Building PowerView Pages

A PowerView page displays information from different locations in an application on a single page, allowing users to access and update commonly-used information quickly.

This section explains how to build and configure PowerView pages.

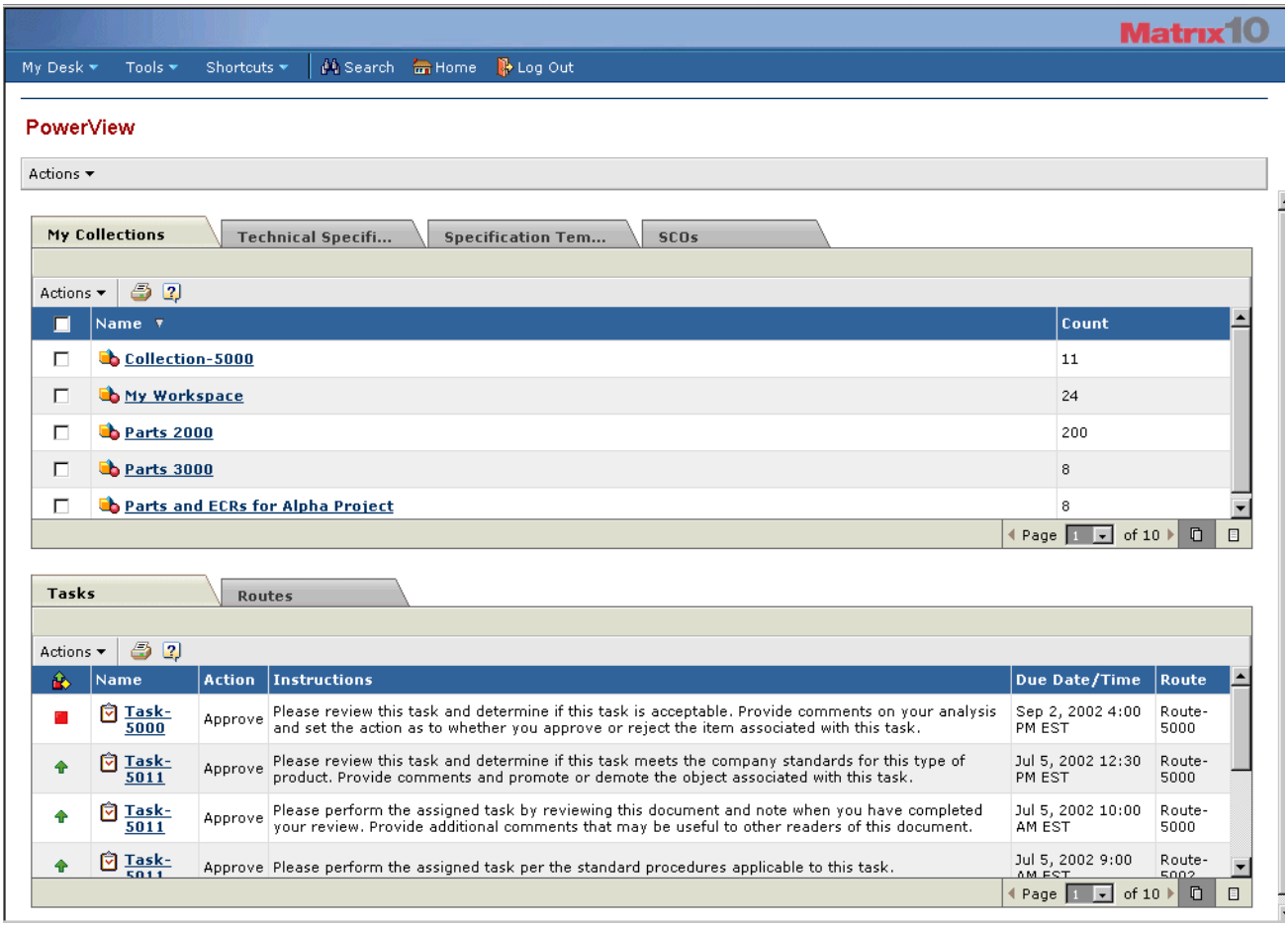
- *Two Kinds of PowerView Pages*
- *Overview of PowerView Pages*
- *Building a PowerView Page*
- *Parameters and Settings for Portal Objects*
- *Parameters and Settings for Tab Command Objects*
- *URL Parameters Accepted by emxPortal.jsp*

Two Kinds of PowerView Pages

There are two kinds of PowerView pages:

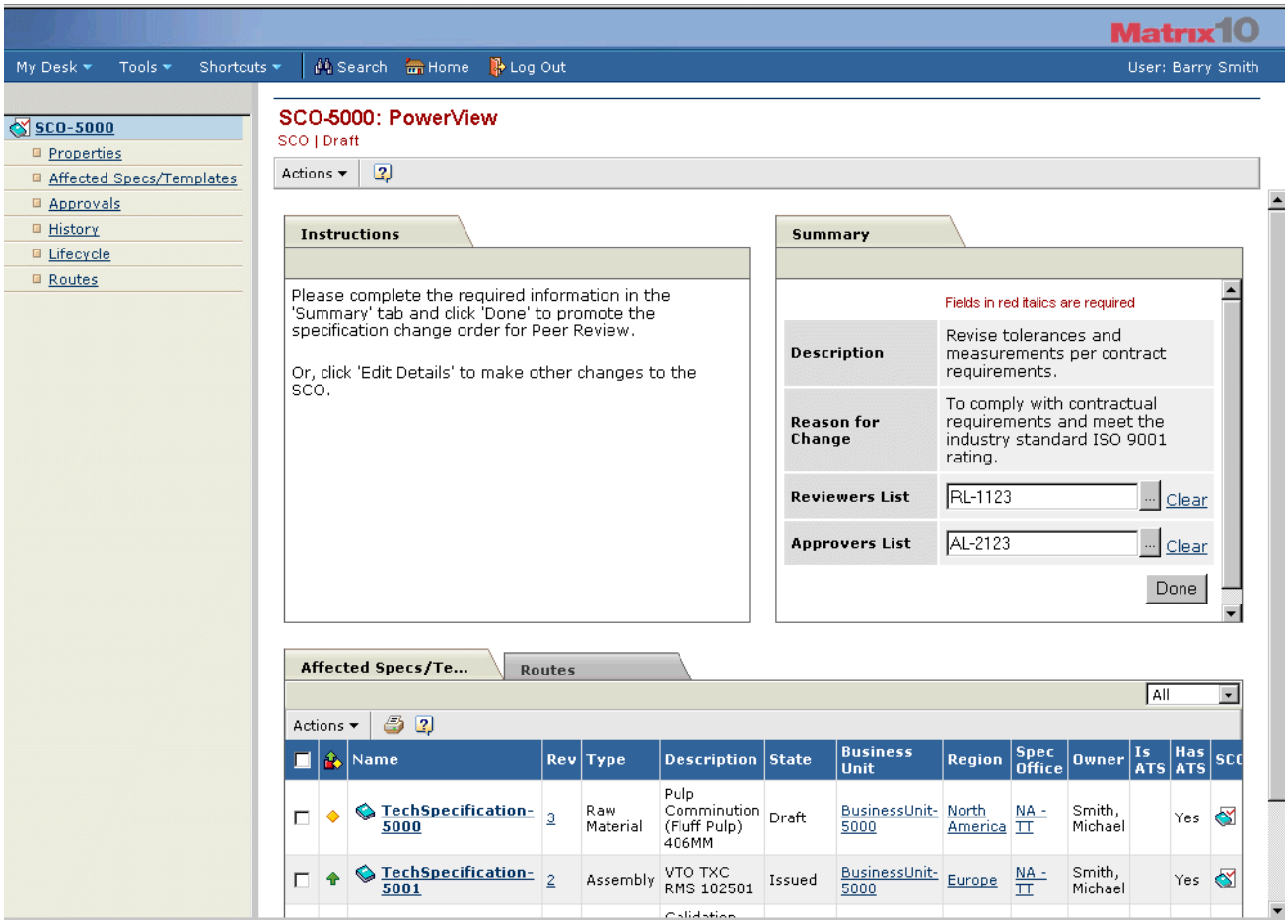
- Home PowerView

  Users can access their PowerView page using the Tools menu in the global toolbar. If the user's Home page preference is set to the PowerView page, the user can also access it using the Home tool. The PowerView lets users directly access items that are important to their daily use of the applications.



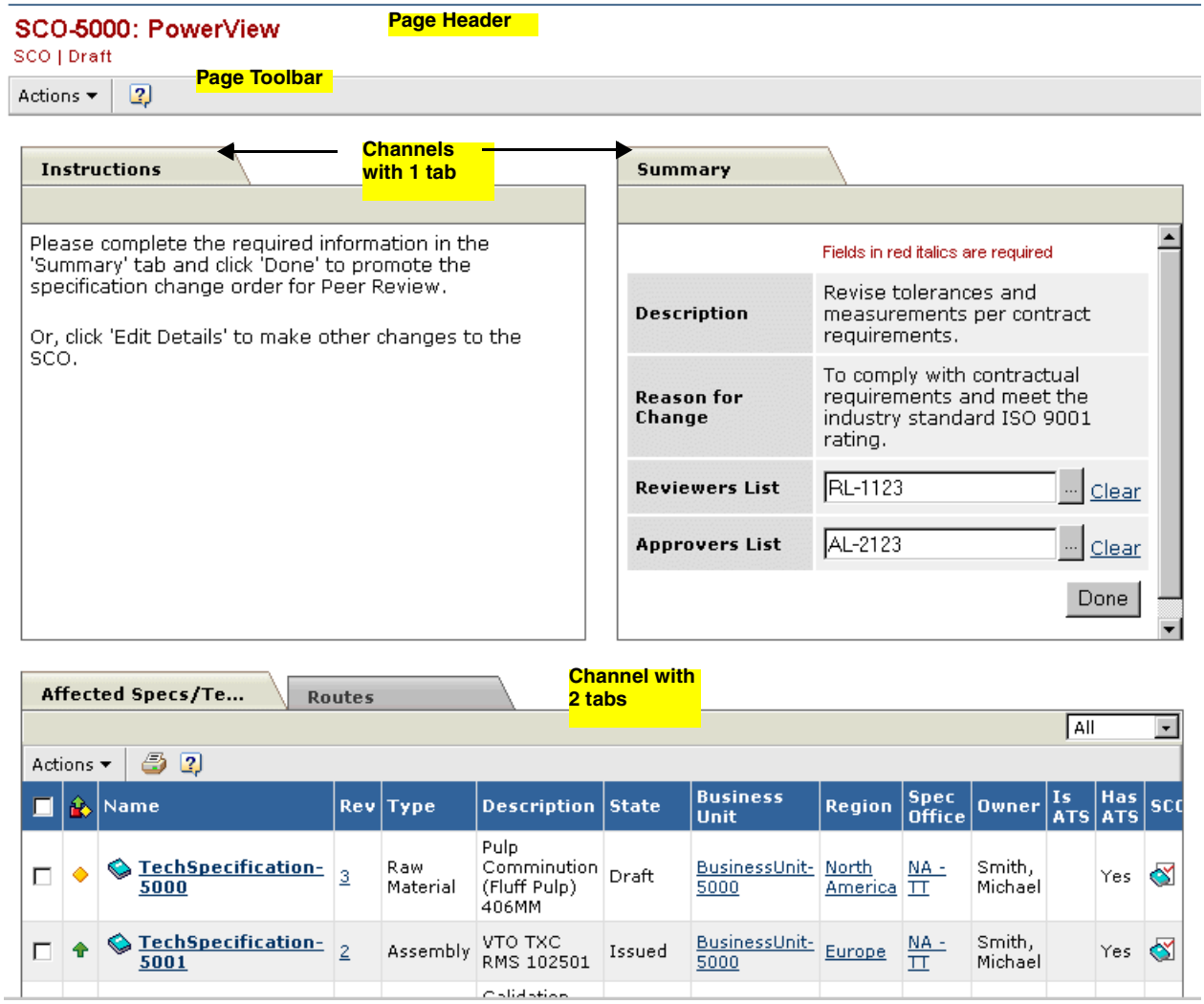
- Object PowerView for object types

  An object type can have a PowerView page defined for it that aggregates information about that object onto a single page. The PowerView can be set by the Business Administrator as the default page to display when users initially access objects of that type. The Context Navigator displays in the left frame and the PowerView page displays in the content frame on the right.



Overview of PowerView Pages

The following graphic shows the main components of a PowerView page. This is an example of a PowerView page for an SCO. The Context Navigator, which displays to the left of the PowerView, is not shown. The components of the Home PowerView are similar except it is not specific to an object type and does not include a Context Navigator on the left.



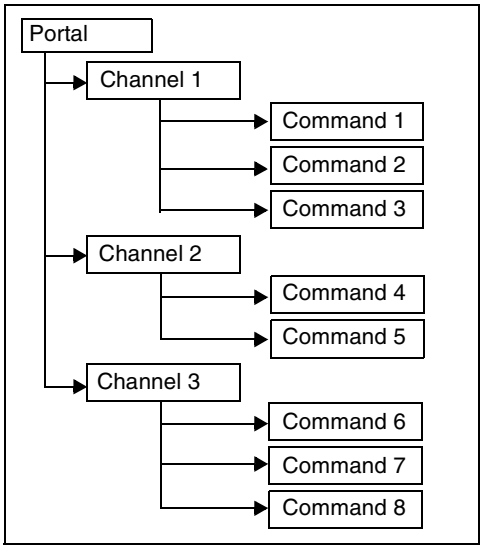
The channel tabs contain the actual content for users to view or work on. The tabs can contain any information that is normally displayed on an application page, such as an Actions menu, a configurable table with pagination controls and filter list, a read-only or editable configurable form, a custom JSP, or any Web page. For example, the channel in the upper right of the PowerView shown above contains an editable form page, and the selected tab on the bottom channel contains a table page, both of which were built with configurable components.

Clicking object links in an object PowerView page behaves the same as clicking a link displayed on any application page. The object is displayed in the main content area of the window and added to the category list under the appropriate category.

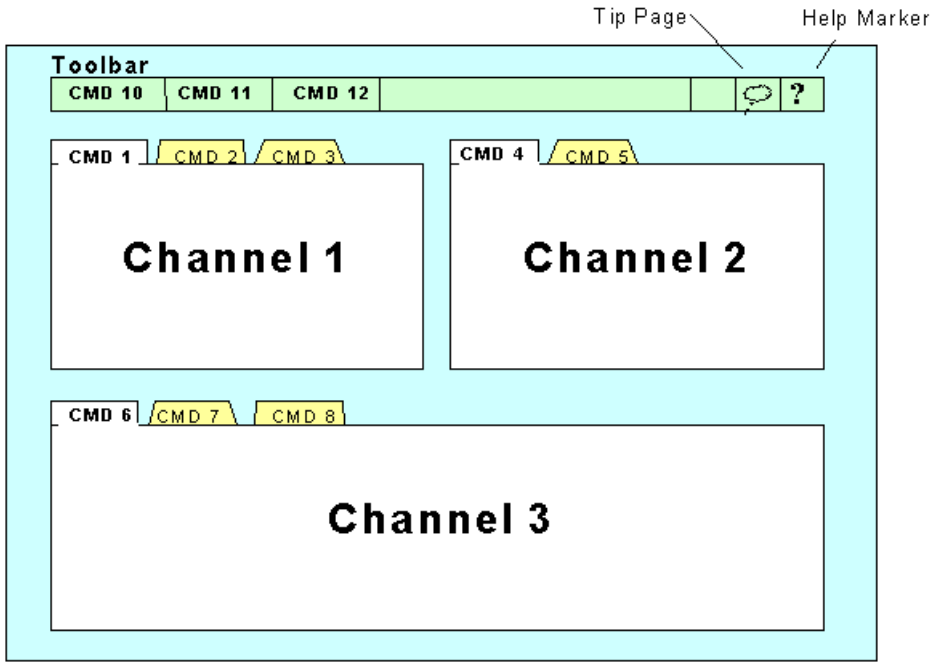
Mechanisms that Define a PowerView Page

The configurable JSP emxPortal.jsp creates PowerView pages. This JSP accepts parameters that define the content and behavior of the page. These parameters are passed to the JSP through the href URL.

This graphic shows the admin objects needed to define a PowerView page. The main components of the PowerView page are defined as portal, channel, and command admin objects. The names of the top-level PowerView object and the toolbar menu object (not shown below) are passed to the emxPortal.jsp through its URL parameters.



The above admin objects would create a PowerView page like the one shown below. Note that the page toolbar is also defined using menu and command objects but these objects are not shown in the above graphic. For information on page toolbars, see *Building Toolbars*.



There should be a portal admin object for each unique PowerView page and a channel admin object for each unique channel. Channel admin objects can be assigned to more than one portal object.

How the Home PowerView Page is Linked to the Application

The Framework installs the portal admin object called AEFPowerView, and the first channel of the Home PowerView page, called AEFPowerViewChannel. Each installed MatrixOne Application adds channel objects to this object to define channels appropriate for that application. For example, Specification Central might add a channel that contains tabs for collections, specifications, and SCOs. You can configure the channels and tabs assigned to AEFPowerView as needed but this object must be used for the Home PowerView.

The order in which the channel objects are assigned to the portal object determines where the channel displays on the PowerView page. Therefore, the order in which the applications are installed determines the default order the channels display in. For example, the channels added for the first MatrixOne Application installed will be listed first on the Home PowerView page, the channels for the second application installed will be listed next and so on. You can change the order by changing the order in which they are assigned to the portal object.

Users typically access the Home PowerView page by clicking Tools> PowerView from the global toolbar or by clicking the Home tool from the global toolbar when the Home page preference is set to PowerView. The component that calls the home PowerView page must specify the configurable JSP for PowerView pages, emxPortal.jsp, in the href parameter. The href must include a parameter that specifies AEFPowerView, which is the object installed by the Framework that represents the PowerView. The href should include any other parameters needed to configure the page, such as a toolbar menu object, as described in *URL Parameters Accepted by emxPortal.jsp*.

For example, when entered in the href parameter for an appropriate command admin object, the following URL creates a home PowerView page:

```
${COMMON_DIR}/
emxPortal.jsp?portal=AEFPowerView&toolbar=AEFPowerViewToolbar&H
elpMarker=HelpMarker&TipPage=tippage&PrinterFriendly=true&Expor
t=true
```

How Object PowerView Pages Are Linked to MatrixOne Applications

Users typically access the PowerView page for an object type by clicking an object name from a table or by searching for a specific object. For example, in Engineering Central, clicking the name of an ECO from a table or search results displays the ECO PowerView page for that object.

The component that calls the object PowerView page must specify the configurable JSP for PowerView pages, emxPortal.jsp, in the href parameter. The href must include a parameter that specifies the portal object defined for the PowerView page. The href should include any other parameters needed to configure the page, as described in *URL Parameters Accepted by emxPortal.jsp*.

For example, when entered in the href parameter for an appropriate command admin object, the following URL creates a PowerView page:

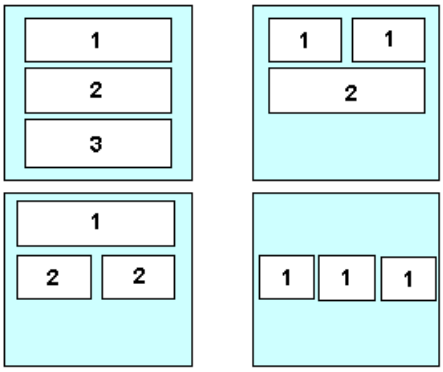
```
${COMMON_DIR}/
emxPortal.jsp?portal=SPCSCOPortal&toolbar=SPCSCOToolbar&header=
SummaryView&HelpMarker=emxhelpscosummary
```

Controlling the Layout of Channels

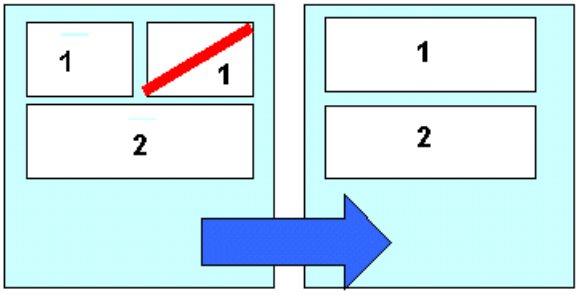
A channel can contain one or more tabs. There is no limit for the number of channels that can be displayed in a row or on a single PowerView page but displaying more than 2 channels in a row or more than 3 rows would make the page cumbersome.

The row a channel appears in is determined by how it is assigned to the portal object. If you are using Business Modeler to create portal objects, channels can be arranged in the Items tab to reflect how they will appear on the page. In MQL, use the Channel clause of the Create Portal or Modify Portal command to arrange the channels. See the *Matrix Business Modeler* or *MQL Guide* for details.

For example, channels can appear one below the other, or they can be grouped to appear side by side.



The PowerView page automatically compensates if the user does not have access to any of the tabs in the channel by hiding the channel and rearranging the display of channels. For example, if the user does not have access to the second channel in the first row, as shown in the left of the figure below, the layout shown on the right displays instead.



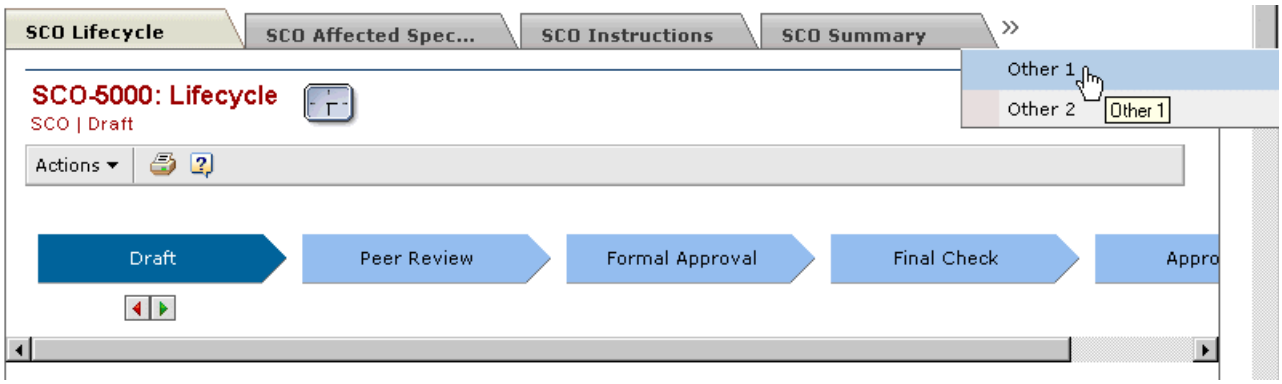
Controlling the Layout of Tabs within a Channel

The tabs within channels contain the actual content to be viewed or worked on. There is no limit to the number of tabs that can be assigned to a channel, but more than 4 or 5 tabs would be cumbersome.

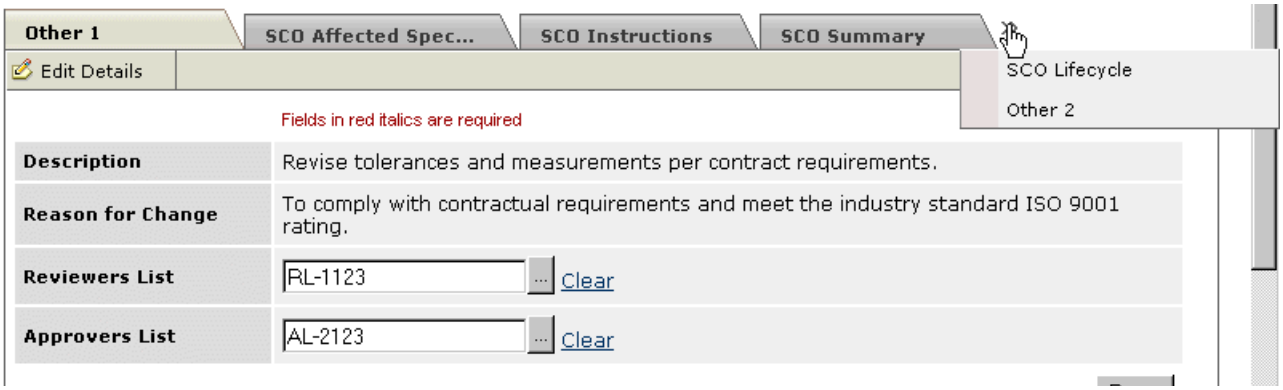
The order in which tabs appear within a channel is determined by the order the tab commands are added to the channel object.

Up to 4 tabs can be displayed on a full, page-width channel and 2 tabs for half page-width channels. For channels that have more tabs, a small arrow tab is displayed. This tab provides a drop-down menu of the remaining tabs. When the user selects one of the items in the drop-down menu, the currently-selected tab is replaced by the newly-selected tab. The tab it replaces is then placed in the drop-down menu.

For example, this graphic shows a page-width channel that has 6 tabs. The 5th and 6th tabs are accessed using the >> drop-down menu. When the 5th tab (Other 1) is selected...



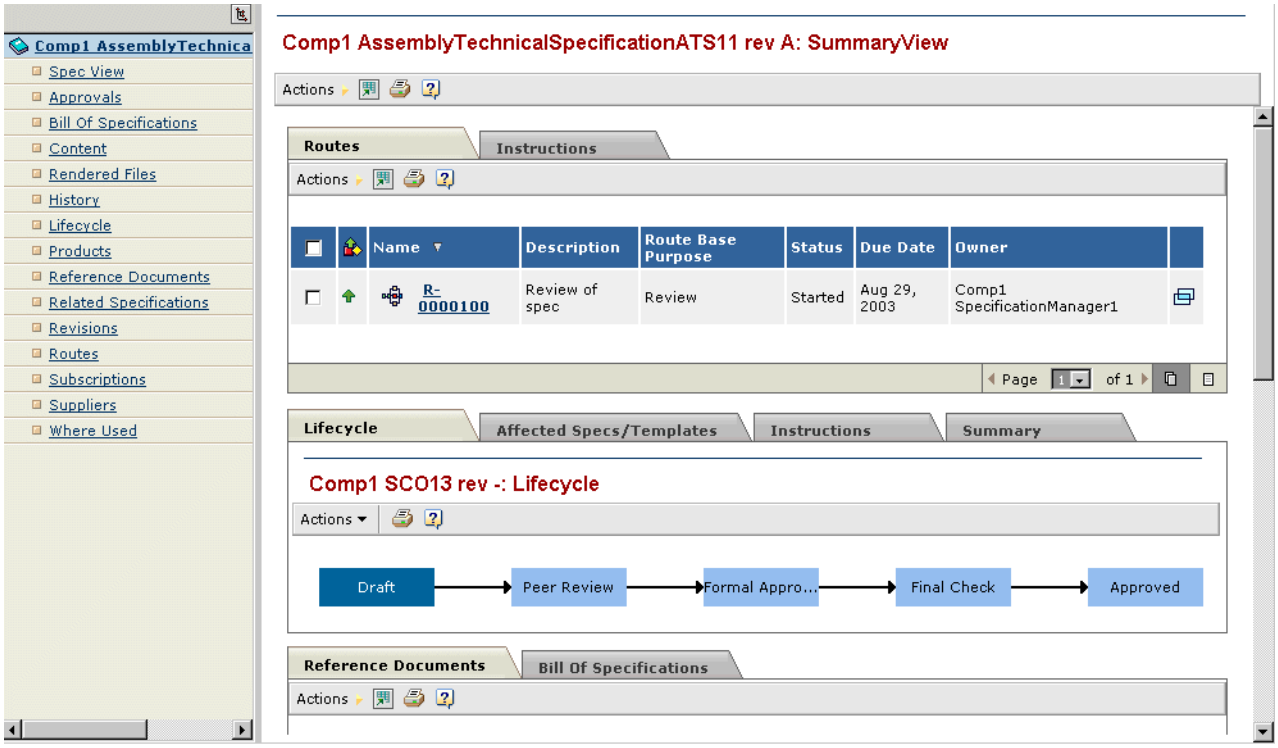
...it replaces the previously-selected tab (SCO Lifecycle). The previously-selected tab is then listed in the >> drop-down menu.



Steps for Building a PowerView Page and Linking it to an Application

The following procedure lists the main steps needed for creating a PowerView page and connecting it to an application. The procedure explains how to create all the components needed for a PowerView page from scratch. You would use this procedure if you wanted to create a PowerView page for an object type that doesn't have one by default. Since the PowerView installs by default with the Framework and each installed application adds to it, you wouldn't build a PowerView page for the Home PowerView from scratch. To configure the PowerView portal page or an existing object PowerView page, you add and remove portal, channel, menu and command objects and change settings for the objects.

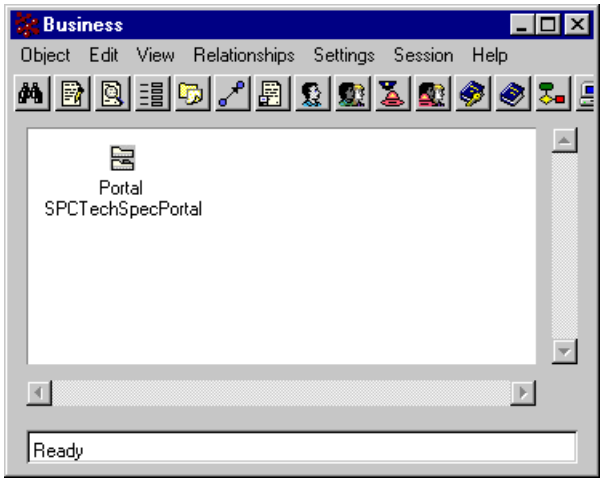
The graphics included in the procedure represent the administrative objects needed to build an object PowerView page for technical specifications, as shown in this graphic. This sample PowerView page has 3 channels, each positioned in its own row. The first and third channels have 2 tabs (commands) each and the second has 4 tabs.



To create a PowerView page and add it to an application

1. Create a portal object to represent the PowerView page.

   For naming conventions, see *Naming Conventions for UI Administrative Objects*.

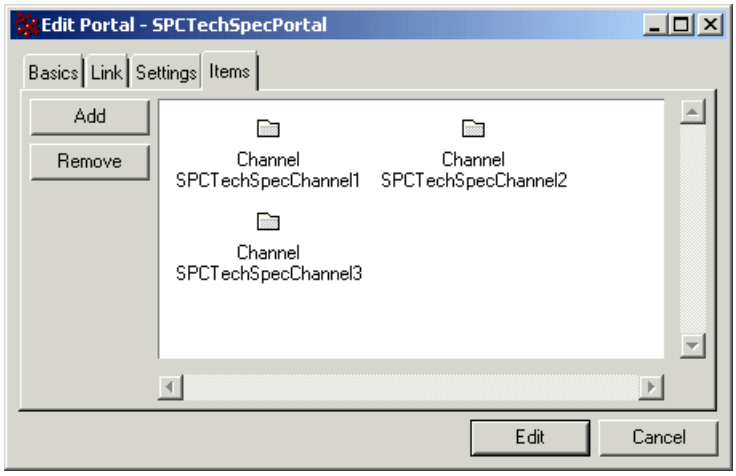


2. Create a channel object for each channel that should be on the PowerView page.

   When you create a channel object, you define whether it appears alone in a row on the page or in a row with other channels. For a description of how to fill in parameters and settings for channel objects, see *Parameters and Settings for Portal Objects*. For naming conventions, see *Naming Conventions for UI Administrative Objects*.

3. Assign the channel objects to the portal object that you created in Step 1.

You can drag the channel objects up and down or right and left within the Items tab to rearrange them in the order you want them displayed on the PowerView page. For details, see *Controlling the Layout of Channels*.

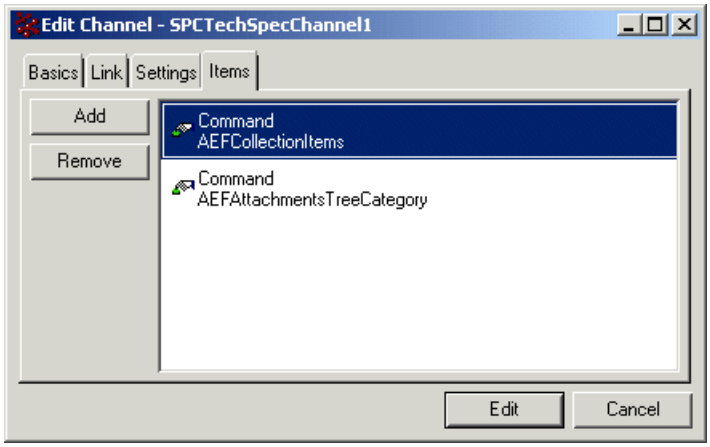


4. Create a command object for each tab that should be included for each channel.

   For a description of how to fill in parameters and settings for channel objects, see *Parameters and Settings for Tab Command Objects*. For naming conventions, see *Naming Conventions for UI Administrative Objects*.

5. Assign the tab command objects to the channel that it belongs to.

   The order you add the tab commands to the channel object determines the order they display on the channel. You can drag the command objects up and down within the Channel Items tab to rearrange them in the order you want them displayed.



6. Make sure the admin objects are created for the action bars, toolbars, table pages, and form pages needed for the PowerView page and any of its channels/tabs.

7. In the href parameter for the menu or command object that should call the table page (or in the JSP if a JSP is calling the page), enter emxPortal.jsp and specify the parameters as needed to display the page. Since emxPortal.jsp is in the ematrix/common folder, the first part of the URL should usually be: ${COMMON_DIR}/emxPortal.jsp. Here is an example of an href value for an object PowerView page. For a description of the parameters, see *URL Parameters Accepted by emxPortal.jsp*.

```
${COMMON_DIR}/
emxPortal.jsp?portal=SPCSCOPortal&toolbar=SPCSCOToolbar&header=
SummaryView&HelpMarker=emxhelpscosummary
```

8. If you are working with the Web-based user interface as you are making changes and want to see your changes in the user interface, click the **Reload Cache** tool in the toolbar and click the browser **Refresh** button.

   The cache is refreshed automatically when the component age expires. This setting is in emxSystem.properties.

Only persons assigned to the Administration Manager role have access to the Reload Cache tool.

Parameters and Settings for Portal Objects

This table describes how to fill in the parameters for portal, channel, and command objects that represent a PowerView page. For specific instructions on how to create objects using Business Modeler or MQL, refer to the *Business Modeler Guide* or *MQL Guide*.

Parameter	Description	Accepted Values/Examples
Alt	Not applicable to menu objects for portals or channels.	—
Commands (specified in the channel Items tab in Business Modeler)	The command objects that represent the tabs in the channel. The order of the commands is the order the tabs appear in the channels in the user interface.	Names of command objects, such as: SPCSCOInstructions SPCRoutesTreeCategory
Href	Not applicable to objects for portals or channels.	emxDynamicAttributes.jsp ${COMMON_DIR}/ emxPartInfo.jsp ${SUITE_DIR}/ emxViewPartAttr.jsp ${ROOT_DIR}/emxHome.jsp
Icon	Not applicable to objects for portals or channels.	The name of an image file, such as Part.gif.
Label	Not applicable to objects for portals or channels.	$<type> $<name> $<attribute[attribute_Weight].value > Engine - $<type> $<name> Connected ECR emxFramework.Common.Part ${TYPE} ${NAME}

Parameters and Settings for Tab Command Objects

This table describes how to fill in the parameters for command objects used for tabs that appear within channels on PowerView pages. For specific instructions on how to create

command objects using Business Modeler or MQL, refer to the *Business Modeler Guide* or *MQL Guide*.

Parameter	Description	Accepted Values/ Examples
Icon	Icon for the object within the Matrix core. This setting does not display an image in the UI.	The name of an image file, such as Part.gif.
*Label	The text that should display on the tab. The label can be made up of: • Plain text, such as "ECRs". • A string resource ID, such as "emxEngineeringCentral.common.portal". To internationalize the text, you must use a string resource ID. See *Internationalizing Dynamic UI Components*. The system first looks for a string resource ID that matches the entered value. If it finds one, it uses the value for the ID. If it doesn't find one, it displays the entered text and/or output from the select expression. The label can be up to 17 characters in length. If it exceeds 17 characters, it is truncated to17 characters and an ellipsis ("…") is added to the label. The HTML "Title" attribute is defined so that when the user mouses over the tab, the full label displays. 	Properties EBOM Lifecycle emxEngineeringCentral.common.portal
Href	The URL to display when a user selects the tab. Tabs can display content from the following sources: • configurable table • configurable form • JSP page • URL If the specified URL contains a header parameter, it is not used because the header would be redundant with the tab label. For example, if the URL contains emxTable.jsp with a header specified, the table header is not used.	emxTable.jsp?xxxxx emxForm.jsp?xxxxx Sample.jsp?xxxxx www.matrixone.com
Alt	The Alt text to appear over the tab label when the user mouses over the label.	—
Access	The persons, roles, and groups who can access the tab. To make the tab available to all users, regardless of role/group assignments, choose All. Note that if no users are assigned access, the system assumes all users have access. If a user doesn't have access to any tabs in a channel, the channel is not displayed. Also see *Controlling User Access to User Interface Components*.	Names of group, role, person administrative objects. *Or* All (default)
Settings	Additional settings that define the behavior and appearance of the tab command. For a list of the accepted settings, see the table below.	Name/value pairs, as defined in the table below.

This table lists and describes the settings for command objects used for tabs. Note that the name and value for each setting are case sensitive.

Setting	Description	Accepted Values/Examples
Access Expression Access Function Access Mask Access Program	All these settings can be used to control access to tabs just as they can be used to control access to other UI components. For details, see *Controlling User Access to User Interface Components*.	—
*Registered Suite	The application the command belongs to. The system looks for files related to the command in the registered directory for that application, which is specified in emxSystem.properties. Based on the application name, the system passes the following parameters in the href URL: • suiteKey • emxSuiteDirectory • StringResourceFileId	The value must be set without any spaces, for example, EngineeringCentral or Framework. The value must be set to the suite name as defined in the key eServiceSuites.DisplayedSuites within emxSystem.properties. If the suite name starts with "eServiceSuite" then this prefix can be skipped and assign the remaining text to the setting. For example, if the suite name in emxSystem.properties is "eServiceSuite**EngineeringCentral**", then the word "**EngineeringCentral**", can be assigned as "Registered Suite". In the href URL that is called when the tab is clicked, the system passes a parameter called "suiteKey". The value for the parameter is the property name from emxSystem.properties that maps to the setting's value.
***Required Setting**		

URL Parameters Accepted by emxPortal.jsp

This table lists the parameters that emxPortal.jsp can use. You can add these parameters to the href parameter for the component that calls the portal jsp. For example, when you specify the emxPortal.jsp to be called from a tree category, you can add these parameters to the href parameter for the menu object.

Parameter	Description	Accepted Input Values
portal	Specifies the portal admin object that represents the top-level menu, which defines the channels.	Name of channel objects
toolbar	Specifies the menu admin object that represents the toolbar, which appears in the page header.	Name of menu admin object.
alternateCmd	Specifies the command admin object. This is used only for unsupported browsers (Netscape 4.7xx and Internet Explorer 5.2 in Macintosh) to use as the alternate command to display a specific page configured in the alternate command instead of the portal page. This is an optional parameter.	Name of command admin object. Default: The first available command in the channel will be used for displaying the page.

Parameter	Description	Accepted Input Values
HelpMarker	Specifies the name of the help marker to call for context-sensitive help. For information about implementing help, see *Implementing Context-Sensitive Help*.	String The naming convention for help markers "emxhelp" followed by the object or feature and then the action, for example, emxhelproutecreate and emxhelpprojectedit. The marker is all lowercase with no spaces.
TipPage	Specifies whether the page should include the Tip Page tool and call a specific html or JSP when a user clicks the tool. If this setting is not included, the Tip tool is not included on the page.	Name of a custom html or JSP page, including any path. The starting point for the directory reference is the content directory. For example, if you want to call an html file in ematrix/doc/customcentral and the content directory is ematrix/customcentral, you would add this parameter to the table.jsp: TipPage=../doc/customcentral/tippage.html

URL Parameters Accepted by emxTable.jsp

Displaying table component listings in half page-width channels will typically cause the user to scroll horizontally to the view all columns in the table. For this situation, you can create an alternate table with fewer columns for display specifically within a channel tab. For example, if a standard ECO New Parts table contains 10 columns, you could define an alternate ECO New Parts table that contains just 4 columns. The `portalTable` parameter can be used to pass in to emxTable.jsp the name of the alternate table to be displayed within a channel tab.

You can add this parameter to the href parameter for the component that calls the table.

Parameter	Description	Accepted Input Values
portalTable	Specifies the alternate table that can be used within a channel tab.	Name of alternate table

Naming Conventions

[add the following rows to the existing section on naming conventions for configurable components]

Admin Object Type, Usage	Convention	Examples
portal objects for object PowerView	3-letter standard abbreviation for application name (see table below), followed by the feature name, followed by “SummaryView”.	SCSRFQSummaryView
channel objects	3-letter standard abbreviation for application name (see table below), followed by feature name, followed by “Channel”.	SCSRFQLineItemsChannel
command objects for channel tabs	3-letter standard abbreviation for application name (see table below), followed by feature name.	SCSRFQLineItemsAttachments

Objects and Menus

[add the following row(s) to the commands table in the schema definitions chapter]

Command	Description	Configured As:
AEFCollections	Command installed by the AEF that will be the first command of the first channel (AEFPowerViewChannel) of the PowerView page.	Tab for channel menu
AEFLogoutToolbar	Command installed by the AEF that will be the only command actions toolbar (AEFPowerViewActionsToolbar) of the PowerView page.	Action bar for page

[add the following row to the menus table in the schema definitions chapter]

Menu	Description
AEFPowerViewToolbar	Represents the toolbar menu that is installed by the AEF. This is used as the toolbar for the Home PowerView page. AEFPowerViewActionsToolbar is the actions menu that is installed by the AEF. This is used by the toolbar AEFPowerViewToolbar for its actions.
AEFPowerViewActionsToolbar	Represents the actions menu that is installed by the AEF. This is used by the toolbar AEFPowerViewToolbar for its actions.

[add the following row to the objects table in the schema definitions chapter]

Business Object	Description
AEFPowerView	Portal object installed by the AEF, used for Home PowerView page. Applications can add their own channels to AEFPowerView.
AEFPowerViewChannel	Channel object that contains the first channel of the PowerView. All other applications can add tabs to this channel.