

HP Service Request Catalog

For the Windows[®] and Unix operating systems

Software Version: 9.32

Customization Guide

Document Release Date: August 2013

Software Release Date: August 2013



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 2010 - 2013 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

<http://www.hp.com/go/hpsoftwaresupport>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

| | |
|--|----|
| Contents | 5 |
| Overview | 9 |
| Configuration Files | 9 |
| Working With the Manifest File | 9 |
| Using the Manifest.xml validator tool | 10 |
| Using the Upgrade Tool | 11 |
| What Does the Upgrade Tool Do? | 11 |
| Upgrade Tool Usage | 11 |
| Service Request Catalog Customization | 13 |
| Customizing the Interface | 13 |
| Embed SRC in an HTML Frame | 13 |
| Customize the Landing Pages | 14 |
| Use the Launcher to Create Third Party Links | 15 |
| Customized Error Messages | 19 |
| Hide the Service or Support Tabs | 19 |
| Configure the Your Requests widget and Your Service Request pane | 20 |
| Styling the Interface | 20 |
| Customize the Application Name | 20 |
| Customize Colors and Graphics | 21 |
| Modifying panel.swf | 22 |
| Modifying the favorite icon | 23 |
| Customize the Text Format | 23 |
| Text Manipulation Overview | 24 |
| Converting styles.css to styles.swf | 24 |
| Add a New Language | 24 |
| Configuration Steps | 25 |
| Service Manager Configuration | 27 |
| Tailoring Best Practices | 27 |
| Tailoring Check-Out Panels | 27 |
| Checkout Panels | 28 |

| | |
|--|----|
| Configurations | 28 |
| Sections | 28 |
| Custom Fields | 29 |
| Add a Configuration | 31 |
| Edit an Existing Configuration | 33 |
| Localize a configuration | 34 |
| Add a Section to an Existing Configuration | 34 |
| Modify an Existing Section | 35 |
| Delete a Configuration | 35 |
| Multi-Company Mode | 36 |
| Example: Multi-Company Mode | 36 |
| Miscellaneous Tasks | 37 |
| Enable Knowledge Management | 37 |
| Subscriptions | 38 |
| Enable Approval Delegation | 38 |
| Improve SRC Performance | 41 |
| Additional Performance Tips | 47 |
| Look-up Fields | 47 |
| Date Time Validation | 48 |
| Update Service Manager Lists | 48 |
| Supporting Other Locales | 49 |
| Authoring Bi-directional Text | 54 |
| Understanding Unexpected Text Order | 55 |
| Input Bidirectional Text in the Service Manager Client | 56 |
| Configure Additional Details for Service Requests | 57 |
| Enable Journal Updates | 58 |
| Require Users to Read Catalog Item Details. | 58 |
| Enable Line Item Status | 58 |
| Configure IPv6 | 59 |
| Enable FIPS for Service Request Catalog | 59 |
| Using the FIPS standard in SRC | 59 |

| | |
|--|-----------|
| Common Access Card Introduction | 60 |
| Enable SRC for Common Access Cards | 60 |
| Auto-Suggest/Type Ahead | 63 |
| Appendix A: Launcher Examples | 64 |
| Example: Add Bomgar Live Chat | 64 |
| Example: Cloud Service Automation Launcher | 65 |
| Example: Get | 67 |
| Appendix B: Service Catalog in Service Manager | 68 |
| Dynamic User Options in Catalog Items | 68 |
| Create a Dynamic User Option | 68 |
| Notes on Expressions in Dynamic Fields | 69 |
| Service Request Catalog HTML Tag Support | 73 |
| Refining the Service Catalog | 76 |
| Remove Empty Categories | 76 |
| Add Images for Catalog Items | 76 |
| Separate Catalog Items for Different Companies in Multi-Company Mode | 76 |
| Working with the Support Catalog | 77 |
| Search for a Support Catalog Category or Item | 78 |
| Create a Support Catalog Category | 78 |
| Create a Support Catalog Item | 79 |
| Delete a Support Item | 81 |
| Appendix C: XML Tag Reference | 82 |
| Appendix D: Interface References | 89 |
| Client Customization Reference | 89 |
| Customize the SRC Header | 91 |
| Customize the SRC Footer | 92 |
| Customize the Dashboard | 93 |
| Customize the Services Page | 95 |
| Customize the Support Page | 96 |
| Customize the Account View Page | 97 |
| Customize the Details View Page | 99 |

| | |
|--|-----|
| Customize the Approval History | 100 |
| Customize the Request Inbox Welcome Page | 101 |
| Customize the Browse Catalog Button | 102 |
| Customize Subscriptions | 103 |
| Customize the Approvals Page | 105 |
| Customize the Service Approval Details Page | 107 |
| Customize the Change Approval Details Page | 109 |
| Customize the Quote Approval Details Page | 111 |
| Customize the Search Results Page | 113 |
| Customize the Request on Behalf Of | 114 |
| Customize the Shopping Cart | 115 |
| Customize the SRC Settings | 117 |
| Customize the Approval Delegation Details Page | 118 |
| Customize the Recurring Cost Labels | 120 |
| Customize the Login Panel | 122 |
| Customize the User Account Settings | 123 |
| Customize the Request Summary Panel | 124 |
| Customize the Submission Panel | 125 |
| Server Customization Reference | 125 |
| Customize the Search Results | 126 |
| Customize the Order Information | 129 |
| Customize the Subscriptions Page | 130 |
| Customize the Knowledge Management Results | 132 |
| Customize: "Your Approvals" | 134 |
| Customize the Request For Panel | 135 |
| SRC Service Manager Customization Reference | 136 |
| The Style.css File | 138 |

Overview

There are many aspects of Service Request Catalog which are customizable to your organization. The contents of this Customization Guide are to help you reference the information you are looking for quickly. It is not intended as a step-by-step guide. Instead, each section provides loosely coupled concepts to assist you in achieving an immediate goal, such as enabling knowledge management in SRC, configuring approval rights for a user, or customizing the user interface.

Configuration Files

There are configuration files that enable you to take advantage of the customization features. From the installation directory structure on the server where you installed Service Request Catalog, the following files are important when you design your customization. You will define a company folder to contain some of these resource files.

Configurable Files

| File Path and Name |
|--|
| <p>...\src\secure\configuration\manifest.xml</p> <p>Identifies customization resources and settings.</p> |
| <p>...\src\resources\client\en\Custom.properties</p> <p>Localizable property file that contains your customization information.</p> |
| <p>...\src\WEB-INF\classes\ApplicationContext.properties</p> |
| <p>...\src\WEB-INF\classes\cacConfiguration.properties</p> |
| <p>...\src\secure\configuration\default\panel.swf</p> |
| <p>...\src\secure\configuration\default\styles.swf</p> |
| <p>Note: <code>styles.swf</code> must be generated from the <code>styles.css</code> file. For instructions on how to do this, see "Customize the Text Format" on page 23.</p> |

Working With the Manifest File

This file contains information about the presentation of the Dashboard and other landing pages for your company. This file also points to other resources that contribute to the look and feel of your user interface, such as your company name, color palette, and other presentation styles.

The `manifest.xml` file has basic sections for capturing this information.

```
<configuration name="">
  <companyDisplayNameKey>
  <resources>
  <styles>
```

```
<launchersGroup>  
  <widgetsGroup>  
</configuration>
```

See the "[Appendix C: XML Tag Reference](#)" on page 82 for more information about configuring the individual tags in each section.

Using the Manifest.xml validator tool

SRC comes with a prepackaged validation tool that you can use to help ensure that your manifest.xml file is valid and usable. The validator tool can be found in the SRC release .zip package. The vtool consists of the following two script files for Windows and Linux operating systems:

- validator.bat
- validator.sh

For example, you would enter the following command on a Windows system to use the validation tool:

```
validate.bat -c "<path>"
```

Note: The *<path>* placeholder tag represents the file system path where you have the manifest.xml file.

After running this command, you will receive the following message if your manifest.xml file is valid:

```
"manifest.xml is valid!"
```

If your manifest file is not valid, you will receive an error message. The following table demonstrates some of the error messages you may receive:

| ERROR: Cannot read manifest File: <path> | Description |
|--|---|
| org.xml.sax.SAXParseException: cvc-datatype-valid.1.2.1: 'abc' is not a valid value for 'integer'. | A string value is present in some element that requires an integer value. |
| org.xml.sax.SAXParseException: cvc-complex-type.2.4.b: The content of element 'configurations' is not complete. One of '{configuration}' is expected. | The <configurations> parent node must contain at least one <configuration> child node. |
| org.xml.sax.SAXParseException: cvc-complex-type.2.4.a: Invalid content was found starting with element 'resources'. One of '{styles, widgetsGroup, launchersGroup}' is expected. | The <resources> element can contain only one of each of the following: <styles>, <widgetsGroup>, <launchersGroup> |

| ERROR: Cannot read manifest File: <path> | Description |
|--|---|
| org.xml.sax.SAXParseException: cvc-complex-type.4: Attribute 'displayOn' must appear on element 'launchers'. | The displayOn attribute is not present in the <launchers> element. |
| org.xml.sax.SAXParseException: cvc-enumeration-valid: Value 'CHOICE' is not facet-valid with respect to enumeration '[TEXT, TEXTAREA, DROPDOWN, HIDDEN]'. It must be a value from the enumeration. | Type of fields in the <launcher> must be a value from the enumeration [TEXT, TEXTAREA, DROPDOWN, HIDDEN]. |

Using the Upgrade Tool

The Service Request Catalog Migration Tool is a dedicated tool to preserve your SRC customizations as you migrate to Service Manager.

What Does the Upgrade Tool Do?

Some files in SRC are open for customization. Without the Migration Tool, you must migrate any customizations manually to complete the migration process. Doing this manually exposes the process to potential mistakes. The Migration Tool copies the customized files to the new SRC release directly and merges certain customized files into the SRC 9.32 release.

Upgrade Tool Usage

\$TOOL_HOME refers to the folder to which you extracted this tool.

\$SOURCE_SRC refers to the folder in which the older version of SRC is deployed and which contains the customizations you wish to migrate.

\$TARGET_SRC refers to the folder in which you are deploying the new version of SRC.

Linux Usage:

1. Grant execution permission for migrateSRC.sh. To do this, navigate to \$TOOL_HOME and then run the following command:

```
chmod u+x bin/migrateSRC.sh
```

2. Launch the Migration Tool by running the following command:

```
bin/migrateSRC.sh $SOURCE_SRC $TARGET_SRC
```

3. Follow the onscreen instructions to migrate the specific migrations you wish to move to the new version of SRC.

Windows Usage:

1. Double-click the `migrateSRC.bat` file in the `$TOOL_HOME\bin\` directory.
2. Follow the onscreen instructions to migrate the specific migrations you wish to move to the new version of SRC.

Note: Currently, you can only use this tool to migrate from SRC 1.40 to SRC 9.32.

Service Request Catalog Customization

You can make many changes to brand Service Request Catalog with your own corporate identity. Read these sections to learn more about each task:

Interface

["Embed SRC in an HTML Frame" below](#)

["Customize the Landing Pages" on the next page](#)

["Use the Launcher to Create Third Party Links" on page 15](#)

["Customized Error Messages" on page 19](#)

Styling

["Customize the Application Name" on page 20](#)

["Customize the Text Format" on page 23](#)

["Customize Colors and Graphics" on page 21](#)

Customizing the Interface

Service Request Catalog allows you to customize the interface through a variety of mechanisms. You can show or hide any of the five primary widgets, or use the launcher to add links to third party Web sites, applications, videos, or anything else you prefer.

Embed SRC in an HTML Frame

If you wish to embed Service Request Catalog within your corporate intranet, you can embed SRC within standard HTML Web pages by using the FRAME html tag. To do this, create a Web page that contains a FRAME element, and specify the source property of the FRAME tag as the url of your SRC installation.

```
<HTML>
<HEAD>
<TITLE>Service Request Catalog</TITLE>
</HEAD>
<FRAMESET rows="25,*" cols="*">
  <FRAME name="Top" src="corporateHeader.html" FRAMEborder=0 noresize scrolling=no
  o FRAMEspacing=0 border=0>
  <FRAME name="Content" src="http://<domainname>:<port>/<srcDeploymentName>/secur
  e/main.jsp" FRAMEborder=0 noresize scrolling=no FRAMEspacing=0 border=0>
</FRAMESET><noframes></noframes>
</HTML>
```

In this example, the `<domainname>` placeholder represents the domain on which you installed SRC, the `<port>` placeholder represents the port number that was specified during installation of SRC, and the `<srcDeploymentName>` represents the name you used to deploy SRC. Therefore, a complete example for a deployed installation of SRC may resemble the following:

```
http://myCompany:1701/src/secure/main.jsp
```

Note: The `corporateHeader.html` file that is referenced in the example is an example file that your organization can create to crown the frame that contains SRC.

Customize the Landing Pages

You can specify how many information objects appear in the Dashboard, Services, and Support pages, and their order of appearance. The order is defined in rows that appear left-to-right, and top-to-bottom. For example, you might want to hide the object that lists popular bundles or popular items. There are seven information objects (*widgets*) that can appear on any landing page. The following list describes the name and configuration ID of these widgets.:

- Pending Approvals (APPROVAL_STATUS)
- Request Status (REQUEST_STATUS)
- Popular Items (POPULAR_ITEMS)
- Request On Behalf (REQUEST_ON_BEHALF)
- Assistance (ASSISTANCE)
- Your Subscriptions (SUBSCRIPTION)
- Support (SUPPORT)

To customize the order of appearance of these widgets, follow these steps:

1. In the `manifest.xml` file, locate the `<widgetsGroup></widgetsGroup>` section. Within this section, you can configure the number of widgets for each landing page and the display order (left to right and top to bottom).

```
- <widgetsGroup>  
  + <widgets displayOn="DASHBOARD">  
  + <widgets displayOn="SERVICE">  
  + <widgets displayOn="SUPPORT">  
</widgetsGroup>
```

There are sub-parameters for each available object.

2. Locate the `<widgets></widgets>` parameter. There are sub-parameters for each available object. For example, your `manifest.xml` may resemble the following:

```
<widgets displayOn="pagename">  
  <widget displayOrder="1"
```

```
    type="APPROVAL_STATUS" visible="true"/>
    <widget displayOrder="2"
    type="REQUEST_STATUS" visible="true"/>
    <widget displayOrder="3"
    type="POPULAR_ITEMS" visible="true"/>
    <widget displayOrder="4"
    type="ASSISTANCE" visible="true"/>
</widgets>
```

3. You cannot change the name of the object, but you can change the order that they appear in by changing the number assigned to the object. For example, to make the REQUEST_STATUS object appear first, change the value of its displayOrder attribute from 2 to 1.

Note: Make sure to change the order value assigned to APPROVAL_STATUS.

4. Make sure each object has a unique displayOrder value.
5. To hide a widget, change its visible attribute from **true** to **false**.
6. Save your changes, but do not close the file until you are done.

Use the Launcher to Create Third Party Links

The Launcher is an empty space directly below the Dashboard, Services, Support, and Subscription menu bar. You can use the Launcher to link to third party applications, such as video training links, an Internet search Web site, an internal HR Web site, or a chat support Web application. You can customize the presentation to show just an icon, or you can display a small amount of text with each icon.

To create a third party application launcher that appears on the Dashboard, Support, or Services landing pages, you will need to modify the following two files:

- manifest.xml
- Custom.properties

To create a third party link from the Dashboard, Services, or Support pages, follow these steps.

Important Note: In the manifest.xml file, there are many xml elements that end with the term "key." Elements that end with the term "key" correspond to a property specified in the Custom.properties file. For example, the value of the <titleKey> element in the manifest.xml file is specified as "launcher_title" which corresponds to the "launcher_title" property in the Custom.properties file.

1. Open this file with a text editor.

```
C:\...\src-9.32\resources\client\en\Custom.properties
```

2. Locate the template example in the file.
3. Follow the directions to copy the template section and paste it below the template. Customize only the pasted section. It is a good idea to add a comment line above the new section to identify its purpose.
4. Follow the guidelines in the value table to make sure this section contains values that you want to appear in the Service Request Catalog Dashboard. Note that:
 - The identifier can be user-defined; however, when you edit the manifest.xml file, the identifiers have to match.
 - The .value for each identifier is exactly what will display in the user interface.

In this example, we will build a launcher for the XYZ third party business application.

```
/* New Launcher Example */  
launcher_title = XYZ  
launcher_desc = You can link to XYZ here.  
dropdown_list_label = Select  
dropdown_value_1 = I want information  
dropdown_value_2 = I want to order  
dropdown_value_x = I want to see status  
text_label = Type Your Comments  
action_label = Send  
action_type = GET
```

5. Save and the file but keep it open for reference as you configure the next file.
6. Open this file with a text editor.

C:\...\src-9.32\secure\configuration\manifest.xml

Use the following lines as a reference for the remaining steps:.

```
<launcher launcherId="XYZ_Launcher" displayOrder="5" showLogoBox="true" show  
LogoDropShadow="true">  
  <logo>XYZ_image.png</logo>  
  <titleKey>launcher_title</titleKey>  
  <descriptionKey>launcher_desc</descriptionKey>  
  <launcherType>APP</launcherType>  
  <fields>  
    <field fieldId="text_label">  
      <labelKey>text_label</labelKey>  
      <type>TEXT</type>  
    </field>  
    <field fieldId="text_area">  
      <labelKey>text_area</labelKey>  
      <type>TEXTAREA</type>
```

```

</field>
<field fieldId="dropdown_list_Label">
  <labelKey>dropdown_list_Label</labelKey>
  <type>DROPDOWN</type>
  <options>
    <option>
      <labelKey>value_1</labelKey>
      <value>X</value>
    </option>
    <option>
      <labelKey>value_2</labelKey>
      <value>Y</value>
    </option>
    <option>
      <labelKey>value_2</labelKey>
      <value>Z</value>
    </option>
  </options>
</field>
</fields>
<action>
  <labelKey>action_Label</labelKey>
  <link><![CDATA[http://www.XYZ.com]]></link>
  <method>GET</method>
</action>
</launcher>

```

7. Locate the <launchers></launchers> section. The <launchers> element may contain one or more <launcher></launcher> sub-sections that describe the attributes of a third party application(s) that you want to display on the Dashboard. Replace the blue italicized text between each tag with a valid value. Follow the guidelines in the value table to make sure this section is syntactically correct and contains values that can be processed successfully.
8. Save the file, but keep it open for reference as you configure the next file.
9. If the application is running, log out and log in again to confirm that the third party application link appears in the dashboard.

Notes:

- For a detailed example of how to configure the launcher to start a support chat option, see ["Example: Add Bomgar Live Chat" on page 64](#)
- If you chose to link to use the Launcher to play a video, note that SRC uses an open source video player class from Adobe. Therefore acceptable video formats are .flv, .f4v, and mp4 - based container formats. For more information regarding the video player, reference the following link:

http://help.adobe.com/en_US/flex/using/WSc78f87379113c38b-669905c51221a3b97af-8000.html

Customized Error Messages

You can customize error messages that appear when you launch third party applications. These messages have default content but you may want to change them. To customize error messages, follow these steps:

Tip: It is always a good idea to make a backup of any system files before you make changes to them. If an error occurs later, you can always return the file to its original state.

1. If necessary, stop the server.
2. Open this file with a text editor.

```
C:\...\src-9.32\resources\client\en\Custom.properties
```

3. Locate this section:

```
# Customizable error message properties
```

4. Each message consists of a property ID that you should not change, and a property value that you can change.

Property ID Property Value

```
default_error_title           = Error
default_success_title        = Success
default_error_message        = Message {0}.
default_success_form_message = Your information is posted.
```

Edit one or all property values as necessary.

5. Save and close the file.

Hide the Service or Support Tabs

You can show or hide the Service, Support, and Generic Support widgets, including any references to those services in other areas of Service Request Catalog. To hide any of these tabs, set the value of the following parameters to false as appropriate. By default, these parameters are shown, which means that the default values are true:

```
src.sm.enableService=false //hides Service tab and hides the Services section from the Your Requests widget and Request Inbox.
src.sm.enableSupport=false //hides Support tab and hides the Support section from the Your Requests widget and Request Inbox
src.sm.canRequestGenericSupport=false //hides the generic support widget.
```

Note: You must still hide the Services and Support widgets by using the steps outlined in ["Customize the Landing Pages" on page 14](#)

Configure the Your Requests widget and Your Service Request pane

You can choose to show or hide the Pending, Closed, Denied, or Approved menu items for either the Your Requests widget or the Your Service Request pane. These settings are controlled by the following parameters in the `applicationsContext.properties` file. The default values for these parameters is `true`, which means the menu item will be shown. You can hide an item by setting the appropriate parameter to `false`.

Your Service Requests

```
src.ui.service.enablePendingInbox=true  
src.ui.service.enableApprovedInbox=true  
src.ui.service.enableClosedInbox=true  
src.ui.service.enableDeniedInbox=true
```

Your Requests

```
src.ui.service.enablePendingOnWidget=true  
src.ui.service.enableApprovedOnWidget=true  
src.ui.service.enableClosedOnWidget=false  
src.ui.service.enableDeniedOnWidget=false
```

Note: The Your Requests widget can show a maximum of three of the menu items. This is due to the height limitation of the widget.

Styling the Interface

You can style Service Request Catalog to show your Company logo, change the application name, or set text formatting and the background as needed.

Customize the Application Name

You can replace the application name, Service Portal, with the name of your company or a customized application name. For example, you would like the application name to reflect your company branding and your company name is XYZ.

1. Open this file with a text editor.

```
src\resources\client\en\Custom.properties
```

2. Locate the `// Company Display Name` section.
3. Change the name of the default company name to your file name. For example, change:

```
default_company_name_label = Service Central
```

to

```
default_company_name_label = XYZ Company
```

You can change the ID on the left side to some other ID. For example, you could change this:

```
default_company_name_label = XYZ Company
```

to

```
my_company_name = XYZ Company
```

4. Save and close the file.

Edit manifest.xml

The manifest.xml file contains a section that identifies the key of the company name to be displayed.

1. Open this file with a text editor:

```
...\src\secure\configuration\manifest.xml
```

2. Locate this section:

```
<companyDisplayNameKey>default_company_name_label  
</companyDisplayNameKey>
```

3. If you changed the ID value in Custom.properties, make the corresponding change in this file.

Example: Changing default_company_name_label to my_company_name in the Custom.properties file. requires the new ID to appear in the manifest.xml file.

```
<companyDisplayNameKey>my_company_name  
</companyDisplayNameKey>
```

4. Save manifest.xml. The changes will be captured and propagated to all distributed servers the next time the server synchronization scripts run. To see them immediately, log out and log back into your local instance of Service Request Catalog

Customize Colors and Graphics

You can change the interface colors and introduce customization that echoes your company's signature branding and color palette.

Service Request Catalog packages files that you can edit to change the basic application design and color palette. You can also redesign the backgrounds for the header, main, and footer areas. You do not need to be a designer or a programmer to make these changes. Obtain the required tools and follow the directions in this section. A reference for the panel.swf file can be found in ["Appendix D: Interface References" on page 89](#)

Before You Begin

Customization of most of the graphics and images of Service Request Catalog is performed by modifying the symbols in the `panel.fla` file and replacing the `panel.swf` used in SRC. The symbols in the `panel.fla` file each correspond to a graphic, image, or text area in the SRC.

You can find the `panel.swf` file and the `panel.fla` in the following location:

```
<SRC_Root_Folder>\secure\configuration\default
```

Important Notes:

- `<SRC_Root_Folder>` refers to the Web application server where you installed SRC.
- Before you attempt to replace the `panel.swf` file, we recommend that you make a copy and store it in a convenient location. You should perform all your adjustments on the copy first.
- You need to obtain Adobe® Flash® Professional CS5.x (or a later release) to edit the relevant files. You may also need an image editing application, such as Adobe® Illustrator® or Adobe® Photoshop®.
- Make sure you have your corporate branding color values (expressed in hexadecimal numbers) handy.

Modifying panel.swf

To modify `panel.swf` and customize Service Request Catalog, follow these steps:

Note: You cannot change the name of the `panel.swf`. SRC will search only for the name `panel.swf`.

1. Open `panel.fla` in Flash Professional CS5.x, and note the various symbols. You should see symbols for the background, header, footer, logo, and other elements of the SRC user interface.
2. Select a symbol that you wish to modify.
3. By default, Flash Professional CS 5.x provides numerous panels on the right side of the screen. These panels allow you to access the properties of the symbol that you selected. You can choose to adjust the color, gradient of the various symbols. For example, you may click on the symbol for the background, and use the color panel to specify a background that matches your organization's branding.
4. When you have finished modifying the symbols, make sure that all symbols are under the "Illustrator Symbols" folder in the Library panel in Flash Professional. During the process of manipulating the symbols, Flash Professional may have moved the symbols from their original hierarchy in the Library Panel.
5. Export the file as a `panel.swf` file.
6. Stop any running instances of SRC.

7. Copy the `panel.swf` file that you exported in Step 4, and then navigate to where the current `panel.swf` is installed. By default, this folder is in the following location:

```
<SRC_Root_Folder>\secure\configuration\default
```

8. Rename the current `panel.swf` to `panel.OLD` to ensure that you maintain an original copy, and then paste your modified `panel.swf` into the same directory. Move the `panel.OLD` file to a safe location.
9. Restart Service Request Catalog. You should see the modifications you made to the SRC user interface.

Note: By manipulating the symbols in the `panel.swf` file, you can customize almost all of the graphical aspects of SRC. This includes the your corporate logo, which you can adjust by replacing the current logo in `panel.swf`. However, you should take care not to adjust the following:

- **Size:** Adjusting the size of the symbols can lead to unpredictable results and may cause a poor user experience. SRC is designed with most elements at specific sizes.
- **Nine scale slicing:** Manipulating Nine-scale slicing can cause some symbols to become distorted in unintended ways, so that the symbols may be too large or small, or distorted.
- **Names:** Do not change the names of any of the symbols or their corresponding associations.

Handling Empty Symbols

Some symbols in `panel.swf` may be blank or empty by default. For example, the `backgroundImage` symbol contains no image by default. However, you may wish to add a picture to that will appear as the backdrop to your users' SRC experience.

Similarly, you may wish to remove some symbols. To do this, simply reset the symbol with a blank graphic.

Warning: Do not delete any symbol.

Modifying the favorite icon

The favorite icon is displayed in the browser tab header. By default, this icon is the HP logo. However, you can replace this logo with your organization's logo by replacing the `favicon.ico` file located in the SRC root folder. Note that the size for this icon must be 16 x 16 pixels.

Customize the Text Format

You may want to make changes to the appearance of text in the application. You can control fonts, text color, and size in the header and footer areas by customizing the `style.css` file. To reference the `styles.css` file, and for some graphics that show the location of the different text boxes, see ["Appendix D: Interface References" on page 89](#)

Text Manipulation Overview

Text styles, font, weight and other characteristics are controlled by the `styles.swf` file. You can locate the `styles.swf` file at the following location:

```
<SRC_Root_Folder>\src\secure\configuration\default
```

However, the `styles.swf` file is an auto-generated `.swf` file derived from the corresponding `styles.css` file. Because, Cascading Style Sheets are a well-known standard you can use easily configure attractive looking text according to your organization's branding guidelines. To convert the `styles.css` file to a `styles.swf` file, you must use the Adobe Flex Software Development Kit (SDK) compiler. You can download this open-source compiler from the following Web site:
www.Adobe.com

Converting `styles.css` to `styles.swf`

Once you have downloaded the compiler, you can use it to convert the `styles.css` file to a `styles.swf` file. To do this, follow these steps:

1. Extract the compiler to an easily accessible folder.
2. Add the location of the mxml compiler to the Path environment variable.
3. Open a command prompt, and then navigate to the location where you stored the `styles.css`.
4. Enter the following command at the command prompt:

```
mxmlc styles.css
```

5. When you do this, the compiler should generate the `styles.swf` file. Copy the `styles.swf` file, and replace the original `styles.swf` in the following location:

```
<SRC_Root_Folder>\src\secure\configuration\default
```

Note: Make sure that you make a copy of the original `styles.swf` file and store it in a safe place.

Add a New Language

SRC is delivered in 16 language packages, which can meet most localization requirements. However, if you wish to add additional languages, you can do so on your own by using the following procedure. To do this, you must modify the following two areas:

- The resource files located in the following directory:

```
[release war]\resources\client and [release war]\resources\server.
```

All resources, messages, and user interfaces labels are stored in these properties files.

- Service Manager Server: The resources for configurable items, such as catalog items, user options, and checkout panels are stored here.

Configuration Steps

In this example, we will convert SRC to Italian.

Add new language into mlu.properties

1. Find resource under [release war]\resources\mlu.properties
2. Add language code for key "supportedLanguages"

```
supportedLanguages=US,DE,IT,FR,ES,JP,BRAZILIAN,ZH,NL,KO,RU,CS,HU,PL,AR,HE
```

3. Add lines like below for new language

```
IT.languageId=ITALIAN
IT.localeLanguage=it
IT.localeCountry=IT
IT.displayName=Italiano
```

Notes:

- If you do not care about using the country code for Italian, just put IT as the language code, and all resource files must be appended as follows:

```
abc_it.properties.
```

- If you do wish to put the country code (for example Switzerland (CH)) as the localeCountry for this language, then all resource properties files must be appended as follows:

```
abc_it_CH.properties;
```

Create a folder for Italian

If you do not care about using the country code for Italian, create a new folder under [release war] \resources\client\ named: it

If you do wish to put the country code, as the localeCountry for this language, create a new folder under [release war] \resources\client\ named: it_CH

Copy all client resource files

1. Copy all resource files from the [release war] \resources\client\en folder to the folder that you created in the previous step.
2. Append "_it" or "_it_CH" to each client resource file, according to your decision in the previous step.

Copy all server resource files

1. Copy all server resource files from the [release war]\resources\server\en to the folder the you created previously.
2. Append "_it" or "_it_CH" to each server resource file, according to your decision.

Translate each properties file

1. Translate the property files by hand.

Note: Please heed those files that are marked by the "DO NOT TRANSLATE THIS FILE" statement; Similarly, there are several lines in the files that are marked "DO NOT TRANSLATE". If you are not sure what lines should be translated or not, you can compare the existing files between a non-English version and the English version.

Localize the resources in Service Manager

Service Manager provides extensive language support. For details on how localize Service Manager, refer to *SM_Open_L10N_UG.pdf*.

Restart Tomcat

1. Restart Tomcat, and then navigate to SRC.
2. Set the Browser Language Preference in SRC.

Reference Materials

| | |
|---|-----|
| SRC Service Manager Customization Reference | 136 |
| Server Customization Reference | 125 |
| Client Customization Reference | 89 |

Service Manager Configuration

Some SRC customization tasks require you to make changes to the Service Manager server to see the result in Service Request Catalog. Make sure you have a Service Manager client that connects to the same server that you use for Service Request Catalog.

Tailoring Best Practices

Making changes to a Service Request Catalog tailoring configuration can create problems. The best practice recommendation is to invest in a complete pre-production test cycle to be sure you have all the fields you need and that the names of these fields are meaningful to your user community. It is important to know that when these user fields are available and populated with information by users, if you make post-production changes to a tailoring configuration, there is an impact to existing requests.

| If I Do This | This Is the Result |
|--|--|
| Add new fields to an existing configuration. | Existing requests will not display those fields when you check their status. |
| Remove any (or all) custom fields from an existing configuration and add new ones. | Existing requests will not display the data associated with the deleted fields. The new fields will not appear in those requests. The collected data associated with the deleted fields exists in the Service Manager database, but cannot display in the Service Request Catalog user interface. Best Practice: Do not remove a custom field once it is used in a production environment. |
| Add a new field that is mandatory. | Existing requests will not display this field as long as there is no attempt to edit and resubmit a request. If the user resubmits the request, the new mandatory field appears and the user must supply any required information or form validation fails. Best Practice: Set a default value for new mandatory fields. |

Tailoring Check-Out Panels

If you are using Service Request Catalog to submit a catalog or support request, you may want to customize Service Request Catalog to your specific business requirements. You can use a simple Service Manager wizard to add new custom Checkout panels, define custom fields and labels that will appear on the Service Request Catalog Checkout panels when you submit a request. You can also completely modify all Out-of-box Checkout panels in the SRC UI.

Checkout Panels

In Service Request Catalog, users can submit different types of requests through various navigation paths. Each of these paths brings users to a specific request panel, which is called a Checkout panel. The following are the default Checkout panels in an OOB implementation of Service Request Catalog:

| Panel name in SRC UI | Description |
|----------------------|--|
| Service Catalog | This checkout panel is used to request for an item that is in the Service Catalog. To access this panel, click the Services tab on the Service Request Catalog home page, click Browse Catalog , and then browse to a service catalog item/bundle. |
| Support Catalog | This checkout panel is used to request for an item that is in the Service Catalog. To access this panel, click the Support tab on the Service Request Catalog home page, click Browse Catalog , and then browse to a Support Catalog item/bundle. |
| General Support | This checkout panel is used to request for an item that is in the Support Catalog. To access this panel, click the Support tab on the Service Request Catalog home page, click Create , and then create a custom Support Request. |

Note:

You can add custom fields for the three checkout panels. However, for each company, you can create only one configuration for each checkout panel.

If you upgraded from an earlier version of SRC by using the Migration Tool, you should note that only a limited migration is performed upon the checkout panel. Therefore, you should verify your checkout panel configuration after the migration, and adjustment as necessary. If you have also corresponding configurations for this company, these configurations will be used. If there are no corresponding configurations for this company, the default configuration will be used.

Configurations

A Checkout panel is defined by a configuration in the SRC Tailoring section of Service Manager. A configuration is associated with one or more sections, and each section can contain one or more custom fields that will be added to a Service Request Catalog checkout panel. You must specify a label for each section and field, which will display on the relevant checkout panel in the Service Request Catalog UI.

Sections

A new section in the checkout panel can have a title and a combination of these objects:

- Fields that exist in a Service Manager table
- Labels that exist in a Service Manager form or a user-defined label
- Radio buttons
- Checkboxes
- Lines of text for a user description text box
- Service Manager lookup fields
- Combo boxes
- Date fields
- Number fields
- Time fields*

*Time is usually specified by taking the system from the Service Manager server and adjusting for the clients location by using the GMT adjustment in the users browser.

Custom Fields

Custom fields are useful when you need to gather more information for a certain type of catalog request. Custom fields appear as a new accordion section in the checkout panel after the standard accordion sections for the request type. The data collected from the user with the custom fields is stored in the appropriate Service Manager table that contains each field. New fields and labels appear in Service Request Catalog as a new section in the checkout panel. You can use this feature to collect additional information from the user that is relevant to the item or service fulfillment business logic.

Notes on Creating Custom Fields

To create a custom field, you must specify the following:

- The checkout panel to which the field belongs.
- The section of the checkout panel to which the field belongs.
- A label for the field.
- User restrictions on the field.
- The field type
- Default values for the field, if any.
- Mandatory requirements for the field, if any

You can specify the data type for a custom field and Service Manager will validate the user input against that data type and return an error message if there is a mis-match. In this case, the user can try again to provide data that is consistent with the assigned data type.

You must have the SysAdmin capability word in your user profile to access the Service Request Catalog tailoring wizard.

If a custom field was optional, and the user does provide a response or a value, the custom field does not appear in the status view.

Lookup Tables and Lookup Fields

Some fields have a lookup table associated to it because of the business logic in Service Manager. For example, if you want to add a custom field named City to a checkout panel, users should not be allowed to type a city name in this field. Instead, users should select a value retrieved from the list of city field values in the location table. To achieve this, specify the location table as the lookup table, and the city field in this table as the lookup field for the custom field.

Note: To use a lookup table and field, you must specify the Display Type of the custom field as Text.

You must also specify restrictions on the custom fields to limit the user's ability to change the content of the custom field.

Custom field Restrictions

| Restriction: | Restriction Description: |
|-------------------|--|
| Read Only | Setting a custom field as read-only means that the user will not be able to modify this field. If no default value is specified, this field will not appear on the checkout panel. |
| Create Only | Specifying a custom field as Create only means that the user can only add the field and its associated information when creating the request. This field will be disabled if the user is viewing their requests. |
| Create and Update | Create and Update means that user can add the field to their service request, and that this field can also be updated from the Service Request Catalog. |

When the user checks the status of a request that contains a custom field, the field is read-only information until the user resubmits that request. At that point, the user can edit the custom field value and any other fields to be updated.

To make custom fields visible in the Service Request Catalog checkout panel, your Service Request Catalog `applicationContext.properties` file must point to the same Service Manager server that you used to create the custom fields.

Custom Fields and the cache

It is possible that an Administrator adds new fields to a Service Catalog item at the same time that user is attempting to order that item. If the Administrator implements a new custom field before the user submits a request, an inconsistency may be introduced to the system. To prevent this, user data is cached on the server side. Therefore, if an administrator modifies a panel or item while a user is actively using SRC, the user's data will be preserved and repopulated.

Note: We recommend that the Administrator avoid modifying custom fields while users are actively engaged in using the system.

Add a Configuration

A *configuration* is the description of one or more new sections with custom fields that will appear in a Service Request Catalog checkout panel. Use the examples to create a sample configuration of your own.

To complete any of the Service Request Catalog tailoring tasks, start with these basic steps.

1. Start a Service Manager Windows client session. Make sure the Service Manager Windows client connects to a Service Manager 9.32 server. The client can be an earlier version, but the server must be a 9.32 server.
2. Expand the left Navigation pane.
3. Click **Tailoring > SRC Tailoring**. Service Manager displays an SRC Configuration wizard. The first page of the wizard lists existing configurations. There can be a maximum of three configurations that add new sections with custom fields to the Support catalog, Services catalog, or Support checkout panel.

To add a configuration, follow these steps:

4. From the SRC Configuration Wizard home page, click **Add a Configuration**.
5. Create a **Name** for the new configuration.

Example: Type Support Custom Fields.

6. Click the drop-down list to specify which checkout panel will display the new custom fields. Choose **Service Catalog**, **Support Catalog**, or **General Support**.

Example: Select **Support Catalog**.

Note: A General Support request is a request for an item or service that is not listed in the regular Service Catalog or Support Catalog.

7. If your Service Manager instance is in multi-company mode, choose a specific **Company** from the drop-down list.
8. Click **Next**.
9. You can choose an existing label for a new section, or you can create a new one. Click **Select a label** or **Add a label**.

Example: Click **Add a label** and type CONTACT NUMBERS for the name of the new label.

10. You can choose an existing label for a new field, or you can create a new one. Click **Select a**

label or **Add a label**.

Example: Click **Add a label** and type FAX for the name of the new label.

11. Click **Next**.
12. Click the **Field Name** drop-down to choose the field that you want populate with new information. The values of the drop-down list can be changed in the “incidents” table from the Database Dictionary (DBdict).

Example: Choose Fax from the drop-down list.

Note: If you select a lookup table in step 16, then you must use the lookup field name here.

13. In the **Modifiability** field, you can set the field permission to enable the user to do one of the following:
 - **Create-only** to change the field only when submitting a new request. It cannot be updated if the request is resubmitted.
 - **Read-only** to display data that is never changed during the initial submission or during resubmission.
 - **Create and Update** to change the field when submitting a new request and updating if the request is resubmitted.

Example: Select **create-only** from the drop-down list.

14. In the **Display Type** list, select the type of field that you want to appear in the Service Request Catalog checkout panel. Choose a display that is consistent with the type of data that you want to gather. For example, if you want the user to provide information, choose **Text**. If you want the user to choose from a pre-defined list, choose **Pick List**.

Example: Choose the **Text** option.

15. Click **Next**.
16. In the **Lookup Table** field, select the table that contains the field to be populated with (or display) new information.

Example: Leave this blank if you want to create a plain text field.

Note: If you use a lookup table here, then you must use the lookup field name in step 12.

17. Select **Mandatory** if the user must provide this information. If the information is optional, skip this step.

Example: Click the **Mandatory** checkbox.

Note: If you selected specified the field permissions in step 12 as **Read-only**, **Mandatory** will not be an available option.

18. Select **Default Value** if you want to specify a default value for the user.

Example: Type `None` as the default value (in case the user has no Fax number).

19. Click **Next**.

20. The Wizard displays the custom field that you created. You can click **Add New Custom Field** to add another.

Example: Click **Add New Custom Field**.

21. Repeat step 10 through step 21 to add another custom field.

Example: Create one more custom field for a Phone Number using the contacts table and Alternate Phones field.

22. When you have more than one custom field, you can:

- Change the order of appearance by selecting a field and clicking **Move Field Up** or **Move Field Down**.
- Select a field and click **Edit Custom Field** to make changes.
- Click **Remove Custom Field** to delete that field.

23. When you are satisfied with the result, click **Finish**.

Edit an Existing Configuration

You can edit an existing configuration for Checkout panel, including the default out-of-box Checkout Panels. You can use the following steps to modify the default Checkout panels to do any of the following:

- Enable or disable attachments.
- Change the name of the configuration as it appears in Service Manager.
- Change the name of the Checkout Panel as it appears in the SRC UI.
- Change the Company for which the Checkout panel applies.

To complete any of the Service Request Catalog tailoring tasks, start with these basic steps.

1. Start a Service Manager Windows client session. Make sure the Service Manager Windows client connects to a Service Manager 9.32 server. The client can be an earlier version, but the

server must be a 9.32 server.

2. Expand the left Navigation pane.
3. Click **Tailoring > SRC Tailoring**. Service Manager displays an SRC Configuration wizard. The first page of the wizard lists existing configurations. There can be a maximum of three configurations that add new sections with custom fields to the Support catalog, Services catalog, or Support checkout panel.

Then, to edit the configuration, follow these steps:

1. Click **Edit Configuration**.
2. Edit the field you wish to change.
3. Click **Next**.

Localize a configuration

To localize a configuration, follow these steps:

1. Log on to the Service Manager client as a system administrator.
2. Type `db` in the command box, type `svcSrcConfig` in the **Table** field, click **Search**
3. Double-click `svcSrcConfig` in the search list, and then click **Search**.
4. Select the checkout panel configuration you want to localize.
5. In the details panel, find the **label id** under **Options** fields that you want to localize.
6. Type `message` in the command box, type `srcconfig` in the **class** field, `23` in the **Message Number** field, and then click **Search**.
7. Modify the text and language code that you want to localize. For example for the `fr` language code, enter the appropriate text in French, and then click **Add**.

Add a Section to an Existing Configuration

To complete any of the Service Request Catalog tailoring tasks, start with these basic steps.

1. Start a Service Manager Windows client session. Make sure the Service Manager Windows client connects to a Service Manager 9.32 server. The client can be an earlier version, but the server must be a 9.32 server.
2. Expand the left Navigation pane.
3. Click **Tailoring > SRC Tailoring**. Service Manager displays an SRC Configuration wizard. The first page of the wizard lists existing configurations. There can be a maximum of three

configurations that add new sections with custom fields to the Support catalog, Services catalog, or Support checkout panel.

Follow these steps to add a new section to a custom configuration.

1. From the SRC Configuration Wizard home page, double-click an existing configuration.

Example: If you added a section to the Support Catalog checkout panel, double-click the **Support Catalog** configuration.

2. To create a new section in that configuration, click **Add a Section**.
3. You can **Select a Label** from labels that you already created or you can click **Add a Label** to create a new section heading.

Example: Type `More Information` as the new section heading.

4. Click **Next**.
5. Repeat step 10 through step 19 in "[Add a Configuration](#)" on page 31 to add one or more custom fields to the new section.
6. When the new section is complete, click **Finish**.

Modify an Existing Section

You can make some editorial changes to an existing section.

- To change the name that identifies that section in the checkout panel, select a label and click **Edit Section Label**.
- To delete the section from the checkout panel, select a label and click **Remove Section**.
- To change the order of a section's appearance, select a label and click **Move Section Up** or **Move Section Down**.

Delete a Configuration

To complete any of the Service Request Catalog tailoring tasks, start with these basic steps.

1. Start a Service Manager Windows client session. Make sure the Service Manager Windows client connects to a Service Manager 9.32 server. The client can be an earlier version, but the server must be a 9.32 server.
2. Expand the left Navigation pane.
3. Click **Tailoring > SRC Tailoring**. Service Manager displays an SRC Configuration wizard. The first page of the wizard lists existing configurations. There can be a maximum of three

configurations that add new sections with custom fields to the Support catalog, Services catalog, or Support checkout panel.

Follow these steps to delete a custom configuration.

1. From the SRC Configuration Wizard home page, right-click an existing configuration.
2. Click **Remove Configuration**.

Multi-Company Mode

If you already operate in multi-company mode, your configuration of custom fields changes slightly. In this case, you can define a custom field configuration for each type of request (Service Catalog, Support Catalog, and Support request) for each company.

You can verify whether multi-company mode is configured by following these steps:

1. Log into Service Manager as a system administrator.
2. Click **System Administration > Base System Configuration > Miscellaneous > System Information Record**.
3. On the **General** tab, make sure that **Run in Multi-Company Mode** is selected.

If you already configured multi-company mode, you also created the individual companies and associated those companies with users. You can verify the settings for any specific company by logging into Service Request Catalog as a user assigned to that company.

Example: Multi-Company Mode

The following steps provide an example of how you might configure multi-company mode for SRC:

1. Create two new users in Service Manager. To do this, navigate to **System Administration > Ongoing Maintenance > User Quick Add Utility**.
2. Enter the required information to create a new user named `kuser`, click **Next**, and then follow the onscreen instructions to save the operator record. Repeat this step and create another user named `duser`.
3. Create two new companies. To do this, navigate to **System Administration > Base System Maintenance > Companies**.
4. Click **Add** and then enter the required information to create a new company named `kcompany`. Repeat this step and create another company named `dcompany`.
5. Assign the new users to the new companies. To do this, type `operator` in the command line, press the **Execute** button.

6. Search for kuser. In the Operator Record for kuser, select kcompany from the **Default Company** compfill control. Repeat this step for duser, but set the **Default Company** compfill control to dcompany.
7. In the manifest.xml file for your SRC deployment, modify the configuration section as shown in the following sample:

```
<configurations>
  <configuration name="kcompany">
    ...
  </configuration>
  <configuration name="dcompany">
    ...
  </configuration>
</configurations>
```

8. To customize SRC for each company, add the theme resources (for example, panel.swf and style.swf) to each company's folder.

Miscellaneous Tasks

The tasks in this section are optional tasks, which you may need to complete according to your organizations requirements. You should read and familiarize yourself with these tasks in case you wish to return to perform additional configuration of Service Request Catalog at a later time.

Enable Knowledge Management

Before a user creates a support request, you may want to encourage them to search an external knowledge database to find a documented solution or recommendation. This feature is optional, but can be useful to help your user community resolve problems quickly without opening formal support requests.

Before you try to enable Knowledge Management from Service Request Catalog, verify the following:

- You have a Knowledge Management license. Without this license, SRC cannot display a Knowledge Management Search page.
- Knowledge Management is configured according to the recommendations in the Knowledge Management documentation.

When you enable the Knowledge Management link, the interface displays a Search Knowledgebase window where the user can type a search phrase.

To enable the Knowledge Management link, follow these steps:

1. Open this file with a text editor:

```
C:\...\WEBINF\classes\applicationContext.properties
```

2. Locate and then set the **src.km.enabled** property to **true**.
3. Save and close the file.
4. Restart the Tomcat application server to make the changes take effect.

After enabling Knowledge Management you must re-index the SRC Knowledge Base index. To do this, follow these steps:

1. In the Service Manager client, navigate to **Knowledge Management -> Administration -> Environment**.
2. Check the **SRC?** box.
3. Verify that the correct server is specified in the **Search Server Name:** drop-down box. If not, select the correct server from this drop-down box.
4. Click the **Full Re-index** button.

Subscriptions

The configuration of Subscriptions in Service Request Catalog is performed in the standard Service Manager Service Catalog. For documentation on how to set up Service Catalog items including Subscriptions, consult your HP Service Manager documentation.

Note: It is not required that a user of SRC must make or have a Subscription to request an item.

Enable Approval Delegation

To enable other operators to delegate approval authority, a System Administrator must make the following changes to an operator's application profile:

| To delegate approval authority for this application | Requires enabling this application profile option |
|--|---|
| Change Management | Can Delegate Approvals |
| Request Management | Delegate Approvals |
| Service Catalog (controlled by Service Desk profiles) | Delegate Approvals |

The Delegate Approvals or Can Delegate Approvals profile option only controls whether an operator can view the Approval Delegation wizard. Separate application profile options control whether an operator can view or approve objects in an application.

Enable Delegation for Service

To enable Service Request Catalog users to delegate approval authority, a System Administrator must update each operator's Service Profile record with the Delegate Approvals and View options. There is no additional capability word required for a Service Request Catalog user to delegate their approval rights. You can access the Service Profile settings from the Operator record.

1. Click **System Administration > Ongoing Maintenance > Operators**.
2. On the **General > Application Profiles** tab, make sure you specify self service in the **Service Profile** field.
3. Click **Save**.
4. On the **Security** tab, locate the Template Information section.
5. If Template_SelfService appears in the Template field, delete it.
6. Click **Save**.
7. Return to the **General** tab.
8. Place the cursor in the **Service Profile** field.
9. Click **Find** on the toolbar. Service Manager displays the Service Desk Security Profile form.
10. On the **Security** tab, select **Delegate Approvals**. This enables the user to assign their approval rights to someone else.

Service Desk Security Profile

Profile name :

◆ Security ◆ Forms

◆ Rights

New

Close

Update: Always

View

Allowed statuses:

Approval groups: Service Manager

Advanced search

Use operator full name

Can create personal views

Can create system views

Can notify

Lock on display

Alternate views

Modify Templates

Template Mass Update

Complex Mass Update

Mass Close

Delegate Approvals

11. If necessary, select **View**.

12. Click **Save**.

Enabling Approval Delegation for Change

The following example illustrates how to enable approval delegation for the Change Management application:

1. Log in to Service Manager with a System Administrator account.
2. Click **System Administration > Ongoing Maintenance > Profiles > Change Management Profiles**. Service Manager displays your custom Change Management profile form.
3. In Profile Name text box, type the name of a profile you want to grant approvals. For example, enter "change approver."
4. Click **Search**.
5. Service Manager displays the current Change Management Security Profile.

6. In the Approvals/Groups tab, select **Can Delegate Approvals**.
7. Click **Save**.

Note: Best practice is to enable the Delegate Approvals or Can Delegate Approvals option only for operators who can also view and approve objects in the application.

Enabling Approval Delegation for Quote

The following example illustrates how to enable approval delegation for the Request (Quote) Management application:

1. Log in to Service Manager with a System Administrator account.
2. Click **System Administration > Ongoing Maintenance > Profiles > Request Management Profiles**.
3. Service Manager displays your custom Request Management profile form.
4. In Profile Name text box, type the name of a profile you want to grant approvals . For example, type "request approver."
5. Click **Search**.
6. Service Manager displays the current Change Management Security Profile.
7. In the Alert/Approval tab, select **Can Delegate Approvals**, and then make sure that the check box "Approvals" is checked.
8. Click **Save**.

Note: Best practice is to enable the Delegate Approvals or Can Delegate Approvals option only for operators who can also view and approve objects in the application.

Improve SRC Performance

Note: These steps only improve performance only if you do not already have the an ApprovalA1 table in your installation of Service Manager. In Service Manager 9.32, this table already exists.

The following changes to the Service Manager database can improve the performance of Web Service requests from Service Request Catalog. These steps should be completed by an experienced Service Manager administrator.

These changes remedy problems that occur when the Service Manager server cannot fully translate Service Manager queries to an SQL server when fields that appear in the query are mapped to large object (LOB) type fields. The changes prevent inefficient scans by the Service Manager server when it runs queries against tables that reference LOB type fields.

Note: If you already mapped any repeating group in a table that is referenced by an Array table, you will have an Array table that uses the A1 alias. In this case, use a different alias, such as A2, and append that alias value to the base table name to form the table name. For example, you may

already have an Array table for the Approval file. If you already have an A1 table, use alias A2 and name the table APPROVALA2 instead.

Create the ApprovalA1 Table

Remap the current `.pending.groups` array from a CLOB/TEXT field to an Array table, creating APPROVALA1 table. Use the Dbdict utility, not the System Definition utility or the SQLMapping utility. To edit the dbdict for the Approval file, follow these steps:

1. In the **Fields** pane, scroll to and then double-click on the `current.pending.groups` array definition line.
2. Click the edit field.
3. Set `SQLTable` to `a1`.
4. Click **Next**. You should be positioned on the dbdict entry for the `current.pending.groups` character field.
5. Set `SQLTable` to `a1`.
6. Change the SQL type from CLOB to `VARCHAR(120)` and click **OK**.
7. Select the **SQL Tables** tab and add a new line with alias, name, and type of `a1`

`APPROVALA1 oracle10.`
8. Click **OK**. A pop-up dialog displays the DDL to create the new table.

Important: Do not modify the DDL.
9. Copy this DDL to the clipboard for future reference.
10. Click **User Alters**. Do *not* click SM Alters.
11. Click **OK** on the main dbdict window to update the dbdict.

Note: After you click **User Alters**, Service Manager displays a dialog warning that you must now alter the database using the displayed DDL. However, Service Manager also tries to modify the database directly, using that DDL. If it succeeds, nothing further is required.

The operation will succeed provided the `sqllogin` account information in the Service Manager `sm.ini` file is for a database user with the rights to issue `CREATE TABLE` and `CREATE INDEX` commands. If the `sqllogin` account information in the Service Manager `sm.ini` file is for a database user who does not have these rights, the operation will fail, and an appropriately authorized user must manually run the DDL that you copied to the clipboard against the Service Manager database schema.

12. Verify that Service Manager succeeded in creating the new table and index by examining the Service Manager database in Oracle to confirm that the new APPROVALA1 table was created.
13. Investigate any SQL error messages that appear.

14. Use Oracle SQL Developer or another database management tool to verify that the APPROVALA1 table was created.

Remap the parent.tree Field

The next task is to remap the parent . tree field in the capability file to an Array table. Use the Dbdict utility, not the System Definition utility or the SQLMapping utility. To edit the capability file to an Array table, follow these steps:

1. Select the **parent.tree** type array.
2. Click the **edit** field.
3. Set **SQLTable** to a1.
4. Click **Next**. You should be positioned on the parent . tree type character.
5. Change the **SQL** type to VARCHAR(50).
6. Change **SQLTable** to a1.
7. Click **OK**.
8. Go to **SQLTables**.
9. Add **a1 CAPABILITYYA1 oracle10**.
10. Click **OK**. A pop-up dialog displays the DDL to create the new table.

Important: Do not modify the DDL

11. Copy this DDL to the clipboard for future reference.
12. Click **User Alters**. Do not click SM Alters.
13. Click **OK** on the main dbdict window to update the dbdict.

Note: After you click **User Alters**, Service Manager displays a dialog warning that you must now alter the database using the displayed DDL. However, Service Manager also tries to modify the database directly, using that DDL. If it succeeds, nothing further is required.

The operation will succeed, provided the sqllogin account information in the Service Manager sm.ini file is for a database user with the rights to issue CREATE TABLE and CREATE INDEX operations. If the sqllogin account information in the Service Manager sm.ini file is not for a database user who does not have these rights, the operation will fail, and an appropriately authorized user must manually run the DDL that you copied to the clipboard against the Service Manager database schema.

14. Verify that Service Manager succeeded in creating the new table and index by examining the Service Manager database in Oracle to see if the new CAPABILITYYA1 table was created.

15. Investigate any SQL error messages that appear.
16. Use Oracle SQLDeveloper or another database management tool to verify that the CAPABILITYA1 table was created.

Remap the access.list Field

The next step is to remap the `access.list` field in the `svcCatalog` file to an Array table. Use the Dbdict utility, not the System Definition utility or the SQLMapping utility. To edit the `svcCatalog` file to an Array table, follow these steps:

1. Select the `access.list` type array.
2. Click the **edit** field.
3. Set **SQLTable** to `a2`.
4. Click **Next**. You should be positioned on the `access.list` type character.
5. Change the **SQL** type to `VARCHAR(50)`.
6. Change **SQLTable** to `a2`.
7. Click **OK**.
8. Go to **SQLTables**.
9. Add `a2 SVCCATALOGA2 oracle10`.
10. Click **OK**. A pop-up dialog displays the DDL to create the new table.

Important: Do not modify the DDL.

11. Copy this DDL to the clipboard for future reference.
12. Click **User Alters**. Do not click SM Alters.
13. Click **OK** on the main dbdict window to update the dbdict.

Note: After you click **User Alters**, Service Manager displays a dialog warning that you must now alter the database using the displayed DDL. However, Service Manager also tries to modify the database directly, using that DDL. If it succeeds, nothing further is required.

The operation will succeed, provided the `sqllogin` account information in the Service Manager `sm.ini` file is for a database user with the rights to issue `CREATE TABLE` and `CREATE INDEX` operations. If the `sqllogin` account information in the Service Manager `sm.ini` file is not for a database user who does not have these rights, the operation will fail, and an appropriately authorized user must manually run the DDL that you copied to the clipboard against the Service Manager database schema.

Remap the operators Field

The next task is to remap the operators field in the `kmgroup` file to an Array table. Use the Dbdict utility, not the System Definition utility or the SQLMapping utility. To edit the `kmgroup` file to an Array table, follow these steps:

1. Select the operators type array.
2. Click the edit field.
3. Set **SQLTable** to `a1`.
4. Click **Next**. You should be positioned on the operators type character.
5. Change the **SQL** type to `VARCHAR(60)`.
6. Change **SQLTable** to `a1`.
7. Click **OK**.
8. Go to **SQLTables**.
9. Add `a1 KMGROUPA1 oracle10`.
10. Click **OK**. A pop-up dialog displays the DDL to create the new table.

Important: Do not modify the DDL.

11. Copy this DDL to the clipboard for future reference.
12. Click **User Alters**. Do not click SM Alters.
13. Click **OK** on the main dbdict window to update the dbdict.

Note: After you click **User Alters**, Service Manager displays a dialog warning that you must now alter the database using the displayed DDL. However, Service Manager also tries to modify the database directly, using that DDL. If it succeeds, nothing further is required.

The operation will succeed, provided the `sqllogin` account information in the Service Manager `sm.ini` file is for a database user with the rights to issue `CREATE TABLE` and `CREATE INDEX` operations. If the `sqllogin` account information in the Service Manager `sm.ini` file is not for a database user who does not have these rights, the operation will fail, and an appropriately authorized user must manually run the DDL that you copied to the clipboard against the Service Manager database schema.

14. Verify that Service Manager succeeded in creating the new table and index by examining the Service Manager database in Oracle to see if the new `KMGROUPA1` table was created.
15. Investigate any SQL error messages that appear.
16. Use Oracle SQL Developer or another database management tool to verify that the `KMGROUPA1` table was created.

Remap the members Field

Re-map the `members` field in the `cm3groups` file to an Array table. To edit the `cm3groups` file to an Array table, follow these steps:

1. Select the `members` type array.
2. Click the `edit` field.
3. Set **SQLTable** to `a1`.
4. Click **Next**. You should be positioned on the `members` type character.
5. Change the **SQL** type to `VARCHAR(60)`.
6. Change **SQLTable** to `a1`.
7. Click **OK**.
8. Go to **SQLTables**.
9. Add `a1 CM3GROUPSA1 oracle10`.
10. Click **OK**. A pop-up dialog displays the DDL to create the new table.

Important: Do not modify the DDL.

11. Copy this DDL to the clipboard for future reference.
12. Click **User Alters**. Do not click SM Alters.
13. Click **OK** on the main `dbdict` window to update the `dbdict`.

Note: After you click **User Alters**, Service Manager displays a dialog warning that you must now alter the database using the displayed DDL. However, Service Manager also tries to modify the database directly, using that DDL. If it succeeds, nothing further is required.

The operation will succeed, provided the `sqllogin` account information in the Service Manager `sm.ini` file is for a database user with the rights to issue `CREATE TABLE` and `CREATE INDEX` operations. If the `sqllogin` account information in the Service Manager `sm.ini` file is not for a database user who does not have these rights, the operation will fail, and an appropriately authorized user must manually run the DDL that you copied to the clipboard against the Service Manager database schema.

14. Verify that Service Manager succeeded in creating the new table and index by examining the Service Manager database in Oracle to see if the new `CM3GROUPSA1` table was created.
15. Investigate any SQL error messages that appear.
16. Use Oracle SQL Developer or another database management tool to verify that the `CM3GROUPSA1` table was created.

Additional Performance Tips

If the user receives a “max session exceed” error when running SRC with large number of concurrent users, increase the Service Manager max threads per node. To do this, follow these steps:

1. Edit the sm.ini in the root Service Manager directory: \$SM_install_dir\$\server\RUN
2. Locate the Threadspersprocess parameter, and then increase the number. If the parameter is not present in the sm.ini file, add the parameter by using the following format:

```
Threadspersprocess : {number}
```

3. Restart the Service Manager server.

For Oracle databases, you can improve the SRC login and approval performance by adding an expression index:

1. Log in to the Oracle Database under the sysop role.
2. Locate the “APPROVALM1” table.
3. Create the following expression index on this table:

```
FILE_NAME || '-' || COMPONENT || '-' || UNIQUE_KEY || '-' || NAME
```

Look-up Fields

Service Request Catalog displays certain information, such as contact information, based on the corresponding QBE lists in Service Manager.

All Service Request Catalog users must have complete contact information specified in their Service Manager operator and contact records for the contact look-up to search successfully for a match. The contact record and the operator record must contain the first name, last name, and company information for the user. Service Manager generates a formatted contact ID that may also be used in the contact look-up.

Example: The Service Manager user Jennifer Falcon has this information in her company and operator records:

- Last name = Falcon
- First name = Jennifer
- Company = HP

Her Service Manager generated contact ID is FALCON, JENNIFER.

If you are using the Contact look-up in Service Request Catalog, you can find her by typing one of the following:

- Her first name only
- Her last name only
- Both first and last name
- Her contact ID (in upper case)

Date Time Validation

This version of Service Request Catalog allows you to use the Date/Time Validation mechanism of Service Manager. You can use the `tod()` to define an appropriate expression for a Date/Time user selection.

For example, suppose you want to specify a minimum 5-day delivery date for the delivery of a Catalog Item. To do this, you can configure a "Deliver By:" user selection in the Catalog Item to use the following expression (in the "greater than" field):

```
tod()+ '5 00:00:00'
```

When a user tries to order this Catalog Item, the user must choose a date that is greater than 5 days from the current system time. Otherwise, an warning message will appears, and the user must pick another date.

Update Service Manager Lists

Service Manager provides list information to Service Request Catalog that appear in different sections as the user completes a request. The items in the lists are configured in Service Manager. Therefore, if you want to change the contents of certain lists that appear in Service Request Catalog, you must use the Service Manager Windows or Web client to change the appropriate Service Manager record.

The Service Manager Help Server has all the information and steps to make changes to global lists.

1. Obtain the URL for the Service Manager Help Server.
2. Type or paste it into a browser window.
3. Type "global lists" in the Search area of the Help Server interface to locate all information about these lists.

For Service Request Catalog, you can make changes to these lists:

- Callback Type (to specify the users preferred contact method)
- Interactions Activity Types (such as "Operator update")
- Resolution Code (to categorize the reason for closure)
- Urgency (such as "Critical" or "High")

Note: Service Manager localizes these lists to make sure they render correctly in different locales. If you customize any list value, you must create manual translations of your new values and store them in the appropriate locale folder.

Supporting Other Locales

You may have a global community of users who want to view the service catalog in their preferred language. Service Request Catalog provides complete out-of-box localization for multiple languages. You must complete tasks to configure support for your selected languages.

Service Request Catalog also supports the currency code and symbol for each of these languages. Complete the following tasks to enable more than one locale. The only constraint is that each locale must be supported by Service Manager. Complete the first three tasks with a Service Manager Client that connects to your Service Manager server.

Review Out-of-box Supported Languages

| Language | Language Code |
|----------------------|---------------|
| Arabic | ar_SA |
| Brazilian Portuguese | pt_BR |
| Czech | cs_CZ |
| Dutch | nl_NL |
| English | en_US |
| French | fr_FR |
| German | de_DE |
| Hungarian | hu_HU |
| Italian | it_IT |
| Japanese | ja_JP |
| Korean | ko_KR |
| Polish | pl_PL |
| Russian | ru_RU |
| Simplified Chinese | zh_CN |
| Spanish | es_ES |

Task 1: Configure the Language in Service Manager

The first task is to make the selected language available when the user logs in to Service Request Catalog. Some languages may already be enabled on the Service Manager side. Others may not be

enabled. Follow these steps to verify that your target language is enabled, or complete the steps to enable it.

1. From the Service Manager client, expand the left navigation and click Tailoring > Database Manager.

2. From the Table field, type or select:

language

3. Click the **Search** icon.

Service Manager returns a list of Service Manager language files.

4. Double-click the desired language file.

5. From the Language Identification dialog, click the Search icon.

A list of available languages appears. If the language you want has an active flag set to true, the language is enabled, and your task is complete. If the flag is false, proceed to the next step.

6. Select the language that you want to enable.

7. Check the empty **Active for logins** check box.

8. Click **Save** to update the language record.

9. Click **OK**.

10. Repeat step 1 through step 9 to add more languages.

11. Click **Back** to exit this task.

12. Restart the Service Manager client to verify your changes.

Task 2: Configure the User Profile

The second task is to associate a default language with the individual user in their Service Manager user profile. For users in a multi-lingual environment, one might prefer to search the catalog in French, while another user would be more comfortable with Spanish.

Note: If the user does not exist in the Service Manager system, you can follow these steps to navigate to the Operator Record dialog, but you must create the user before you complete the language association task.

1. Verify the target language is configured in Service Manager by reviewing [Task 1](#) in the Support Other Locales section.
2. In the **System Navigator** pane, click **System Administration > Ongoing Maintenance > Operators** to edit the user's operator record.
3. Click **Search**.

4. Select the user from the **Login Name** or **Full Name** column.
5. Click the **Login Profile** tab.
6. Click the **Language** drop-down list to select the default language for that user.
7. Click **Save**.
8. Click **OK**.
9. Repeat step 2 through step 8 if you have more than one operator record to configure.
10. Click **Back** to exit this task.

Task 3: Localize the Item Record

1. To display any item in the user's preferred language, you must add the specific language attribute to the item record. If you skip this step, the item appears in the default language.
2. In the **System Navigator** pane, click **Service Catalog > Manage Catalog** to locate the record to be localized.
3. Click **Search**. Service Manager displays a list of the item records in the catalog.
4. Scroll to find the target record in the list.
5. Click the record in the list to display all the record information in the Category Definition form.
6. When Service Manager displays the record, right-click anywhere in the form and select **Localize this record** from the context menu.
7. Service Manager displays the *Create Localized Data* wizard.
8. Choose the language from the drop-down list. **Example:** select German.
9. Replace the English **Name**, **Short Description**, and **Long Description** with equivalent values in the language you selected in step 6.
10. Click **Finish**. Service Manager displays a confirmation message above the Category Definition form.
11. Repeat step 3 through step 8 to create localized versions of each catalog record.
12. When you are finished, click **OK**.
13. Click **Back** to exit the task.

Task 4: Configure the Currency

All currencies related to supported locales are available in the Service Request Catalog application. Currency values can appear with a symbol, such as the dollar symbol (\$), or a code, such as USD

(United States Dollars). The default display for the English locale is the currency symbol. The default display for non-English locales is the currency code.

Changing a Currency Code or Symbol

You can change the way currency displays, or you can add a new code or symbol.

Example: You would like to display using a different currency code or symbol, or even add a new currency.

Changing the Default Currency Symbol for Service Request Catalog

The Service Manager server normally provides localized item descriptions and converts the item cost to your locale currency symbol and value without intervention. You can override the currency display by overriding the default value.

1. On the server where you deploy the Service Request Catalog, navigate to the Control Panel > Administrative Tools > Services dialog.
2. Navigate to this folder:

Windows: C:\...\src-9.32\resources\client\
Unix: /opt/.../src-9.32/resources/client/

All localization sub-folders contain translated text for the application.

3. Open the folder for your locale.

Example: Hungarian translations are in this folder:

Windows: C:\...\src-9.32\resources\client\hu\
Unix: /opt/.../src-9.32/resources/client/hu/

4. Open the CurrencySymbol_hu.properties file.
5. Locate the currency code and symbol that you want to display and remove the comment character from that entry.
6. Save and close the file.

Adding a New Currency to Service Manager

If you want to display items in the new locale using the correct currency value and symbol, browse the Service Manager documentation to locate “Displaying currency in the Service Catalog.” Follow the recommendations in this and related topics to make sure the:

- Display currency field is set correctly in the affected operator records
- Currency table contains the correct currency definition record
- Currency conversion rate, established by the daily currency exchange rate, appears in the currency conversion table.

Adding a New Code and Symbol to Service Request Catalog

In this example, we want to add the Korean code and symbol.

Follow these steps on the server where you deploy the Service Request Catalog .war file.

1. On the server where you deploy the Service Request Catalog, navigate to the **Control Panel > Administrative Tools > Services** dialog.

2. Navigate to the following folder:

Windows: C:\...\src-9.32\resources\client\
Unix: /opt/.../src-9.32/resources/client/

Unix: /opt/.../src-9.32/resources/client/

All localization sub-folders contain translated text for the application.

3. The /ko folder contains all property files for the Korean locale. Open this file with a text editor:

Windows: C:\...\src-9.32\resources\client\ko\CurrencySymbol_ko.properties
Unix: /opt/.../src-9.32/resources/client/ko/CurrencySymbol_ko.properties

Unix: /opt/.../src-9.32/resources/client/ko/CurrencySymbol_ko.properties

4. Add the correct currency code and currency symbol to this file.

Example: To add Korean currency, add this value:

KRW=\u20a9

5. Save and close the file.

Task 5: Display The Currency for a User

Activate the target currency. To do this, follow these steps:

1. In the Service Manager Windows client, navigate to **System Administration -> Base system Configuration -> Currencies**.
2. Enter the target currency name (for example, CNY) and then click **Search**.
3. Check the **Active?** checkbox and click then click **OK**.

Add conversion rates. To do this, follow these steps:

1. Navigate to **System Administration-> Base system Configuration -> Conversion Rates**.
2. Search for the target currency Conversion Rates. Add one if not exist. Generally, the Root Currency Code should be USD.

Add the Currency to the User Profile.

1. Log in to Service Manager and enter **operator** in command line.
2. Input a valid SRC username and click **Search** to open the user profile.
3. Click the **Login Profiles** tab and change the **Display Currency** value to the target one.

4. Click **Ok**.
5. Log in to SRC using the username from step 2 and set the language to one consistent with the currency. For example, to test Chinese Yuan, set the language to **Chinese**.
6. Open the catalog item that has a specified cost in SM and verify that the correct currency symbols, converted price and currency format are correct.

Task 6: Configure the Browser for a Locale

To view Service Request Catalog in a preferred language, users should configure their browser settings. The following general guidelines are for Microsoft Internet Explorer and Mozilla Firefox. If you use an unsupported browser, consult the browser documentation.

Microsoft Internet Explorer

1. Click **Tools > Internet Options**.
2. Click the **General** tab.
3. Click **Languages**.
4. Click **Add**.
5. Choose a supported language and click **OK**.
6. Select that language and move it to the top of the preference list.
7. Click **OK**.

Mozilla Firefox

1. Click **Tools > Options**.
2. Click **Content**.
3. In the **Languages** section, click **Choose**.
4. Click **Select a language to add**.
5. Choose a supported language and click **Add**.
6. Select that language and move it to the top of the preference list.
7. Click **OK**.

Authoring Bi-directional Text

Starting from Service Manager 9.32, Service Request Catalog supports right-to-left direction for both Arabic and Hebrew language environments. However, without certain modifications, you may find that some text is displayed left-to-right (LTR) or right-to-left (RTL). The following sections

provide a brief introduction on bidirectional text, and show you how to input bidirectional texts in the Service Manager client.

Understanding Unexpected Text Order

Service Manager stores texts in Unicode encoding. Service Request Catalog displays texts using the Unicode standard, which defines the order in which text should be displayed by specifying a particular base direction.

The base direction influences the order of the display of text of different directions, and the display of directionally-neutral text (i.e., characters or sequences of characters that do not have inherent directionality, as defined in the Unicode Character Standard).

In SRC, if the language is Arabic or Hebrew, the base direction is RTL; otherwise, LTR.

For example, suppose a user writes the following text in English as the name of a catalog item. In an LTR language in SRC, the text is displayed exactly as shown:

```
Printing (North America)
```

However, in a Hebrew or Arabic language environment, where the base direction is RTL, the text displays as follows:

```
(Printing (North America
```

This behavior occurs because a parenthesis is directionally neutral in the Unicode standard and does not have an inherent direction. The Unicode text engine first checks whether a parenthesis can inherit a direction from the surrounding text. If a parenthesis cannot inherit a direction from the surrounding text, it defaults to the base direction. In the example text, the first parenthesis is embedded in a single run of LTR text. Therefore, it adopts the direction of its surrounding text. The second parenthesis, which is at the end of the text, is not surrounded by text of a certain direction. Therefore, the second parenthesis defaults to the RTL base direction.

Thus, the text can be divided into two parts with different directions, wherein the first part is LTR:

```
Printing (North America
```

And, the second part is RTL:

```
(
```

Because “)” displays as “(” in the RTL direction, and the two divided text parts are ordered in the RTL base direction, the text is displayed as shown:

```
(Printing (North America
```

To resolve this issue, the Unicode standard contains two special characters that are used in bi-direction-enabled text.

| Unicode Character | Code | HTML | Name | Description |
|-------------------|------|------|------|-------------|
|-------------------|------|------|------|-------------|

| | | | | |
|-----|--------|------------------------------|-----------------------|---------------------------------------|
| LRM | U+200E | ‎ ‎ ‎ | LEFT-TO-RIGHT MARK | Left-to-right zero-width character |
| RLM | U+200F | ‏ ‏ ‏ | RIGHT-TO-LEFT MARK | Right-to-left zero-width character |

These zero-width invisible characters act as surrounding text. These characters enable directionally-neutral characters (such as parentheses) to inherit the correct direction from the surrounding text instead of defaulting to the base direction.

In the prior example, you can add a left-to-right mark after the second parenthesis to fix the problem as shown in the following example:

```
Printing (North America)LRM
```

Because both parentheses are surrounded by LTR text, the whole text is treated as a single run of LTR text. Therefore, the text is displayed as we want in Hebrew or Arabic environment:

```
Printing (North America)
```

There are other control characters related to bi-directional text in the Unicode standard which are rarely used. For full instruction on the Unicode Bidirectional Algorithm and the related control characters, see the following links:

<http://www.w3.org/International/articles/inline-bidi-markup/uba-basics>

<http://www.unicode.org/L2/L2012/12173-bidi-paren.pdf>

<http://www.unicode.org/reports/tr9/>

Input Bidirectional Text in the Service Manager Client

The text input boxes in the Service Manager clients use Unicode encoding. However, unlike other third-party text editing tools where you can easily insert a left-to-right mark or other Unicode control characters by using a keyboard shortcut or context menu, the Service Manager clients are somewhat limited.

You can use any of the following three methods to input bidirectional text in the Service Manager clients.

Copy from Text Editing Tools (Recommended)

If you are familiar with authoring bidirectional text, you probably have a text editing tool that you often use. You can copy previously-authored bidirectional text from your text editing tool and paste directly into the text input box in the Service Manager client.

Note: Make sure you that you do not the invisible Unicode characters, especially at the start or

the end of the text.

Insert a Single Unicode Control Character

If you already have some text in Service Manager, and you just want to insert several LRM or RLM marks to make that text bidirectional, you can create these marks easily using WordPad in Windows. To do this, use one of the following methods:

Method 1

1. Open a new document in WordPad.
2. Type 200E (200E is the hexadecimal code of LRM) and then press the Alt+X keyboard command. The “200E” should disappear and the LRM is inserted.
3. Press CTRL+A to select all, and then press CTRL+C so you can paste it back into Service Manager.

Method 2

- Open a new document in Wordpad.
- Press and hold the ALT key, and then type 8206 (the decimal code of LRM) by using the numeric keypad. When you release the ALT key, the LRM is inserted.
- Press CTRL+A to select all, and then press CTRL+C so you can paste it back into Service Manager.

Rich Text Input Box in Service Manager Client

There are a few rich text input boxes in Service Manager client, such as the detailed description of Catalog items. Rich text input boxes have two editing modes: “normal” and “source”

In the “normal” mode, you can input bidirectional text as you would in other text input boxes.

In the “source” mode, you must use the HTML escaped form of special Unicode characters. For example, to input a RLM (right-to-left mark), you should input one of the following:

‏

‏

&r1m;

Configure Additional Details for Service Requests

You can configure Service and Support request to display additional details from Service Manager.

Enable Journal Updates

You can enable Journal Updates to appear in SRC for each request to provide additional information to the user on the history of their request. Journaling is a feature of Service Manager that allows additional details or updates to a record. If Journaling is enabled on your Service Manager server, and you want to enable those journal updates for SRC, follow these steps:

1. Open the `applicationContext.properties` file, and specify the following properties as **true**:

| Property | Description |
|--|---|
| <code>src.sm.service.journalUpdatesEnabled=true</code> | A value of true enables the Journal Updates for service. Change the value to false to hide Journal Updates information |
| <code>src.sm.support.journalUpdatesEnabled=true</code> | A value of true enables the Journal Updates for support. Change the value to false to hide Journal Updates information. |

Edit the custom definition label for Journal Updates

By default, the label for Journal Updates is "Request Updates." However, you can modify this label. To do this, edit the `Custom.properties` file, and specify a new label for the `request.update` property.

Require Users to Read Catalog Item Details.

In Service Request Catalog the **Request Now** button allows a user who knows what they are ordering to quickly place a Catalog item in the shopping cart and proceed to check out. However, in some cases, you may wish to force the user to read the details of the Catalog item. To do this, you can set the `enableDirectRequest` parameter for each of the Service and Support sections to false.

```
src.ui.support.enableDirectRequest = false  
src.ui.service.enableDirectRequest = false
```

Note: A value of false means that users must read the Catalog item description. A value of true allows users to use the **Request Now** button to place an item in their shopping cart.

Enable Line Item Status

You can use the following parameter to enable the display of the status for individual line items of a request. If this option is enabled, users can see the status of individual components of a request when they view the status of their request.

```
src.ui.service.enableLineItemStatus=true
```

Configure IPv6

To configure IPv6 for Service Request Catalog (SRC), follow these steps:

1. Enable IPv6 support in Tomcat. To do this, see the Tomcat section of the Configure the Web clients for IPv6 page in the Service Manager Help Center.
2. Enable IPv6 in Service Manager. To do this, see the Configure the Service Manager server for IPv6 page in the Service Manager Help Center.
3. In folder where SRC is deployed, set the `sm.hostname` parameter in the `applicationContext.properties` file to a Full Qualified Domain Name or to a valid IPv6 address

```
sm.hostname=[2620:0:a17:e006:4a8:91c1:c9cb:93dd]
```

4. You can access the SRC application by entering the valid IP address in your browser, as shown in the following example:

```
http://[2620:0:a17:e006:4a8:91c1:c9cb:93dd]:8080/src-9.32
```

Enable FIPS for Service Request Catalog

FIPS (Federal Information Processing Standards) are a set of standards that describe document processing, encryption algorithms and other information technology standards for use within U.S. non-military government agencies and by U.S. government contractors and vendors who work with such agencies.

As of version 9.32, Service Manager supports the FIPS 140-2 approved Advanced Encryption Standard (AES) 256-bit data encryption algorithm. To use FIPS together with SRC you must first enable FIPS mode in Service Manager. For information on how to do this, see the following sections of the *Service Manager Help Center*.

- "FIPS mode"
- "Configure FIPS mode in Service Manager"

Using the FIPS standard in SRC

SRC provides one tool to generate encrypted password. To support the FIPS standard encryption, this tool has been upgraded from SRC 9.32.

Note: For the details of this tool, see the "Create an Encrypted Password for the Service Manager Administrator" in the *SRC 9.32 Interactive Installation Guide*.

By default, the `runme.bat` script generates an encryption key that meets the FIPS standard. More specifically, the `runme.bat` script uses an Advanced Encryption Standard (AES) 256-bit data encryption algorithm. However, to remain compatible with previous versions of Service Request

Catalog, an optional parameter (PBE) is now accepted by the *runme.bat* script, which uses the MD5 and DES encryption of earlier versions of Service Request Catalog. Therefore, the following commands are valid:

- `Runme.bat //uses the AES 256-bit algorithm`
- `Runme.bat PBE //uses the earlier MD5 and DES algorithm`

Common Access Card Introduction

The Common Access Card (CAC) is the United States Department of Defense (DoD) smart card for multi-factor authentication. CACs are issued as standard identification for active-duty military personnel, reserve personnel, civilian employees, non-DoD government employees, state employees of the National Guard and eligible contractor personnel. In addition to its use as an ID card, a CAC is required for access to government buildings and computer networks.

If you have not already configured CACs for Service Manager, you must do so before you can enable CAC for Service Request Catalog. To do this, refer to the following sections of the Service Manager Help Center:

- *Common Access Card (CAC) sign-on*
- *Requirements for CAC sign-on*

You may also want to review the following example in the Service Manager Help Center:

- *Example: Enabling CAC sign-on*

If you have already configured CACs for Service Manager, you can skip ahead to the following section:

["Enable SRC for Common Access Cards" below](#)

Enable SRC for Common Access Cards

To enable Common Access Cards for SRC, follow these steps:

1. Enable CAC in Service Manager. To do this, use the `cacsignon` parameter from a Service Manager command line, or specify it in the `sm.ini` file.

```
Command line: sm -httpPort:13080 -cacsignon:1
```

```
Initialization file: cacsignon:1
```

For more information on this parameter, see the following section of the *Service Manager Help Center*:

- "Parameter: `cacsignon`"
2. Complete the following tasks from the "Example: enabling CAC sign-on" section of the Service Manager Help Server.

- Task 1. Enable required SSL encryption and trusted clients.
- Task 2. Enable CAC sign-on in the SM server.
- Task 5. Connect your web application server to the web server.
- Task 6. Configure your web application server to use SSL.
- Task 7. Configure the web server to use SSL.
- Task 8. Create operator records for CAC users
- Task 9. Configure web browsers to enable web client users to use CAC.
- Task 10. Test your CAC sign-on setup.

Note:

- When you perform "Task 1. Enable required SSL encryption and trusted clients" you will be referred to "Example: Enabling required SSL encryption and trusted clients" section of the Service Manager Help Center. You will need to copy the keystore files generated at Step 9 of this procedure to the WEB-INF folder in your SRC deployment or, if needed, to a folder that you plan to specify in the applicationContext.properties file.
- Tasks 3 and 4, which are missing from the preceding list, refer to the configuration of the Service Manager Web tier and Windows client, and are not a consideration for Service Request Catalog.
- If you choose to use CAC, make sure that you have configured SRC to use SSL and either TSO or LW-SSO (as appropriate for your environment) as described in the *Service Request Catalog Interactive Installation Guide*.

The following tables show a summary of how your property files should be configured after you have enabled CAC:

ApplicationContext.properties

| Configure these properties | Comments |
|--|---|
| <pre>sm.protocol= sm.hostname= sm.port=13443 src.trustStore= src.trustStorePassword= src.keyStore= src.keyStorePassword=</pre> | <p>Enables two-way SSL between SRC and Service Manager.</p> <div data-bbox="716 1423 1344 1564" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The keys are generated when you complete "Task 1. Enable required SSL encryption and trusted clients." See the previous note for details.</p> </div> <pre>sm.protocol=https sm.hostname=[FQDN] sm.port=13443 src.trustStore=C:\\client\\cacerts src.trustStorePassword=changeit src.keyStore=C:\\client\\src.keystore src.keyStorePassword=clientkeystore</pre> |

ApplicationContext.properties, continued

| | |
|--|---|
| src.security.mode=cac src.security.cac=true | Enables CAC. |
| src.security.secureLogin=true src.security.sslPort=8443 | Enables SSL Connection between SRC and the end user's client browser. |

cacConfiguration.properties

| Configure these properties | Comments |
|--------------------------------------|---|
| certificateFieldExtractDN=Subject.CN | This parameter is used to get the field in the certificate that stores the login name. If the DN is from one of the fields in the Subject, set this parameter using the following format: Subject.CN Subject.CN means extracting the DN from the Common Name field in Subject. |
| validationStrategy=1 | Client Certificate Verification Sequence is as follows: <ol style="list-style-type: none"> 1. Check the revocation status using local CRL. 2. Check revocation status using an online CRL. 3. Check revocation status using OCSP. 4. Check if the status is expired. 5. Check if the certificate type is smart card. |
| RootCertPath= | Specifies file path to the issuer CA certificate. This is a mandatory property. Example: RootCertPath=C:/client/CAC/hp_ca_for_cac.cer |
| CRLRefreshScheduler= | Specifies a schedule in minutes for CRL refresh, if CRL check is enabled. Example: CRLRefreshScheduler=1440 |

cacConfiguration.properties, continued

| | |
|-------------------|---|
| CRLDownloadURL= | The URL that identifies the location of the CRL. If it is not specified, the CRL Distribution Point from the certificate being validated will be used. This behavior will slow performance dramatically. CRLDownloadURL=http://onsitecrl.example.com/LatestCRL.crl |
| CRLStoreLocation= | Specifies the CRL file path. |
| OCSPResponderURL= | The URL that identifies the location of the OCSP responder. By default, the location of the OCSP responder is determined implicitly from the certificate being validated. This property is used when the Authority Information Access extension (defined in RFC 3280) is absent from the certificate or when it requires overriding. OCSPResponderURL=http://ocsp.example.net:80 |

Auto-Suggest/Type Ahead

Service Request Catalog supports an auto-suggest/type-ahead feature to suggest the top ten matching Catalog items when users enter information in the Search Box. You can enable or disable this feature by modifying the `src.ui.enableAutoComplete` property to true or false in the `applicationsContext` file, as shown in the following example:

```
src.ui.enableAutoComplete=true
```

Note: A value of false disables the type ahead feature.

Appendix A: Launcher Examples

The following pages demonstrate various ways you can use the Launcher. All configuration for the Launcher is performed by editing the manifest.xml file. Some examples are included in the manifest.xml file, but are commented out by default.

Example: Add Bomgar Live Chat

Service Request Catalog supports integration only with a Bomgar™ chat solution. As such, you must independently purchase a Bomgar™ chat solution for your organization. For more information about the Bomgar API and API documentation, visit www.bomgar.com.

A live chat interface can enhance your catalog support strategy. If a user has a question about an item or a request, live chat support provides immediate assistance. You can embed a live chat link at the top of the Dashboard, Services, or Support landing pages. The link launches a chat window in a new tab or window, depending on the user's browser settings. To create a live chat link from the Dashboard, Services, or Support pages, follow these steps.

Tip: It is always a good idea to make a backup of any system files before you make changes to them. If an error occurs later, you can always return the file to its original state.

1. Open this file with a text editor.

```
C:\...\src-9.32\resources\client\en\Custom.properties
```

2. Locate the chat example in the file.
3. If necessary, change `chat_launcher_title`, and the `chat_launcher_description` display text. Note that:

- The identifier can be user-defined; however, when you edit the manifest.xml file, the identifiers have to match.
- The value for each identifier is exactly what will display in the user interface.

4. Save and the file but keep it open for reference as you configure the next file.
5. Open this file with a text editor.

```
C:\...\src-9.32\secure\configuration\manifest.xml
```

Use the following lines as a reference for the remaining steps.

```
<launcher launcherId="Chat_Launcher" showLogoBox="true"
showLogoDropShadow="true">
  <logo>chat_image.png</logo>
  <titleKey>chat_launcher_title</titleKey>
  <descriptionKey>chat_launcher_description</descriptionKey>
  <launcherType>APP</launcherType>
  <action>
    <labelKey></labelKey>
    <link><![CDATA[http://bomgar_server_name/api/start_session.ns]]></lin
k>
    <method>GET</method>
  </action>
</launcher>
```

6. Locate the `<launchers></launchers>` section, which may contain one or more `<launcher></launcher>` sub-sections that describe the attributes of third party applications that you want to display on the Dashboard.
7. Copy the Live Chat Launcher template section and insert it as a new `<launcher></launcher>` section.
8. Omit the `<fields></fields>` section. It is not relevant for a live chat link.
9. The `<labelKey>` section is not essential. You do not need to insert a value into this tag. The user will click the chat icon to launch the chat window.
10. Verify the `<link>` element. This link must pass the correct parameter values to connect to the chat server. For more information about the correct parameters in the link, consult the Bomgar documentation.
11. If the application is running, log out and log in again to confirm that the third party application link appears in the dashboard.

Example: Cloud Service Automation Launcher

To create a Cloud Service Automation launcher from the Dashboard, Services, or Support pages, follow these steps.

Tip: It is always a good idea to make a backup of any system files before you make changes to them. If an error occurs later, you can always return the file to its original state.

1. Open this file with a text editor.

```
C:\...\src-9.32\resources\client\en\Custom.properties
```

2. Locate the chat example in the file.

3. Change `chat_launcher_title` and the `chat_launcher_description` display text to `csa_launcher_title` and to `csa_launcher_description` respectively. When you do this, note the following:
 - The identifier can be user-defined; however, when you edit the `manifest.xml` file, the identifiers must match.
 - The value for each identifier is exactly what will display in the user interface.
4. Save and the file but keep it open for reference as you configure the next file.
5. Open this file with a text editor.

C:\...\src-9.32\secure\configuration\manifest.xml

Use the following lines as a reference for the remaining steps.

```
<launcher launcherId="CSA_Launcher" showLogoBox="true" showLogoDropShadow="true">
  <logo>CSA.png</logo>
  <titleKey>CSA_Launcher_title</titleKey>
  <descriptionKey>CSA_Launcher_description</descriptionKey>
  <launcherType>APP</launcherType>
  <action>
    <labelKey></labelKey>
    <link>
      <!-- Specify the IP Address of the CSA Launcher here -->
      <![CDATA[http://<IPAddressCSA>:<port>/csa/Login]]>
    </link>
    <method>GET</method>
  </action>
</launcher>
```

6. Locate the `<launchers></launchers>` section, which contains one or more `<launcher></launcher>` sub-sections that describe the attributes of third party applications that you want to display on the Dashboard.
7. Copy the Live CSA Launcher template section and insert it as a new `<launcher></launcher>` section.
8. Omit the `<fields></fields>` section.
9. The `<labelKey>` section is not essential. You do not need to insert a value into this tag. The user will click the CSA icon to navigate to your organization's CSA Web site.
10. Verify the `<link>` element. This link must specify the correct address to CSA server.
11. If the application is running, log out and log in again to confirm that the third party application link appears in the dashboard.

Example: Get

The following example shows a launcher configuration that specifies a Launcher item to trigger a Google search.

Note: POST HTTP Requests are not supported due to issues with security that occur when executed across multiple Web application servers. However, if the target of the POST request is on the same Web application server instance as SRC, these security issues are not present.

```
<launcher displayOrder="2"
  launcherId="getLauncher"
  showLogoBox="true"
  showLogoDropShadow="true"
  >
  <logo>test2_app.png</logo>
  <titleKey>get_launcher_title</titleKey>
  <descriptionKey>get_launcher_description</descriptionKey>
  <launcherType>APP</launcherType>
  <fields>
    <field fieldId="q">
      <labelKey>get_search</labelKey>
      <type>TEXT</type>
      <defaultValueKey>get_search_value</defaultValueKey>
    </field>
  </fields>
  <action>
  <labelKey>get_action_label</labelKey>
  <link>
    <![CDATA[http://www.google.com/search]]>
  </link>
  <method>GET</method>
  </action>
</launcher>
```

Appendix B: Service Catalog in Service Manager

Dynamic User Options in Catalog Items

You can configure SRC to use dynamic user options when a user is making a request. Dynamic user options provide a mechanism by which an SRC administrator can gather additional information about a Catalog item from a user based on some criteria the user previously entered. For example, users might need to have a request fulfilled within a certain time frame. In this case, the SRC administrator can implement a check box (for example, one that uses the mandatory condition) that asks users whether they have a such a requirement. If the user does select the check box, another text field may appear (one which uses the visible condition), where the user can enter in such a requirement, specify the date, and so on.

Create a Dynamic User Option

Notes:

- The later steps (after Step 6) in this guideline may vary somewhat, depending on which rules and conditions you have selected. For example, if you select a Pick List, the screen on which you specify the Mandatory and Visible conditions appears after the screen on which you specify the values of the Pick List. However, the functionality remains the same.
- The fields condition and query fields leverage Service Manager's Dynamic View Dependency (DVD) mechanism. Through this mechanism, you can enter an expression to evaluate some value, and determine whether the Mandatory or Visible condition is met. For example, suppose you have a check box (CheckBox1). Depending on whether that check box is selected, you want a text box to appear, so that a user can add additional information. To do this, you would set the Visible Condition to an expression that evaluates the selected status of the check box. For more information on how to write these conditions and their behaviors, see "[Notes on Expressions in Dynamic Fields](#)" on the facing page and see the Dynamic View Dependencies entry in the Service Manager Help Center.

To specify a mandatory or visible condition on a dynamic field, use the following guideline:

1. In the Service Manager client, navigate **Service Catalog > Manage Catalog**.
2. Enter the name of the Catalog item you wish to modify and the click **Search**.
3. Select the Catalog item from the list.

Note: You must select an item of the "item" type. You cannot select a bundle or category.

4. In the **Catalog Item Definition**, click **User Selections**.
5. Click **Add User Selection**.
6. Specify a Name, Label, and the Display Type and then click **Next**.
7. Specify the Type (if present), Mandatory Condition, and Visible Condition as required by your business needs.
8. Click **Next**. If you have specified the Display Type as Text and the Type as String, you will be brought to the Validation Rule screen. Here, you may specify a no validation or a validation based on a query run on some table ("Record in Table").

Notes on Expressions in Dynamic Fields

Before you write expressions for the Dynamic fields in a Catalog item, it is important that you note the following limitations and behaviors.

Supported Operators

The Condition and Query rules follow the Service Manager standard, except that only the following operators are supported:

- **&** (the AND operator)
- **|** (the OR operator)
- **!=** (NOT EQUAL TO operator)

Mandatory Conditions

You can only specify a Mandatory condition for the Text, Multi-Text, and Pick List Display Type. The Mandatory condition can be specified as any of the following four types:

- Hard coded: **true, false**
- Variable condition: **name in \$lo.operator="falcon"**

Note: Only global variables are supported. The global variable list is generated only when a user logs in to SRC.

- Dependent on other widget in the user selection: **[widget_name] ="falcon"**

In this example, *widget_name* is name of another widget in the user selection. If the dependent widget does not exist, an error message will appear.

- Dependent on the fields of a catalog item outside of the user selection:
`$requested.for="FALCON, JENNIFER"`

Note: Only the `$requested.for`, `$requested.for.dept`, `$delivery`, and `$service.sla` variables are supported.

Visible Conditions

You can specify a Visible condition for all Display Types.

Mandatory and Query Condition Sample

| Type: | Possible values: | Notes: |
|--|---|---|
| Hard code condition: | true false | |
| Variable condition: | company in \$lo.operator="advantage" | The company on the right is an element of the object. Only Global variables are supported |
| Dependent on other widget: | [<i>widget name</i>]="specificValue" | [<i>widget name</i>] is a placeholder for the name of another widget in the user selection. |
| Dependent on the fields of catalog item outside of the user selection: | <code>\$requested.for="FALCON, JENNIFER"</code> | contact.name is a column of the selected table. Only the following values are supported i SRC: <code>\$requested.for</code> <code>\$Requested.type</code> <code>\$requested.for.dept</code> <code>\$delivery</code> <code>\$service.sla</code> |

Query Fields

The Query field only appears when you have specified the following attributes for your dynamic user option field:

- Display Type: Text
- Type: String
- Validation Rule: Record in Table

If your Dynamic field meets these criteria, you must then specify the following:

- Table Name: *some_table*
- Field Name: *some_field*
- Query Condition:
- Group by Fields: This field is not supported by SRC.

Note: If you click the **Edit** button, the advanced query window appears.

Query Sample

| Type: | Possible values: | Notes: |
|----------------------------|---|--|
| Hard code condition: | true false contact.name="FALCON, JENNIFER" | contact.name is a column of the selected table. |
| Variable condition: | display.currency.code=\$G.my.currency company = company in \$lo.operator | The company on the left is a column of the selected table. The company on the right is an element of the object. |
| Dependent on other widget: | name =[widget name] | [widget name] is a placeholder for the name of another widget in the user selection. name is a column of the selected table. |

Query Sample, continued

| | | |
|--|------------------------------|---|
| Dependent on the fields of catalog item outside of the user selection: | contact.name=\$requested.for | contact.name is a column of the selected table. Only the following values are supported in SRC: \$requested.for \$Requested.type \$requested.for.dept \$delivery \$service.sla |
|--|------------------------------|---|

Other Behaviors

For any set custom Dynamic user option fields, you can very easily create a complicated set of rules. Therefore, it is important that you recognize the following behaviors.

- By default, the Mandatory condition is false. If the condition format is not correct, or any JavaScript function call does not exist, then the user option is not mandatory.
- By default, the Visible condition is true. If the condition format is not correct, or any JavaScript function call does not exist, then the user option is visible.
- If a Mandatory condition for a user option is true, but the visible condition is false, a request for that item can still be submitted successfully.
- If the condition format is not correct, or any JavaScript function call does not exist, an empty list is returned.
- If a user option has dependencies on two other user selected fields and the first field has no value, whether second field returns all or empty is decided by the condition expression of the second field. String, number, check box, and date may all behave differently.
- A field that is dependent on another field is cleared automatically if the second field is cleared.
- The Mandatory designation only exists in the SRC UI. In Service Manager, an error message will display when a user does not set a value for mandatory field.
- The Visible behavior only exists in the SRC UI.
- In the case of a circular query condition (where A depends on B, and B depends on A), an empty set is returned.
- In the case of a circular Mandatory condition (where A depends on B, and B depends on A), an empty set is returned.

- In the case of a circular Visible condition (where A depends on B, and B depends on A), an empty set is returned.
- You cannot delete a user selection from Service Manager if another field or user selection depends on it.
- The Query Condition does not support the currency exchange rate. This is due to a limitation in Service Manager, which supports dollar values as numbers, instead of currency. To work around this issue, you can display the currency as a number together with a list that specifies the actual currency (for example, USD or RMB), and the approving manager's name based on your business requirements (if the request is over 500 USD, for example).

Service Request Catalog HTML Tag Support

Service Manager supports a variety of HTML tags as part of the Service Catalog. However, Service Request Catalog supports only a subset of the HTML tags. Therefore, we recommend that you refine the Service Catalog in Service Manager to ensure that catalog items are rendered as expected in Service Request Catalog. The following table shows the tags that are supported in SRC.

Note: HTML tags that are not listed in the following table are not supported

| TAG | DESCRIPTION |
|----------|---|
| Anchor | <p>The <a> tag creates a hypertext link, and supports the following attributes:</p> <ul style="list-style-type: none"> • target: This attribute is ignored by SRC. By default, all URLs are opened in a new browser window to prevent SRC from being unloaded. • href: Specifies a URL. The URL can be either absolute or relative to the location of main.jsp of SRC. An example of an absolute reference to a URL is http://www.hp.com; an example of a relative reference is index.html. <p>Note: You must prefix a URL with http(s):// to designate it as an absolute URL. Otherwise, SRC treats them as relative URLs.</p> |
| Bold | The tag renders text as bold. |
| Break | The tag creates a line break in the text. |
| Emphasis | The tag displays the tagged text in italics. An italic typeface must be available for the font used. |

| TAG | DESCRIPTION |
|----------|---|
| Font tag | <p>The tag specifies a font or list of fonts to display the text. The font tag supports the following attributes:</p> <ul style="list-style-type: none"> • color: Only hexadecimal color (#FFFFFF) values are supported. • face: Specifies the name of the font to use. You can specify a list of comma-delimited font names, in which case SRC selects the first available font. If the specified font is not installed on the local computer, SRC selects a substitute font. • size: Specifies the size of the font. The valid value is an integer between 1 and 7; or a relative increase from base font size 3 (for example, "+1" means 4 and "-2" means 1). All relative sizes also belong to the scale of 1 to 7. • letterspacing: Specifies the tracking (manual kerning) in pixels to be applied to the right of each character. • kerning: Specifies whether kerning is enabled or disabled. A non-zero value enables kerning, while zero disables it. |
| Image | <p>The tag lets you embed external image files (JPEG, GIF, PNG) and SWF movies inside text. The img tag supports the following attributes:</p> <p>src: Specifies the URL to an image or SWF file. This attribute is required; all other attributes are optional. The images or SWF movies do not show until they are downloaded completely.</p> <p>Note: Embedded SWF movies cannot access the network unless there's no sandbox security violation.</p> <p>width: The width of the image being inserted, in pixels.</p> <p>height: The height of the image being inserted, in pixels.</p> |
| Italic | <p>The <i> tag displays the tagged text in italics. An italic typeface must be available for the font used.</p> |

| TAG | DESCRIPTION |
|-----------------|--|
| List | <p>The tag (ordered list) and tag (unordered list) are rendered in an identical manner except that ordered list items are numbered and unordered list items are not. Both types of lists are made up of sequences of list items defined by the tag (list item). SRC does not support any attributes of the list tags. Lists can be nested as shown in the following examples:</p> <pre data-bbox="428 485 1154 877"> ... level one, number one... ... level two, number one... ... level two, number two... ... level three, number one... </pre> |
| Strike | The <s> and <strike> tags render strike-through style text. |
| Strong | The tag renders text as bold. |
| Paragraph | <p>The <p> tag creates a new paragraph. The <p> tag supports the following attributes:</p> <p>align: Specifies alignment of text within the paragraph; valid values are left, right, justify, and center.</p> |
| Text format tag | <p>The <textformat> tag is not a standard HTML tag but a customized tag in SRC. It lets you use a subset of paragraph formatting attributes including the following:</p> <p>indent: Specifies the indentation from the left margin to the first character in the paragraph. Both positive and negative numbers are acceptable.</p> <p>blockindent: Specifies the indentation applied to all lines of the paragraph.</p> <p>leftmargin: Specifies the left margin of the paragraph, in points.</p> <p>rightmargin: Specifies the right margin of the paragraph, in points.</p> <p>leading: Specifies the leading (line height) measured in pixels between a line's ascent and the previous line's descent.</p> |
| Underline tag | The <u> tag underlines the tagged text. |

Refining the Service Catalog

Remove Empty Categories

Empty Service Manager categories (that contain no items) will cause a performance degradation when Service Request Catalog attempts to populate existing categories with child items. Service Catalog administrators should verify that the Service Manager Service Catalog contains no empty categories.

Add Images for Catalog Items

You will obtain optimum results if all catalog images are the same size and in a similar format. Catalog items display as a “thumbnail” image. Follow these basic rules for attaching a thumbnail image to a catalog item:

- The recommended size for an image is 196x140 pixels. Use a good image utility to crop or resize your images to a consistent size.
- The default background color for a smaller image is white. For consistency, consider adding an appropriate background to maintain the same image dimensions for all items.
- The recommended file type is .jpg or .png. Using other formats may produce unpredictable results.

Separate Catalog Items for Different Companies in Multi-Company Mode

These steps should be completed by an experienced Service Manager administrator. An administrator can configure Service Manager to support Multi-Company mode to filter the information that Service Request Catalog users see when making requests. In Multi-Company mode, Service Request Catalog users see only request items for their own company.

An administrator can also enable the Mandanten feature with Multi-Company mode to store company catalog data in its own secure database.

Use the Service Manager Windows Client to complete these steps.

1. To enable Multi-Company mode in the System Information Company record, click **System Administration > Base System Configuration > Miscellaneous > System Information Record > General**
2. Verify that you have a complete company record with Multi-Company enabled. Click **Tailoring > Database Manager > Table (*company*) > Search > *company* > Search > Show Company in Multi-Company Lists**

3. Add a new “company” field of type character type to the svcCatalog tables and to the svcDisplay table.
4. Enable Mandanten security by specifying a filtering condition for the svcCatalog and svcDisplay tables. Use the Mandanten Field Restriction form to specify the company value (that you specified in step 3) in the Mandant Field Name field. Create two records: one for File Name svcCatalog and one for File Name svcDisplay. Click **System Administration > Ongoing Maintenance > Mandanten > Mandanten Field Restrictions**
5. Create a Mandanten security group. Make sure the Security ID is a an upper case value. For example, SRC is an upper case value. Click **System Administration > Ongoing Maintenance> Mandanten > Mandanten Security Groups**
6. Type the “company” value in the first row of the Include Value List. Leave the Exclude Value List empty.
7. Click **Add** to save the security group record.
8. If you enable Mandanten security, add the name of the Mandanten security group to the Security Groups tab on the operator record of each Service Request Catalog user.
9. Assign the company to each category, sub-category, package, and item. This property does not cascade.
Note: If you are an advanced user, you can use these RAD Debugger commands to accomplish this task:
 - `d company in $L.file`
 - `x company in $L.file="company name"`
10. Click **Save**.
11. If you have more than one company defined in Service Manager, make sure each company has categories, subcategories, and producttypes assigned associated with service catalog.

For the detailed steps to complete each Service Manager task, see the Service Manager Help server.

Working with the Support Catalog

Service Manager self-service users can request support only through the Service Request Catalog. It is not available from the Enterprise Self Service (ESS) portal. The Service Manager administrator must configure the support categories and category items that the user will select in the Service Request Catalog interface.

For an introduction to this process, you can begin by starting a Service Manager Windows client session and expand the left navigation pane. Follow the Example suggestions to create a sample

category and item. Your actual Service Catalog and item configuration will be dependant on your particular organization's requirements.

For more information, see the Service Catalog section of the Service Manager Help system.

Search for a Support Catalog Category or Item

1. Click **Service Catalog > Administration > Manage Catalog**.
2. Type any available information into the appropriate fields.
3. Click the **Support Item** checkbox to narrow the search scope.
4. Click **Search**. Service Manager displays a list of all support categories and items in the catalog.

Create a Support Catalog Category

Create Support Catalog categories that contain only sub-categories and items that are related to support.

1. Click **Service Catalog > Administration > Manage Catalog**.
2. Click **Add New Category**. Service Manager displays a wizard that prompts you for the information about a new category.
3. Type a **Category Name**.
Example: Type PCHelp
4. Type a **Display Name** that will appear in the interface as a category that you can search for items.
Example: Type PC Help
5. Type a **Description** for the new category.
Example: Type This is a new category.
6. Click **Next**.
7. Make the new category a **Top Level** category, or you can click the drop-down list to make it a **Subcategory** of a selected Top Level category.
Example: Choose **Top Level**.
8. Specify whether the new category should contain Subcategories, or Items and/or Bundles. Support categories can contain only items, not bundles.
Example: Choose **Items and/or Bundles** to allow support items to be added to the category.

9. Click **Next**.
10. When Service Manager displays a message that it added a new Service Catalog category, click **OK**. Service Manager displays the Category Definition interface where you can refine the category by choosing an Owner or even attaching an image.
11. To create a Support category, select the **Non-cart category** checkbox.
12. Select the **Support category** checkbox.
13. Click the **Access** tab.
14. Click the first empty row in the **Available to** table.
15. Select the Service Catalog group capability word that you want to associate with this new category.

Example: svcCatEmployeeRequester.
16. Add as many capability word groups to the list as necessary.
17. Click **Save**.

Create a Support Catalog Item

You can create individual support items for the Support Catalog, but you cannot create bundles or packages of support items.

1. Click **Service Catalog > Administration > Manage Catalog**.
2. Click **Add New Support Catalog Item**. Service Manager displays a wizard that prompts you for the information about a new item.
3. Type an **Item Name**.

Example: Type Repair Software.

4. Select the checkbox to **Restrict request to a single item**. Support catalog items should be ordered only once.
5. Type a **Display Name** that will appear in the interface as a selectable item.

Example: Type Repair Software

6. Optional. Select the checkbox for an **Information-only** item if the item is to be read-only in the catalog and not to be requested.

Example: A temporarily unavailable item might appear in the catalog but not be selectable. Information-only items can also be instructions that appear in the catalog with an image,

description, or even attachments, but they do not have an associated price and they cannot be requested.

7. Type a brief **Description** of the new item.

Example: Type Repair email software.

8. Type a **Detailed Description** of the new item.

Example: Type Reinstall or upgrade.

9. Type the **Cost** of the item.

Example: Type 100.

Tip: For support items, you may need to gather labor and material costs to accurately reflect the true cost of the support item.

10. Select the Currency unit from the drop-down list.

Example: Use the default currency that appears.

11. Click **Next**.

12. Click the lookup icon to display available categories.

13. Choose a **Category** from the list to specify what type of record Service Manager should create when a user selects this item. Choose:

- Complaint
- Incident
- Request for change
- Request for information

Example: Choose **Request for change**.

14. Select a category **Subarea**:

- New service
- Upgrade/new release

Example: Choose **Upgrade/new release**.

Service Manager displays the New Support Catalog Item Wizard.

15. If necessary, change the Category, Area, or Subarea values. Click **Next**.

16. Choose a Connector from the drop-down list. The connector is the type of record that Service

Manager will create. If you do not select a connector, Service Manager creates a new interaction with the Category, Area, and Subarea values that you specified for this Support Catalog item.

Example: Leave this blank.

17. Choose a **Category** from the drop-down list.

Example: Choose **Applications**.

18. Click **Next**.
19. When Service Manager displays a message that it added a new Service Catalog item, click **OK**.
20. Service Manager displays the Catalog Item Definition interface where you can add more information about the item, add an attachment or an image.
21. If you make changes, click **Save**

Delete a Support Item

1. Click **Service Catalog > Administration > Manage Catalog**.
2. Type any available information into the appropriate fields.
3. Click the **Support Item** checkbox to narrow the search scope.
4. Click **Search**. Service Manager displays a list of all categories and items in the catalog.
5. Browse the list to find the category or item that you want to delete.

Example: Repair software

6. Click **Delete**.

Appendix C: XML Tag Reference

| For This XML Tag | Insert This Value |
|---|---|
| <configurations> </configurations> | Define one or more <configuration></configuration> sections |
| <configuration name="default"> </configuration> | Define all the necessary values to describe company customization values. The company name that you configure should match the default company name that is in the end user operator record. |
| <companyDisplayNameKey> </companyDisplayNameKey> | <p>company_name is defined in ...\\resources\\client\\en\\Custom.properties</p> <p>Custom.properties Example:company_name = XYZ</p> <p>Example: <companyDisplayNameKey>company_name </companyDisplayNameKey></p> |
| <resources></resources> | Define one or more <resource></resource> sections within this parameter |
| <resource></resource> | <p>panel.swf should be in this folder: ...\\secure\\configuration\\default\\</p> <p>Example:...\\secure\\configuration\\default\\panel.swf</p> <p>panel.swf contains your custom color selections.</p> <p>Example:<resource>panel.swf</resource></p> |
| <styles></styles> | Define one or more <style><style> sections. |

| For This XML Tag | Insert This Value |
|--|--|
| <style></style> | <p>styles.css should be in this folder ...\\secure\\configuration\\default\\ Example:...\\secure\\configuration\\default\\styles.swf</p> <p>styles.xml contains your custom display settings. Example:<style>styles.swf</style></p> |
| <launchersGroup> </launchersGroup> | Define the launchers that will appear on each landing page. |
| <launchers displayOn="Landingpage" showIconOnly="false" showButtons="true"> </launchers> | <p>Identify the third party launchers that will appear on the Dashboard and other landing pages.</p> <p>displayOn=DASHBOARD SERVICE SUPPORT showIconOnly= "true" "false" showButtons="true" "false"</p> |
| <launcher displayOrder="n" launcherId="id" showLogoBox="true" showLogoDropShadow="true"> | <p>You can show the logo as an icon and apply a drop shadow to the icon.</p> <p>displayOrder="n" - order of appearance (left-to-right and top-to-bottom) launcherid="id"- a unique user-defined value for each launcher. showLogoBox="true" "false" showLogoDropShadow="true" "false"</p> |
| <logo></logo> | <p>Your launcher logo file should be in this folder: ...\\secure\\configuration\\default\\applications</p> <p>The optimum size is 83x83px and the resolution should be 72 dpi. Example:...\\secure\\configuration\\default\\applications\\abc.png</p> <p>The file format must be .png, .gif, or .jpg Example:<logo>abc.png</logo></p> |

| For This XML Tag | Insert This Value | | | | |
|---|---|----------|---------------|----------------|--|
| <pre><titleKey></titleKey></pre> | <p>titleKey value from the Custom.properties file.</p> <p>Example: <titleKey>launcherName</titleKey></p> <p>The inserted ID value cannot contain blanks. Use underscore characters as separators if necessary. The display value can contain blank characters.</p> <p>Custom.properties Example:</p> <table border="0"> <thead> <tr> <th data-bbox="722 500 877 526">Id Value</th> <th data-bbox="905 500 1079 526">Display Value</th> </tr> </thead> <tbody> <tr> <td data-bbox="722 553 877 579">launcherName =</td> <td data-bbox="905 553 1079 579">XYZ Company</td> </tr> </tbody> </table> | Id Value | Display Value | launcherName = | XYZ Company |
| Id Value | Display Value | | | | |
| launcherName = | XYZ Company | | | | |
| <pre><descriptionKey> </descriptionKey></pre> | <p>descriptionKey value from the Custom.properties file.</p> <p>Example: <descriptionKey>launcherDesc</descriptionKey></p> <p>The inserted ID value cannot contain blanks. Use underscore characters as separators if necessary. The display value can contain blank characters.</p> <p>Custom.properties Example:</p> <table border="0"> <thead> <tr> <th data-bbox="722 862 877 888">Id Value</th> <th data-bbox="905 862 1079 888">Display Value</th> </tr> </thead> <tbody> <tr> <td data-bbox="722 915 877 941">launcherDesc =</td> <td data-bbox="905 915 1619 941">The XYZ Company gives you information about XYZ widgets!</td> </tr> </tbody> </table> | Id Value | Display Value | launcherDesc = | The XYZ Company gives you information about XYZ widgets! |
| Id Value | Display Value | | | | |
| launcherDesc = | The XYZ Company gives you information about XYZ widgets! | | | | |
| <pre><launcherType> </launcherType></pre> | <p>Specify the type of third party link:</p> <ul style="list-style-type: none"> • APP to link to an application, like Wikipedia. • MEDIA to open a video window. <p>The link type value must be upper case.</p> <p>Example: <launcherType>MEDIA</launcherType></p> | | | | |
| <pre><fields> </fields></pre> | <p>Optional: Define one or more individual fields to appear in a popup window.</p> | | | | |

| For This XML Tag | Insert This Value | | | | | | | | |
|---|---|-----------------|----------------------|-------------------------|--|----|--|---|--|
| <pre><field fieldID="ID"></pre> | <p>Define a field that will appear in a popup window</p> <p>fieldId is a unique user-defined value for each field.</p> <p>Example: <field fieldId="GetName"></p> | | | | | | | | |
| <pre><labelKey> </labelKey></pre> | <p>labelKey value from the Custom.properties file.</p> <p>Typically, this is an informational label or text string that might precede a text box or just communicate information.</p> <p>Example: <labelKey>textLabel</labelKey></p> <p>Custom.properties Example:</p> <table border="0"> <tr> <td>Id Value</td> <td>Display Value</td> </tr> <tr> <td>textLabel = Select One:</td> <td></td> </tr> <tr> <td>or</td> <td></td> </tr> <tr> <td>textLabel = You are about to open a new window.</td> <td></td> </tr> </table> | Id Value | Display Value | textLabel = Select One: | | or | | textLabel = You are about to open a new window. | |
| Id Value | Display Value | | | | | | | | |
| textLabel = Select One: | | | | | | | | | |
| or | | | | | | | | | |
| textLabel = You are about to open a new window. | | | | | | | | | |

| For This XML Tag | Insert This Value |
|---|--|
| <code><type></type></code> | <p>Type of field. Choose one:</p> <ul style="list-style-type: none"> • TEXT to display any text • TEXTAREA to gather user information • DROPDOWN for list information • HIDDEN for passing information, like a token <p>Note: Make sure you type these values in upper case.</p> <p>Examples:</p> <pre><type>TEXT</type> <type>TEXTAREA</type> <type>DROPDOWN</type> <type>HIDDEN</type></pre> |
| <code><defaultValue> </defaultValue></code> | <p>Specify a constant value to display, such as a list item or other value that you want to pre-populate with static content.</p> <p>Examples:<code><defaultValue>Welcome!</defaultValue></code></p> |
| <code><options> </options></code> | <p>Insert individual <code><option></option></code> values to define values for a dropdown list.</p> |

| For This XML Tag | Insert This Value | | | | | | |
|-----------------------|---|----------|---------------|-----------|---------|-----------|-------|
| <option></option> | <p>For each <option></option> tag set, insert <labelKey></labelKey> and <value></value> tags and related values.</p> <p>Example:</p> <pre><option> <labelKey></labelKey> <value></value> </option></pre> | | | | | | |
| <labelKey></labelKey> | <p>labelKey value from the Custom.properties file. In an option list, specify the key name of a single list value that you want to display in the user interface.</p> <p>Example:<labelKey>dropdown_1</labelKey></p> <p>Custom.properties Example:</p> <table border="0"> <thead> <tr> <th data-bbox="724 841 850 867">Id Value</th> <th data-bbox="867 841 1041 867">Display Value</th> </tr> </thead> <tbody> <tr> <td data-bbox="724 893 871 919">dropdown1</td> <td data-bbox="867 893 947 919">= Black</td> </tr> <tr> <td data-bbox="724 945 871 971">dropdown2</td> <td data-bbox="867 945 932 971">= Red</td> </tr> </tbody> </table> | Id Value | Display Value | dropdown1 | = Black | dropdown2 | = Red |
| Id Value | Display Value | | | | | | |
| dropdown1 | = Black | | | | | | |
| dropdown2 | = Red | | | | | | |
| <value></value> | <p>This value can be the same as you defined in the Custom.properties file, or might be a different value to be passed to the third party application server.</p> <p>Example:</p> <pre><value>color</value></pre> | | | | | | |
| <action></action> | <p>The action describes how data will be passed to the third party application.</p> | | | | | | |
| <link></link> | <p>Insert the URL for the link in a CDATA tag.</p> <p>Example:<![CDATA[www.google.com]]></p> <p>Example:<link><![CDATA[http://www.XYZ.com]]></link></p> | | | | | | |

| For This XML Tag | Insert This Value |
|-------------------|--|
| <method></method> | <p>Specify the appropriate action:</p> <ul style="list-style-type: none"> • GET enables form data to be encoded by the browser into a URL. • POST enables form data to be encapsulated in a message. <p>Note: If the Post request is to access a different domain (server URL), you must confirm that there is a cross-domain policy file configured at the root directory of the remote domain. The cross domain policy file grants access to the sender passing transactional data to that server.</p> <p>Example: Successful connection to a third-party application in a remote domain, like www.XYZ.com requires a crossdomain.xml file to exist at this location:</p> <pre>www.XYZ.com/crossdomain.xml</pre> <p>The crossdomain.xml file would contain a section like this:</p> <pre><cross-domain-policy> <allow-access-from domain="*.ABC.com" secure="false"/> </cross-domain-policy></pre> <p>Examples:</p> <pre><method>GET</method> <method>POST</method></pre> |

Appendix D: Interface References

The following graphics and keys show the different labels of the SRC user interface that you can customize. For details on how to customize the user interface, see the "[Customize Colors and Graphics](#)" on page 21 section.

Client Customization Reference

The following user interface labels can be customized in SRC Client customization file, which you can find in the following default location:

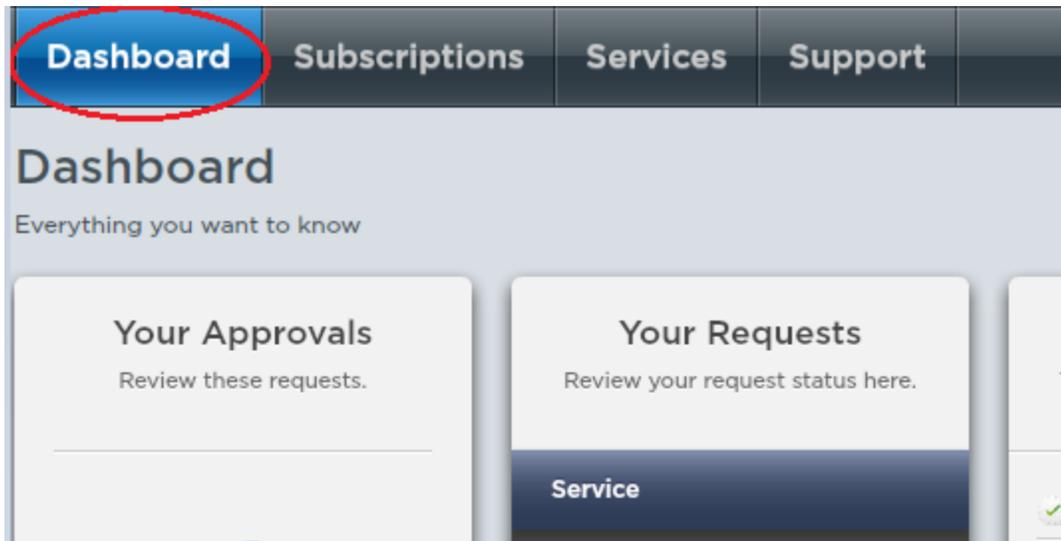
`/resources/client/xx/custom.properties`

Note:

- In this example path, the xx placeholder represents the language code of your SRC installation.
- For any custom property file, we recommend that you use Unicode encoding for any non-English characters. English characters in a non-English resource files will render in plain text as expected.

Example:

Suppose you want to change the default "Dashboard" label in the following screen shot to say "Welcome":



To implement this change, follow these steps:

1. In the file, locate the following UI text label key:

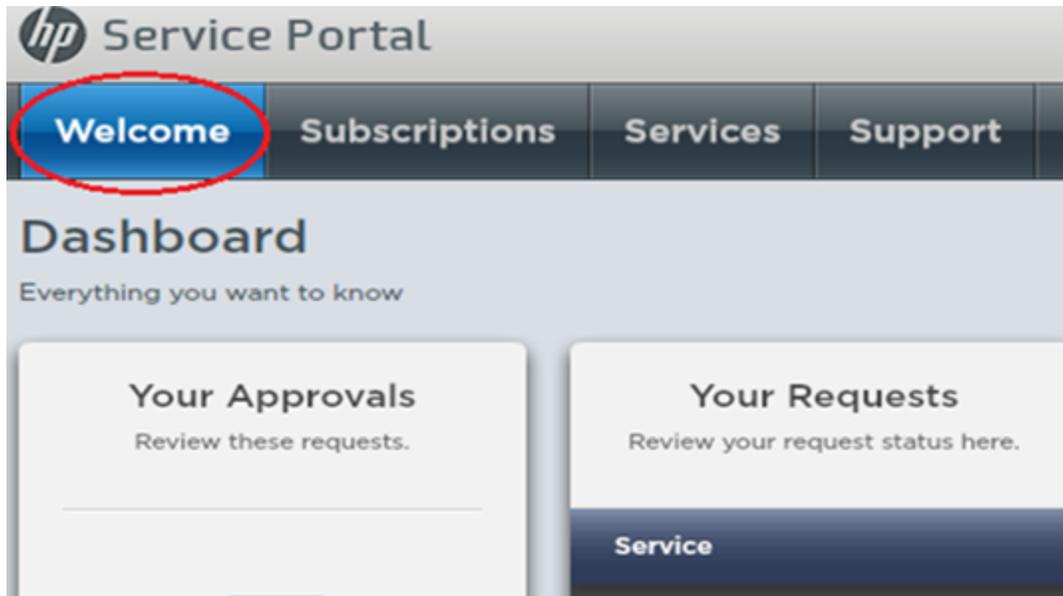
```
header.navigation.tab.dashboard = Dashboard
```

2. Change the "Dashboard" value to "Welcome" and then press CTRL+C to copy this line.
3. Paste the line into the custom properties file:

```
/resources/client/xx/custom.properties
```

4. Restart the Service Request Catalog service.

Your Service Request Catalog should appear as follows:



Note: If the label has not changed, clear your browser cache and then restart Service Request Catalog service again.

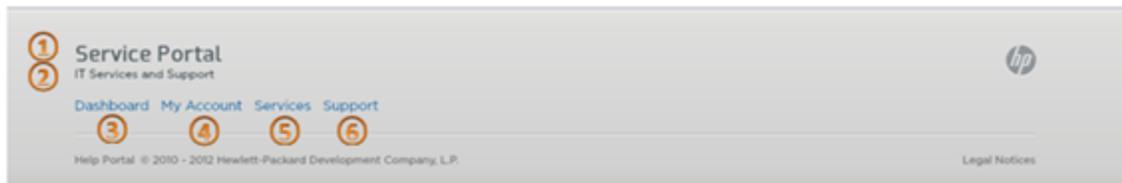
Customize the SRC Header



| Id | Key | Value |
|----|--|----------------|
| 1. | Configured in the manifest.xml, default key : default_company_name_label | Service Portal |
| 2. | header.navigation.tab.dashboard | Dashboard |

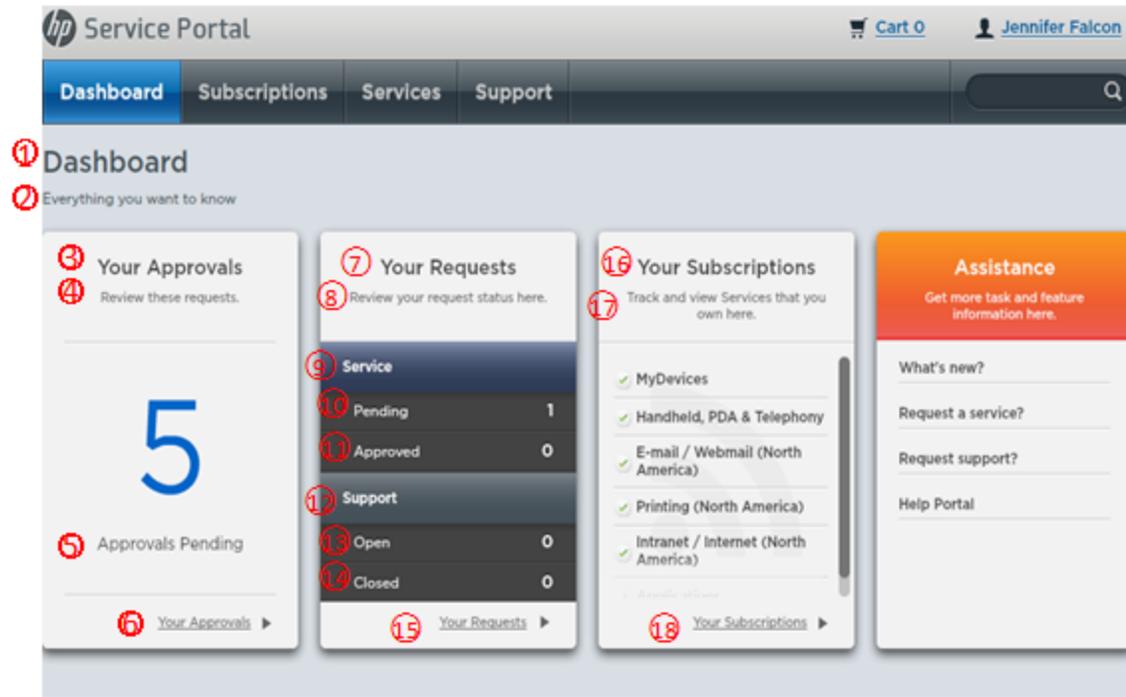
| | | |
|----|------------------------------------|---------------|
| 3. | header.navigation.tab.subscription | Subscriptions |
| 4. | header.navigation.tab.services | Services |
| 5. | header.navigation.tab.support | Support |
| 6. | header.savedItem.button.label | Cart 0 |

Customize the SRC Footer



| Id | Key | Value |
|----|---|------------------------|
| 1. | default_company_name_label (Configured in the manifest.xml, default key) | Service Portal |
| 2. | footer.slogan.label | IT Service and Support |
| 3. | footer.navigation.link.dashboard | Dashboard |
| 4. | footer.navigation.link.account | My Account |
| 5. | footer.navigation.link.services | Services |
| 6. | footer.navigation.link.support | Support |

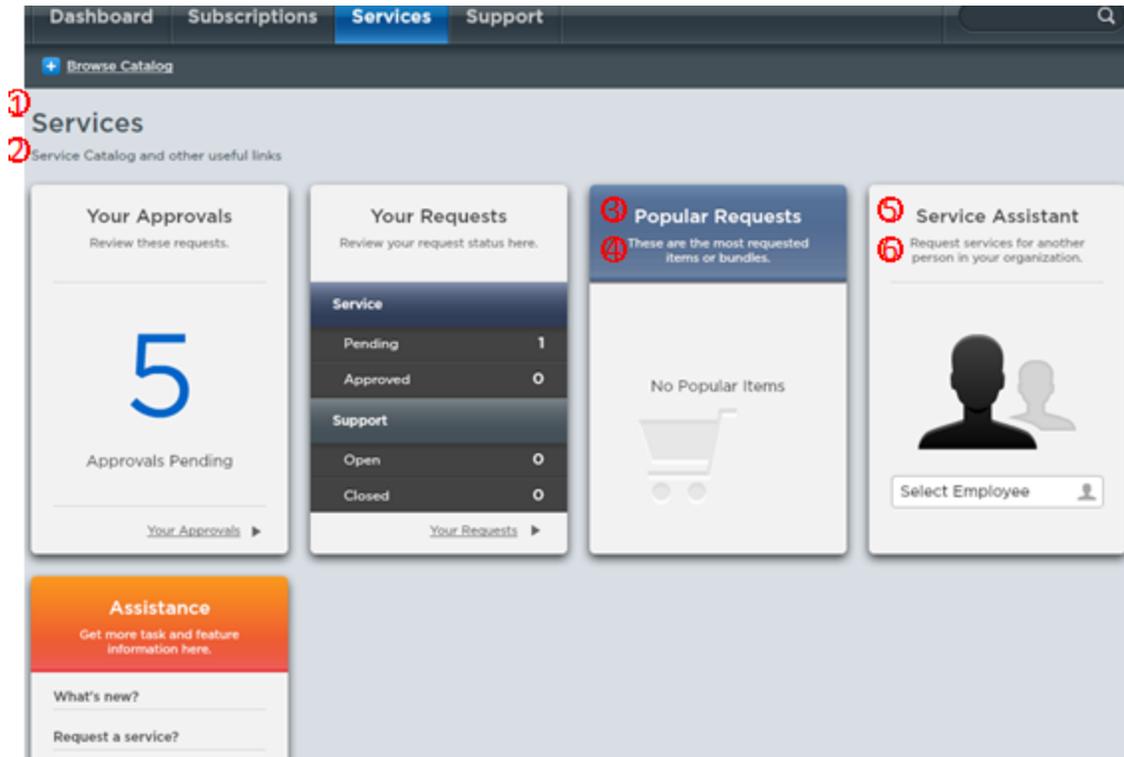
Customize the Dashboard



| Id | Key | Value |
|----|---------------------------------------|-------------------------|
| 1. | dashboard.dashboard_landing.title | Dashboard |
| 2. | dashboard.dashboard_landing.message | Everything you want ... |
| 3. | dashboard.approval_status | Your Approvals |
| 4. | dashboard.approval_widget_instruction | Review these requests |
| 5. | dashboard.view_approval_details | Approval Pending |

| | | |
|-----|--|-----------------------------|
| 6. | dashboard.approval_status | Your Approvals |
| 7. | dashboard.request_widget_title | Your Request s |
| 8. | dashboard.sm_services_request_widget_instruction | Review your request ... |
| 9. | dashboard.sm_services_request_status | Service |
| 10. | account_view.status_pending | Pending |
| 11. | account_view.status_approved | Approved |
| 12. | dashboard.sm_services_request_status | Support |
| 13. | account_view.status_open | Open |
| 14. | account_view.status_closed | Closed |
| 15. | dashboard.request_widget_title | Your Request s |
| 16. | dashboard.subscription_widget_title | Your Subscriptions |
| 17. | dashboard.subscription_widget_instruction | Track and view Services ... |
| 18. | dashboard.view_widget_details | Your Subscription |

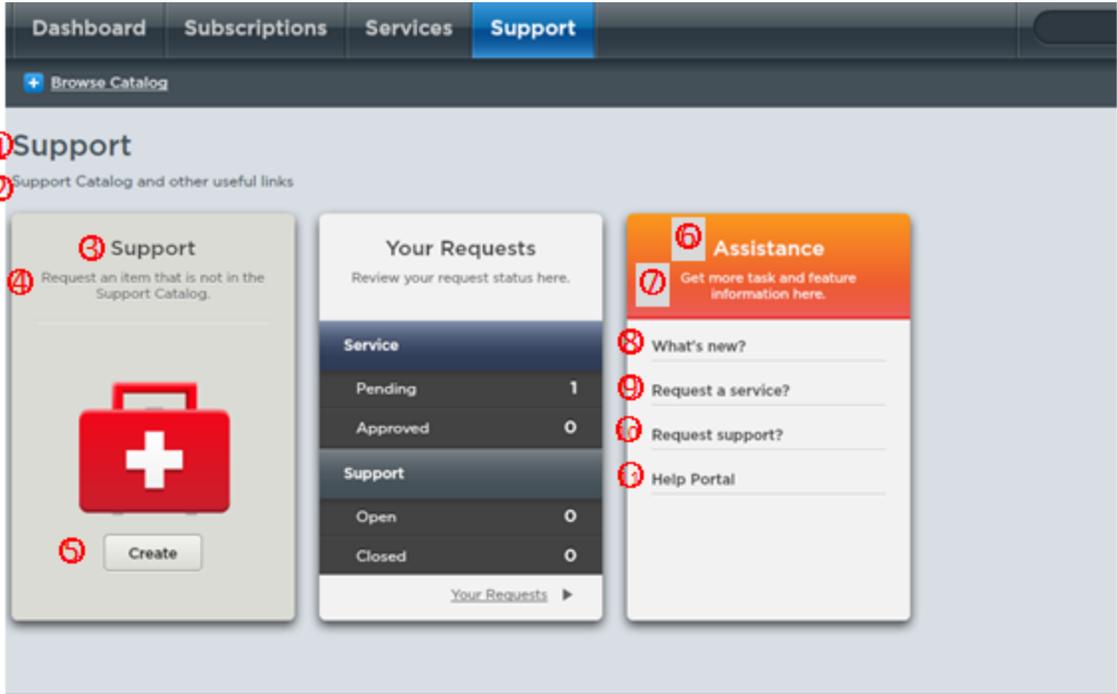
Customize the Services Page



| Id | Key | Value |
|----|-------------------------------------|----------------------------------|
| 1. | dashboard.dashboard_landing.title | Services |
| 2. | dashboard.service_landing.message | Service Catalog and ... |
| 3. | dashboard.popular_items | Popular Request s |
| 4. | dashboard.popular_items_instruction | There are the most requested ... |
| 5. | dashboard.request_on_behalf_title | Service Assistant |

| | | |
|----|---|----------------------------------|
| 6. | dashboard.request_on_behalf_instruction | Request Services for another ... |
|----|---|----------------------------------|

Customize the Support Page



| Id | Key | Value |
|----|-----------------------------------|--------------------------|
| 1. | dashboard.support_landing.title | Support |
| 2. | dashboard.support_landing.message | Support Catalog and ... |
| 3. | dashboard.support_title | Support |
| 4. | dashboard.support_instruction | Request an item that ... |

| | | |
|-----|----------------------------------|-----------------------|
| 5. | buttonTextCreate | Create |
| 6. | dashboard.assistance | Assistance |
| 7. | dashboard.assistance_instruction | Get more task and ... |
| 8. | dashboard.getting_started | What's new? |
| 9. | dashboard.faqs | Request a service? |
| 10. | dashboard.information | Request support? |
| 11. | dashboard.asssitance_portal | Help Portal |

Customize the Account View Page

The screenshot shows a 'Request Inbox' interface with a yellow header bar. Below the header, there are several sections and a main table. The sections include 'Your Service Requests' (with sub-sections: Approved, Pending, Denied, Closed, Review) and 'Your Support Requests' (with sub-sections: Open, Closed). The main table has columns for Request ID, Request Name, Price, and View. Red circles with numbers 1-11 are overlaid on the interface to indicate the locations of the keys listed in the table below.

| Id | Key | Value |
|----|----------------------------------|-----------------------|
| 5 | buttonTextCreate | Create |
| 6 | dashboard.assistance | Assistance |
| 7 | dashboard.assistance_instruction | Get more task and ... |
| 8 | dashboard.getting_started | What's new? |
| 9 | dashboard.faqs | Request a service? |
| 10 | dashboard.information | Request support? |
| 11 | dashboard.asssitance_portal | Help Portal |

| Id | Key | Value |
|----|----------------------------------|-----------------------|
| 5 | buttonTextCreate | Create |
| 6 | dashboard.assistance | Assistance |
| 7 | dashboard.assistance_instruction | Get more task and ... |
| 8 | dashboard.getting_started | What's new? |
| 9 | dashboard.faqs | Request a service? |
| 10 | dashboard.information | Request support? |
| 11 | dashboard.asssitance_portal | Help Portal |

| | | |
|-----|---|-----------------------|
| 1. | account_view.order_id | Request ID |
| 2. | account_view.title | Request Name |
| 3. | account_view.price | Cost(Price) |
| 4. | account_view.view | View |
| 5. | account_view.your_service_request | Your Service Requests |
| 6. | account_view.your_support_request | Your Support Requests |
| 7. | account_view.status_approved | Approved |
| 8. | account_view.status_pending | Pending |
| 9. | account_view.status_denied | Denied |
| 10. | account_view.status_closed | Closed |
| 11. | account_view.status_review | Review |
| 12. | account_view.status_open | Open |
| 13. | account_view.status_closed | Closed |
| 14. | account_view.request_inbox | Request Inbox |
| 15. | *This box takes its value from numbers, 7,8,9,10, and 11, as appropriate. | N/A |

Customize the Details View Page

Pending Approval

SD10325
df

1 Request placed on March 4, 2013 10:39:39 AM

Resubmit

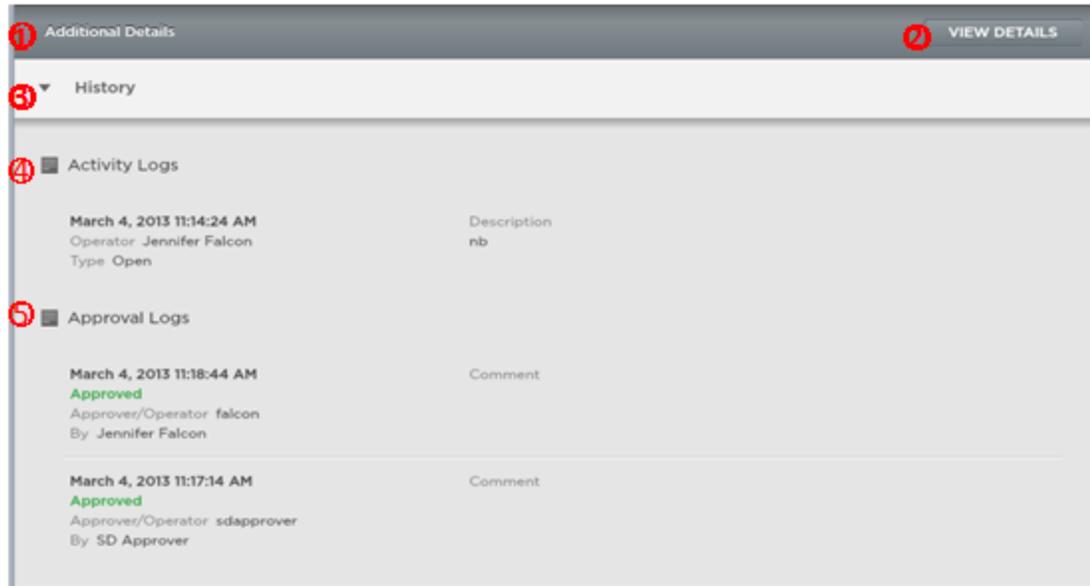
2 Approvers
SD Approver (2. Pending)

3 Requested By Jennifer Falcon 4 Contact Information sales@sm.hp.com

5 Request Description df

| Id | Value | Key |
|----|----------------------|----------------------------------|
| 1. | Request place on ... | account_view.request_placed_on |
| 2. | Approvers | account_view.approvers |
| 3. | Requested By | account_view.request_by |
| 4. | Contact Information | account_view.contact_info |
| 5. | Request Description | account_view.request_description |

Customize the Approval History

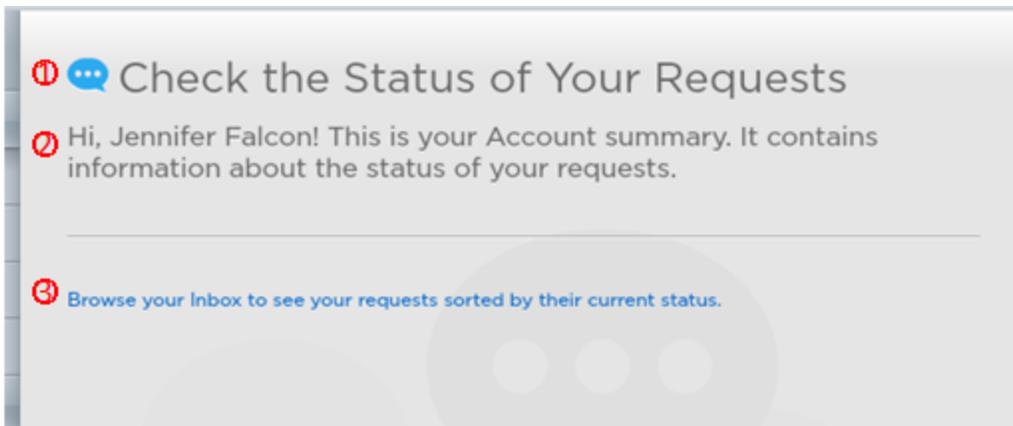


| Id | Key | Value |
|----|--|--------------------|
| 1. | account_view.header_approval_status | Additional Details |
| 2. | account_view.button_view_approval_history | View Details |
| 3. | interaction_history.section_name.history | History |
| 4. | interaction_history.section_name.activity_logs | Activity Logs |
| 5. | interaction_history.section_name.approval_logs | Approval Logs |



| Id | Key | Value |
|----|---|--------------------|
| 1. | account_view.header_approval_status | Additional Details |
| 2. | account_view.button_hide_approval_history | View History |
| 3. | account_view.header_line_items | Line Item |

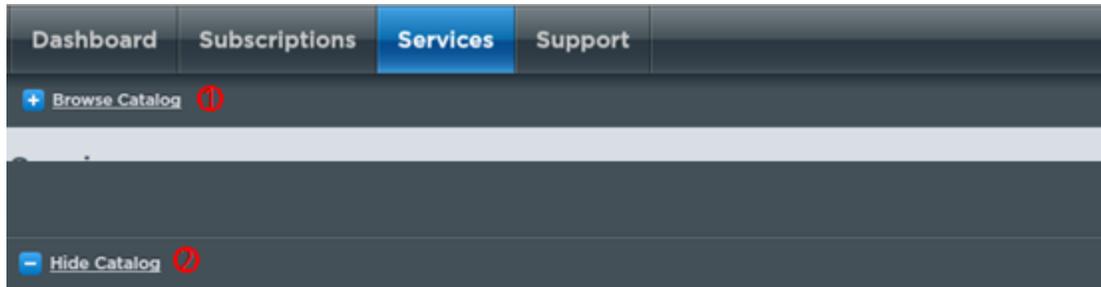
Customize the Request Inbox Welcome Page



| Id | Key | Value |
|----|-----------------------------|-------------------------|
| 1. | account_view.welcome.header | Check the Status of ... |

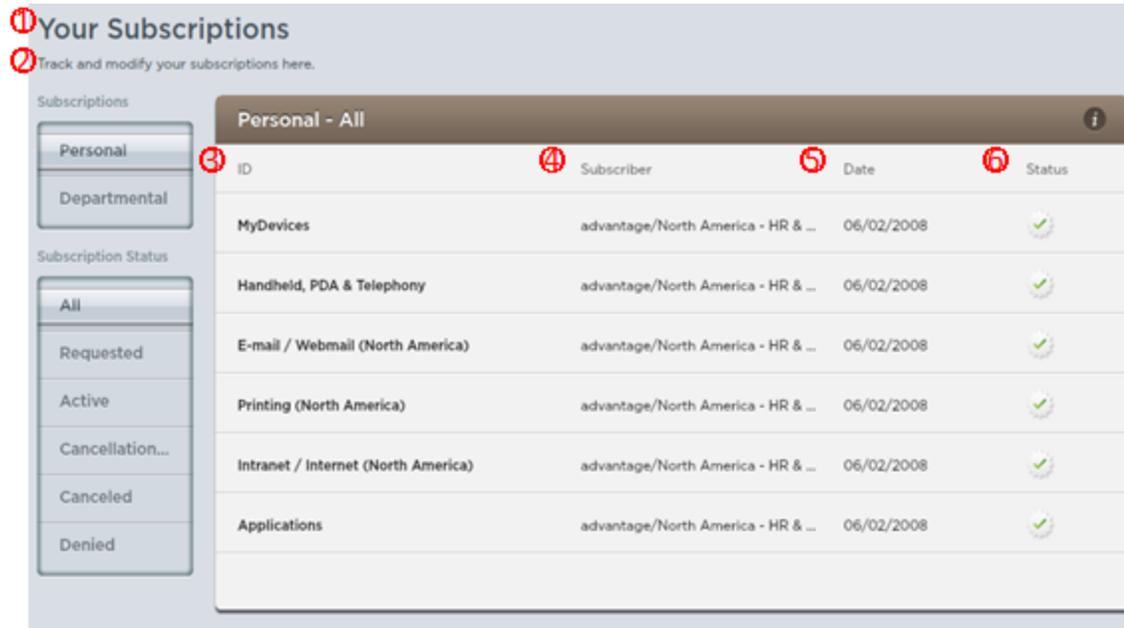
| | | |
|----|--------------------------|------------------------|
| 2. | account_view.welcome | Hi, Jennifer ... |
| 3. | account_view.instruction | Browser your inbox ... |

Customize the Browse Catalog Button

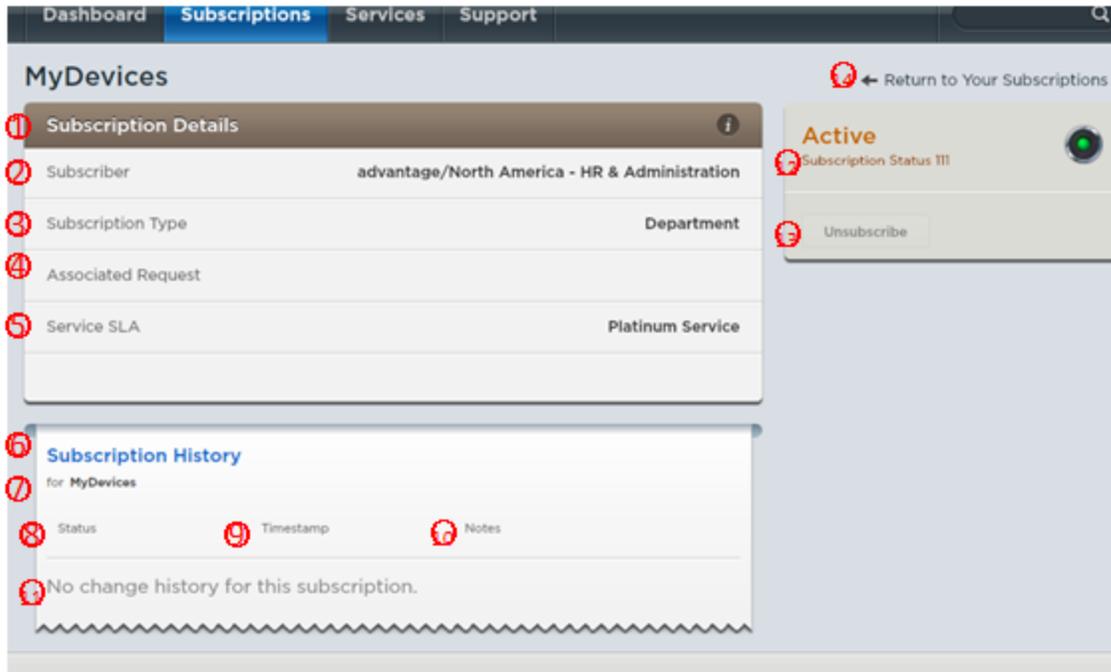


| Id | Key | Value |
|----|--|----------------|
| 1. | header.search.openCatalog.button.label | Browse Catalog |
| 2. | header.search.closedCatalog.button.label | Hide Catalog |

Customize Subscriptions



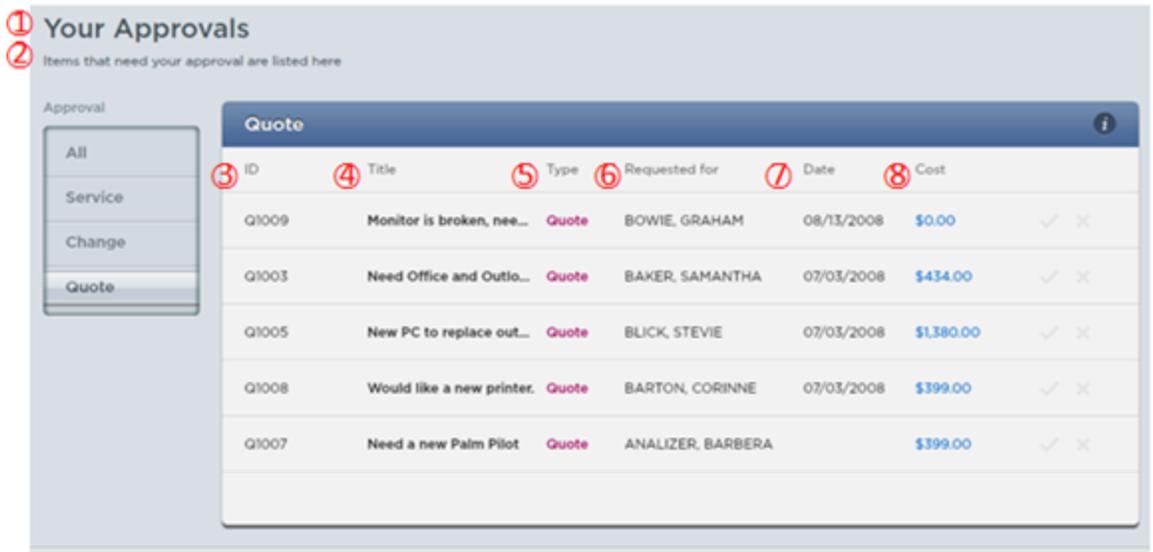
| Id | Key | Value |
|----|---|----------------------|
| 1. | subscription.header.title | Your Subscriptions |
| 2. | subscription.header.description | Track and modify ... |
| 3. | subscription.itemlist.header.name | ID |
| 4. | subscription.itemlist.header.subscriber | Subscriber |
| 5. | subscription.itemlist.header.date | Date |
| 6. | subscription.itemlist.header.status | Status |



| Id | Key | Value |
|----|---|----------------------|
| 1. | subscription.itemDetail.header.title | Subscription Details |
| 2. | subscription.itemDetail.subscriber | Subscriber |
| 3. | subscription.itemDetail.subscriptionType | Subscription Type |
| 4. | subscription.itemDetail.associatedRequest | Associated Request |
| 5. | subscription.itemDetail.serviceSLA | Service SLA |
| 6. | subscription.itemDetail.history.title | Subscription SLA |
| 7. | subscription.itemDetail.history.for | for |
| 8. | subscription.itemDetail.history.status | Status |

| | | |
|-----|---|------------------------------|
| 9. | subscription.itemDetail.history.timestamp | Timestamp |
| 10. | subscription.itemDetail.history.Notes | Notes |
| 11. | subscription.itemDetail.history.noresult | No Change history for ... |
| 12. | subscription.itemDetail.contolPanel.statusLabel | Subscription Status |
| 13. | subscription.itemDetail.contolPanel.unsubscribeButton | unsubscribe |
| 14. | subscription.itemDetail.header.return | Return to Your Subscriptions |

Customize the Approvals Page

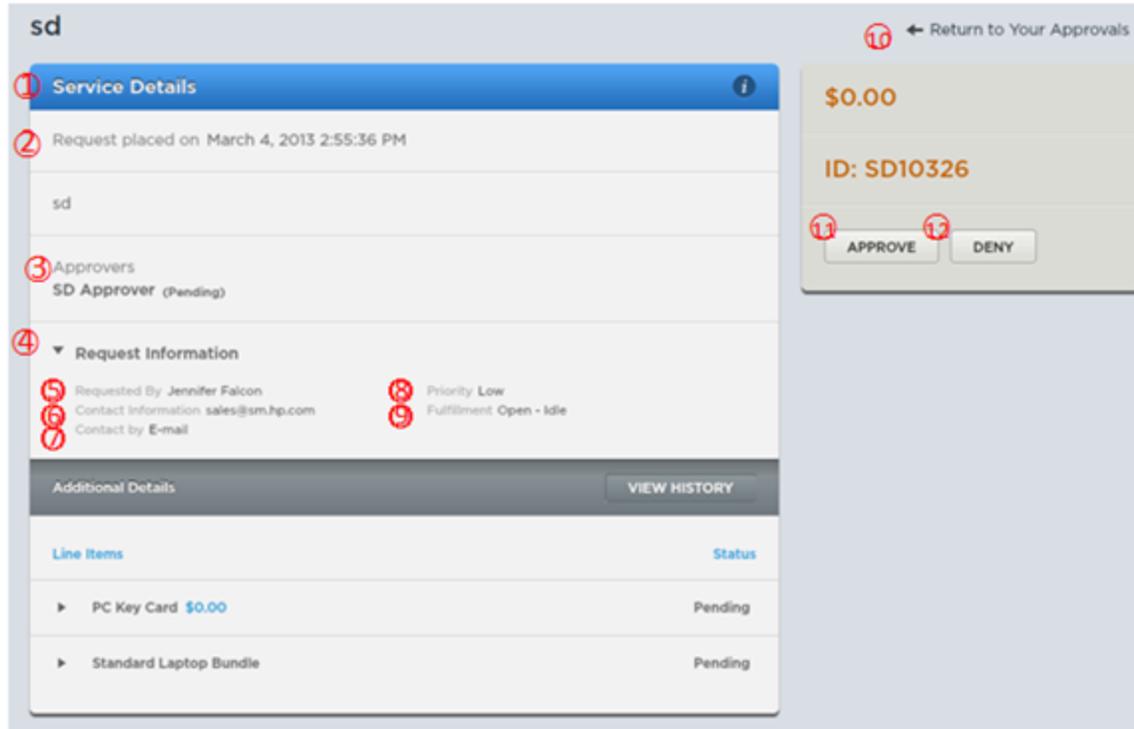


| Id | Key | Value |
|----|--------------------|--------------------|
| 1. | header.title | Your Approvals |
| 2. | header.description | Item that need ... |

| | | |
|----|--------------------------|---------------|
| 3. | list.column.id | ID |
| 4. | list.column.title | Title |
| 5. | list.column.type | Type |
| 6. | list.column.requestedFor | Requested for |
| 7. | list.column.date | Date |
| 8. | list.column.cost | Cost |

Note: The Approvals Page also requires some customization on the SRC server side. For more information, see ["Customize: "Your Approvals"" on page 134](#)

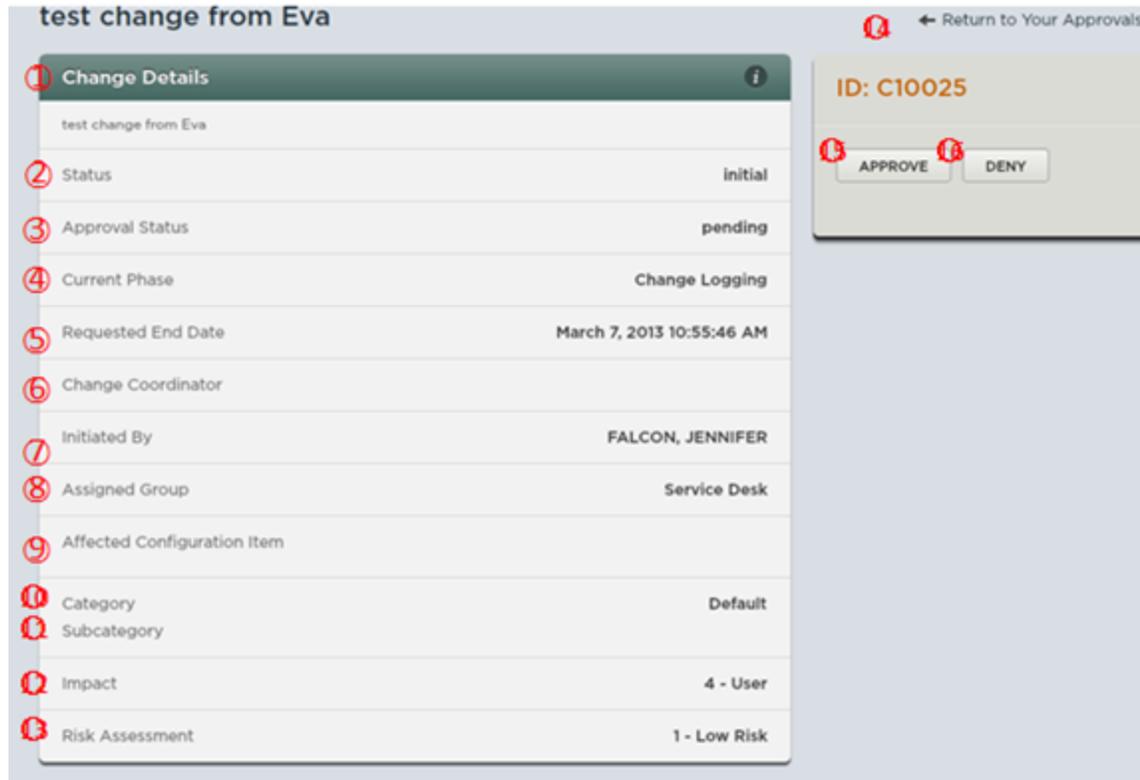
Customize the Service Approval Details Page



| Id | Key | Value |
|----|--------------------------------|---------------------|
| 1. | service.detail | Service Details |
| 2. | account_view.request_placed_on | Request Placed on |
| 3. | account_view.approvers | Approvers |
| 4. | request.information | Request Information |
| 5. | account_view.request_by | Requested By |

| | | |
|------------|----------------------------------|--------------------------|
| 6. | account_view.contact_info | Contact Information |
| 7. | request.information.callbackType | Contact by |
| 8. | request.information.urgency | Priority |
| 9. | request.information.status | Fulfillment |
| 10. | detail.back | Return to your Approvals |
| 11. | account_view.approve_button | APPROVE |
| 12. | account_view.deny_button | DENY |

Customize the Change Approval Details Page



| Id | Key | Value |
|----|---|-----------------|
| 1. | approval.change.detail.title | Change Details |
| 2. | approval.change.detail.label.status | Status |
| 3. | approval.change.detail.label.approvalStatus | Approval Status |
| 4. | approval.change.detail.label.currentPhase | Current Phase |

| | | |
|-----|--|-----------------------------|
| 5. | approval.change.detail.label.requestedEndDate | Requested End Date |
| 6. | approval.change.detail.label.changeCoordinator | Change Coordinator |
| 7. | approval.change.detail.label.initiatedBy | Initiated By |
| 8. | approval.change.detail.label.assignedGroup | Assigned Group |
| 9. | approval.change.detail.label.affectedConfigurationItem | Affected Configuration Item |
| 10. | approval.change.detail.label.category | Catalog |
| 11. | approval.change.detail.label.subcategory | Subcategory |
| 12. | approval.change.detail.label.impact | Impact |
| 13. | approval.change.detail.label.riskAssessment | Risk Assessment |
| 14. | detail.back | Return to your Approvals |
| 15. | account_view.approve_button | APPROVE |
| 16. | account_view.deny_button | DENY |

Customize the Quote Approval Details Page

Monitor is broken, need to schedule a tech to come look at it.

[Return to Your Approvals](#)

Quote Details

Requested For: BOWIE, GRAHAM

Requested By: BOWIE, GRAHAM

Requested On: February 5, 2001 10:43:04 PM

Status: initial

Approval Status: pending

Approver: Financial Approval

Current Phase: Manager Approval

Bundles

| ID | Quantity | Description | Cost |
|----|----------|-------------|--------|
| 1 | 1 | Repairs | \$0.00 |

Items

| ID | Quantity | Description | Status | Cost |
|-----------|----------|-------------|-----------|--------|
| Q1009-001 | 1 | Repairs | requested | \$0.00 |

\$0.00

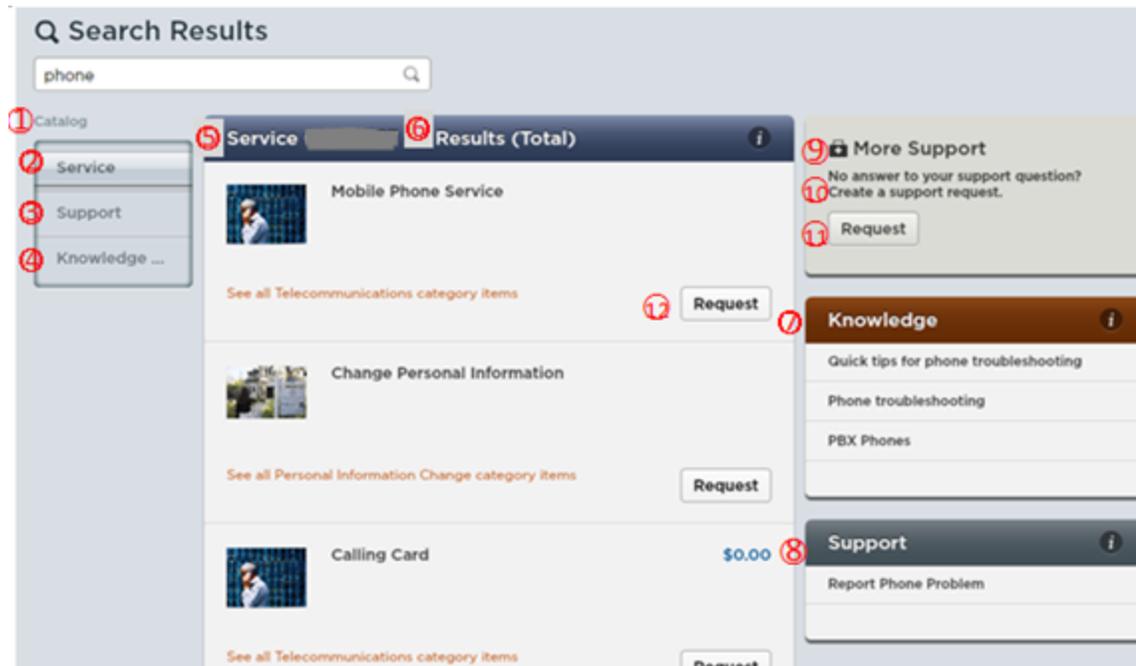
ID: Q1009

APPROVE DENY

| Id | Key | Value |
|----|------------------------------------|---------------|
| 1. | approval.quote.detail.title | Quote Details |
| 2. | approval.quote.detail.requestedFor | Request For |
| 3. | approval.quote.detail.requestedBy | Request By |
| 4. | approval.quote.detail.requestedOn | Request On |

| | | |
|-----|--------------------------------------|--------------------------|
| 5. | approval.quote.detail.status | Status |
| 6. | approval.quote.detail.approvalStatus | Approval Status |
| 7. | list.column.approvals | Approver |
| 8. | approval.quote.detail.currentPhase | Current phase |
| 9. | approval.quote.bundle.title | Bundles |
| 10. | approval.quote.bundle.id | ID |
| 11. | approval.quote.bundle.quantity | Quantity |
| 12. | approval.quote.bundle.description | Description |
| 13. | approval.quote.bundle.price | Cost |
| 14. | approval.quote.item.title | Items |
| 15. | approval.quote.item.id | ID |
| 16. | approval.quote.item.quantity | Quantity |
| 17. | approval.quote.item.description | Description |
| 18. | approval.quote.item.status | Status |
| 19. | approval.quote.item.price | Cost |
| 20. | approval.quote.return | Return to your Approvals |
| 21. | account_view.approve_button | APPROVE |
| 22. | account_view.deny_button | DENY |

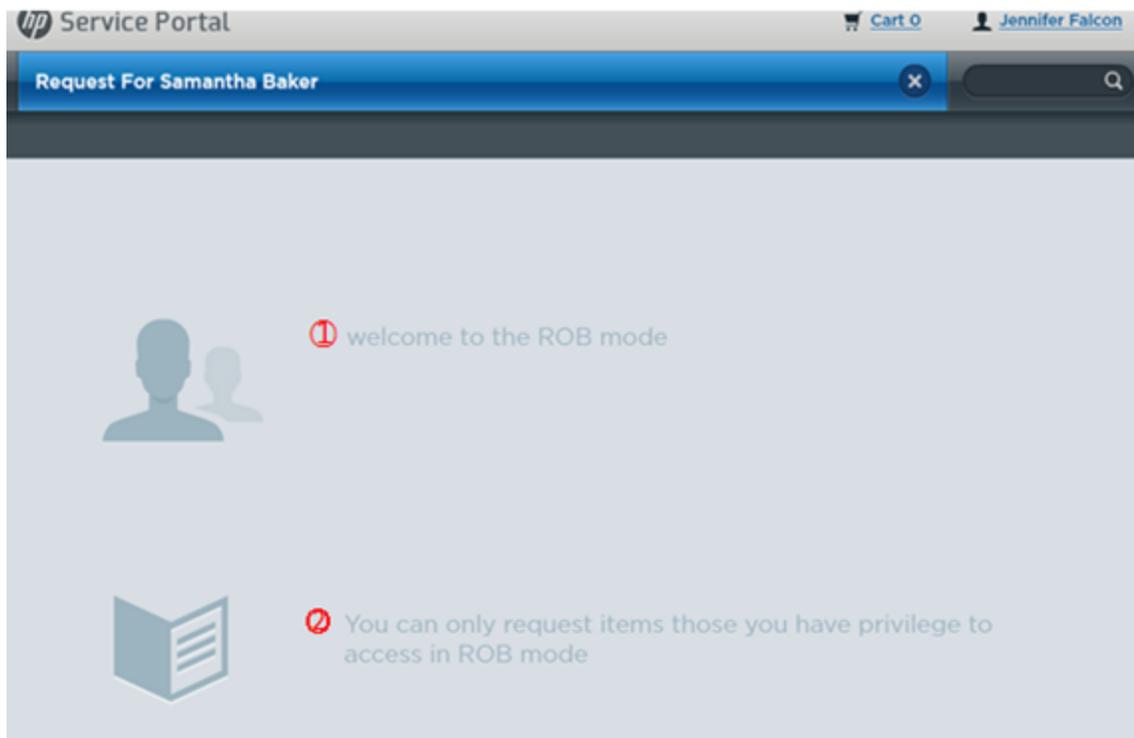
Customize the Search Results Page



| Id | Key | Value |
|----|---------------------------|-----------------|
| 1. | searchView.catalog | Catalog |
| 2. | searchView.service | Service |
| 3. | searchView.support | Support |
| 4. | searchView.knowledge | Knowledge ... |
| 5. | searchView.header.service | Service |
| 6. | results.number | Results (Total) |

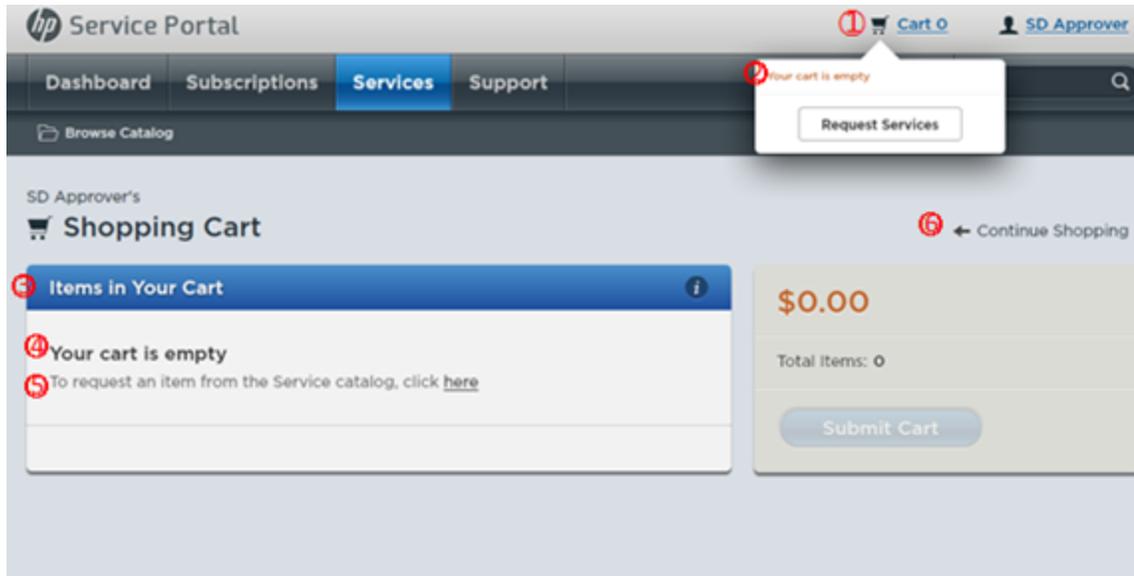
| | | |
|-----|---|-------------------|
| 7. | searchView.header.knowledge | knowledge |
| 8. | searchView.header.support | Support |
| 9. | support.generic.request.widget.title | More Support |
| 10. | support.generic.request.widget.message | No answer to ... |
| 11. | support.generic.request.widget.button.label | Request(Support) |
| 12. | mainview.item_request_button | Request (Service) |

Customize the Request on Behalf Of



| Id | Key | Value |
|----|-------------------------------------|---------------------------|
| 1. | services.impersonate.welcome.header | Welcome to ... |
| 2. | services.impersonate.instruction | You can only request |

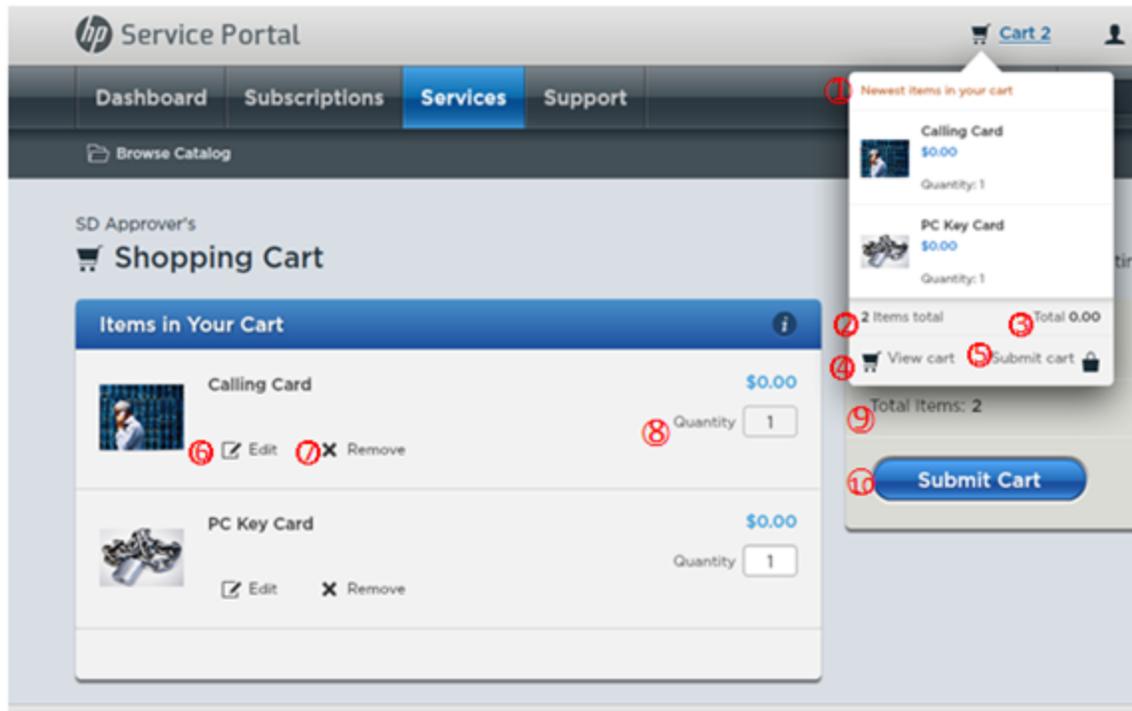
Customize the Shopping Cart



| Id | Key | Value |
|----|----------------------------------|--------------------|
| 1. | header.savedItem.button.label | Cart 0 |
| 2. | cart_view.cart_empty | Your cart is empty |
| 3. | cart_view.cart.cart_items_header | Items in Your cart |
| 4. | cart_view.cart.empty | Your cart is empty |

| | | |
|----|-----------------------------|------------------------|
| 5. | cart_view.cart.empty_prompt | To request an item ... |
| 6. | cart_view.continue_shopping | Continue shopping |

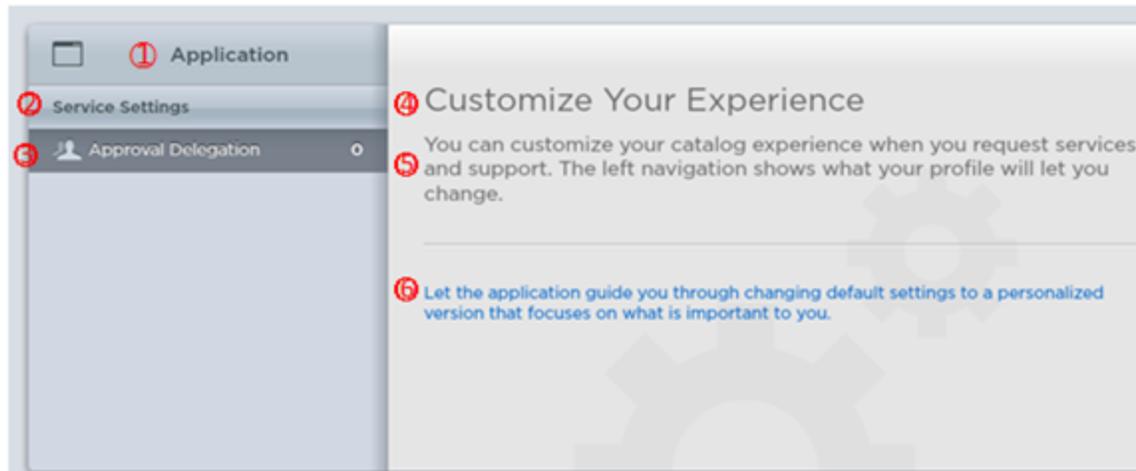
Shopping Cart View



| Id | Value | Key |
|----|--------------------------|-------------------------------|
| 1. | Newest Item in your cart | cart_view.mouse_hover_header |
| 2. | * Items total | cart_view.items_total.several |
| 3. | Total | cart_view.price_total |
| 4. | View Cart | cart_view.view_cart |

| | | |
|-----|-------------|-----------------------------|
| 5. | Submit cart | cart_view.checkout |
| 6. | Edit | cart_view.edit_item_label |
| 7. | Remove | cart_view.remove_item_label |
| 8. | Quantity | cart_view.item_quantity |
| 9. | Total Items | quantity.cart.label |
| 10. | Submit Cart | button.cart.label |

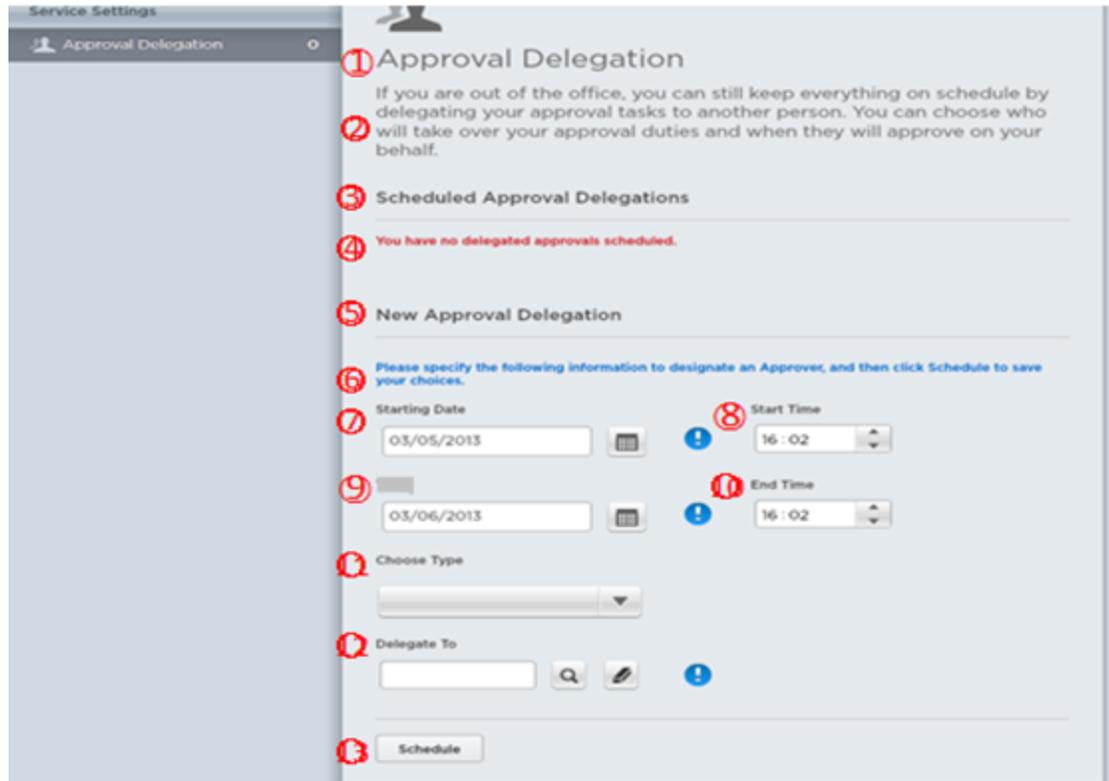
Customize the SRC Settings



| Id | Key | Value |
|----|--|---------------------|
| 1. | settings.inbox.title | Application |
| 2. | settings.service.title | Service Settings |
| 3. | settings.approvalDelegation.welcome.header | Approved Delegation |

| | | |
|----|-------------------------|-------------------------------|
| 4. | settings.welcome.header | Customized Your Experience |
| 5. | settings.welcome | You can customize ... |
| 6. | settings.instruction | Let the application guide ... |

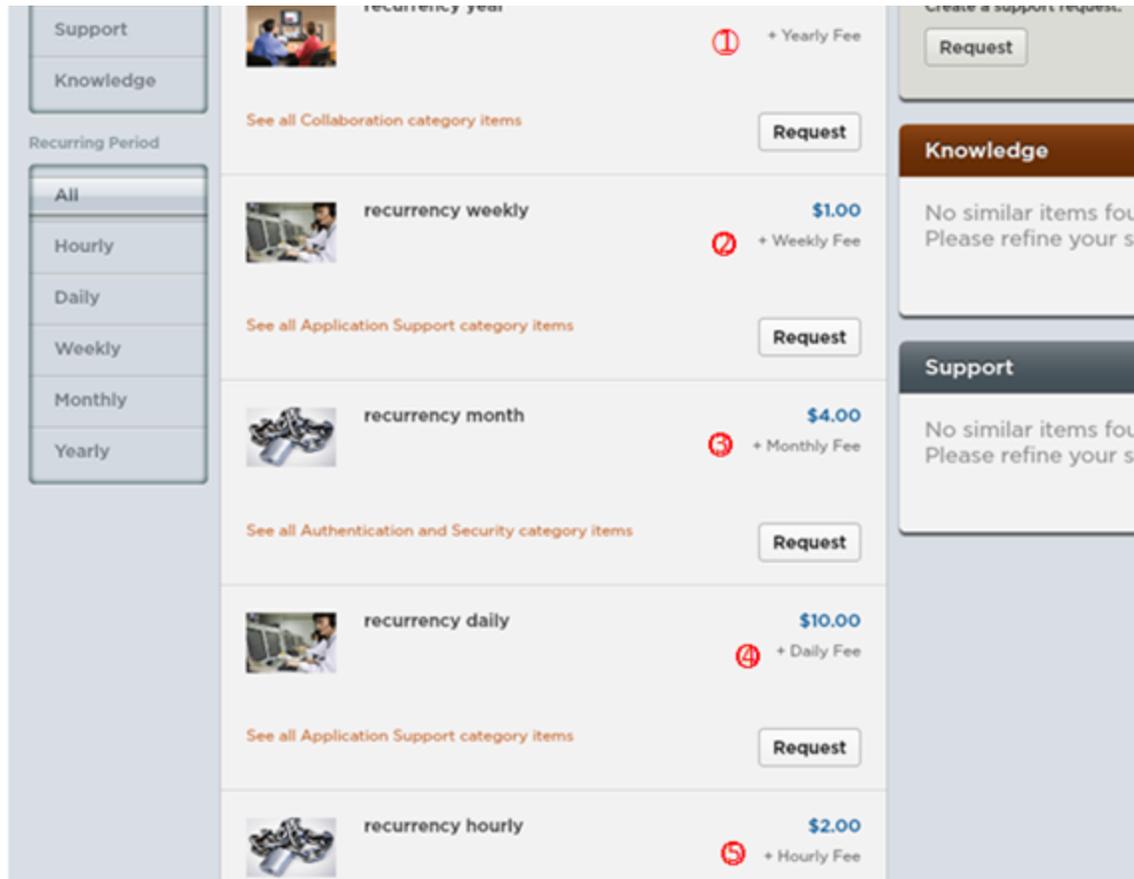
Customize the Approval Delegation Details Page



| Id | Key | Value |
|----|--|---------------------|
| 1. | settings.approvalDelegation.welcome.header | Approval Delegation |

| | | |
|-----|--|------------------------------|
| 2. | settings.approvalDelegation.welcome | If you are out of office ... |
| 3. | settings.approvalDelegation.list.header | Scheduled Approval ... |
| 4. | settings.approvalDelegation.list.no.delegation | You have no ... |
| 5. | settings.approvalDelegation.create | New Approval Delegation |
| 6. | settings.approvalDelegation.form.description | Please specify the ... |
| 7. | settings.approvalDelegation.startDate.label | Starting Date |
| 8. | settings.approvalDelegation.startTime.label | Start Time |
| 9. | settings.approvalDelegation.endDate.label | End Date |
| 10. | settings.approvalDelegation.endTime.label | End Time |
| 11. | settings.approvalDelegation.chooseType | Choose Type |
| 12. | settings.approvalDelegation.delegateTo.label | Delegate To |
| 13. | settings.approvalDelegation.createButton.label | schedule |

Customize the Recurring Cost Labels



| Id | Key | Value |
|----|-----------------------------------|--------------|
| 1. | itemList.recurringPrice.yearly.sm | + Yearly Fee |
| 2. | itemList.recurringPrice.weekly.sm | + weekly Fee |

| | | |
|----|------------------------------------|---------------|
| 3. | itemList.recurringPrice.monthly.sm | + Monthly Fee |
| 4. | itemList.recurringPrice.daily.sm | + Daily Fee |
| 5. | itemList.recurringPrice.hourly.sm | + Hourly Fee |

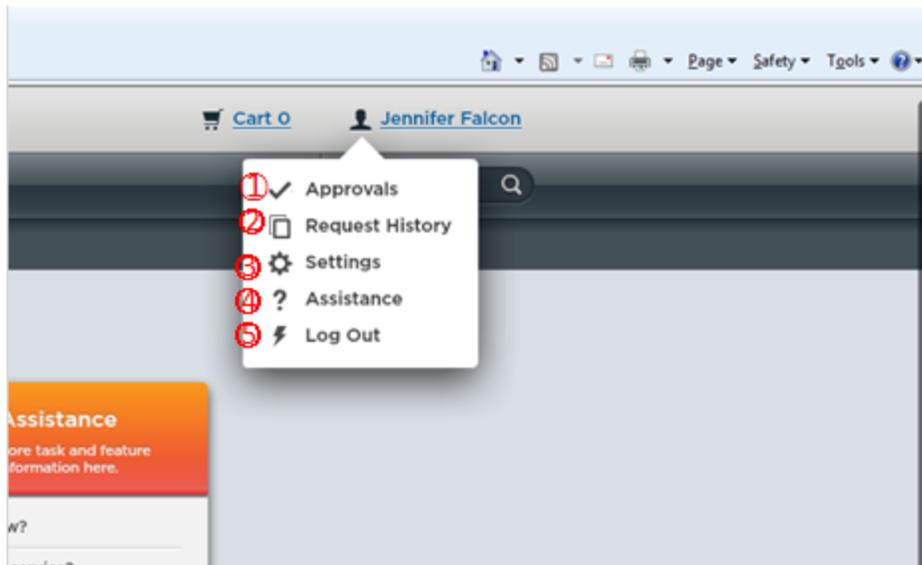
Customize the Login Panel

The screenshot shows the HP Service Portal login interface. At the top left is the HP logo and the text "Service Portal". Below this is a dark grey banner with the text "Hello and Welcome! Please Log In" and an information icon. The main login area contains a "User Name" label and input field (callout 2), a "Password" label and input field (callout 3), a checkbox for "Keep me logged in" (callout 4) with the text "Do not select if this is a shared computer." (callout 5), and "Submit" (callout 6) and "Clear" (callout 7) buttons. At the bottom, there is a copyright notice: "© 2010 - 2012 Hewlett-Packard Development Company, L.P."

| Id | Key | Value |
|----|-----|-------|
|----|-----|-------|

| | | |
|----|--------------------------|-----------------------|
| 1. | login.label.head | Hello and Welcome ... |
| 2. | login.label.username | User Name |
| 3. | login.label.password | Password |
| 4. | login.label.remember | Keep me logged in. |
| 5. | login.label.rememberNote | Do not select if ... |
| 6. | login.label.submit | Submit |
| 7. | login.label.reset | Clear |

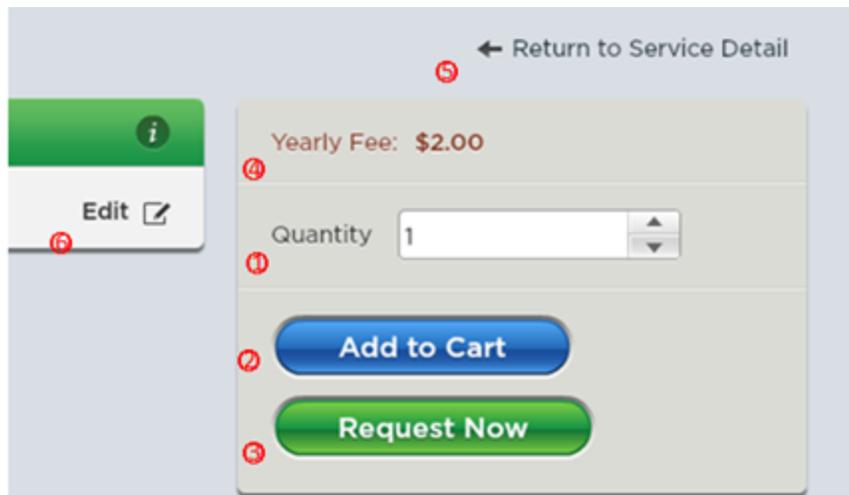
Customize the User Account Settings



| Id | Key | Value |
|----|-----|-------|
|----|-----|-------|

| | | |
|----|--------------------------------|-----------------|
| 1. | header.user_account.approvals | Approvals |
| 2. | header.user_account.requests | Request History |
| 3. | header.user_account.settings | Settings |
| 4. | header.user_account.logout | Assistance |
| 5. | header.user_account.assistance | Logout |

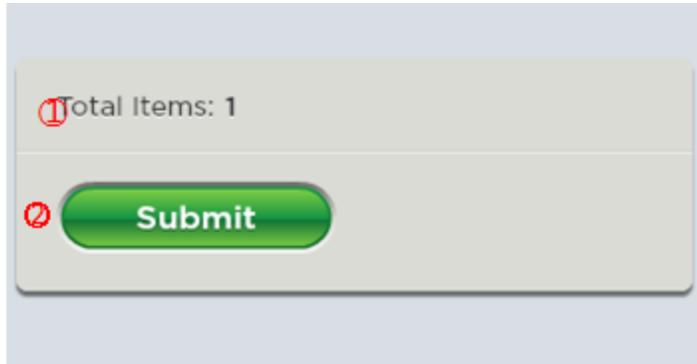
Customize the Request Summary Panel



| Id | Value | Key |
|----|-------------|--------------------------|
| 1. | Quantity | quantity.request.label |
| 2. | Add to Cart | button.add.cart.label |
| 3. | Request Now | button.requestNow.label |
| 4. | Yearly Fee. | recurring.fee.year.label |

| | | |
|----|--------------------------|-------------------------|
| 5. | Return to Service Detail | button.return.to.detail |
| 6. | Edit | button.edit |

Customize the Submission Panel



| Id | Value | Key |
|----|-------------|-----------------------|
| 1. | Total Items | quantity.cart.label |
| 2. | Submit | button.checkout.label |

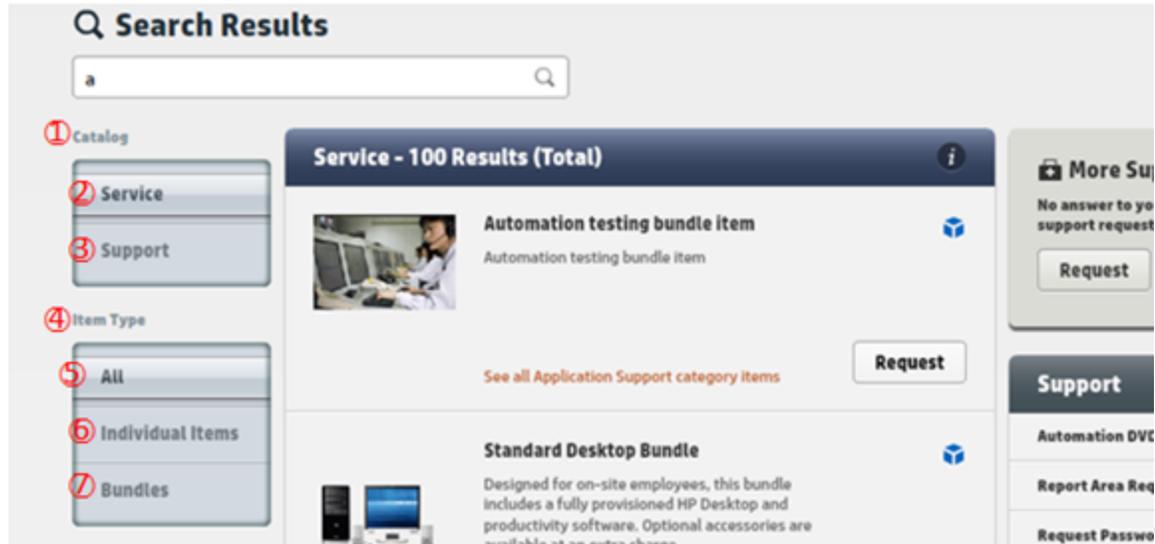
Server Customization Reference

The following user interface labels can be customized in SRC Server customization file, which by default, you can find in the following location:

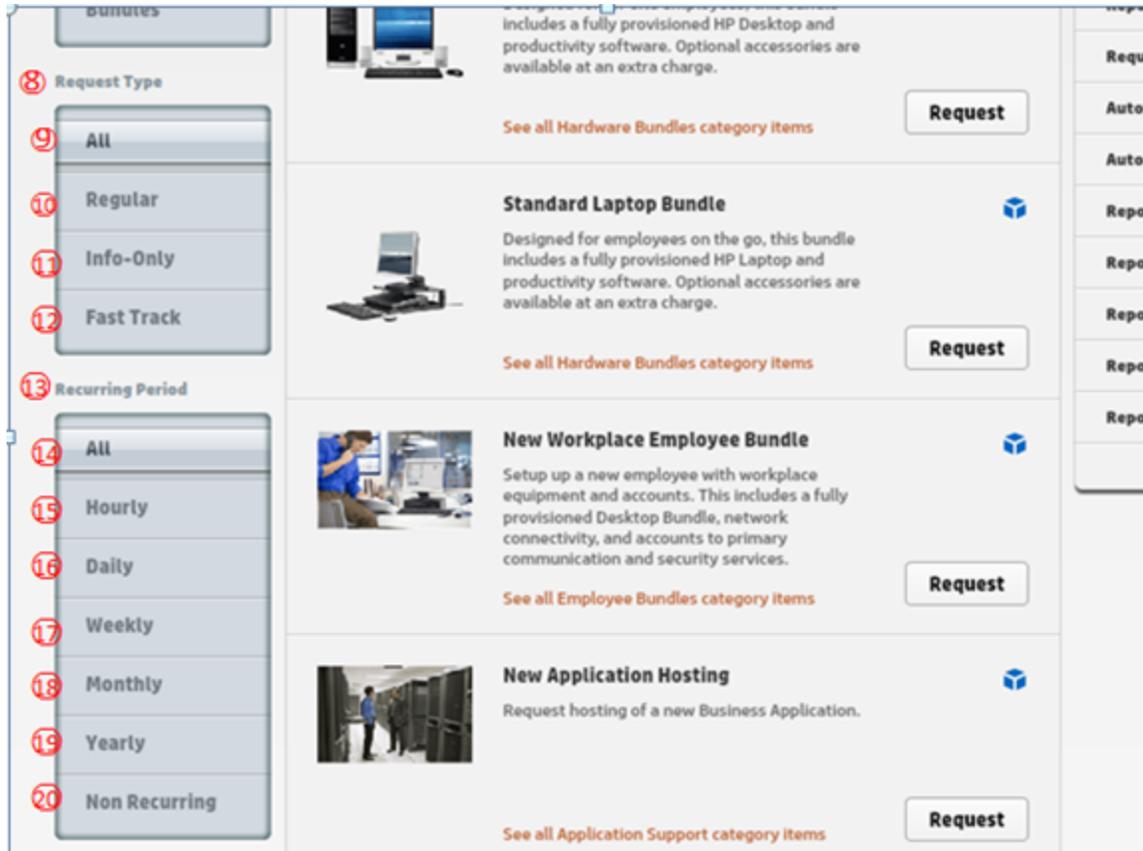
`/resources/server/xx/CustomServer_xx.properties`

Note: In this example path, the `xx` placeholder represents the language code of your SRC installation.

Customize the Search Results



| Id | Key | |
|----|------------------|--|
| 1. | Catalog | Configure in client custom.properties file |
| 2. | Service | Configure in client custom.properties file |
| 3. | Support | Configure in client custom.properties file |
| 4. | Item Type | search_facet.service.item_type |
| 5. | All | search_facet.facet_value_all |
| 6. | Individual Items | search_facet.service.item_type.item |
| 7. | Bundles | search_facet.service.item_type.bundle |



| Id | Key | |
|-----|--------------|---|
| 8. | Request Type | search_facet.service.item_status |
| 9. | All | search_facet.facet_value_all |
| 10. | Regular | search_facet.service.item_status.standard |
| 11. | Info-Only | search_facet.service.item_status.infoOnly |
| 12. | Fast Track | search_facet.service.item_status.nonCart |

| | | |
|-----|------------------|--|
| 13. | Recurring Period | search_facet.service.recurring_period |
| 14. | All | search_facet.facet_value_all |
| 15. | Hourly | search_facet.service.recurring_period.hour |
| 16. | Daily | search_facet.service.recurring_period.day |
| 17. | Weekly | search_facet.service.recurring_period.week |
| 18. | Monthly | search_facet.service.recurring_period.month |
| 19. | Yearly | search_facet.service.recurring_period.year |
| 20. | Non Recurring | search_facet.service.recurring_period.nonrecurring |

Customize the Order Information

Business Continuity

1 Order Information i

2 Request Type

Individual !

Department

3 This request is for

🔍 ✎ !

4 Delivery SLA

!

5 Service SLA

!

! Required Fields. ▶ Next

6 Item Options i

This task depends on the results of the previous task.

Fixed Cost

Quantity

| Id | Key | |
|----|-------------------|-------------------------------------|
| 1. | Order Information | group_info.order_info.label |
| 2. | Request Type | order_info.requested_for_type.label |

| | | |
|----|---------------------|--------------------------------|
| 3. | This request is for | order_info.requested_for.label |
| 4. | Delivery SLA | order_info.delivery_sla.label |
| 5. | Service SLA | order_info.service_sla.label |
| 6. | Item Options | group_info.item_options.label |

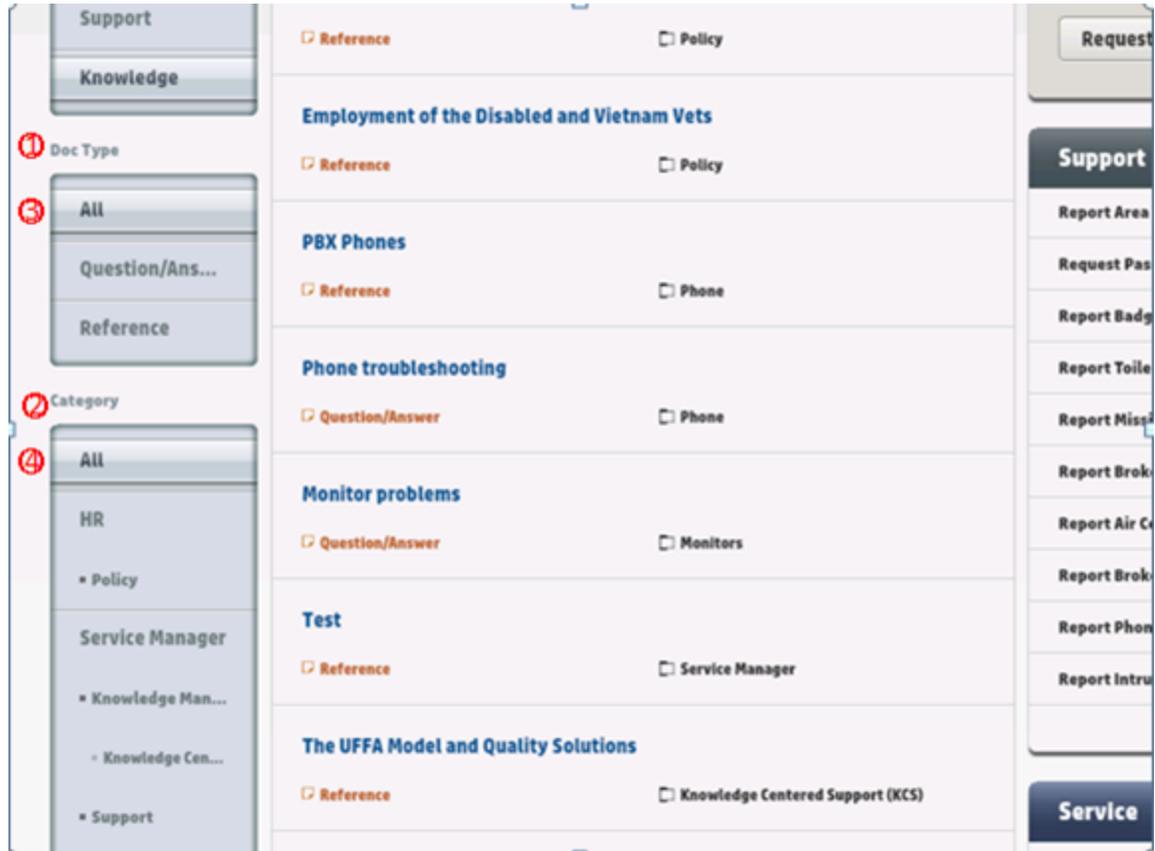
Customize the Subscriptions Page

The screenshot shows the HP Service Portal interface for the Subscriptions page. The navigation bar includes 'Dashboard', 'Subscriptions', 'Services', and 'Support'. The main content area is titled 'Your Subscriptions' and includes a sub-heading 'Track and modify your subscriptions here.' On the left, there are two filter menus: 'Subscriptions' with options 'Personal' and 'Departmental', and 'Subscription Status' with options 'All', 'Requested', 'Active', 'Cancellation R...', 'Canceled', and 'Denied'. On the right, a table titled 'Personal - All' displays a list of subscriptions with columns for Name and Subscriber.

| Personal - All | |
|--|---|
| Name | Subscriber |
| MyDevices | advantage/North America - HR & Adminis... |
| Handheld, PDA & Telephony | advantage/North America - HR & Adminis... |
| E-mail / Webmail (North America) | advantage/North America - HR & Adminis... |
| Printing (North America) | advantage/North America - HR & Adminis... |
| Intranet / Internet (North America) | advantage/North America - HR & Adminis... |
| Applications | advantage/North America - HR & Adminis... |

| Id | Key | |
|-----------|---------------------|---|
| 1. | Subscriptions | subscription.inbox.group.displayname |
| 2. | Personal | subscription.inbox.group.myuse.displayname |
| 3. | Departmental | subscription.inbox.group.department.displayname |
| 4. | Subscription Status | subscription.inbox.status.displayname |
| 5. | All | subscription.status.all |
| 6. | Personal - All | Dynamic value |

Customize the Knowledge Management Results



| Id | Key | Value |
|----|------------------------------|----------|
| 1. | search_facet.km.doctype | Doc Type |
| 2. | search_facet.km.category | Category |
| 3. | search_facet.facet_value_all | All |
| 4. | search_facet.facet_value_all | All |

The screenshot displays a service catalog interface with several facets and a list of services. On the left, there is a 'Category' facet with options: All, HR, Policy, Service Manager, Knowledge Man..., Knowledge Cen..., Support, Monitors, Telecom, and Phone. Below it is a 'Date' facet with options: All, Past Week, Past Month, and Past Year, each marked with a red circled number (1-5). The main content area shows search results for 'Quick tips for phone troubleshooting' (External, HR, HR, Phone), 'The UFFA Model and Quality Solutions' (Reference, Knowledge Centered Support (KCS)), 'What are the KCS Roles and Responsibilities?' (Question/Answer, Knowledge Centered Support (KCS)), 'Searching in Knowledge Management' (Reference, Knowledge Management), and 'Monitor problems' (Question/Answer, Monitors). On the right, there is a 'Service' facet with various service cards including 'Standard', 'New Mobi', 'New Work', 'New Appli', 'Standard', 'PC Backup \$0.00', and 'Performa'.

| Id | Key | Value |
|----|-----------------------------------|-----------|
| 1. | search_facet.km.last_updated | Date |
| 2. | search_facet.facet_value_all | All |
| 3. | search_facet.km.last_updated.week | Past Week |

| | | |
|----|------------------------------------|------------|
| 4. | search_facet.km.last_updated.month | Past Month |
| 5. | search_facet.km.last_updated.year | Past Year |

Customize: "Your Approvals"

The screenshot shows the 'Your Approvals' section of a user interface. On the left, there is a filter menu with the following options: 'Approval' (indicated by a red circle with '1'), 'All' (indicated by a red circle with '2'), 'Service' (indicated by a red circle with '3'), 'Change' (indicated by a red circle with '4'), and 'Quote' (indicated by a red circle with '5'). The main area displays a table of approval items under the 'All' filter. The table has columns for ID, Title, Type, Requested for, and Date. The items listed are:

| ID | Title | Type | Requested for | Date |
|-------|--------------------------------|-------|-------------------|-------|
| Q1009 | Monitor is broken, need to ... | Quote | BOWIE, GRAHAM | 08/13 |
| Q1003 | Need Office and Outlook | Quote | BAKER, SAMANTHA | 07/03 |
| Q1008 | Would like a new printer. | Quote | BARTON, CORINNE | 07/03 |
| Q1007 | Need a new Palm Pilot | Quote | ANALIZER, BARBERA | |

| Id | Key | |
|----|----------|------------------------------------|
| 1. | Approval | search_facet.approval.type |
| 2. | All | search_facet.facet_value_all |
| 3. | Services | search_facet.approval.type.service |

| | | |
|----|--------|-----------------------------------|
| 4. | Change | search_facet.approval.type.change |
| 5. | Quote | search_facet.approval.type.quote |

Note: The Approvals Page also requires some customization on the SRC client side. For more information, see ["Customize the Approvals Page" on page 105](#)

Customize the Request For Panel

 Required Fields.
▶ [Next](#)

Contact Information


This task depends on the results of the previous task.

Additional Information


This task depends on the results of the previous task.

1 Order Information


This task depends on the results of the previous task.

2 Details


This task depends on the results of the previous task.

3 Attachments


| Id | Key | |
|-----|--|---|
| 1. | Order Information | group_info.order_info.label |
| 1.1 | This request is for | order_info.requested_for.label |
| 2. | Details | group_info.details.label |
| 3. | Attachments(Support catalog/General support) | support_attachments.support_attachments_label |
| 4. | Attachments(Service catalog) | attachments_info.attachments_info_label |

SRC Service Manager Customization Reference

You must customize the following fields in Service Manager. To do this, edit the SRC checkout panels by navigating to **Tailoring** > **SRC Tailoring** in Service Manager. For more information on how to customize and configure the SRC checkout panels, see "[Tailoring Check-Out Panels](#)" on page 27.

1 Delivery Information i

Title

Description

! Required Fields. ▶ Next

2 Contact Information i

This task depends on the results of the previous task.

3 Additional Information i

This task depends on the results of the previous task.

Attachments i

This task depends on the results of the previous task.

| Id | Key |
|------------|-------------------------------|
| 1 | Delivery Information |
| 1.1 | Title |
| 1.2 | Description |
| 2 | Contact Information |
| 2.1 | Contact for this request |
| 2.2 | Contact Method |
| 2.3 | Request for (General support) |
| 3 | Additional Information |
| 3.1 | Urgency |
| 3.2 | Affected Service |

The Style.css File

The content below is provided as a convenient reference for the default style.css file that is included with SRC.

```

/* CSS file */

@font-face {
  fontFamily: "Collator";
  src: url("font/Collator.otf");
  embedAsCFF: true;
}

@font-face {
  fontFamily: "Gotham";
  src: url("font/Gotham-Book.otf");

```

```
embedAsCFF: true;
}
/*
text shape
*/
.headerTitle {
font-family: "Collator";
font-size: 21px;
color: #002561;
text-drop-shadow-alpha: 0.25;
}
.headerCartLink{
font-family: "Gotham";
font-size: 14px;
fontWeight: bold;
color: #336699;
textDecoration: "underline";
text-drop-shadow-alpha: 1;
}
.headerUserDropDownLabel{
font-family: "Gotham";
font-size: 14px;
fontWeight: bold;
color: #336699;
```

```
textDecoration: "underline";
text-drop-shadow-alpha: 1;
}
.footerTitle{
font-family: "Collator";
font-size: 21px;
color: #002561;
text-drop-shadow-alpha: 0.25;
}
.footerSubTitle{
font-family: "Gotham";
font-size: 12px;
color: #212F38;
text-drop-shadow-alpha: 0.25;
}
.footerLink{
font-family: "Gotham";
font-size: 14px;
color: #3c6296;
}
.footerHelpLink{
font-family: "Gotham";
font-size: 11px;
color: #666666;
```

```
text-drop-shadow-alpha: 0.25;
}
.footerCompanyLabel{
font-family: "Gotham";
font-size: 11px;
color: #666666;
text-drop-shadow-alpha: 0.25;
}
.footerLegalNoticeLabel{
font-family: "Gotham";
font-size: 11px;
color: #666666;
text-drop-shadow-alpha: 0.25;
}
.navigationLabelUp{
font-family: "Gotham";
font-size: 17px;
fontWeight: bold;
color: #336699;
text-drop-shadow-alpha: 0.25;
}
.navigationLabelSelected{
font-family: "Gotham";
font-size: 17px;
```

```
fontWeight: bold;
color: #FFFFFF;
text-drop-shadow-alpha: 0.25;
}
.navigationLabelOver{
font-family: "Gotham";
font-size: 17px;
fontWeight: bold;
color: #336699;
text-drop-shadow-alpha: 0.25;
}
.navigationLabelDisabled{
font-family: "Gotham";
font-size: 17px;
fontWeight: bold;
color: #CCCCCC;
text-drop-shadow-alpha: 0.25;
}
```