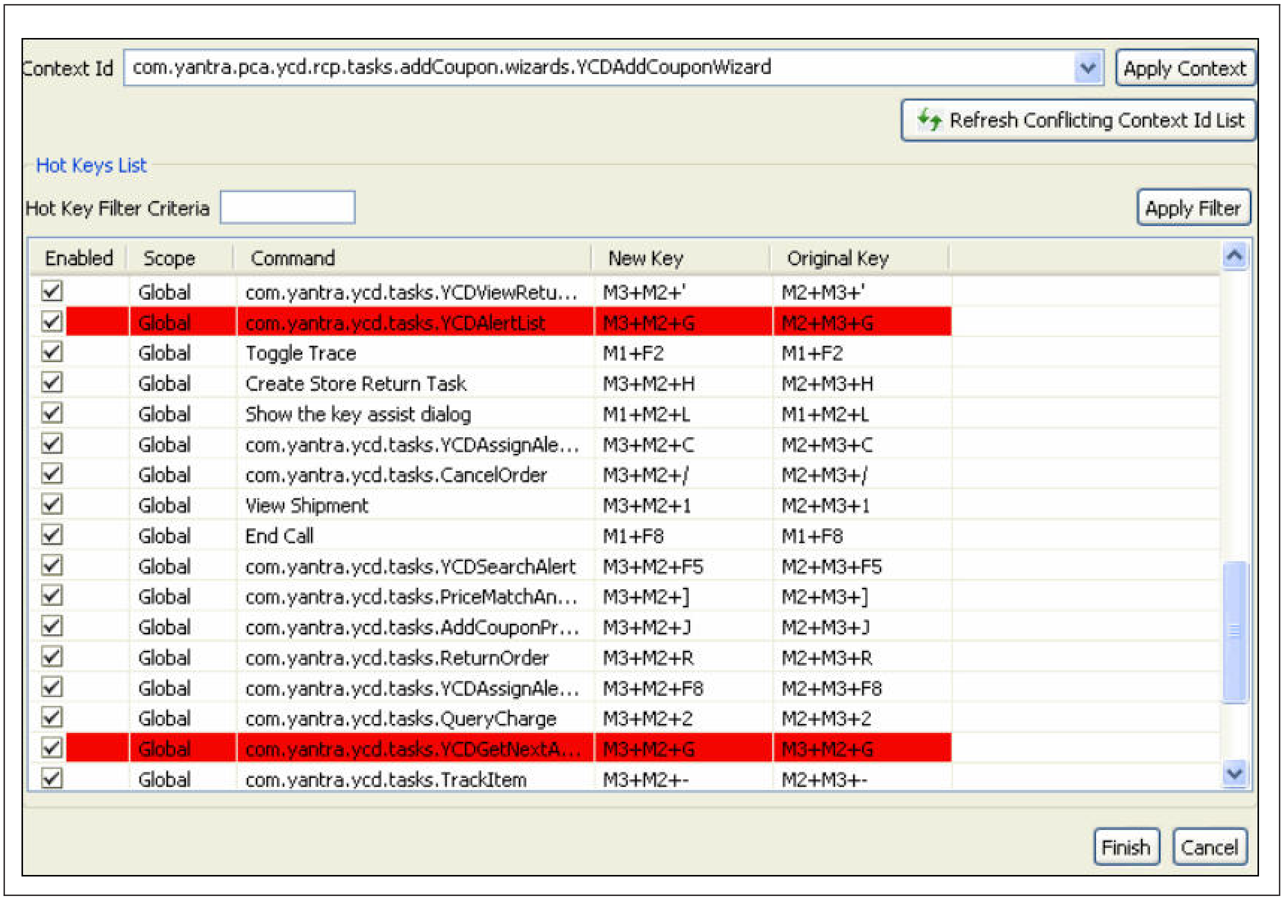
Using the Rich Client Platform Extensibility Tool, you can view a list of conflicting hot keys and resolve conflicts. In the Hot Keys List panel, the specific table cells that display the hot keys with conflicts are highlighted in red.

To resolve a hot key conflict:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

and click once on the screen. The Hot Keys window displays.



Field	Description
Context Id	Select the context identifier for which you want to view the list of conflicting hot keys.
Apply Context	Click this button to view all hot keys defined for the selected context identifier. A list of hot keys defined in the global context also displays.
Refresh Conflicting Context Id List	Click this button to refresh the list of conflicting context identifiers.
Hot Keys List	
Hot Key Filter Criteria	Enter the custom criteria based on which you want to filter the hot keys displayed in the New Key column. You can use regular expressions for filtering the hot keys. For example, if you want to view hot keys that start with M1, specify "M1.*".
Apply Filter	Click this button to view a list of hot keys as a result of your filter criteria. The specific table cells that displays conflicted hot keys are shaded in red.
Enabled	Uncheck the appropriate box of the hot key that you want to disable. **Note:** When you uncheck any box, the complete row is grayed out.

Field	Description
Scope	Displays the scope or context of a particular hot key. Valid values are: • Local—indicates that the hot key is applicable for a specific screen. • Global—indicates that the hot key is applicable for any screen.
Command	Displays the name of the command defined for a hot key.
New Key	Displays the new key sequence defined for the hot key. You can modify the new key sequence, if necessary. If you uncheck the box, you cannot modify the new key sequence.
Original Key	Displays the original key sequence defined for the hot key. You cannot modify the original key sequence.
Finish	Click this button to save the changes made to the hot key configurations in the plugin.xml file of the plug-in project.

Chapter 5. Modifying New and Existing Controls

About Modifying New Controls

You can modify the newly added fields by disabling or hiding them, changing their default values, and so forth. The Rich Client Platform enables you to modify layout settings, binding attributes, and themes. You can also delete the newly added fields from the screen.

Note: You cannot modify the control name of any field. However, you can set the text and combo box controls as mandatory fields.

Modifying a New Label

About this task

To modify a newly added label:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added label and click once. The Edit New Label window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
3. Click **Finish**. The label is updated with the style, layout data, binding attributes, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The label is updated with the layout data, binding attributes, and theme.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The label is updated with the binding attributes and theme.
8. Click **Next** to modify the binding options.
9. Click **Finish**. The label is updated with the theme binding.
10. Click **Next** to modify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the checkbox, synchronize the resource files.

    For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources".

Modifying a New Button

About this task

To modify a newly added button:

Procedure

1. Start the Rich Client Platform Extensibility Tool.

2. Select the newly added button and click once. The Edit New Button window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."

   Note: If you attempt to modify a button that is placed in the row layout container, the Layout Setup Page window is disabled.
3. Click **Finish**. The button is updated with the style, layout data, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The button is updated with the layout data, and theme.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The button is updated with the theme binding.
8. Click **Next** to modify the theme binding.
9. Click **Finish**.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the button, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying a New Checkbox

About this task

To modify a newly added checkbox:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select the newly added checkbox and click once.

   The Edit New Checkbox window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."

   Note: If you attempt to modify a checkbox that is placed in the row layout container, the Layout Setup Page window disabled.
3. Click **Finish**. The checkbox is updated with the style, layout data, binding attributes, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The checkbox is updated with layout data, binding attributes, and theme binding.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The checkbox is updated with the binding attributes and theme binding.
8. Click **Next** to modify the binding options.
9. Click **Finish**. The checkbox is updated with the theme binding.
10. Click **Next** to modify the theme binding.
11. Click **Finish**.
12. Click  to save the changes made to the extension file.

13. If you have specified the default value, bindings, or theme entry for the checkbox, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying a New Radio Button

About this task

To modify a newly added radio button:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added radio button and click once.

   The Edit New Radio Button window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."

   Note: If you attempt to modify a radio button that is placed in the row layout container, the Layout Setup Page window is disabled.
3. Click **Finish**. The radio button is updated with the style, layout data, binding attributes, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The radio button is updated with the layout data, binding attributes, and theme binding.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The radio button is updated with the binding attributes and theme binding.
8. Click **Next** to modify the binding options.
9. Click **Finish**. The newly added radio button is updated with the existing theme.
10. Click **Next** to modify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the radio button, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources".

Modifying a New Text Box

About this task

To modify a newly added text box:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added text box and click once. The Edit New Text Box window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."

3. Click **Finish**. The text box is updated with the style, layout data, binding attributes, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The text box is updated with the layout data, binding attributes, and theme binding.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The text box is updated with the binding attributes and theme binding.
8. Click **Next** to modify the binding options.
9. Click **Finish**. The text box is edited with the theme binding.
10. Click **Next** to modify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the text box, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources".

Modifying New StyledText Component

About this task

To modify newly added styled text:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added styled text and click once. The Edit New Styled Text window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
3. Click **Finish**. The styled text is updated with the style, layout data, binding attributes, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The styled text is updated with the layout data, binding attributes, and theme binding.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The styled text is updated with the binding attributes and theme binding.
8. Click **Next** to modify the binding options.
9. Click **Finish**. The styled text is updated with the theme binding.
10. Click **Next** to modify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the styled text, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying a New Combo Box

About this task

To modify a newly added combo box:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added combo box and click once. The Edit New Combo Box window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
3. Click **Finish**. The combo box is updated with the style, layout data, binding attributes, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The combo box is updated with the layout data, binding attributes, and theme binding.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The combo box is updated with the binding attributes and theme binding.
8. Click **Next** to modify the binding options.
9. Click **Finish**. The combo box is updated with the theme binding.
10. Click **Next** to modify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the combo box, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying a New List Box

About this task

To modify a newly added list box:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added list box and click once. The Edit New List Box window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
3. Click **Finish**. The list box is updated with the style, layout data, binding attributes, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The list box is updated with the layout data, binding attributes, and theme binding.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The list box is updated with the binding attributes and theme binding.

8. Click **Next** to modify the binding options.
9. Click **Finish**. The list box is updated with the theme binding.
10. Click **Next** to modify the theme binding.
11. Click **Finish**.
12. Click to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the list box, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying a New Table Column

About this task

To modify a newly added table column:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added column in a table and click once. The Edit New Table Column window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
3. Click **Finish** to add the new column with the default column layout data, theme, and null binding attributes.
4. Click **Next** to specify the column layout options.
5. Click **Finish**. The table column is updated with the binding attributes and theme binding.
6. Click **Next** to modify the binding options.
7. Click **Finish**. The table column is updated with the theme binding.
8. Click **Next** to modify the theme binding.
9. Click **Finish**.
10. Click to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the table column, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying a New Link

About this task

To modify a newly added link:

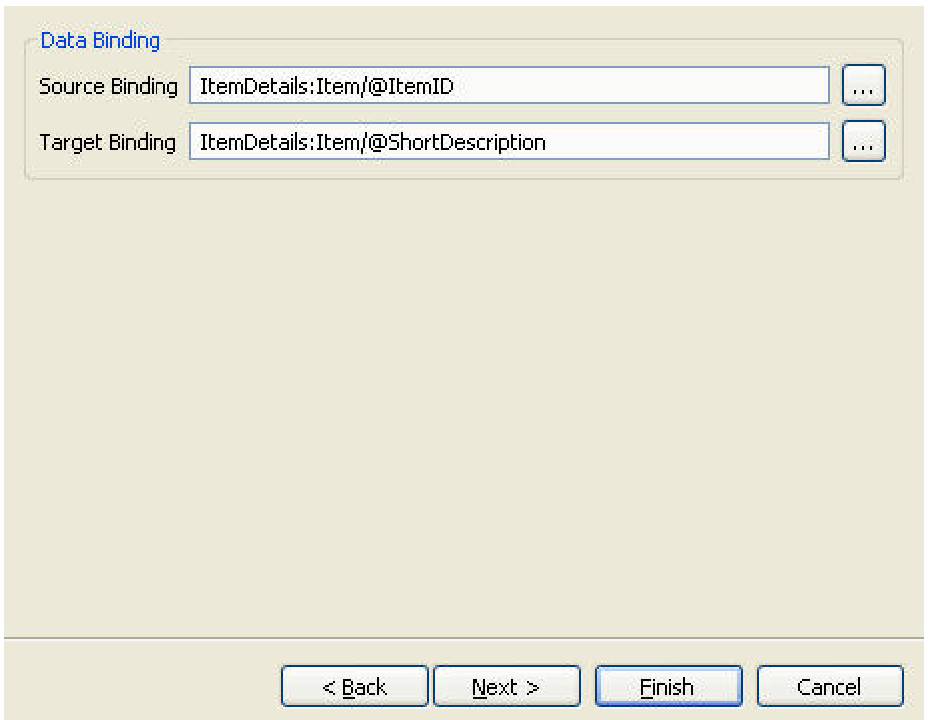
Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added link and click once. The Edit New Link window displays.

   Note: If you attempt to modify a link that is placed in the row layout container, the Layout Setup Page window disabled.

For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."

3. Click **Finish**. The link is updated with the style, layout data, binding attributes, and theme.
4. Click **Next** to modify the style options.
5. Click **Finish**. The link is updated with the layout data, binding attributes, and theme binding.
6. Click **Next** to modify the layout data options.
7. Click **Finish**. The link is updated with the binding attributes and theme binding.
8. Click **Next** to modify the binding options.



9. Click **Finish**. The link is updated with the theme binding.
10. Click **Next** to modify the theme binding.
11. Click **Finish**.
12. Click  to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the link, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying a New Composite

About this task

To modify a newly added composite:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added composite and click once. The Edit New Composite window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
3. Click **Finish**. The composite is updated with the style, layout data.
4. Click **Next** to modify the style options.
5. Click **Finish**. The newly added composite is edited with the existing layout data.
6. Click **Next** to modify the layout data options.
7. Click **Finish**.
8. Click  to save the changes made to the extension file.

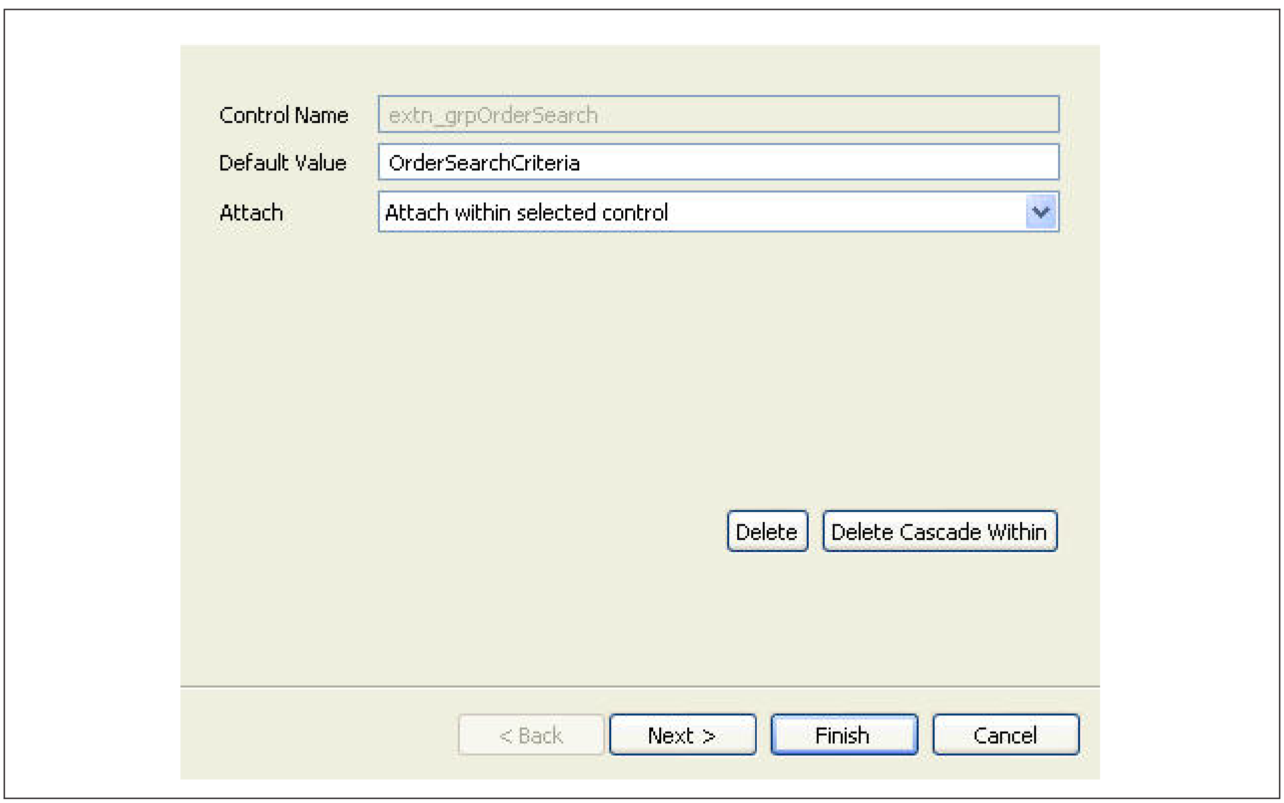
Modifying a New Group

About this task

To modify a newly added group:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added group and click once. The Edit New Group window displays.



For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."

3. Click **Finish**. The group is updated with the layout data.
4. Click **Next** to modify the layout data options.
5. Click **Finish**.
6. Click  to save the changes made to the extension file.
7. If you have specified the default value, bindings, or theme entry for the group, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

About Modifying Existing Controls

You can modify the fields provided with the Rich Client Platform application by disabling them, hiding them, changing their default values, and so forth. The Rich Client Platform provides the ability to modify layout settings and theme binding. You can also delete the existing fields from the screen.

Note: You cannot modify the control name of any existing fields.

Note: Provided fields that are mandatory cannot be made optional.

Modifying an Existing Label, Composite, or Group

About this task

To modify an existing label, composite, or group:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click  to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
4. Click **Next** to modify the layout data options.
5. Click **Next** to modify the theme binding.
6. Click **Finish**.
7. Click  to save the changes made to the extension file.
8. If you have specified the theme entry for the field, synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources".

Modifying an Existing Button, Checkbox, Radio Button, List Box, or Link

About this task

To modify an existing button, checkbox, radio button, list box, or link:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click  to show hidden and disabled fields, if applicable.

3. Select an existing field and click once. The Edit Existing Field window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
4. Click **Next** to modify the layout data options.
5. Click **Next** to modify the theme binding.
6. Click **Finish**.
7. Click to save the changes made to the extension file.
8. If you have specified the default theme entry for the field, you must synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying an Existing Text Box

About this task

To modify an existing text box:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
4. Click **Next** to modify the layout data options.
5. Click **Next** to modify the theme binding.
6. Click **Finish**.
7. Click to save the changes made to the extension file.
8. If you have specified the default value, bindings, or theme entry for the field, you must synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying an Existing StyledText Component

About this task

To modify an existing StyledText component:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
4. Click **Next** to modify the layout data options.
5. Click **Next** to modify the theme binding.
6. Click **Finish**.
7. Click to save the changes made to the extension file.

8. If you have specified the default value, bindings, or theme entry for the field, you must synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying an Existing Table Column

About this task

To modify an existing table column:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing column in the table and click once. The Edit Existing Field window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
4. Click **Next** to modify the column layout options.
5. Click **Next** to modify the theme binding.
6. Click **Finish**.
7. Click to save the changes made to the extension file.
8. If you have specified the theme entry for the table column, you must synchronize the resource files. For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

Modifying an Existing Combo Box

About this task

To modify an existing combo box:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays.

   For descriptions of the window fields referenced in this procedure, see "New and Existing Controls Field Definitions."
4. Click **Next** to modify the layout data options.
5. Click **Next** to modify the theme binding.
6. Click **Finish**.
7. Click to save the changes made to the extension file.
8. If you have specified the theme entry for the field, you must synchronize the resource files.

   For more information about synchronizing resource files, see "Synchronizing New and Existing UI Resources."

New and Existing Controls Field Definitions

Edit New Label Window

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Delete	Click this button to delete the newly added field from the form and extension file.

Modify Style Fields

Field	Description
BORDER	Include this property to set the border for the control.
RIGHT	Include this propety to left-align the control.
CENTER	Include this property to center-align the control.
LEFT	Include this property to center-align the control.
WRAP	Include this property to wrap the text in the control.
READ_ONLY	Include this property to make the control read only.
MULTI	Include this property for using multi-selection in lists and multiple line on text fields.

Modify Layout Data Options

Field	Description
Layout Tab This tab is used to change the layout properties for the control you want to modify. Based on the control, you can change the layout properties for the following: • Parent Composite—If you are modifying a control that is not a composite or group, the layout tab enables you to change the layout properties of the control (composite or group) to which the control belongs. You can change the position and size of the controls in the parent control. The positioning and sizing of a field depends on the layout of the parent control. • Composite or Group—If you are modifying a composite or group, the layout tab enables you to change the layout properties of the composite or group itself. **Note:** If you are modifying a control that is not a composite or group and the control name for its parent control (composite or group) is not set, the layout tab is disabled.	

Field	Description
Control Name	Disabled field. If you are modifying a control that is not a composite or group, this field displays the name of the parent control (composite or group) to which the control belongs. If you are modifying a composite or group, this field displays the name of the composite or group itself.
Grid Columns	
Layout	Select the layout for the control. The valid value is "GridLayout".
Number of Columns	Enter a number or click to increase or decrease the number of columns for the layout. **Note:** This property enables you to add one or more controls to the same row by changing the number of columns in the layout.
Make Columns Equal	Check this box if you want all columns in this layout to be of equal width.
Spacing	
Horizontal Spacing	If you increase or decrease the horizontal spacing, the horizontal space between the two neighboring columns also increases or decreases.
Vertical Spacing	If you increase or decrease the vertical spacing, the vertical space between the two neighboring columns also increases or decreases.
Margin Width	If you increase or decrease the margin width, the horizontal margin along the left and right edges of the layout also increases or decreases.
Margin Height	If you increase or decrease the margin height, the vertical margin along the top and bottom edges of the layout also increases or decreases.
Component Tab Change the properties of the layout data of the control.	
Pick Layout Like	To apply the same layout as another field in the same parent, select the field from the drop-down list. The drop-down list contains the control names that are siblings of the selected control.
Pick This	When you click this button, depending on the layout of the control you selected in the Pick Layout Like combo box, the values in other fields automatically displays.
Height	Enter any value between -1 and 9999 for the control.
Width	Enter any value between -1 and 9999 for the control.
Horizontal Span	If you increase the horizontal span by specifying any value between 1 and 25, the new control spans to the right of its current position.
Vertical Span	If you increase the vertical span by specifying any value between 1 and 25, the new control spans to the cell below its current position.
Grab Horizontal Space	Check this box if you want the new control to grab the extra horizontal space.
Grab Vertical Space	Check this box if you want the new control to grab the extra vertical space.

Field	Description
Horizontal Alignment	Select the appropriate horizontal alignment type for the new control. Valid values are 'BEGINNING', 'CENTER', 'END', or 'FILL'. • BEGINNING aligns the new control with the left side of the horizontal space. • CENTER centers the new control within the horizontal space. • END aligns the new control with the right side of the horizontal space. • FILL fills the excess horizontal space.
Vertical Alignment	Select the appropriate vertical alignment type for the new control. Valid values are 'BEGINNING', 'CENTER', 'END', or 'FILL'. • BEGINNING aligns the new control with the top of the vertical space. • CENTER centers the new control within the vertical space. • END aligns the new control with the bottom of the vertical space. • FILL fills the excess vertical space.
Horizontal Indent	Set the horizontal indentation for the left side of the control.

Modify Data Binding Fields

The exact contents will vary according to the control you are modifying.

Field	Description
Source Binding	Enter the XML path to populate the new field, if applicable. Click ... to view the available XML paths. The Source Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are modifying an advanced control, this field is not displayed.
Target Binding	Enter the XML path to send data to the API from the new field, if applicable. You can specify multiple target bindings by using a semicolon. Click ... to view the available XML paths. The Target Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are modifying an advanced control, this field is not displayed.
Checked Binding	Enter the XML path to specify the checked binding attribute, if applicable. Click ... to view the available XML paths. The Checked Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are modifying an advanced control, this field is not displayed.

Field	Description
Unchecked Binding	Enter the XML path to specify the unchecked binding attribute, if applicable. Click ... to view the available XML paths. The Unchecked Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are modifying an advanced control, this field is not displayed.
List Binding	Enter the XML path to get a list of items for the new field. Click ... to view the available XML paths. The List Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are modifying an advanced control, this field is not displayed.
Code Binding	Enter the XML path to send the value of an attribute to the API for the new field. Click ... to view the available XML paths. The Code Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are modifying an advanced control, this field is not displayed.
Description Binding	Enter the XML path to specify the value to display on the screen. Click ... to view the available XML paths. The Description Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are modifying an advanced control, this field is not displayed.
Attribute Binding	Enter the XML path to specify the binding attribute, if applicable. Click ... to view the available XML paths. The Attribute Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are modifying an advanced control, this field is not displayed.

Modify Theme Binding Options

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td colspan="2">You cannot define theme for the following RCP UI controls. This is an eclipse SWT limitation:
Button—You can only change the font (size , name , and style) for a button. The background and foreground color cannot be changed as it is OS specific.
Combo Box—You can only change the foreground , background and font of the contents of a combo box. The combo box color cannot be changed as it is OS specific.
Table Column—Theme cannot be applied for a table column.</td></tr>
<tr><td>Theme Name</td><td>Enter the theme name you want to apply for the field, if applicable.

Click  to view the available themes. The Themes Tree pop-up window displays. Select the appropriate theme name you want to apply. You can also press Ctrl+Space and select the name of the theme you want to apply from the drop-down list.

Note: If you are modifying an advanced column, this field is not displayed.</td></tr>
</table>

Edit New Button, Checkbox, or Radio Button, or Link Window Options

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Validation Required?	Check this box to validate a new field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
Delete	Click this button to delete the newly added field from the form and extension file.

Edit New Text Box, StyledText, Combo Box, or List Box Window Options

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.

Field	Description
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Validation Required?	Check this box to validate a new field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
DB Localized?	Check this box to retrieve data for the field from a localized database (DB) column, if applicable. **Note:** If you are modifying an advanced control, this field is not displayed.
Is Mandatory?	Check this box if you want to make the field mandatory. **Note:** If you are modifying an advanced control, this field is not displayed.
Delete	Click this button to delete the newly added field from the form and extension file.

Edit New Table Column Window Options

Field	Description
Control Name	Disabled field. Control Name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control".
DB Localized?	Check this box to retrieve data for the field from a localized database (DB) column, if applicable. **Note:** If you are modifying an advanced column, this field is not displayed.
Movable?	Check this box to move table columns, if applicable. **Note:** If you are modifying an advanced column, this field is not displayed.
Sort Required?	Check this box to sort the data in a particular order, if applicable. **Note:** If you are modifying an advanced column, this field is not displayed.
Filter Required?	Check this box to filter the data, if applicable. **Note:** If you are modifying an advanced column, this field is not displayed.
Delete	Click this button to delete the newly added field from the form and extension file.

Edit New Composite Window Options

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". Note: If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Delete	Click this button to delete the field from the form and extension file.
Delete Cascade Within	Click this button to delete the field and its sibling controls from the form and extension file.

Edit New Group window

Field	Description
Control Name?	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Delete	Click this button to delete the newly added field from the form and extension file.
Delete Cascade Within	Click this button to delete the newly added field and its sibling controls from the form and extension file.

Edit Existing Field Window Options

The values displayed may vary according to the control being modified.

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Enabled?	Uncheck this box if you want to disable the field.

Field	Description
Visible?	Uncheck this box if you want to hide the field. **Note:** Note: You cannot hide a composite or group if it contains mandatory fields.
Validation Required?	Check this box to validate a field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
Delete	This button displays if you attempt to modify an existing field that is already modified. When you click this button, the field is deleted from the extension file. Additionally, all the modifications made to this field are deleted. The default settings for this field are automatically restored.
Is Mandatory?	Check this box if you want make this field mandatory.
DB Localized?	Check this box to retrieve data for the field from a localized database (DB) column, if applicable.
Column Width	Increase or decrease the width of the table column, if applicable.

Chapter 6. Synchronizing New and Existing UI Resources

Synchronizing Bundle Entries

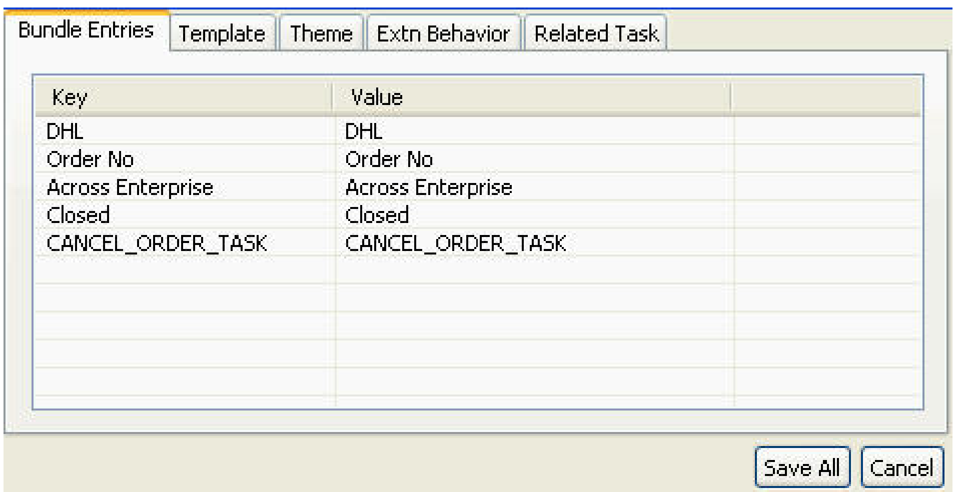
About this task

Whenever you add new fields, you must synchronize the existing bundle file with the new bundle entries.

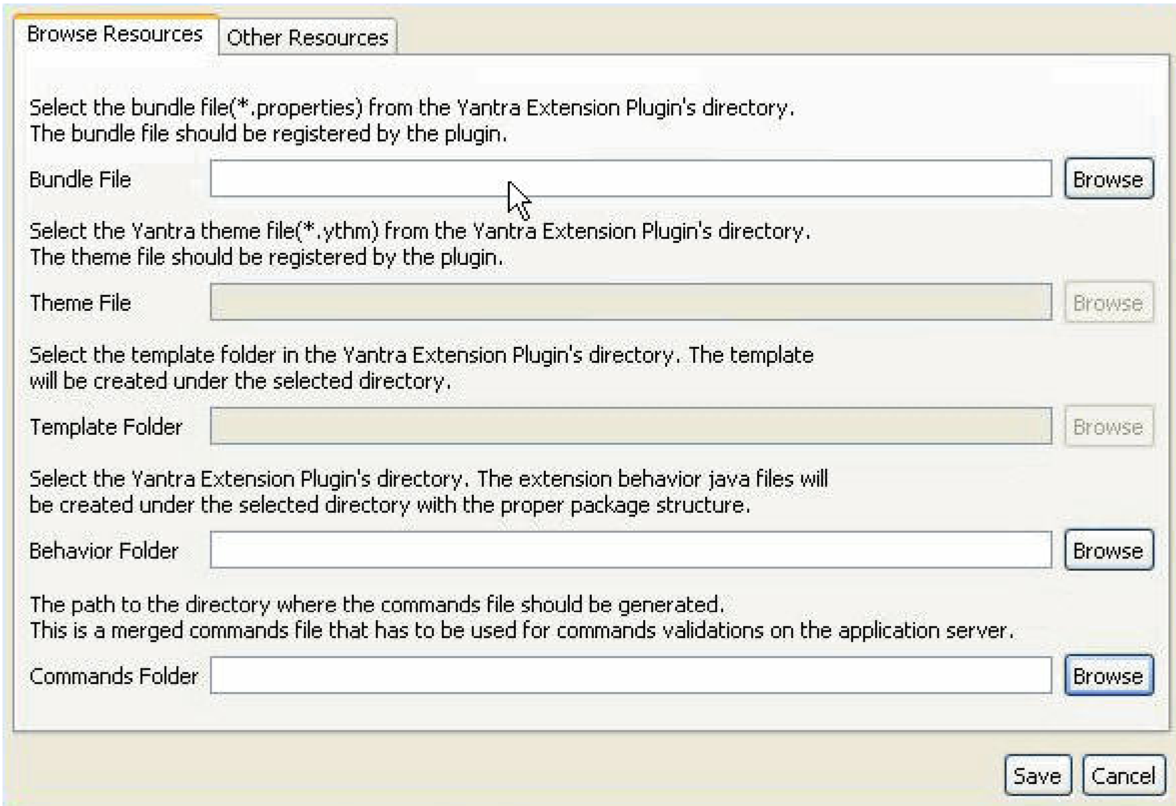
To synchronize the bundle file:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Bundle Entries tab.



3. Click **Save All**. The Save All window displays.



Note: You should save all files, folders, and packages in the plug-in directory.

4. Click **Save**.
5. Click **Close**.

Synchronizing Templates

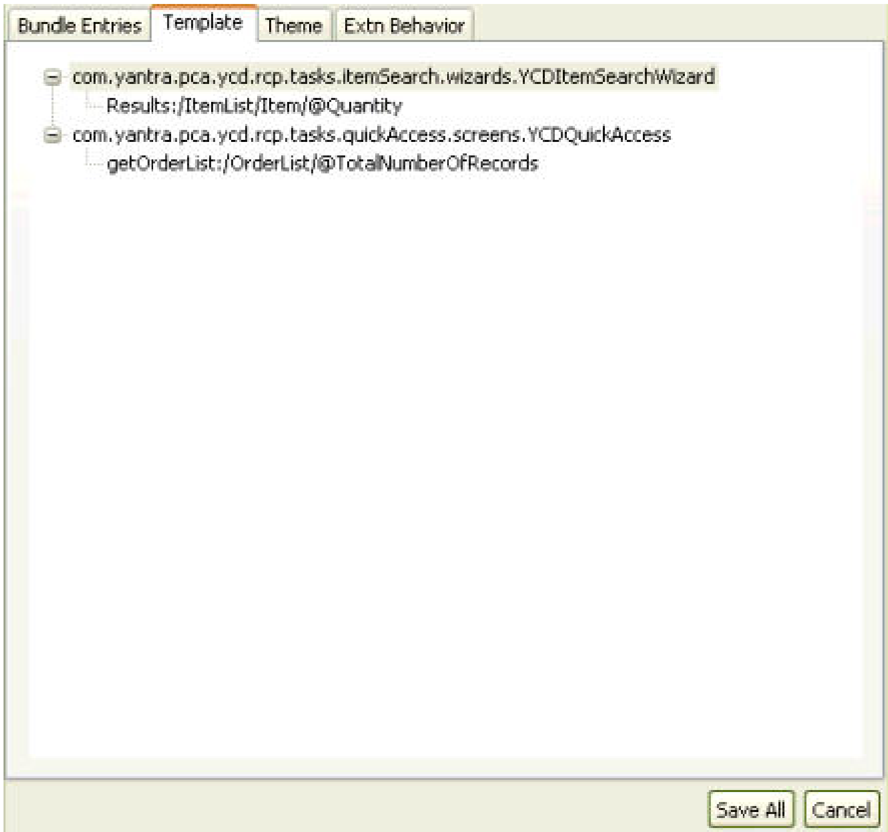
About this task

Whenever you specify bindings for newly added fields, you must synchronize the template folder with new templates that are defined for binding the newly added fields.

To synchronize the templates:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Template tab. You can view the newly added bindings for the newly added fields under the respective form id.



3. Click **Save All**. The Choose File for Saving Diffs pop-up window displays.
4. Click **Close**.

Synchronizing Theme Files

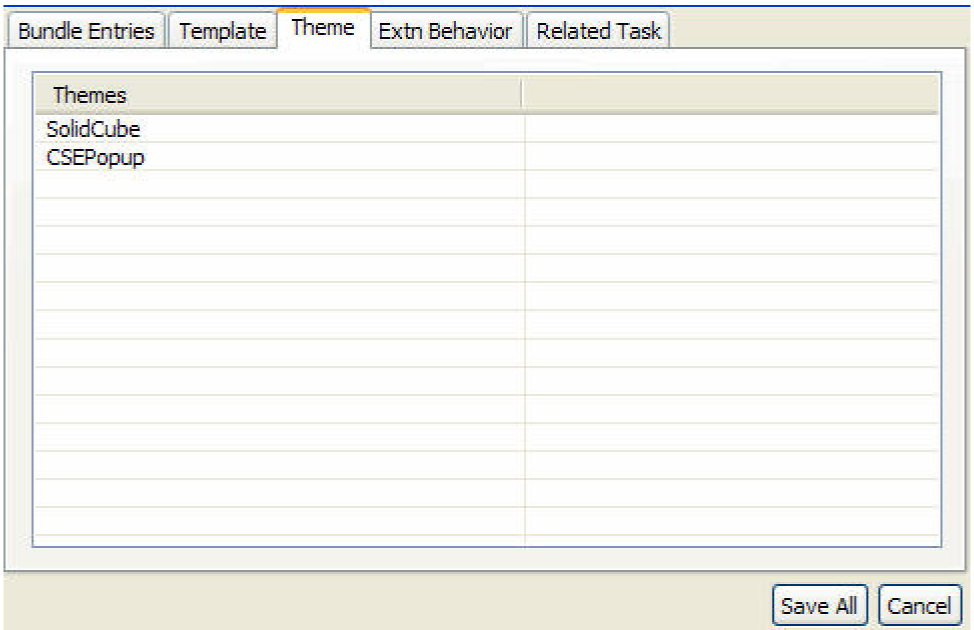
About this task

Whenever you specify a new theme entry for newly added fields, you must synchronize the theme file.

To synchronize the theme file:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click  . The Synchronize pop-up window displays. Select the Theme tab.



This window displays the new theme entries.

3. Click **Save All**. The Choose File for Saving Diffs pop-up window displays.
4. Click **Close**.

Synchronizing Extension Behavior

About this task

When you create an extension behavior, you must synchronize the differences.

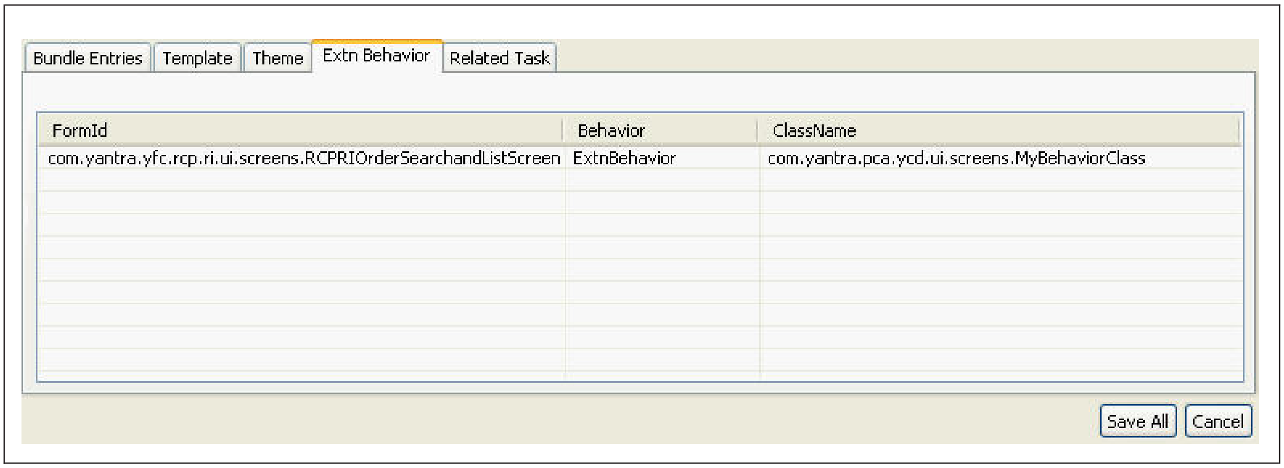
There are two types of extension behavior:

- Normal Extension Behavior (YRCExtensionBehavior)—The system creates this behavior when you create an extension behavior for extending a Rich Client Platform screen that is not a wizard. When synchronizing a normal extension behavior, specify the name of the implementation class (extends YRCExtensionBehavior). You can add the logic to call an API or service or set the extension model in the init() method of the YRCExtensionBehavior class.
- Wizard Extension Behavior (YRCWizardExtensionBehavior)—The system creates this behavior when you create an extension behavior for extending a Rich Client Platform screen that is a wizard. When synchronizing the wizard extension behavior, specify the name of the implementation class (extends YRCWizardExtensionBehavior). You can add the wizard-specific logic to call an API or service or set the extension model in the initPage (String pageId) method of the YRCWizardExtensionBehavior class.

To synchronize extension behavior:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Extn Behavior tab.



3. Click **Save All**. The Choose File for Saving Diffs pop-up window displays.
4. Click **Close**. The extension behavior class is created in the package that you specified.
5. To retrieve the appropriate data for this newly added field or advanced column, write the code to call the required API or service, set the extension model, and setting the bindings for an advanced control in the newly created extension behavior class.

Calling APIs or Services

After synchronizing the extension behavior for a newly added field, you may want to retrieve data to populate the new field by calling an API or service.

Setting the Extension Model

You must set the extension model for the new namespaces that you define for the extended screens.

Setting Bindings for an Advanced Control or an Advanced Column

You must set bindings for the extended table, advanced column, and advanced controls that you added using the Rich Client Platform Extensibility Tool.

Synchronizing Related Tasks

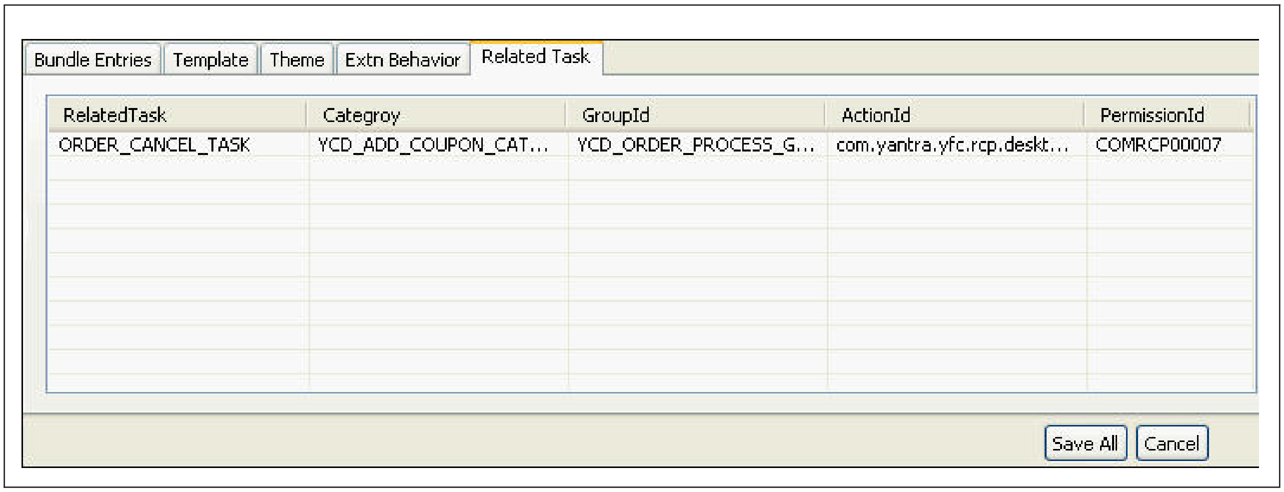
About this task

Whenever you specify bindings for the newly added fields, you must synchronize the template folder with new templates that are defined for binding the newly added fields.

To synchronize the templates:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Related Task tab. You can view the newly added related tasks along with the other details such as Category Id, Group Id, Action Id, and Permission Id .



3. Click **Save All**. The Choose File for Saving Diffs pop-up window displays.
4. Click **Close**.

Displaying Hidden and Disabled Controls

About this task

You can show the hidden and disabled fields using the Rich Client Platform Extensibility Tool.

To show the hidden and disabled fields:

Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The hidden and disabled fields display on the screen.

Viewing Shared Tasks

About this task

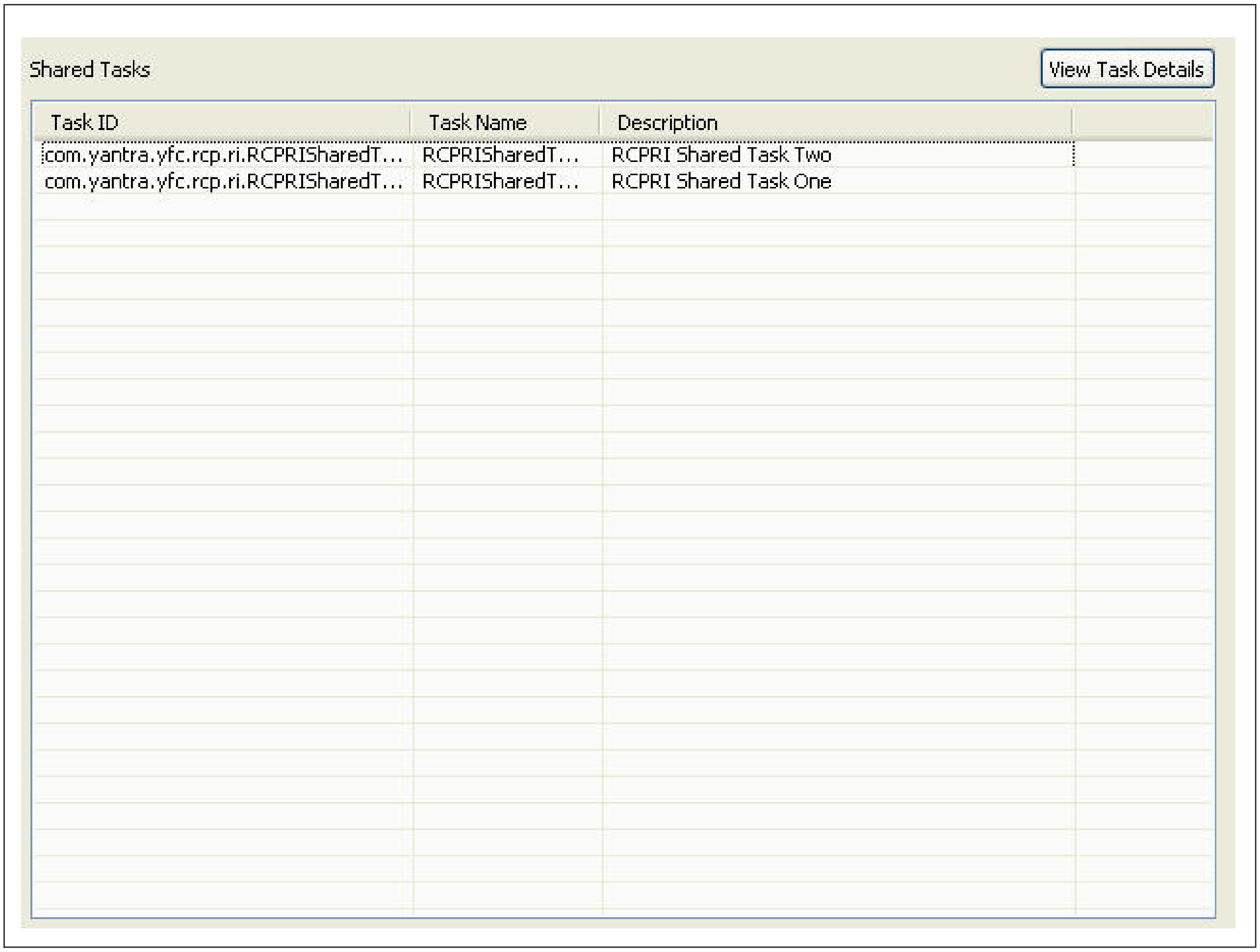
Using the Rich Client Platform Extensibility tool, you can view the shared task details used across applications or plug-ins. You can view the:

- Identifier of the shared task
- Name of the shared task
- A brief description of the shared task
- Template of the input XML model
- Template of the output XML model

To view the shared task details:

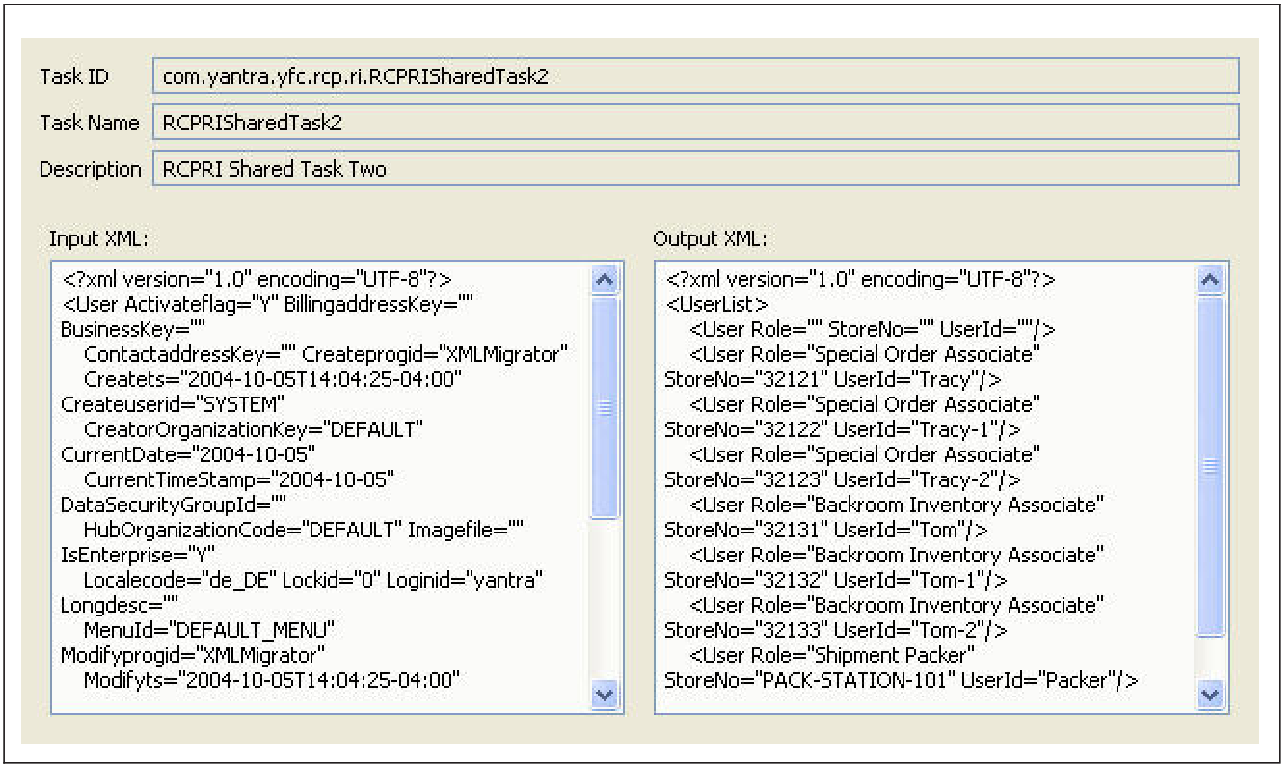
Procedure

1. Start the Rich Client Platform Extensibility Tool.
2. Click View Shared Tasks and click once on the screen. The Shared Tasks window displays.



Field	Description
Task ID	Displays the unique identifier of each shared task.
Task Name	Displays the name of each shared task.
Description	Displays the description of each shared task.

3. If you want to view more details about a particular shared task, select the shared task and click View Task Details. The Shared Task Details Panel pop-up window displays.



Field	Description
Task ID	Displays the unique identifier of the shared task.
Task Name	Displays the name of the shared task.
Description	Displays the description of the shared task.
Input XML	Displays the template structure that is used as an input XML by the shared task.
Output XML	Displays the template structure that is used as an output XML by the shared task.

Synchronizing New and Existing UI Resources Field Descriptions

Choose File for Saving Diffs Pop-up Window Options

Field	Description
Browse Resources Tab This tab is used to save all the differences in the bundle and theme entries in their respective bundle and theme files. Also, the differences in the templates and behavior class in the respective template and behavior folders.	
Bundle File	Click Browse. The Save File dialog box displays. Select the bundle file in which you want to save the new bundle entries.
Theme File	Click Browse. The Save File dialog box displays. Select the theme file in which you want to save the new theme entries.
Template Folder	Click Browse. The Choose Folder dialog box displays. Select the template folder in which you want to save the new template.
Extn Behavior Folder	Click Browse. The Choose Folder dialog box displays. Select the folder in which you want to save the new extension behavior class.

Field	Description
Other Resources Tab Displays the path of the plugin.xml file in which all your related task changes are saved.	
Update Resource	Uncheck this box if you don't want to save the changes in the plugin.xml file.

Synchronize Pop-up Window - Bundle Entry Options

Field	Description
Key	Displays the control names of all the newly added fields for which the corresponding bundle entry does not exist. **Note:** If you add new related tasks, this field displays the related task identifier and the related task group identifier.
Value	Editable field. Enter the value associated with the key, if applicable. This value is used when retrieving the value for the keys from the bundle file. **Note:** For a related task and related task group, this field corresponds to the respective name.

Synchronize Pop-up Window - Extn Behavior Tab Options

Field	Description
FormId	Displays the identifier of the form that you are extending.
Behavior	The type of behavior class automatically displays in the drop-down list. For example, WizardExtnBehavior or ExtnBehavior. • If you have created extension behavior for a Rich Client Platform screen that is a wizard, the drop-down list displays "WizardExtnBehavior". • If you have created extension behavior for a Rich Client Platform screen that is not a wizard, the drop-down list displays "ExtnBehavior".
ClassName	Enter the path of the behavior class. For example: `com.yantra.pca.ycd.ui.screens.MyBehaviorClass` where `com.yantra.pca.ycd.ui.screens` is the package name and `MyBehaviorClass` is the name of the behavior class.

Synchronize Pop-up Window - Template Options

Field	Description
RelatedTask	Displays the identifier of the new related task that you added.
Category	Displays the identifier of the identifier of the category in which the new related task is interested.
GroupId	Displays the identifier of the group in which you want to display the related task, so that the new related task can be displayed in the Related Tasks view.
ActionId	Displays the identifier of the action you want to invoke upon clicking the new related task.

Field	Description
PermissionId	Displays the resource identifier.

Chapter 7. Building and Deploying Rich Client Platform Extensions

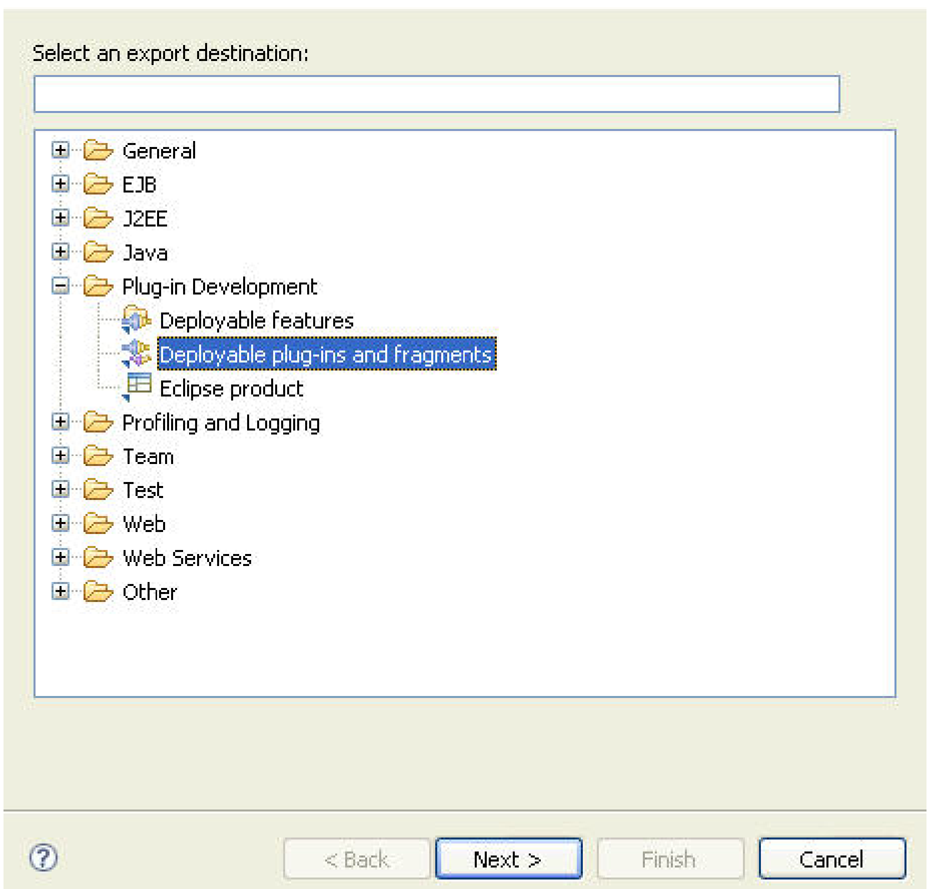
Building Rich Client Platform Extensions

About this task

Building the Rich Client Platform extensions is as follows:

Procedure

1. Start the Eclipse SDK.
2. From the menu bar, select **Window** → **Show View** → **Navigator**. The plug-in project is displayed in the Navigator view.
3. Right-click on the plug-in project that you want to build and deploy.
4. Select **Export...** from the pop-up menu. The Export window displays.



5. From the list of export destinations, under Plug-in Deployment, select **Deployable plug-ins and fragments**.
6. Click **Next**.
7. In the Destination tab, choose **Archive file:**.

8. Click **Browse** and browse to the folder where you want to store the exported plug-in compressed file.
9. In the Options tab, make sure that the Package plug-ins as individual JAR archives box is checked.
10. Click **Finish**. The plug-in jar is generated and stored in the plugins folder in the compressed file specified in step 8.

Deploying Rich Client Platform Extensions

About this task

After you build the Rich Client Platform extensions plugin jar, you must deploy this plug-in.

To deploy the Rich Client Platform extensions, copy the plugin jar that you built to the plugins directory of the *RCP_EXTN_FOLDER* folder. For more information about deploying and updating Rich Client Platform Application, see the Sterling Selling and Fulfillment Foundation:: *Installation Guide*

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise™, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce™, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Product Number: xxxx-xxx

Printed in USA