



Custom Themes

This chapter contains the following topics:

- [Overview, page 1](#)
- [Custom Style Sheets, page 1](#)
- [Style Summary and Recommended Practices, page 13](#)
- [Custom Headers and Footers, page 21](#)

Overview

This chapter describes the capabilities provided to customize the Service Catalog web pages. The customer-facing modules in the application can be customized using Cascading Style Sheets (CSS) and custom headers and footers.

The pages which may be customized include:

- Pages displayed in the Cisco Prime Service Catalog, My Services, and Service Manager modules including service forms dynamically generated, based on definitions specified via Service Designer
- The portals for Reporting and Advanced Reporting
- The login pages
- Preconfigured and custom portal pages in the Service Portal solution

The appearance of modules used by service designers and administrators to configure and manage Service Catalog cannot be customized. These modules include Service Item Manager, Service Designer, Organization Designer, Administration, Catalog Deployer, Portal Designer, and Service Link.

Custom Style Sheets

Contents of the Service Catalog application are presented as web pages formatted using HTML5. Cascading Style Sheets (CSS) offer the ability to customize the web pages by changing the definition of themes used to display the pages, rather than having to edit the pages themselves.

Custom themes allow designers to customize Service Catalog web pages, headers and footer. Custom themes may be applied to all users of an application instance, or different themes may be applied to users based on their home organizational unit.

Prerequisites

- You must have access to the file system of the application server, specifically to the “custom” directory of the RequestCenter.war archive and its subdirectories. You need both read and write access to this directory and to its subdirectories.
- You must have a user role which includes the Administration capability to “Manage Global Settings” in order to turn on or off the use of custom style sheets, headers, and footers.
- Browser page caching must be turned off in order for you to test style sheet changes.
- Ideally, you should have access to an application instance where you can test your changes without disturbing the work of other analysts or developers.
- A style sheet editor and other editing tools are highly recommended, but not required.

Customizing Built-In Modules

The procedure below gives the basic steps to follow in order to customize the styles used in the built-in modules, namely My Services, Service Catalog, Service Manager, and Reporting. Additional details on these styles are given in the following sections.

- 1 Create a directory on the application server, under the RequestCenter.war/custom directory, where the files required for the custom styles will reside. In a Linux deployment this directory would be, `/opt/CiscoPrimeServiceCatalog/jboss-as-7.1.1.Final/ServiceCatalogServer/deployments/RequestCenter.war/custom`. The directory will typically have an images subdirectory, for any custom images. The directory name should indicate the tenant/organization name to which the style will apply.
- 2 If you use the Service Catalog module as the end user module, copy all the files located in the custom/ServiceCatalogExamples directory into the new directory created in Step 1. In a Linux deployment this directory would be, `/opt/CiscoPrimeServiceCatalog/jboss-as-7.1.1.Final/ServiceCatalogServer/deployments/RequestCenter.war/custom/SmallCompany`. If you use My Services module, copy all the files located in the custom/CustomExamples instead. Location of this archive will vary, based on your application server and installation setting.
- 3 In the new directory, remove the “example_” prefix of files in the directory as well as in the application directory under it. Modify the css files and add image files as needed to tailor the look and feel of the user interface as needed.
- 4 Use the Custom Styles page in the Administration module to define the style, specify the directory on which required files reside, and assign the organizations to which the style name applies.
- 5 Use the Settings page in the Administration module to turn on custom stylesheets.
- 6 Restart the browser session of Service Catalog—the pages should appear with the customizations applicable to the logged in user. You must exit and restart the Service Catalog session when custom stylesheets are initially activated. To test subsequent changes to the styles, it is sufficient to copy the revised style sheet to the application server and refresh the current page. The new styles will be applied, provided page caching is not in effect.

**Caution**

Once you change the Administration Settings to use custom stylesheets, the custom.css file should be present on the specified directory. If the file is not present, Service Catalog will use its standard styles. Similarly, if the option to use a custom header or footer is turned on, the corresponding files must be present on the specified directory.

Defining a Custom Theme

A CSS Style designer will put the CSS in a directory on the server. In the example below you name a custom Style, and then associate it with a Style Directory, enter the description, and specify whether this style should apply to every user for the site, the sub-OU's under the associated Organizational Units, and whether it should apply to the Service Catalog module.

Fill in the properties as follows:

To start using custom style sheets or headers and footers:

-
- Step 1** Log in to Service Catalog, choose the **Administration** module, and go to the **Settings** tab. The Customizations page appears.
- Step 2** At the right side of the screen, choose the **Custom Themes** option from the option list.
- Step 3** Click **Add** to create a new theme. The Custom Theme Properties page appears. See the below table to understand the Custom Theme Properties.
- Step 4** Fill in the properties and click **Create** to create the theme. You can then edit the theme, to specify the organizations to which the theme applies.
- Step 5** Click **Add** in the Associated Organizational Units pane. The Organizational Unit Search window appears. Choose one or more organizational units.
- Note** You must create an user (see [Adding a Person](#) section) and add that user to the organizational unit. Only the user who is a member of the organizational unit can view the customization.
- Step 6** You may edit the theme definition or the business units to which it applies at any time.
-

Table 1: Custom Theme Properties

Field	Description
Name	The theme name should reflect the OU or group of OUs the organizations to which the styles will apply.
Description	(Optional) Provide a description for the custom theme.
Apply this Theme to all subOUs	If a hierarchical structure is used in the organizations, you may specify that a theme is inherited by all child OUs of a parent.

Field	Description
Make this Theme the default for the entire site	One theme may be designated as the default. If a default is specified, it is used for any user whose home organization (OU) has not been assigned a theme. If no default is specified, the system-defined style sheets is used.
Classic Custom	You may choose the Theme Directory from any directory under <i>RequestCenter.war\custom</i> . The directory must exist before you can create the theme. The default directory named 1, already exists.
Website for Service Catalog	Select this check-box to enable the browse option. For both Desktop and Mobile custom theme, you may choose the theme directory from the Website Directory pop-up. The default directory named ServiceCatalogWebsite, already exists. Note The directory must exist under RequestCenter.war\Website before you can create the theme.
Website for Tenant Management	Select this check-box to enable the browse option. For both Desktop and Mobile custom theme, you may choose the theme directory from the Website Directory pop-up. The default directory named TenantManagementWebsite, already exists. Note The directory must exist under RequestCenter.war\Website before you can create the theme.
Website for Cloud Integrations	Select this check-box to enable the browse option. For both Desktop and Mobile custom theme, you may choose the theme directory from the Website Directory pop-up. The default directory named CloudIntegrationsWebsite, already exists. Note The directory must exist under RequestCenter.war\Website before you can create the theme.
Website for User Management	Select this check-box to enable the browse option. For both Desktop and Mobile custom theme, you may choose the theme directory from the Website Directory pop-up. The default directory named UserManagementWebsite, already exists. Note The directory must exist under RequestCenter.war\Website before you can create the theme.

Customizing Customer Facing Modules

Website model provides enhanced capabilities to customize users facing modules of Prime Service Catalog such as, Service Catalog, Tenant Management, Cloud Integrations, and User Management. By default the GUI for these modules is launched from the website model. Using website model you now have complete control to redesign the UI by applying customization (except for the Service Form), such as, adding another bootstrap file, and adding routing html and js files. Website model is also supported for mobile UI, similar to web UI you can reference the customization files for mobile UI for each module.

To customize website model, perform the following:

Before You Begin

- 1 Add all the customized files in a directory under *RequestCenter:war\Website*. By default, the directories *UserManagementWebsite*, *ServiceCatalogWebsite*, *TenantManagementWebsite*, and *CloudIntegrationsWebsite* are available, which contains the out-of-box files.

- 2 Entry points for these modules are:

Module	Entry Point
Tenant Management	index.html
Service Catalog	index.html
User Management	usermanagement.html
Cloud Integrations	<i>dashboard.html</i> and <i>admin.html</i>

Example: <http://<ServerURL>/RequestCenter/website/ServiceCatalogWebsite/application/index.html>

- 3 In the routing html files,

- Use the following script to load all css dependancies by importing headcss.html into all routing html files:

```
<script type="text/javascript" src="common/js/vendor/jquery/1.9.1/jquery.min.js"></script>
<script>
    $(function(){ $("head").load("headcss.html") });
</script>
```

- Use the following scripts to get the dependent details like localized strings and user dependent details.

- For Service Catalog module use the script:

```
<script type="text/javascript"
src="/RequestCenter/servicecatalog/api/v11/strings.action"></script>
<script type="text/javascript"
src="/RequestCenter/servicecatalog/api/v11/uservars.action"></script>
```

- For Tenant Management module use the script:

```
<script type="text/javascript"
src="/RequestCenter/tenantmanagement/api/v11/strings-tenant.action"></script>
<script type="text/javascript"
src="/RequestCenter/tenantmanagement/api/v11/uservars-tenant.action"></script>
```

- For Cloud Integration module use the script:

```
<script type="text/javascript"
src="/RequestCenter/clouddashboard/api/v11/strings.action"></script>
<script type="text/javascript"
src="/RequestCenter/clouddashboard/api/v11/uservars.action"></script>
```

- For User Management module use the script:

```
<script type="text/javascript"
src="/RequestCenter/usermanagement/api/v11/strings.action"></script>
<script type="text/javascript"
src="/RequestCenter/usermanagement/api/v11/uservars.action"></script>
```

- Use the following script to load the requirejs:

```
<script type="text/javascript"
src="common/js/vendor/requirejs/2.1.6/require.js"></script>
```

- 4 Add or modify the vendor file details in *common-config.js* file.

**Note**

You must place *js/configs/common-config.js* file on the website directory only.

-
- Step 1** Log in to Service Catalog, choose the **Administration** module, and go to the **Settings** tab. The Customizations page appears.
- Step 2** At the right side of the screen, choose the **Custom Themes** option from the option list.
- Step 3** Click **Add** to create a new theme. The Custom Theme Properties page appears.
- Step 4** Enter the **Name** and **Description** (optional).
- Step 5** Check the **Website for Service Catalog** option to customize Service Catalog module, and browse to choose the Website Directory and click **OK**.
You can reference the customization files for Desktop or Mobile alone, or for both at the same time.
- Remember** If you uncheck the Website for Service Catalog checkbox, the Browse... option will be grayed out. When you check the Website for Service Catalog again, then the previously chosen website directory is selected by default.
- Step 6** Repeat **Step 6** to customize required modules such as Tenant Management, Cloud Integration, and User Management.
- Step 7** Click **Create** to create the theme. You can then edit the theme, to specify the organizations to which the theme applies.
- Step 8** Click **Add** in the Associated Organizational Units pane. The Organizational Unit Search window appears. Choose one or more organizational units.
- Step 9** You may edit the theme definition or the business units to which it applies at any time.
-

Enabling Custom Style Sheets and Headers/Footers

The appearance of customer-facing modules in the application can be customized using Cascading Style Sheets (CSS) and custom headers and footers.

Choose the Administration module, and go to the Settings tab. The Customizations page appears. The Common settings include parameters to “Enable Custom Header Footer” and “Enable Custom Style Sheets”.

The custom Style sheets are enabled by default. You can change the corresponding parameter setting from “On” to “Off.” Save your changes by updating the page. Any specified in the custom.css file (in place on the application server) will be in effect.

To enable custom headers and footers, change the parameter setting for the “Enable Custom Header Footer” parameter to “On.”

Once you start a session with these parameters turned on, there is no need to exit from your session to view Style changes. Once the definition of the Style is changed and the file placed on the specified directory of the application server, refreshing the page will use the new Style definitions. There is a directory under RequestCenter.war/custom named ServiceCatalogExamples which provides the base application CSS files that you can customize for your brand as well as the GUI design.

Modifying Customizations with Browser Cache Enabled

If the Browser Cache setting is enabled in the Administration Settings, changes made to custom style sheets, headers and footers will not take effect until the browser cache has been deleted. To prompt the application users to delete their browser cache, follow the instructions in the [Browser Cache Setting](#) to increment the browser cache version.

Customizing User-Defined Portals

Just like Service Catalog, My Services and other built-in modules, the Service Portal modules can be customized for different organizational units through style sheets. The custom styles are maintained in a different style sheet from the other built-in modules to give you greater flexibility in how you present the Portal Designer modules.

The Service Portal solution also offers different themes that affect the colors of portlets on a portal page. You can allow users to choose their own themes, or give this ability only to portal designers. See the [Cisco Prime Service Catalog Designer Guide](#) for more information regarding portal page themes.

The custom stylesheet for Service Portal module is located in the same custom/ServiceCatalogExamples or custom/CustomExamples directory as Service Catalog and is enabled/disabled along with it.

- 1 Obtain a copy of the file `example_portal-custom-header.css` from the custom directory of the Service Catalog web archive (RequestCenter.war).
- 2 Name the copied file as `portal-custom-header.css`. Change the styles in the file according to the guidelines given in the next section of this chapter.
- 3 Copy this file into the custom directory created for the tenant/organization (see [Customizing Built-In Modules](#), on page 2, Step 1) along with any images used.

Example

The following example describes how to modify the logo (92X33 pixels), product name and background color in the portal header:

```
.reboot2 .xwtBackgroundSimplified {
background: transparent !important;
}
/*style to modify logo image*/
.reboot2 .applicationHeader17 .applicationLogoImage {
background: url("images/SmallCo_logo92x33.gif") no-repeat transparent !important;
/*background-repeat: no-repeat;
background-position: 0 .514em;
background-size: 75px auto;
padding-left: 3em !important;
padding-right: 5em !important;*/
height: 33px !important;
width: 92px !important;
/* Note: un-comment display property to hide the Product Logo if needed */
/*display: none !important;*/
}
/* This style is used to display or hide the Product Title in Portal modules
See Also: .applicationHeaderAppSubTitle, .applicationLogoImage, .applicationHeaderLogoText
style - which will be the new style used to display branding logo
*/
.reboot2 .applicationHeader17 .applicationHeaderAppName {
visibility: hidden;
}
```

The following example describes how to modify the logo (135x70 pixels), product name and background color in the portal header:

```
.reboot2 .xwtBackgroundSimplified {
background: transparent !important;
}
.reboot2 .applicationHeader17 {
padding-left: 5px !important;
}
/*style to modify logo image*/
.reboot2 .applicationHeader17 .applicationLogoImage {
background: url("/RequestCenter/custom/style_dir/images/custom_logo.png") no-repeat scroll
-2px -8px transparent !important;
background-size: auto auto !important;
height: 73px !important;
line-height: 70px !important;
top: 1px !important;
width: 130px !important;
}
/*This style will reduce the left, right and top padding for logo container*/
.reboot2 .applicationHeader17 .applicationHeaderLogo {
margin-left: 2px !important;
margin-right: 2px !important;
padding-top: 5px !important;
}
.reboot2 .applicationHeader17 .applicationHeaderAppSubTitle {
color: #202020;
cursor: default;
font-size: 17px !important;
font-weight: bold !important;
font-family: CiscoSans,Arial,"Helvetica Neue",Helvetica,sans-serif !important;
text-shadow: 0 0.071em 0 #585858 !important;
}
```

The following example describes how to modify the product name and customize the PortalFullpagePrimeUi.js file. On Linux you can locate the file from the path
`/opt/CiscoPrimeServiceCatalog/jboss-as-7.1.1.Final/ServiceCatalogServer/deployments/RequestCenter.war/ns360.js/PortalFullpagePrimeUi.js`.
 You can search for the function name "getNavItems" and change the name from 'Service Catalog' to 'Cloud Manager':

```
if(defaultSelectedId != ""){ /*existing code*/
navItems.items.defaultSelected = defaultSelectedId;
} /*existing code*/
navItems.items.toolbar.push(getToolbar()); /*existing code*/
/*Overwrite the app header text with custom application name "Cloud Manager"
If text need to add for AppName div then from custom css file its "visibility"
attribute should be "block !important"*/
var appNameDiv = dojo.query('.applicationHeaderAppName')[0];
appNameDiv.style.display = "block !important";
appNameDiv.innerHTML = "Cloud Manager";
var appSubTitleDiv = dojo.query('.applicationHeaderAppSubTitle')[0];
appSubTitleDiv.style.display = "block !important";
appSubTitleDiv.innerHTML = "Cloud Manager";
return navItems; /*existing code*/
```

Customizing Styles for MyServices Module

The custom\CustomExamples directory includes files you can use as starting points for customizations of My Services module. Directory contents are summarized in the table below.

Table 2: Custom styles

custom (folder) Contents	Description
CustomExamples	Folder which contains starting points for custom styles, header, and footer
images	Folder which contains the images currently used by Service Catalog styles which may be replaced via custom style sheets
common_task_bg.gif	Background for the Common Tasks pane
headerGradient.gif	Background for header styles—style which appear at the top of each portlet or pane
logo_bottom.gif	(Deprecated)
lvl1_nav_shade.gif	Background for the tabs which provide top-level navigation through the options available in each Service Catalog module
lvl3_nav_shade.gif	Background for level 3 headers—also recommended for page footers
mark.gif	Denotes a mandatory field on a service form
orange_bullet.gif	Common Tasks bullet
orange_li_bullet.gif	(Not used in example custom.css)
page_footer_shade.gif	Gif available for page footer shading
PopupHeaderGradient.gif	(Not used in example custom.css)
requiredMark.gif	Denotes a mandatory field on all user interface pages other than service forms
tfoot_shade.gif	(Not used in example custom.css)
example.css	Sample file, mirrors default Service Catalog settings to start with, with solid color replacing gradients
example_portal-custom-header.css	Sample file, mirrors default Portal Designer header area settings
example_footer.html	Starting point for developing custom footer
example_header.html	Starting point for developing custom header

The CustomExamples/example.css and example_portal-custom-header.css files (the templates for your custom.css and portal-custom-header.css) are formatted as a standard cascading style sheet file and includes comments to guide you in choosing styles to modify. These comments include brief descriptions of where and how a particular style is used; however, some experimentation is required to fine tune customizations.

The original definitions for customized styles should be retained as comments in the style sheet. This practice is recommended, in case a customized change needs to be backed out and to maintain traceability to the original page appearance.

Page Headers

The page header for the end-user facing modules is governed by the following styles:

- `lvl1_nav` (for built-in modules only): The “Level1 Navigation Bar” provides the background for the application module drop-down menu and menu bar. The application name cannot be modified but can be hidden using the “`lvl1_nav_title`” style if desired.
- `header` (for user-defined modules only): The header style is used with portal modules that are created/maintained using Portal Designer. The usage is similar to the `lvl1_nav` above.
- `headerlogo`, `leftheaderlogo`: The two header logo styles provide flexibility in placing the logo at either the left or right side of the header. When the left logo is used, the background property of the application name must be set to none. The styles governing the application module menu may also be modified so that it can be positioned at the right corner.

Navigation Bars

By default, most of the navigation bars simply specify a background color. However, as with any other background designated in the style sheet, this can be changed to use a banner or graphic.

Other portions of My Services pages use decorations as headers and footers for portions of the page. For example, the “Level3 Navigation” (`lvl3_nav`) and footer styles delimit the page body of the My Services home page, as shown in the illustration above. They should be changed together.

The “Breadcrumb Navigation” (`bread_nav`) provides the background for the breadcrumb area.

Buttons

Buttons appear on service forms and through the Service Catalog user interface. The appearance of buttons is governed via the style `button.primary`. The default style for primary buttons is set to use bold face and can be modified to have more prominent styles if necessary.

Figure 1: Buttons on Service Form

Basic Service with Essential Tasks



A service with a bunch of tasks of different varieties

Add & Review Order

Submit

1	Primary buttons
---	-----------------

Service Forms

The appearance of the fields and captions on service forms is governed by a set of styles, as shown below and summarized in the following table. All .form styles should be changed in unison.

Table 3: Service forms fields

Style	Usage
shortHeader	The dictionary caption
subhead	The bar delineating the start of each dictionary displayed with a caption
.formReq_border	Blank space to the left of the field label, and the line separating one field from the next
.formLabel	The field label
.formElement	Formatting for the input element for the field's HTML representation
.formInfo	Blank space to the right of the form element, and the line separating one field from the next
.formIcon	Grey bar on the right of the form

Figure 2: Service forms

1	shortHeader
2	subhead

**Note**

You can optionally enable the legacy display style for non-grid Dictionary field help texts by setting the `Serviceformelement.display.instructional.helptext` property in the `newscale.properties` file to `true`.

**Note**

You can optionally enable the designers access to `Message_INFO` ID of Service Form input fields through java script, by setting the `serviceformelement.display.instructional.helptext.Message_INFO` property in `newscale.properties` file to `true`.

You must set the `Serviceformelement.display.instructional.helptext` property in the `newscale.properties` file to `true` to enable the access to `Message_INFO` ID.

Preserving Customizations

The custom style sheet file, as well as html files for defining custom page headers and footers, must be part of the application on the application server. Therefore, a mechanism is required for preserving these customizations in that event that an application instance must be upgraded or migrated.

To preserve the customizations when you upgrade or migrate the application:

-
- Step 1** Create an archive file in the Zip format containing the files you have customized. The archive directory structure must match the deployment directory structure. The root directory of the archive file should be the `RequestCenter.war` directory.
 - Step 2** Perform an upgrade of the Service Catalog application.
To avoid losing the customizations, the Service Catalog installation wizard allows you to specify custom content to be included in the installation:
 - Step 3** Run the Service Catalog installation wizard as described in the [Cisco Prime Service Catalog Installation Guide](#), using the **Advanced Installation** type.
 - Step 4** On the Application Server Configuration page, click **Advanced Options**.
 - Step 5** The Advanced Options dialog box appears, as shown below.
 - Step 6** Check **Custom content**.
 - Step 7** Enter the full path to the **Custom content archive** including the name of the archive, or click **Browse** to locate and choose the custom content archive.
 - Step 8** Click **Close**.
 - Step 9** Continue with the installation as described in the [Cisco Prime Service Catalog Installation Guide](#).
 - Step 10** While the Service Catalog installation wizard completes the installation, it extracts your custom content archive into the application deployment directory structure.
-

Known Errors and Omissions

The online help files provided for Service Catalog cannot be customized.

Unknown Errors and Omissions

It is possible that some styles used in Service Catalog pages are not included in the CustomExamples/example.css file. If you find such an omission, please report it to the Cisco Technical Assistance Center (TAC).

A temporary workaround may be possible. View the source for the generated page, noting the class or id of the sections to which the style is to be applied. If this class or id uniquely identified the object whose appearance you want to change, include an appropriate style in your custom stylesheet, or add an appropriate attribute to the style definition. Care should be taken if you elect to use this approach, as any additions to the custom stylesheet may not be supported in subsequent releases.

Upgrading from Previous Versions

The styles used in this version of Service Catalog may have been modified from those used in previous versions. These changes not only update the appearance of the pages but also address performance and consistency issues that had been raised in previous releases.

Style Summary and Recommended Practices

This section describes the various style summary and the recommended practices.

Style Summary – Built-In Modules

The table below summarizes styles available in the custom.css.

Table 4: Available styles

Style/Class Name	Comment
Body and Global Styles	
body	
#lvl3_nav	
#headerlogo	Logo on the right
#leftheaderlogo	Logo on the left
#lvl1_nav_title	Application name
#footer	
.levelTwoNavigation	Tab selection
table#nsLayout.rightMenu td#layoutright	

Style/Class Name	Comment
Navigation Styles	
#lvl1_nav	
#lvl1_nav span#lvl1_nav a	
#lvl1_nav a:hover	
.menuDivider	
#lvl2_nav	
#lvl2_nav a	
#lvl2_nav td.active	
h2#title_nav	
#bread_nav	
#bread_nav a	
#logobottom	Deprecated (dummy image used)
Tab Navigation Control Styles	
.levelTwoNavigation a.tabNavigation a	
.levelTwoNavigation a:hover.tabNavigation a:hover	
.levelTwoNavigation a.selected.tabNavigation a.selected	
.levelTwoNavigation a.selected:hover.tabNavigation a.selected:hover	
.propertyTabNavigation a	
#levelTwoTabDiv img	Left and right-edge images on tab button
.levelTwoNavigation div.levelTwoTab	
.levelTwoNavigation a	
My Services Service Items Tab Styles	
.x-grid3-row	Background-color for grid row

Style/Class Name	Comment
.x-grid3-row TD	Font for grid row
.x-grid3-row-alt	Background color for alternate row
.x-grid3-hd-row td	Font for grid header
ul.x-tab-strip-top	Background color for tabs
.x-tree-node A SPAN	Font for tree
Header and Title Styles	
div.longHeaderdiv.shortHeader	
div.longHeader h4div.shortHeader h4div.longHeader spandiv.shortHeader span	
div.subHeader	
h4.header	
h4.header span	
Button Styles	
buttoninput.primaryinput.secondaryinput.disabled	
button.primaryinput.primary	
button.secondarybutton.helpinput.secondary	
button.disabledinput.disabled	
Catalog and Service Display Styles	
table.browser	
table.browser td.categoryImage	Pixel sizes for the Service Catalog and category images
table.browser td.categoryText	
table.browser td.categoryText	
div.smallshell	
div.service	

Style/Class Name	Comment
table#columns select	
Data Table Styles	
table. halfGrid,fullGrid,footGrid,taskGrid,noGrid	
table. halfGrid,taskGrid,noGrid	
table. footGrid,noGrid	
table.dProcess	
table. halfGrid,fullGrid,footGrid, taskGrid,noGrid,dProcess thead th.first	
table. halfGrid,fullGrid,footGrid, taskGrid,noGrid,dProcess thead th.firstSel	
table. halfGrid,taskGrid,fullGrid tbody td tbody th	
table. footGrid,fullGrid tbody td tbody th	
table.kpi	
table.fullGrid tbody.subHeader td	
table. halfGrid,fullGrid,footGrid, taskGrid,noGrid,smGrid tbody tr.shade td tbody tr.shade th taskGrid tbody tr.current th tbody tr.current td	
table. halfGrid,footGrid,taskGrid,noGrid tbody tr: hover th td	
table. halfGrid,fullGrid,footGrid, taskGrid,noGrid,dProcess,smGrid thead tbody.calendar table#calendar th	
table. halfGrid,fullGrid,footGrid,noGrid tfoot th tfoot td	
table. halfGrid,fullGrid,footGrid,noGrid tbody tr td.select tbody tr th.select	
table. halfGrid,fullGrid,footGrid,noGrid tbody tr td.select + td td.select + th th.select + td th.select + th	

Style/Class Name	Comment
table.fullGrid tbody td.kvpKeyHi:hovertable.fullGrid tbody.th.kvpKeyHi:hovertable.fullGrid tbody td.kvpKeyHi: hover + td.kvpValuetable.fullGrid tbody.th.kvpKeyHi: hover + td.kvpValue	
table.fullGrid tbody td.kvpKeyHitable.fullGrid tbody.th.kvpKeyHi	
table.fullGrid tbody td.kvpValueHi	
Other Tables and Table-Related Content Styles	
div.theaddiv tfoot	
div.tsubfoot	
div.detailHeader	
div.detailHeader td.owi	
div.tfoot input.textBoxdiv.tsubfoot input.textBox	
Comments and History Styles	
table.commentstable	
table.commentsTable div.commentContainer	
Common Tasks and Portlet Styles	
div.shell	
.commonTaskCellul.tasks.li	
ul.tasks	
img.commontaskbullet	
Normal Portlet, Form and Container Styles	
div.servicediv.shelldiv.smallshelldiv smalldiv.commonTaskCellul.tasks.li	
Input Elements	
inputselect	Service form text elements
input.textBoxtextarea	Service form textarea elements

Style/Class Name	Comment
Content-Switching Styles	
ul.MDITabs li:hover	
ul.MDITabs li.active	
Service Form Styles	
tr.error td.formReq formLabel formElement formFlex formInfo	Components of the fields defined in dictionaries used in service forms—the required symbol, the field label, the input element
tr.error td.formIcon	
.formReq_border	
.formLabel	
.formElement	
.formFlex	
.formInfo	
.formIcon	
div#formMonitor div a	
div#formMonitor div.valid	
div#formMonitor div.invalid	
Calendar Styles	
body.calendar table#calendar td	
body.calendar table#calendar td.selected	
Service Manager Styles	
table#SMLayout	
table#SMLayout td#SMTreeFrame	
div.SMToolbar	
table.smGrid tbody tr.highlight td, th	

Style/Class Name	Comment
div.treeHeader treeNode treeItem treeNode span.unselected	
div.treeNode span.selected	
div.treeHeader span	
table.smGrid thead th	
table.smGrid tbody td, th	
table.smGrid tbody tr.shade td, th	
Module Menus	
.modulemenu	
.modulemenu .menuheadingrow	
menuHighlight	

Style Summary – User-Defined Modules

The table below summarizes styles available in the portal-custom-header.css.

Table 5: User defined styles

Style/Class Name	Comment
Header Styles	
#headerlogo	Logo on the right
#leftheadlogo	Logo on the left
#lv11_nav_title	Application name
#header	
#usercontrols	
#cornerpiece	
#moduleMenuDiv	
.modulemenu	

Style/Class Name	Comment
.menuHighlight	
#userinfoandcontrols	
#userinfoforow	
#usercontrolstable	
#userconrolsrow	
#profilef	
#logoutref	
#helpref	

Recommended Practices

After cloning from the example.css to create the initial custom.css, the styles should have no effect on the user interface, and as the individual properties are changed, they should then be evident in the customizable modules.

Some styles used by the default user interface are implemented as background images, rather than color values. Some of these images are duplicated in the custom/CustomExamples/images subdirectory, ready for replacement. They should be replaced with images of the same type, size, shape and name in order to be correctly included in the user interface.

There are a number of places in the custom stylesheet where there is an alternative between using a background image and simply specifying a color value. In each of these places, there are alternate attributes that can be commented in or out to determine which of these is to be used. For example:

```
div.longHeader,
div.shortHeader
{ /* background: #FD2312; */
  background: url(./images/headerGradient.gif);
  border-bottom: 2px solid #cc3333;
}
```

Here, the image providing a shaded header for the portlets is being used. To change that gradient, replace this image in the custom subdirectory. To switch to a simple solid background color, comment out the background that specifies the image using the /* */ pattern, and remove the comment from the background with the hex color.

You can also create new images and modify the custom.css file to point to them. For example:

```
#header
{
  background: #ffffff url(./images/logo.gif) top left no-repeat;
  border-bottom: 1px solid #a7a7a7;
}
```

In this case, a new “logo.gif” could be created and the file replaced, or a completely new image generated such as “acme_logo.gif”. Then, the property declaration could be changed to read

```
background: #ffffff url(./images/acelogo.gif) top left no-repeat;
```

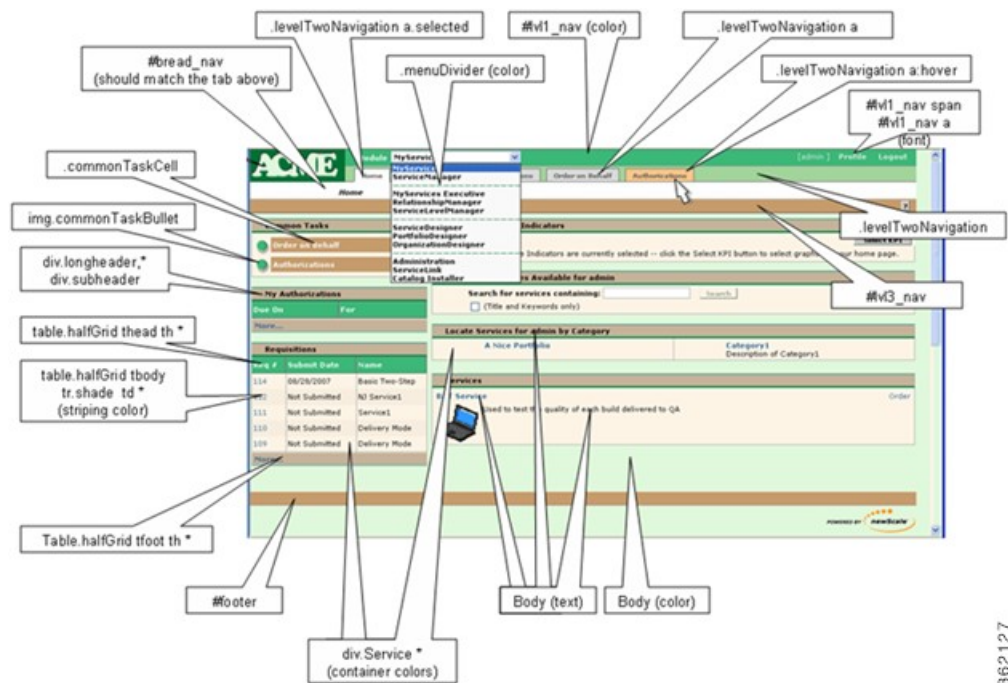
The same goes for any other image used in the look and feel.

For modules defined and maintained in Portal Designer, the portal page body is not affected by the custom stylesheets. Instead the portal page theme can be configured using “Page Settings” to match with the header styles.

Example Screenshot and What Each Style Specifically Affects

This diagram is representative of the effect that the styles have. Although it does not include all styles, it does cover the most commonly customized ones.

Figure 3: Sample Screen



* Indicates that this is the first descriptor in the style grouping, and there are others.

Custom Headers and Footers

This section describes the updates to custom headers and footers.

Overview

Cisco supplies a template for customizing the page headers and footers that appear with Service Catalog web pages.

Procedure

Additional details on these styles are given in the section [Customizing Page Headers and Footers](#), on page 22. To add a custom header or footer to the Service Catalog application:

-
- | | |
|---------------|---|
| Step 1 | Obtain a copy of the files <code>example_header.html</code> and <code>example_footer.html</code> from the custom directory of the Service Catalog web archive (<code>RequestCenter.war</code>). |
| Step 2 | Name the copied files as <code>header.html</code> and <code>footer.html</code> , respectively. |
| Step 3 | Add content to the header or footer files according to the guidelines given in the next section of this chapter. |
| Step 4 | Place your custom header and footer files on the specified directory for the styles to which they apply. Both files must be present. If you are not using a custom header or footer, copy an empty file with the appropriate name and the <code>.html</code> extension to the application server. |
| Step 5 | Use the Site Administration > Site Configuration page in Administration to turn on custom headers and footers, by setting the “Enable Custom Header Footer” site configuration parameter to On . |
| Step 6 | Restart the browser session of Service Catalog—the pages should appear with your customizations. |
-

Customizing Page Headers and Footers

Custom page headers and footers appear in addition to, not instead of, the standard page headers and footers. The header and footer html files may contain any html commands deemed appropriate, including use of default Service Catalog styles.

For example, using a `footer.html` file with the following contents:

```

```

would result in a footer display like that shown below, where the “Technology by Cisco” logo is the standard page footer in My Services.

Figure 4: Footer logo

