HP Service Request Catalog for Service Manager 9.30

for the Windows operating system

Software Version: 1.3

Installation and Configuration Guide

Document Release Date: July, 2011
Software Release Date: July, 2011

Last published date: Jun 01, 2011

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 Notices

© Copyright 2010-2011 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.

Oracle is a registered trademark of Oracle and/or its affiliates.

Documentation Updates

This guide's title page 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

You can visit the HP Software Support 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 Support Online 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 HP Software 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 find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

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

Contents

1 Install the Application

This installation of Service Request Catalog assumes that you have the following:

- An existing installation of Service Manager 9.30 server and client
- Service Manager administration experience
- Access to Service Manager documentation

Before You Begin

Make sure that you complete all installation tasks for upgrading your Service Manager server to version 9.30. Start with the HP Service Manager 9.30 Release Notes and follow the directions to apply listed patches and defect fixes.

When your Service Manager server is ready, read the following sections to install the Service Request Catalog components. Make sure that you meet the requirements before you begin the Service Request Catalog 1.30 installation.

Requirements

Service Request Catalog runs in a Windows or Unix environment using another HP application to serve and store data and requires the support of third party tools.

HP Application Compatibility

Service Request Catalog 1.30 new features work only with Service Manager 9.30. Service Manager 9.30 has enhancements that complement this version of Service Request Catalog. You must have both Service Manager server (RTE) and applications at version 9.30 to run Service Request Catalog 1.30 successfully.

Software and Hardware Requirements

Service Request Catalog supports various configurations that are identified as Tier 1 or Tier 2. Tier 1 recommendations are fully tested and recommended for optimum performance and functionality. Tier 2 recommendations are either earlier supported versions or alternatives to the Tier 1 recommendations. Make sure you review all hardware and software requirements described in the *HP Service Manager 9.30 Release Notes*, and that you check the HP Support site for any post-release patches for this version before you begin the Service Request Catalog installation process.

Third Party Resources

Service Request Catalog 1.30 requires you to install these external resources before you begin the installation process.

- Adobe® Flash® Player 10.2 or a later release
- Apache™ Tomcat™ 7.0.*x*

  Note: You can use an existing instance of Tomcat 7.0.*x*, or you can follow the steps on page 13 to install a new Tomcat server.

- The latest update version of Sun™ Java JDK™ 6 update 20 or a later release

If you plan to generate language packs, Language Builder requires InstallJammer 1.2.13 or a later release.

Servers and Operating Systems

The following table describes supported Service Request Catalog server and operating system combinations.

Tier	Server Platform	Operating System
Tier 1	HP Itanium	HP-UX 11i v3 (11.x)
Tier 1	x86-*xx*	Windows 2008R2 RedHat Enterprise Linux 5 RedHat Enterprise Linux 6
Tier 2	x86-64	Oracle Enterprise Linux 5
Tier 2	x86-32	Oracle Enterprise Linux 5

Client Operating Systems

The following table describes supported Service Request Catalog client configurations.

Tier	Server Platform	Operating System
Tier 1	x86-64	Windows 7 Windows Vista
Tier 2	x86-32	Windows 7 Windows Vista

Application and Web Servers

Apache Tomcat is an open source application server.

Tier	Version
Tier 1	Tomcat 7.0.*x*
Tier 2	Tomcat 6.0.*x*

Apache HTTP Server is an open source web server that you can locate on the same server as Tomcat or on a different server using the Apache JSserv Protocol (AJP) Connector.

Tier	Version
Tier 1	Apache HTTP Server 2.2x

Browsers

Service Request Catalog 1.30 runs on these browser versions.

- Microsoft® Internet Explorer® v8 (Tier 1), or v7 (Tier 2).
- Mozilla Firefox v3.6 (Tier 1).

Skill Requirements

Service Request Catalog deployment requires the administrator to have prior experience with security and authentication configuration tasks. Service Request Catalog supports these protocols:

- Lightweight Directory Access Protocol (LDAP)
- Lightweight Single Sign-On (LWSSO)
- Trusted Sign-On (TSO) using Secure Sockets Layer(SSL)/Transport Layer Security (TSL) protocols

Make sure you are familiar with each and that you are knowledgable about their configuration requirements.

Service Manager

You must have an installed instance of the Service Manager 9.30 server, and Service Manager 9.30 Windows client to complete configuration tasks and make administrative changes to the service catalog items. For more information about administrative functions, see the Service Manager documentation.

The installation process requires the name and encrypted password of a Service Manager system administrator. Before you begin, verify that this administrator has the SOAP API and service catalog capability words in their Service Manager profile.

Installation Contents

The Service Request Catalog installation package contains:

- A web application archive (`src-1.30.war`) file that creates the browser interface when you deploy it using Apache Tomcat.

- A web application archive (`bsf.war`) file that creates the login and authentication framework when you deploy it using Apache Tomcat.
- A simple encryption tool, (`encryptor-1.30.zip`) in a zip archive. The encryption tool encrypts the password of the Service Manager administrator.

Documentation Notes

Specified installation folder and path locations are generally relative to the location of the installed Tomcat instance and deployment location of the `src-1.30.war` file. When you see a path that includes a hard drive letter (C:\), the actual location always depends on the user's discretion. You can substitute the actual drive that you choose. When you see a path that includes an ellipsis (...), it represents the discretionary part of the path and folder structure on your local drive. Example: `C:\...\apache-tomcat-7.0.x`

The `C:\...\` notation assumes that you chose the default installation path for Apache Tomcat. When you encounter this path notation in examples, you can always substitute your local path.

For Unix, the default folder structure is always represented with this convention: `/opt/...`

You can assume all tasks and steps are required unless they are marked *Optional*.

Installation Tasks

The tasks in this chapter enable you to deploy the Service Request Catalog application and configure the authentication environment. The basic tasks and locations are:

Demo Mode

To deploy and configure Service Request Catalog to run in demo mode with no authentication, complete only Tasks 1, 2, 4, and 5.

Task 1. Install Apache Tomcat

Follow these steps to install Apache Tomcat on the Service Request Catalog server.

1. If necessary, navigate to this download site to obtain the zipped installation files for Apache Tomcat `7.0.x`: http://tomcat.apache.org/download-70.cgi

   Download the zip version to deploy manually or the 32-bit/64-bit Windows Service Installer.

2. Do one of the following to install Tomcat as a Windows Service or as a manual deployment:

   Windows Service: Run the `apache-tomcat-7.0.x.exe` installer. Specify a convenient location for Tomcat. Example:

   Windows: `C:\...\apache-tomcat-7.0.x`

   Unix: `/opt/..../apache-tomcat-7.0.x`

   When prompted for the Java location, note the path to the `\jre` folder. You will need this information later when you configure authentication.

   Manual Process: Unzip the files to a preferred location on the designated server. Example: Unzip the files to:

   Windows: `C:\...\apache-tomcat-7.0.x`

   Unix: `/opt/..../apache-tomcat-7.0.x`

3. After you install Tomcat, increase the Java Heap Space setting to prevent out of memory issues. Example:

   a. From the **Windows Start** menu, click **Programs** > **Apache Tomcat 7.0** > **Configure Tomcat** to display the Tomcat Properties dialog.

   b. Click the **Java** tab.

   c. Type a new value of **`1024`** for the **Initial memory pool** and also for the **Maximum memory pool**.

   d. Click **OK**.

Task 2. Deploy and Configure the Service Request Catalog .war File

Follow these steps to deploy and configure the `.war` file.

1. **Windows:** Navigate to the **Control Panel** > **Administrative Tools** > **Services** dialog. Verify the Apache Tomcat service is stopped.

   Unix: Run `/opt/.../apache-tomcat-7.0.x/bin/shutdown.sh`

2. Unzip the contents of the `serviceRequestCatalog.zip` file into an empty folder that you create to be the home location for this application.

   Example: If you create a new folder named `\src-1.30` where you plan to deploy the `.war` file, the result would look like this:

   Windows: `C:\...\src-1.30\war_file_contents`

   Unix: `/opt/.../src-1.30/war_file_contents`

Caution: Do not unzip the `.war` file into

Windows: `C:\...\apache-tomcat-7.0.x\webapps` or

Unix: `/opt/.../apache-tomcat-7.0.x/webapps`

Tip: Use a `.war` file extraction tool, or rename the file to `src-1.30.zip` and unzip it to the target location.

3 Open this file with a text editor:

Windows: `C:\...\Apache\apache-tomcat-7.0.x\conf\server.xml`

Unix: `/opt/.../apache-tomcat-7.0.x/conf/server.xml`

4 Navigate to the `<Host></Host>` section and locate the `Context` parameter. If the section is commented out, uncomment it. If it is missing, add the following.

```
<Context docBase=""
   path=""
   reloadable="false" />
```

5 Specify the complete path to the folder where you unzipped the `src-1.30.war` file in the `docBase=` parameter. Enclose the path in double quotation marks (`" "`). Example:

Windows: `<Context docBase="C:\...\src-1.30"`

Unix: `<Context docBase="/opt/.../src-1.30"`

6 The `path=` parameter enables you to create a simplified path (in the URL) that maps to the actual path where you deploy the `src-1.30.war` application. Enclose the path in double quotation marks (`" "`). Example:

Windows:

```
<Context docBase="C:\...\src-1.30"
   path="/src"
```

Unix:

```
<Context docBase="/opt/.../src-1.30"
   path="/src"
```

Tip: It is acceptable to create a multi-level context path for your deployment. Example:

```
path="/enterprise/xxx/src"
```

7 Save your changes but do not close the file.

Task 3. Deploy and Configure the BSF .war File

The Business Technology Optimization Security Framework (BSF) is an HP product that makes it easy to establish secure communication among HP applications.

1 Unzip the contents of the `bsf.war` file into an empty folder that you create to be the home location for this application.

Example: If you create a new folder named \bsf where you plan to deploy the .war file, the result would look like this:

Windows:

```
C:\...\bsf\
C:\...\bsf\conf
C:\...\bsf\db
```

Unix:

```
/usr/root/.../bsf/
/usr/root/.../bsf/conf
/usr/root/.../bsf/db
```

Caution: Do not unzip the war file into the Tomcat /webapps directory.

Windows: C:\...\apache-tomcat-7.0.*x*\webapps or

Unix: /opt/.../apache-tomcat-7.0.*x*/webapps

2 Return to this file that is already open with a text editor:

Windows: C:\...\apache-tomcat-7.0.*x*\conf\server.xml

Unix: /opt/.../apache/apache-tomcat-7.0.*x*/conf/server.xml

3 Navigate to the <Host></Host> section and locate the Context parameter that you created or changed in step 4 on page 14. Insert a second Context parameter above the original Context parameter.

```
<Context docBase=""
   path=""
   reloadable="false" />
```

4 Specify the complete path to the folder where you unzipped the bsf.war file in the docBase= parameter. Enclose the path in double quotation marks (" "). Example:

Windows: <Context docBase="C:\...\bsf"

Unix: <Context docBase="**/**opt**/...**/bsf"

5 The path= parameter enables you to create a simplified path (in the URL) that maps to the actual path where you deploy the BSF application. Enclose the path in double quotation marks (" "). Example:

Windows:

```
<Context docBase="C:\...\bsf"
   path="/bsf"
```

Unix:

```
<Context docBase="/opt/.../bsf"
   path="/bsf"
```

The final result looks like this:

Windows:

```
<Context docBase="C:\...\bsf"
   path="/bsf"
   reloadable="false" />
```

or

Unix:

```
<Context docBase="/opt/.../bsf"
   path="/bsf"
   reloadable="false" />
```

6 Save your changes and close the file.

Task 4. Create an Encrypted Password for the Service Manager Administrator

Follow these steps to create an encrypted password.

1 From the location where you unzipped serviceRequestCatalog.zip, extract the contents of `...\files\encryptor-1.30.zip` into a separate folder of your choice. Example:

Windows: `C:\...\src-1.30\encrypt`

Unix: `/opt/.../src-1.30/encrypt`

2 Run the executable file to start the encryptor application.

Windows: `C:\...\src-1.30\encrypt\runme.bat`

Unix: `/opt/..../src-1.30/encrypt/runme.sh`

3 At the command line prompt, type your Service Manager administration password. Your administrator rights must include SOAP API and service catalog capability.

```
Enter your password: password_value
Press Enter.
Is this correct? Y/N: Y
Press Enter.
```

4 The encryption tool returns an encrypted password value:

```
This is your encrypted password: encrypted_password
```

Copy this encrypted value and save it to use in a subsequent step.

Task 5. Configure Service Request Catalog Properties

Follow these steps to configure the related properties.

1 Open this file with a text editor:

Windows: `C:\...\src-1.30\WEB-INF\classes\applicationContext.properties`

Unix: `/opt/.../src-1.30/WEB-INF/classes/applicationContext.properties`

2 Change the properties in this file to match your client/server environment:

a Choose http or https protocol. For Trusted Sign-On, specify https.

```
serviceManager.protocol=https
```

b Specify the host server with a fully qualified domain name

```
serviceManager.hostname=hostname.domainName
```

c Specify a unique port number, or 13443, which is the default Service Manager https port number.

```
serviceManager.port=13443
```

3 In the **# Caching** section, locate this entry:

```
serviceManager.adminCredentials=LIST(userName,ENC(encryptedValue))
```

— Replace *userName* with the Service Manager administrator user name that is associated with the password that you encrypted in Task 4. Create an Encrypted Password for the Service Manager Administrator on page 16.

— Replace *encryptedValue* with the encrypted password text that you saved earlier in step 4 on page 16.

Note: If you configured Service Manager to operate in multi-company or multi-tenant mode, then you must specify a separate userName and encrypted password for each company. To enable more than one administrator, repeat step 2 on page 16 through step 4 on page 16 for each additional administrator.

4 Use the following syntax and separate each unique entry with a comma:

```
=LIST(userName,ENC(encryptedValue),userName,ENC(encryptedValue))
```

5 (Optional) In the first **# Performance** section, change this parameter:

```
src.reloadCatalogAfterEvery=21600000
```

if you want the catalog cache refreshed more (or less) frequently than every six hours. Express the value in milliseconds.

Tip: In a pre-production environment where frequent changes occur while you build or test the catalog, you might want a lower setting to update the catalog more often. In a production environment, specify a higher setting that does not impact performance.

6 (Optional) In the second **# Performance** section, change this parameter:

```
src.refreshLookupsAfterEvery=21600000
```

if you want the catalog cache refreshed more (or less) frequently than every six hours. Express the value in milliseconds.

7 (Optional) In the **# Exchange Rates** section, change this parameter:

```
src.refreshExchangeRatesAfterEvery=21600000
```

if you want exchange rates refreshed more (or less) frequently than every six hours. Express the value in milliseconds.

8 (Optional) In the **# SM config properties** section, change these parameters:

```
src.sm.defaultMaxConnectionsPerHost=25
src.sm.maxTotalConnections=25
```

Specify between 25 to 60 connections depending upon the user traffic that you expect. Increase this value by 10 for each additional Service Manager node. Do not exceed 60 regardless of the number of Service Manager nodes because it will impact database performance.

Example: If there are six or more Service Manager nodes supported by Loadbalancer, set this value to 60 connections.

9 (Optional) In the **# Support Catalog properties** section, you can change this parameter if you want to disable the user's ability to request support for anything that is not available through the standard Support Catalog. To disable non-catalog support, change true to false.

```
src.sm.canRequestGenericSupport=true
```

10 (Optional) In the **Contact Display Property** section, you can configure how the user appears in the Service Request Catalog interface when you search for a Service Manager contact. To change the default value, ContactName, to a different display format, you can specify:

- `${N}` to display the user name.
- `${C}` to display the user contact name.
- `${U}` to display the user full name.
- `${F}` to display the user first name.
- `${L}` to display the user last name.

You can define more than one property value and you can separate them with punctuation symbols.

Example: Specify `${L}, ${F}` if you want to display LastName, FirstName (Falcon, Jennifer).

```
src.sm.contactListDisplayPattern=${C}
```

11 Save and close the file.

Task 6. Configure BSF Properties

Make sure you confirm this configuration with your LDAP administrator.

1 Open this file with a text editor:

Windows: `C:\...\bsf\conf\bsf.properties`

Unix: `/opt/..../bsf/conf/bsf.properties`

2 In the #BSF topology section, change the `authentication.provider` parameter value to:

```
authentication.provider=EXTERNAL
```

3 Locate the bsf-test.server.url line.

4 Insert a number sign character (#) at the beginning of the line to make it a comment. The result looks like this:

```
# bsf-test.server.url=http://127.0.0.1:8080/bsf-test/
```

5 Save and close the file.

6 Open this file with a text editor:

Windows: `C:\...\bsf\conf\external-ldap.properties`

Unix: `/opt/..../bsf/conf/external-ldap.properties`

Caution: Make sure you understand LDAP conventions before you proceed with the remaining steps. The recommended configuration changes in subsequent steps are to enable BSF to run successfully. You may need to make other changes to support your local conventions.

7 The beginning of this file contains informational content that is preceded by a number sign (#) to make these comments only.

To begin editing, scroll down to this line where the configuration begins:

```
################## UserManagementLDAP configuration ###########
```

8 Search for this parameter:

```
ldapHost = localhost
```

9 Change the `ldapHost value` to point to the fully qualified domain name of your LDAP server. Example:

```
ldapHost = ldap.hp.com
```

10 Go to the next parameter:

```
ldapPort = 389
```

11 Change the `ldapPort` value to the port number of your LDAP server. Example:

```
ldapPort = 636
```

12 Go to the next parameter and set the correct value for `enableSSL`. Example:

```
enableSSL = true
```

Note: If `enableSSL = true`, make sure the certificate for the LDAP directory exists or is already added to the cacerts file using the `keytool -importcert` command. Example:

```
keytool -importcert -noprompt -alias alias_name -file
c:\certfile_name.crt -keystore cacerts -storepass changeit
```

If you are not sure what to specify, contact your LDAP administrator for assistance.

13 Go to the next parameter and set the correct value for `useAdministrator`. Example:

```
useAdministrator = false
```

Notes: If `useAdministrator = true`, follow the directions to specify the Administrator's user name and password. If you are unsure what to do, contact your LDAP administrator.

If `enableSSL= true`, usually `useAdministrator = false`. Make sure you check with your LDAP administrator to verify that these settings are correct for your environment.

14 Scroll down to this parameter:

```
usersBase =
```

15 Change the `usersBase=` value to your distinguished name. Example:

```
usersBase = ou=People,o=domain.com
```

16 Optional. You can comment out any unused parameters. If necessary, check with your LDAP administrator to verify that no other parameters need to be changed. The default values for the remaining parameters are compatible with a standard LDAP v3 SunOne Directory Server.

17 Save and close the file. If you do not support Active Directory, you can skip the next task and go to Task 8. Configure LWSSO Properties on page 20.

Task 7. Configure Active Directory Properties

For Active Directory, continue editing to complete the steps in this section. The recommended values are essential for a basic Active Directory configuration. You should review the remaining parameters to make sure they are correct for your environment.

1. Go to this parameter:

   `ldapAdministrator = cn=Directory Manager`

2. Change the `ldapAdministrator` value to point to the complete administrator description that is correct for your environment. Example:

   `ldapAdministrator = cn=Administrator, cn=Users, dc=domain, dc=com`

3. Go to this parameter:

   `usersObjectClass = inetOrgPerson`

4. Change the `usersObjectClass` value to the following:

   `usersObjectClass = user`

5. Scroll down to this parameter:

   `usersFilter = (&(uid=*)(objectclass=inetOrgPerson))`

6. Change the `usersFilter` value to point to the correct description for your environment. Example:

   `usersFilter=(&(`*`sAMAccountName`*`=*)(objectclass=user))`

7. Go to the next parameter:

   `usersUniqueIDAttribute = uid`

8. Change the `usersUniqueIDAttribute` value to point to the correct description for your environment. Example:

   `usersUniqueIDAttribute = `*`sAMAccountName`*

9. Go to this parameter:

   `usersLoginNameAttribute = uid`

10. Change the `usersLoginNameAttribute=` value to point to the correct description for your environment. Example:

    `usersLoginNameAttribute = `*`sAMAccountName`*

11. Save and close the file.

Task 8. Configure LWSSO Properties

Follow these steps to configure the LWSSO properties. You will open two files with the same name, but they are in different folders.

1. Open this file with a text editor:

   Windows: `C:\...\bsf\conf\lwssofmconf.xml`

   Unix: `/opt/.../bsf/conf/lwssofmconf.xml`

2 Navigate to the folder where you unzipped the src-1.30 war file archive.

Tip: See step 2 on page 13. Example:

Windows: `C:\...\src-1.30\`

Unix: `/opt/.../src-1.30/`

3 Open this file with a text editor:

Windows: `C:\...\src-1.30\WEB-INF\classes\lwssofmconf.xml`

Unix: `/opt/.../src-1.30/WEB-INF/classes/lwssofmconf.xml`

4 Change the `initstring` value in `C:\...\bsf\conf\lwssofmconf.xml` to match the `initstring` value in

Windows: `C:\...\src-1.30\WEB-INF\classes\lwssofmconf.xml`

Unix: `/opt/.../src-1.30/WEB-INF/classes/lwssofmconf.xml`

The `initstring` passphrase is like a password, but a longer sequence of words or text used to encrypt or decrypt a cookie passed in a session ID. The values must match but should not be the default values. This value MUST be the same with the LWSSO setting of any other HP products you want to integrate with Service Request Catalog.

Task 9. Configure the JAVA Options for Tomcat

Follow these steps to configure the JAVA options.

1 Navigate to this Apache Tomcat installation directory.

Windows: `C:\...\apache-tomcat-7.0.`*x*`\bin`

Unix: `opt/.../apache-tomcat-7.0.`*x*`/bin`

2 Double-click this application to display the Apache Tomcat `7.0.`*x* Properties dialog:

Windows: `C:\...\apache-tomcat-7.0.`*x*`\bin\tomcat7w.exe`

Unix: `opt/.../apache-tomcat-7.0.`*x*`/bin/tomcat7w.exe`

3 Click the **Java** tab.

4 In the Java Options text box, type new path variables and values:

```
-DBSF_HOME=
-Dbsf_home=
-DBSF_CONF=
-Dbsf_conf=
-Dbsf.properties.configuration=true
```

Each path value maps the actual location of these two directories that you created in step 1 on page 14:

```
C:\...\bsf\
C:\...\bsf\conf
```

You are creating case-sensitive path variables for each location. The result should look like this:

```
-DBSF_HOME=C:\...\bsf\
-Dbsf_home=C:\...\bsf\
-DBSF_CONF=C:\...\bsf\conf
-Dbsf_conf=C:\...\bsf\conf
-Dbsf.properties.configuration=true
```

5 Click **Apply**.

6 In the Java Options text box, look for a the Java MaxPermSize variable. If you cannot find it, type the new variable and set this value as a minimum:

```
-XX:MaxPermSize=512m
```

7 Click **OK**.

2 Authentication

HP products offer more than one way to authenticate users.

- Trusted Sign-On (TSO) is a type of authentication supported by a variety of HP products, including Service Manager.
- Lightweight Single Sign-On (LWSSO) is an authentication standard shared by multiple HP software products. It is a modular framework that validates different types of authentication or security tokens. LWSSO extends authentication information in shared environments, such as Service Manager and Service Request Catalog.

For secure interaction between Service Request Catalog and Service Manager 9.30, you can use LWSSO or Trusted Sign-On as your authentication method, depending on the authentication solution selected by Service Manager.

The BTO Security Framework (BSF) is a shared security component that simplifies integration among HP products. It is an authentication framework that performs the actual authentication. You will deploy BSF as a task in the authentication configuration process.

Requirements

It helps to be familiar with these LWSSO requirements before you begin the process to specify configuration settings. You may need to review this table again as you go through the configuration process.

Parameter	Description
GMT Time	All applications participating in an LWSSO integration must use the same GMT time with a maximum difference of 15 minutes.
Multi-domain Functionality	All applications with LWSSO integration should configure the `<trustedHosts>` settings if they are required to integrate with applications in different DNS domains. In addition, they must also add the correct domain in the LWSSO element of the configuration.
Get SecurityToken for URL Functionality	To receive information sent as a SecurityToken for URL from other applications, the host application should configure the correct domain in the LWSSO element of the configuration.

Unix

All commands are shown in Windows format. For Unix installations, convert them to match your local Unix path conventions.

OpenSSL

OpenSSL is an open source toolkit for the SSL protocol that contains a variety of utility functions. You can obtain the OpenSSL toolkit from a variety of sources. You will need it to complete required tasks.

You can obtain OpenSSL if you install Cygwin, which is available at http://www.cygwin.com/

Although Cygwin has multiple components, you need only the following:

- OpenSSL runtime environment
- OpenSSL base environment

Install it in a directory that you can easily navigate to at the command line. Example: `C:\cygwin`.

You need OpenSSL on both the Service Request Catalog and Service Manager servers. You can simplify running openssl commands if you create an environment variable that specifies the openssl location: `cygwin\bin\openssl`.

Cygwin is governed by the GNU General Public License.

Verify: Open the Cygwin Command Prompt window by double-clicking `C:\cygwin\Cygwin.bat`. You will need this window later. Confirm that you have OpenSSL available by typing this command:

```
openssl -help
```

Java

Make sure there is a JAVA_HOME environment variable configured on the Service Manager server and on the Service Request Catalog server. Service Manager installs a complete JRE in this location:

`C:\Program Files\HP\Service Manager 9.30\Server\RUN\jre\bin`.

Verify: Open a Windows Command Prompt window. Confirm that you have the keytool application available by typing this command at the prompt:

```
keytool -help
```

If an error message appears, confirm that you have JAVA_HOME configured correctly and that it includes the `\bin` folder in that path.

On the Service Request Catalog server, JAVA_HOME should point to the same Java instance used by Tomcat.

Important: Each instance of a Java Runtime or Java Developer's Kit contains a `...\lib\security\cacerts` file. Some test or production servers could have more than one version installed. It is important that you consistently reference the same Java instance that is used by Tomcat.

Configuration Tips

During the configuration process, you need to access certain folders and files. Create a separate Windows Explorer window for each.

- You may want to set up a remote desktop connection in advance to the server where the Service Manager server is installed.
  - On the Service Manager server, have the following folder available. The root may vary, depending on your installation conventions.

    ```
    C:\Program Files\HP\Service Manager 9.30\Server\RUN
    ```

- You may want to set up a remote desktop connection in advance to the server where Service Request Catalog is deployed.
  - On the Service Request Catalog server, locate your installation of Java. The root may vary, depending on your installation conventions. You will need to access this file:

    ```
    C:\Program Files\Java\jdk1.6.0_20\jre\lib\security\cacerts
    ```

  - On the Service Request Catalog server, create a new folder that you can locate easily where you can store new security files related to Trusted Sign-On. Example:

    ```
    C:\...\src-1.30\_TSO\
    ```

  - On the Service Request Catalog server, locate the folder where you unzipped the `.war` file. You will need to access these files:

    ```
    C:\...\src-1.30\WEB-INF\classes\applicationContext.properties
    C:\...\src-1.30\WEB-INF\classes\lwssofmconf.properties
    ```

    Tip: Set up separate windows for each of these folders and files ahead of time to speed up the configuration process. Arranging them side by side will make it easy to switch from one to the other.

Verify Server Communication

HP recommends that you verify that the Service Request Catalog server and the Service Manager server can communicate.

1. From the Service Request Catalog server, open a Command Prompt window and ping the Service Manager server using its *fully qualified domain name*.
2. From the Service Manager server, open a Command Prompt window and ping the Service Request Catalog server using its fully qualified domain name.

If you have any communications failures, add the fully qualified domain name and IP of the other server to the local Hosts file.

Tip: Keep both Command Prompt windows open. You will need them to complete subsequent steps.

Troubleshooting

If a command fails when you run openssl commands that reference files in various locations, the problem may be that you are not running the command from the file directory, or you do not have an environment variable defined for openssl that enables you to run from any directory.

Getting Started

Before you begin, verify that you completed all steps in Chapter 1, Install the Application to deploy and configure Service Request Catalog.

There are two possible scenarios that determine what you need to do next.

To use LWSSO for Service Request Catalog authentication and Trusted Sign-On for web service communication, complete these tasks:

Configure Security Settings

These tasks guide you though configuring Service Request Catalog authentication.

Task 1: Configure the applicationContext.properties File

Complete these tasks on the Service Request Catalog server.

1. Open this file with a text editor:

   `C:\...\src-1.30\WEB-INF\classes\applicationContext.properties`

2. Locate the # Service Manager Protocol section.
3. Change the protocol from http to https:

   `serviceManager.protocol=https`

4. Locate the # Service Manager Port section. Change the port number to match the https port number specified by the Service Manager `sm.ini` file.

   Tip: Locate this Service Manager file before you begin this step:

   `C:\Program Files\HP\Service Manager 9.30\Server\RUN\sm.ini`

   Use information in `sm.ini` to specify correct values in `applicationContext.properties`.

   The result should look like this if the default https port value is used.

   `serviceManager.port=13443`

5. Locate the # Security Mode section.

6 Remove the comment symbol (#) and space that precedes this parameter:

```
# src.security.mode=bsfLwssoUiAndTsoWs
```

The result should look like this:

```
src.security.mode=bsfLwssoUiAndTsoWs
```

7 Insert a comment symbol (#) and space before the default mode parameter:

```
src.security.mode=default
```

The result should look like this:

```
# src.security.mode=default
```

8 Save the file but keep it available in the text editor for future reference.

Task 2. Import the CA Certificate for Service Request Catalog

Follow these steps *only* if you do not have a digital security certificate issued by a certificate authority provider, such as Verisign, Thawte, or your corporate certificate authority. The digital certificate contains a public key, the identity of the owner, and a matching private key. The certificate is required to encrypt data sent and received in a "trusted" environment. If you do not have a digital certificate from an external provider, you must create your own certificate to enable encrypted data transfer between the Service Manager server and the trusted Service Request Catalog application.

1 From the Cygwin Command Prompt window, change directories and navigate to the new folder you created to contain Trusted Sign-On files. Example:

```
cd "C:\...\src-1.30\_TSO\"
```

2 From this directory, run the following openssl commands to generate a private key for the certificate authority.

Command	Description
`genrsa`	Generate an RSA private key
`-des3`	A cipher methodology
`-passout pass:`	Specify a password for the created file
`-out` *`filename.pem`*	Create an output file

Copy and paste this command into your Cygwin Command Prompt window:

```
openssl genrsa -des3 -passout pass:changeit -out
certificateAuthorityKey.pem 1024
```

Copy and paste is a shortcut to reduce typing errors and improve efficiency. However, the commands are long with many parameters. Make sure you copy the *entire* command, even when it wraps over multiple lines. When you paste it into the Command Prompt window, it will appear as a single line.

Verify: This folder should contain a new file named `certificateAuthorityKey.pem`.

3 The following openssl commands generate a self-signed certificate for the certificate authority.

Command	Description
`req`	Request a certificate
`-new`	The certificate is new
`-x509`	x509 is a self-signed certificate utility (For more information, see **www.openssl.org**)
`-days 365`	Expires after one year
`-key certificateAuthorityKey.pem`	Specify the certificate file name name
`-passin pass:`	Specify a password for the certificate file
`-subj "/CN=`*`www.xxx.com`*`"`	If you were using an external certificate authority, this would be their domain name. Because we are creating our own, the domain is not relevant but the command must have some reference.
`-out` *`filename.pem`*	Create this certificate file

Copy and paste this command into your Cygwin Command Prompt window:

```
openssl req -new -x509 -days 365 -key certificateAuthorityKey.pem
-passin pass:changeit -subj "/CN=www.xxx.com" -out
certificateAuthorityCert.pem
```

Verify: This folder should contain a new file named `certificateAuthorityCert.pem`

4 From the Windows Command Prompt window, navigate to the new folder you created to contain Trusted Sign-On files. Example:

```
C:\...\src-1.30\_TSO\
```

5 The following keytool commands generate a public/private key pair for Service Request Catalog. The trusted certificate entry is stored in an entity known as a *keystore*. For more information about keytool, see **www.oracle.com**.

Command	Description
`-genkey`	Generate a *key pair*, which contains a public key and associated private key pair
`-dname "CN=`*`fqdn`*`"`	Specify the fully qualified domain name of the Service Request Catalog server
`-validity 365`	Expires after one year
`-alias` *`alias_name`*	Creates a unique name, or alias, for a new trusted certificate entry. In this case, use src as the *alias_name*.
`-keypass`	Specify a password for the certificate file
`-keystore`	Specify the name of the Service Manager keystore file.
`-storepass`	Specify a password for the keystore file.

Copy and paste this command into your Windows Command Prompt window:

Tip: Copy it into a plain text file first, substitute your fully qualified domain name for `fqdn`, and then copy and paste it into the command line. Make sure you use the fully qualified domain name, not an IP address. A fully qualified domain name looks like this: *server.name.qualifier*.

```
keytool -genkey -dname "CN=fqdn" -validity 365 -alias src -keypass
changeit -keystore srcKeystore.jks -storepass changeit
```

Verify: This folder should contain a new file named `srcKeystore.jks`.

If you get a message that says keytool not found, verify that you completed the steps in Java on page 24.

6 The following keytool commands generate a Certificate Signing Request (CSR) for Service Request Catalog.

Command	Description
`-certreq`	Generate a Certificate Signing Request (CSR)
`-alias` *alias_name*	src is the alias name
`-file`	The file name is srcCSR.pem
`-keypass`	Specify a password for the .pem file
`-keystore`	Specify the name of the Service Request Catalog keystore file.
`-storepass`	Specify a password for the keystore file.

Copy and paste this command into your Windows Command Prompt window:

```
keytool -certreq -alias src -file srcCSR.pem -keypass changeit -keystore
srcKeystore.jks -storepass changeit
```

Verify: This folder should contain a new file named `srcCSR.pem`.

7 Return to your Cygwin Command Prompt window.

8 Run the following openssl commands to sign the CSR that you created in step 6.

Command	Description
`x509`	x509 is a self-signed certificate utility (For more information, see **www.openssl.org**)
`-req`	Request a certificate
`-in` *filename.pem*	CSR file name
`-CA` *filename.pem*	File name of the certificate to be signed
`-CAkey` *filename.pem*	File that contains the private key
`-passin pass:`	Specify the password for the key file

Command	Description
`-days 365`	Expires in one year
`-set_serial`	Optional. Serial number of the file (Can replace the -CA parameter)
`-out filename.pem`	Create an output file for the signed certificate

Copy and paste this command into your Cygwin Command Prompt window:

```
openssl x509 -req -in srcCSR.pem -CA certificateAuthorityCert.pem -CAkey
certificateAuthorityKey.pem -passin pass:changeit -days 365 -set_serial 1
-out srcCert.pem
```

Verify: You should see these messages display in the command window:

```
Signature ok
subject=/CN=yourFQDN
Getting CA Private Key
```

This folder should contain a new file named `srcCert.pem`.

9 Return to your Windows Command Prompt window.

10 The following keytool commands import the Certificate Authority certificate into the Service Request Catalog keystore file.

Command	Description
`-importcert`	Read the certificate (or certificate chain) from the file and put it in the *alias* keystore
`-noprompt`	No user input required
`-alias alias_name`	certificateAuthority is the alias name for the **`certificateAuthorityCert.pem file`**
`-keypass`	Specify the password for the **`certificateAuthorityCert.pem`** file
`-file`	The file name is **`certificateAuthorityCert.pem`**
`-keystore`	Specify the name of the Service Request Catalog keystore file
`-storepass`	Specify the password for the Service Request Catalog keystore file.

Copy and paste this command into your Windows Command Prompt window:

```
keytool -importcert -noprompt -alias certificateAuthority -keypass
changeit -file certificateAuthorityCert.pem -keystore srcKeystore.jks
-storepass changeit
```

Verify: You should see the following message display in the command window.

```
Certificate was added to keystore
```

11 The following keytool commands import the signed certificate file for Service Request Catalog into the Service Request Catalog keystore file.

Command	Description
`-importcert`	Read the certificate (or certificate chain) from the file and put it in the *alias* keystore
`-alias alias_name`	src is the alias for the srcCert.pem
`-keypass`	Specify the password for srcCert.pem file
`-file`	The file name is **`certificateAuthorityCert.pem`**
`-keystore`	Specify the name of the Service Request Catalog keystore file
`-storepass`	Specify the password for the Service Request Catalog keystore file.

Copy and paste this command into your Windows Command Prompt window:

```
keytool -importcert -alias src -keypass changeit -file srcCert.pem
-keystore srcKeystore.jks -storepass changeit
```

Verify: You should see the following message display in the command window.

```
Certificate reply was installed in keystore
```

Follow these steps *only* if you do not have a digital security certificate issued by a certificate authority provider, such as Verisign, Thawte, or your corporate certificate authority. The digital certificate contains a public key, the identity of the owner, and a matching private key. The certificate is required to encrypt data sent and received in a “trusted” environment. If you do not have a digital certificate from an external provider, you must create your own certificate to enable encrypted data transfer between the Service Manager server and the trusted Service Request Catalog application.

1 From the Cygwin Command Prompt window, change directories and navigate to the new folder you created to contain Trusted Sign-On files. Example:

```
cd "C:\...\src-1.30\_TSO\"
```

2 From this directory, run the following openssl commands to generate a private key for the certificate authority.

Command	Description
`genrsa`	Generate an RSA private key
`-des3`	A cipher methodology
`-passout pass:`	Specify a password for the created file
`-out` *`filename.pem`*	Create an output file

Copy and paste this command into your Cygwin Command Prompt window:

```
openssl genrsa -des3 -passout pass:changeit -out
certificateAuthorityKey.pem 1024
```

Copy and paste is a shortcut to reduce typing errors and improve efficiency. However, the commands are long with many parameters. Make sure you copy the *entire* command, even when it wraps over multiple lines. When you paste it into the Command Prompt window, it will appear as a single line.

Verify: This folder should contain a new file named `certificateAuthorityKey.pem`.

3 The following openssl commands generate a self-signed certificate for the certificate authority.

Command	Description
`req`	Request a certificate
`-new`	The certificate is new
`-x509`	x509 is a self-signed certificate utility (For more information, see **www.openssl.org**)
`-days 365`	Expires after one year
`-key certificateAuthorityKey.pem`	Specify the certificate file name name
`-passin pass:`	Specify a password for the certificate file
`-subj "/CN=`*`www.xxx.com`*`"`	If you were using an external certificate authority, this would be their domain name. Because we are creating our own, the domain is not relevant but the command must have some reference.
`-out` *`filename.pem`*	Create this certificate file

Copy and paste this command into your Cygwin Command Prompt window:

```
openssl req -new -x509 -days 365 -key certificateAuthorityKey.pem
-passin pass:changeit -subj "/CN=www.xxx.com" -out
certificateAuthorityCert.pem
```

Verify: This folder should contain a new file named `certificateAuthorityCert.pem`

4 From the Windows Command Prompt window, navigate to the new folder you created to contain Trusted Sign-On files. Example:

```
C:\...\src-1.30\_TSO\
```

5 The following keytool commands generate a public/private key pair for Service Request Catalog. The trusted certificate entry is stored in an entity known as a *keystore*. For more information about keytool, see **www.oracle.com**.

Command	Description
`-genkey`	Generate a *key pair*, which contains a public key and associated private key pair
`-dname "CN=`*`fqdn`*`"`	Specify the fully qualified domain name of the Service Request Catalog server
`-validity 365`	Expires after one year

Command	Description
`-alias alias_name`	Creates a unique name, or alias, for a new trusted certificate entry. In this case, use src as the *alias_name*.
`-keypass`	Specify a password for the certificate file
`-keystore`	Specify the name of the Service Manager keystore file.
`-storepass`	Specify a password for the keystore file.

Copy and paste this command into your Windows Command Prompt window:

Tip: Copy it into a plain text file first, substitute your fully qualified domain name for `fqdn`, and then copy and paste it into the command line. Make sure you use the fully qualified domain name, not an IP address. A fully qualified domain name looks like this: *server.name.qualifier*.

```
keytool -genkey -dname "CN=fqdn" -validity 365 -alias src -keypass
changeit -keystore srcKeystore.jks -storepass changeit
```

Verify: This folder should contain a new file named `srcKeystore.jks`.

If you get a message that says keytool not found, verify that you completed the steps in Java on page 24.

6 The following keytool commands generate a Certificate Signing Request (CSR) for Service Request Catalog.

Command	Description
`-certreq`	Generate a Certificate Signing Request (CSR)
`-alias alias_name`	src is the alias name
`-file`	The file name is srcCSR.pem
`-keypass`	Specify a password for the .pem file
`-keystore`	Specify the name of the Service Request Catalog keystore file.
`-storepass`	Specify a password for the keystore file.

Copy and paste this command into your Windows Command Prompt window:

```
keytool -certreq -alias src -file srcCSR.pem -keypass changeit -keystore
srcKeystore.jks -storepass changeit
```

Verify: This folder should contain a new file named `srcCSR.pem`.

7 Return to your Cygwin Command Prompt window.

8 Run the following openssl commands to sign the CSR that you created in step 6.

Command	Description
`x509`	x509 is a self-signed certificate utility (For more information, see **www.openssl.org**)
`-req`	Request a certificate
`-in` *`filename.pem`*	CSR file name
`-CA` *`filename.pem`*	File name of the certificate to be signed
`-CAkey` *`filename.pem`*	File that contains the private key
`-passin pass:`	Specify the password for the key file
`-days 365`	Expires in one year
`-set_serial`	Optional. Serial number of the file (Can replace the -CA parameter)
`-out` *`filename.pem`*	Create an output file for the signed certificate

Copy and paste this command into your Cygwin Command Prompt window:

```
openssl x509 -req -in srcCSR.pem -CA certificateAuthorityCert.pem -CAkey
certificateAuthorityKey.pem -passin pass:changeit -days 365 -set_serial 1
-out srcCert.pem
```

Verify: You should see these messages display in the command window:

```
Signature ok
subject=/CN=yourFQDN
Getting CA Private Key
```

This folder should contain a new file named `srcCert.pem`.

9 Return to your Windows Command Prompt window.

10 The following keytool commands import the Certificate Authority certificate into the Service Request Catalog keystore file.

Command	Description
`-importcert`	Read the certificate (or certificate chain) from the file and put it in the *alias* keystore
`-noprompt`	No user input required
`-alias` *`alias_name`*	certificateAuthority is the alias name for the **`certificateAuthorityCert.pem file`**
`-keypass`	Specify the password for the **`certificateAuthorityCert.pem`** file

Command	Description
`-file`	The file name is **`certificateAuthorityCert.pem`**
`-keystore`	Specify the name of the Service Request Catalog keystore file
`-storepass`	Specify the password for the Service Request Catalog keystore file.

Copy and paste this command into your Windows Command Prompt window:

```
keytool -importcert -noprompt -alias certificateAuthority -keypass
changeit -file certificateAuthorityCert.pem -keystore srcKeystore.jks
-storepass changeit
```

Verify: You should see the following message display in the command window.

```
Certificate was added to keystore
```

11 The following keytool commands import the signed certificate file for Service Request Catalog into the Service Request Catalog keystore file.

Command	Description
`-importcert`	Read the certificate (or certificate chain) from the file and put it in the *alias* keystore
`-alias alias_name`	src is the alias for the srcCert.pem
`-keypass`	Specify the password for srcCert.pem file
`-file`	The file name is **`certificateAuthorityCert.pem`**
`-keystore`	Specify the name of the Service Request Catalog keystore file
`-storepass`	Specify the password for the Service Request Catalog keystore file.

Copy and paste this command into your Windows Command Prompt window:

```
keytool -importcert -alias src -keypass changeit -file srcCert.pem
-keystore srcKeystore.jks -storepass changeit
```

Verify: You should see the following message display in the command window.

```
Certificate reply was installed in keystore
```

Task 3. Import Trusted CA Certificate

If you support Trusted Sign-On authentication, these steps are required. Use either a digital security certificate issued by a certificate authority provider or a self-signed certificate to complete the task.

Complete these steps in the Windows Command Prompt window on the Service Request Catalog server.

1. Navigate to your JAVA_HOME location where the `cacerts` file resides. For example:

   ```
   C:\...\jre\lib\security\cacerts
   ```

2. Navigate to the folder that you created to store new security files related to Trusted Sign-On. Example:

   ```
   C:\...\src-1.30\_TSO\
   ```

3. Copy this file:

   ```
   C:\...\src-1.30\_TSO\certificatAuthoritycert.pem
   ```

4. Paste it in the JAVA_HOME location where the `cacerts` file resides. Example:

   ```
   C:\...\jre\lib\security\certificatAuthoritycert.pem
   ```

5. Copy this file:

   ```
   C:\...\src-1.30\_TSO\certificatAuthorityKey.pem
   ```

6. Paste it in the same JAVA_HOME location.

   Verify: The result should be two new files in the same folder that contains the cacerts file.

   ```
   C:\...\jre\lib\security\cacerts
   C:\...\jre\lib\security\certificatAuthoritycert.pem
   C:\...\jre\lib\security\certificatAuthorityKey.pem
   ```

7. Import the signing certificate authority's public certificate into cacerts to establish a chain of trust:

   ```
   keytool -importcert -noprompt -alias certificateAuthority -keypass
   changeit -file certificateAuthorityCert.pem -keystore cacerts -storepass
   changeit
   ```

   Verify: The keytool application displays a confirmation message that says `Certificate was added to the keystore.`

Task 4. Copy Files to the Service Manager Server

You need to copy some files from the Service Request Catalog server to the Service Manager server before you start the next task.

1. From your JAVA_HOME location on the Service Request Catalog server, copy this file:

   ```
   C:\...\jre\lib\security\cacerts
   ```

2. On the Service Manager server, paste it into this folder:

   ```
   C:\Program Files\HP\Service Manager 9.30\Server\RUN
   ```

3. On the Service Request Catalog server, copy this self-signed certificate file:

   ```
   certificateAuthorityCert.pem
   ```

4. Paste it into this folder:

   ```
   C:\Program Files\HP\Service Manager 9.30\Server\RUN
   ```

5. On the Service Request Catalog server, copy this self-signed certificate key file:

   ```
   certificateAuthorityKey.pem
   ```

6 Paste it into this folder:

C:\Program Files\HP\Service Manager 9.30\Server\RUN

Task 5. Import the CA Certificate for Service Manager

If you support Trusted Sign-On authentication, and you do not have a digital security certificate issued by a certificate authority provider, such as Verisign, Thawte, or your corporate certificate authority, these steps are required. You will need the same certificate authority that you used to create your own keystore and certificate for Service Request Catalog.

Complete this task on the Service Manager server to enable encrypted data transfer between the Service Manager server and the trusted Service Request Catalog application.

1 In a Windows Command Prompt window, navigate to this folder:

```
C:\Program Files\HP\Service Manager 9.30\Server\RUN
```

2 From this directory, run `keytool.exe` to generate a public/private key pair for Service Manager. Copy and paste this command into your Windows Command Prompt window:

```
keytool -genkey -dname "CN=fqdn" -validity 365 -alias sm -keypass changeit
-keystore smKeystore.jks -storepass changeit
```

Tip: Copy it into a plain text file first, substitute your fully qualified domain name, and then copy and paste it into the command line.

Verify: The `C:\Program Files\HP\Service Manager 9.30\Server\RUN` directory should contain a new file named `smKeystore.jks`.

3 Run `keytool.exe` again to generate a Certificate Signing Request (CSR) for Service Request Catalog. Copy and paste this command into your Windows Command Prompt window:

```
keytool -certreq -alias sm -file smCSR.pem -keypass changeit -keystore
smKeystore.jks -storepass changeit
```

Verify: The `C:\Program Files\HP\Service Manager 9.30\Server\RUN` directory should contain a new file named `smCSR.pem`.

4 Open a Cygwin Command Prompt window on the Service Manager server.

5 Navigate to this folder:

```
cd "C:\Program Files\HP\Service Manager 9.30\Server\RUN"
```

6 Run openssl with the following commands to sign the CSR.

Copy and paste this command into your Cygwin Command Prompt window:

```
openssl x509 -req -in smCSR.pem -CA certificateAuthorityCert.pem -CAkey
certificateAuthorityKey.pem -passin pass:changeit -days 365 -set_serial 1
-out smCert.pem
```

Verify: You should see these messages display in the command window:

```
Signature ok
subject=/CN=yourFQDN
Getting CA Private Key
```

This folder should contain a new file named `smCert.pem`.

7 Return to your Windows Command Prompt window.

8 Run `keytool.exe` again with the following commands to import the Certificate Authority certificate into the Service Manager keystore file.

Copy and paste this command into your Windows Command Prompt window:

`keytool -importcert -noprompt -alias certificateAuthority -keypass changeit -file certificateAuthorityCert.pem -keystore smKeystore.jks -storepass changeit`

Verify: You should see the following message display in the command window.

```
Certificate was added to keystore
```

9 Run `keytool.exe` again with the following commands to import the signed certificate file for Service Manager into the Service Manager keystore file.

Copy and paste this command into your Windows Command Prompt window:

`keytool -importcert -alias sm -keypass changeit -file smCert.pem -keystore smKeystore.jks -storepass changeit`

Verify: You should see the following message display in the command window.

```
Certificate reply was installed in keystore
```

10 On the Service Request Catalog server, copy the Service Request Catalog self-signed certificate file:

```
srcCert.pem
```

11 On the Service Manager server, paste it into this folder:

```
C:\Program Files\HP\Service Manager 9.30\Server\RUN
```

12 Run `keytool.exe` with the following commands to import the Service Request Catalog certificate as a trusted client of Service Manager:

```
keytool -importcert -noprompt -alias src -keypass changeit -file
srcCert.pem -keystore clientcerts.keystore -storepass changeit
```

Task 6. Configure Trusted Sign-On Parameters

If you are using a combination of LWSSO and Trusted Sign-On, complete these steps on the Service Request Catalog server.

1 In the `applicationContext.properties` file, locate the `# Trusted Sign-On` section.

2 Locate the `src.trustStore` parameter.

3 Change the `src.trustStore` parameter to point to the cacerts file that contains the public certificates for the Certificate Authorities that signed the Service Manager and Service Request Catalog server certificates.

For example, if the path to the cacerts file is your JAVA_HOME directory:

```
C:\...\jre\lib\security\cacerts
```

the src.trustStore parameter setting would be:

```
src.trustStore=C:\\...\\jre\\lib\\security\\cacerts
```

4 Locate the `src.trustStorePassword` parameter.

5 Change the `src.trustStorePassword=` parameter to specify the password required to edit the trustStore file. Example:

   `src.trustStorePassword=changeit`

6 Locate the `src.keyStore` parameter.

7 Change the `src.keyStore=` value to point to the path and name of the keyStore file used to generate the public/private key-pair for the application. Example:

   `src.keyStore=C:\\SRC\\src-1.30\\_TSO\\srcKeystore.jks`

8 Locate the `src.keyStorePassword` parameter.

9 Change the `src.keyStorePassword=` to specify the password required to edit the keyStore file. Example:

   `src.keyStorePassword=changeit`

10 Save and close the file.

11 On the Service Manager server, in the `sm.ini` file, insert a comment to disable this parameter:

   `sslConnector:0`

   The result looks like this:

   `# sslConnector:0`

12 On the Service Manager server, copy the following parameters and paste them at the bottom of the `sm.ini` file:

```
## enable for SSL connection
sslConnector:1
keystoreFile:smKeystore.jks
keystorePass:changeit
ssl:1
trustedsignon:1
truststoreFile:cacerts
truststorePass:changeit

## enable for SSL/TLS connection
ssl_reqClientAuth:2
ssl_trustedClientsPwd:changeit
ssl_trustedClientsJKS:clientcerts.keystore
```

13 Save and close the file.

Task 7. Configure LWSSO

LWSSO enables a user to log on once and gain access to the resources of multiple software systems without being prompted to log on again. The applications inside the configured group of software systems trust the authentication, and further authentication is not required when moving from one application to another. This is especially useful if you are implementing a solution that involves multiple HP applications. To enable LWSSO, complete these steps.

1 Open this file with a text editor:

   `C:\...\src-1.30\WEB-INF\classes\lwssofmconf.xml`

2 Locate the `initString` parameter within the `<crypto></crypto>` element.

```
<crypto cipherType="symmetricBlockCipher"
        engineName="AES" paddingModeName="CBC" keySize="256"
        encodingMode="Base64Url"
        initString="changeit"></crypto>
```

3 Change the `initString` value to a shared secret passphrase, which can be any phrase you choose. The passphrase is like a password, but a longer sequence of words or text used to encrypt or decrypt a cookie passed in a session ID.

To preserve the validity of the `<crypto></crypto>` element, do not insert an extra line break before the `</crypto>` end tag when you change the `initString` value.

Make sure that this value is identical on all systems where you want to enable seamless LWSSO compatibility.

4 Locate the `<domain>` parameter.

```
<webui>
   <validation>
      <in-ui-lwsso>
         <lwssoValidation id="ID000001">
            <domain>your.enterprise.domain</domain>
```

5 Replace *`your.enterprise.domain`* with your own domain name. Example: `my.corporation.net`

Tip: You can locate your domain value in My Computer properties.

Best Practice: In all cases, the domain should be a fully qualified domain name to pass a cookie in an HTTP session that will be readable by any URL within that domain. The domain name should have at least two segments. (For example, `hp.com` is a valid domain.) If you can reference a more specific domain, such as `americas.hp.com`, you will narrow the scope of the domain.

6 Locate the `validationPoint` section:

```
<validationPoint
   enabled="true"
   refid="ID000002"
   authenicationPointServer="http://domain.name:port/bsf_location"/>
</validation>
```

7 Replace `http://`*`domain.name:port`* with the fully qualified domain name of the server where you deployed the Service Request Catalog `.war` file.

8 Replace */`bsf_location`* with the name of the directory where you deployed the `bsf.war` file.

9 Each application using LWSSO should configure token expiration. The LWSSO Token's expiration value determines the application session validity. Therefore, its expiration value should be at least the same value as that of the application session expiration value configured in this file: `C:\...\src-1.30\WEB-INF\web.xml`.

Example: The `web.xml` file has the `<session-timeout>` parameter set to 30 minutes by default.

```
<session-config>
   <session-timeout>30</session-timeout>
</session-config>
```

To change the session timeout value, locate the <creation> parameter in the lwssofmconf.xml file.

```
<creation>
     <lwssoCreationRef id="ID000002">
        <lwssoValidationRef refid="ID000001"/>
        <expirationPeriod>30</expirationPeriod>
     </lwssoCreationRef>
</creation>
```

10 Locate the multiDomain parameter and create a DNSDomain entry for your domain. In the LWSSO scenario, adding more domains is of no value. Users will not be able to link from one LWSSO-enabled browser session to another LWSSO-enabled browser session in a different domain without proper authentication.

```
<multiDomain>
    <trustedHosts>
        <DNSDomain>enterprise.domain_1</DNSDomain>
    </trustedHosts>
</multiDomain>
```

In all cases, the domain name should have at least two segments and cannot be an IP address. If there are multiple domains, add additional <DNSDomain> entries.

11 Save and close the file.

LWSSO Security Warnings

This section describes security warnings that are relevant to the LWSSO configuration:

- **Confidential InitString parameter in LWSSO.** LWSSO uses Symmetric Encryption to validate and create a LWSSO token. The initString parameter within the configuration is used for initialization of the secret key. An application creates a token, and each application using the same initString parameter validates the token.
  - It is not possible to use LWSSO without setting the initString parameter.
  - The initString parameter is confidential information and should be treated as such in terms of publishing, transporting, and persistency.
  - The initString parameter should be shared only between applications integrating with each other using LWSSO.
  - The initString parameter should have a minimum length of 12 characters.
- **Level of authentication security**. The application that uses the weakest authentication framework and issues a LWSSO token that is trusted by other integrated applications determines the level of authentication security for all the applications.

  It is recommended that only applications using strong and secure authentication frameworks issue an LWSSO token.
- **Symmetric encryption implications.** LWSSO uses symmetric cryptography for issuing and validating LWSSO tokens. Therefore, any application using LWSSO can issue a token to be trusted by all other applications sharing the same initString parameter. This potential risk is relevant when an application sharing an initString either resides on, or is accessible from, an untrusted location.

- **User mapping (Synchronization).** The LWSSO framework does not ensure user mapping between the integrated applications. Therefore, the integrated application must monitor user mapping. We recommend that you share the same user registry (as LDAP/AD) among all integrated applications.

  Failure to map users may cause security breaches and negative application behavior. For example, the same user name may be assigned to different real users in the various applications.

  In addition, in cases where a user logs onto an application (AppA) and then accesses a second application (AppB) that uses container or application authentication, the failure to map the user will force the user to manually log on to AppB and enter a user name. If the user enters a different user name than was used to log on to AppA, the following unexpected behavior can arise: If the user subsequently accesses a third application (AppC) from AppA or AppB, then they will access it using the user names that were used to log on to AppA or AppB respectively.
- **Identity Manager.** Used for authentication purposes, all unprotected resources in the Identity Manager must be configured with the `nonsecureURLs` setting in the LWSSO configuration file.

Troubleshooting

This section describes known issues for LWSSO authentication.

- Security context. The LWSSO security context supports only one attribute value per attribute name.

  Therefore, when the SAML2 token sends more than one value for the same attribute name, only one value is accepted by the LWSSO framework.

  Similarly, if the IdM token is configured to send more than one value for the same attribute name, only one value is accepted by the LWSSO framework.
- Multi-domain logout functionality when using Internet Explorer 7. Multi-domain logout functionality may fail under the following conditions:

  The browser used is Internet Explorer and the application is invoking more than three consecutive HTTP 302 redirect verbs in the logout procedure.

  In this case, Internet Explorer may mishandle the HTTP 302 redirect response and display an "Internet Explorer cannot display the web page" error page instead.

  As a workaround, it is recommended to reduce, if possible, the number of application redirect commands in the logout sequence.

Limitations

Note the following limitations when working with LWSSO authentication:

- Client access to the application.

  If a domain is defined in the LWSSO configuration:

  — The application clients must access the application with a fully qualified domain name in the login URL.

  Example: http://myserver.companydomain.com/WebApp.

  — LWSSO cannot support URLs with an IP address.

  Example: http://192.168.12.13/WebApp.

  — LWSSO cannot support URLs without a domain.

  Example: http://myserver/WebApp.

  If a domain is not defined in the LWSSO configuration: The client can access the application without a fully qualified domain name in the login URL. In this case, a LWSSO session cookie is created specifically for a single machine without any domain information. Therefore, the cookie is not delegated by the browser to another, and does not pass to other computers located in the same DNS domain. This means that LWSSO does not work in the same domain.

- LWSSO framework integration. Applications can leverage and use LWSSO capabilities only if integrated within the LWSSO framework in advance.

- Multi-Domain Support.

  — Multi-domain functionality is based on the HTTP referrer. Therefore, LWSSO supports links from one application to another and does not support typing a URL into a browser window, except when both applications are in the same domain.

  — The first cross domain link using HTTP POST is not supported.

  Multi domain functionality does not support the first HTTP POST request to a second application (only the HTTP GET request is supported). For example, if your application has an HTTP link to a second application, an HTTP GET request is supported, but an HTTP FORM request is not supported. All requests after the first can be either HTTP POST or HTTP GET.

  — LWSSO Token size.

  The size of information that LWSSO can transfer from one application in one domain to another application in another domain is limited to 15 Groups/Roles/Attributes (note that each item may be an average of 15 characters long).

  — Linking from Protected (HTTPS) to non-protected (HTTP) in a multi-domain scenario:

  Multi domain functionality does not work when linking from a protected (HTTPS) to a non-protected (HTTP) page. This is a browser limitation where the referrer header is not sent when linking from a protected to a non-protected resource. For example, see: http://support.microsoft.com/support/kb/articles/Q178/0/66.ASP

- Security Assertion Markup Language 2.0 (SAML2) token.

  — Logout functionality is not supported when the SAML2 token is used.

  Therefore, if the SAML2 token is used to access a second application, a user who logs out of the first application is not logged out of the second application.

— The SAML2 token's expiration is not reflected in the application's session management.

  Therefore, if the SAML2 token is used to access a second application, each application's session management is handled independently.

- Java Authentication and Authorization Service (JAAS) Realm. The JAAS Realm in Tomcat is not supported.
- Using spaces in Tomcat directories. Using spaces in Tomcat directories is not supported.

  It is not possible to use LWSSO when a Tomcat installation path (folders) includes spaces (Example: Program Files) and the LWSSO configuration file is located in the common`\classes` Tomcat folder.
- Load balancer configuration. A load balancer deployed with LWSSO must be configured to use sticky sessions.

3 Next Steps

After you install the Service Request Catalog 1.30 application, there are some configuration steps to follow. Basic tasks include installing and configuring the Service Request Catalog APIs, languages, and security.

Configure Service Manager

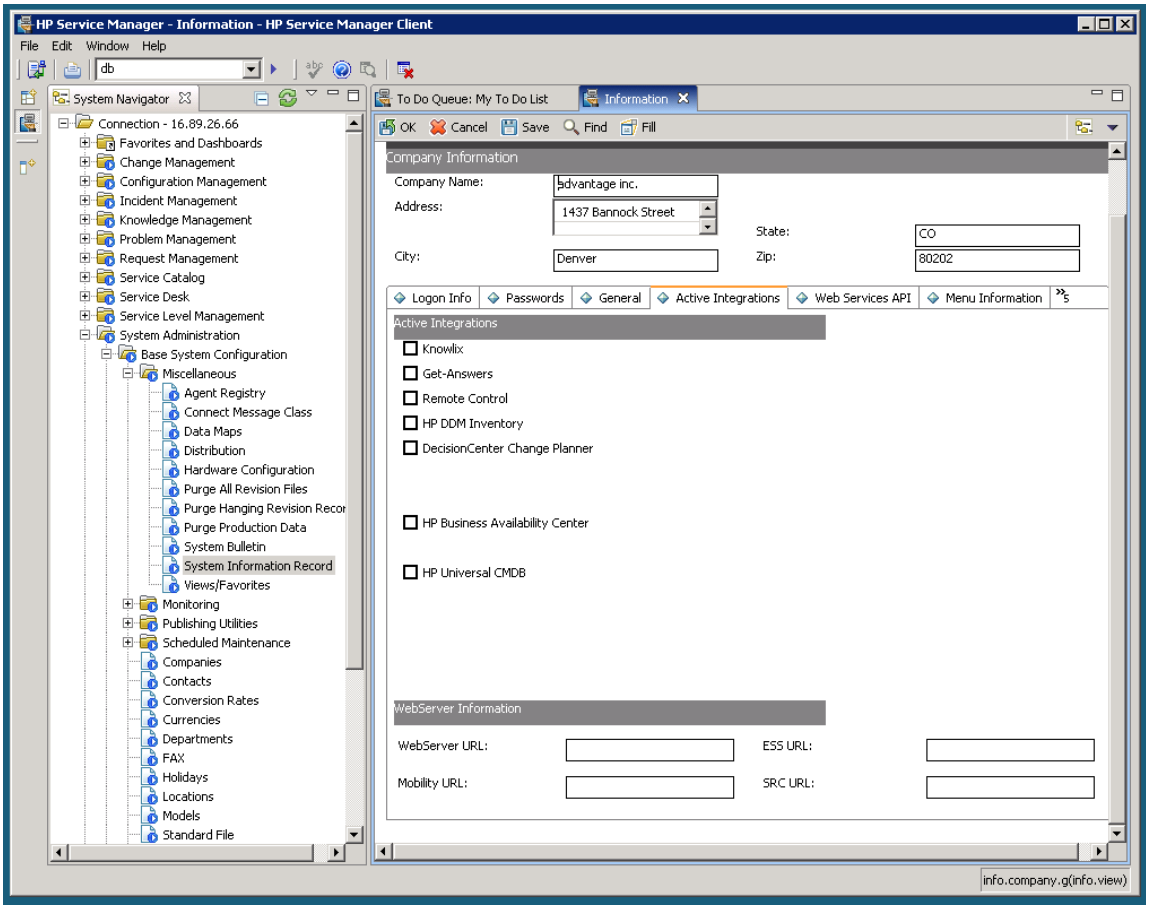
There are a few Service Manager administrative tasks to complete before you begin using Service Request Catalog.

Task 1: Configure the SRC URL

1. Start the Service Manager server.
2. Start a Service Manager Windows client session.
3. Expand the navigation tree in the left pane.
4. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**, shown in Figure 1 on page 46.
5. On the **System Information Definition** form, click the **Active Integrations** tab.
6. Locate the **Webserver Information** section.
7. In the **SRC URL** text box, shown in Figure 1, specify the URL to connect Service Manager to the Service Request Catalog application.

http://*nn.nn.nn.nn*	Internet Protocol address of the Service Request Catalog server.
:*nnnn*	Configured port number on the Service Request Catalog server.
/src-1.30	Configured Service Request Catalog application identifier (specified in step 5 on page 14).

Figure 1 System Information Definition Record



8 Save your changes.

Task 2. Configure User Capability Words

Service Manager best practices recommend assigning user roles that carry all the required capability words to be successful at their tasks. The required capability word for Service Request Catalog is `service catalog`. The `service catalog` capability word allows a user to request items from the service catalog. Other capability words may be required for role-based scenarios.

Related Capability Words

Depending on your task objectives, *at least one* of the following capability words must be part of each Service Request Catalog user profile.

Capability Word	Description
svcCatDeptRequester	An employee can request items from the catalog on behalf of a department.
svcCatEmployeeRequester	An employee can request items from the catalog.
svcCatManagerRequester	A manager can request items from catalog.
svcCatRequestOnBehalf	A user can request items on for another employee.

Request For Another Person

You can order catalog items and services for another person only if the `svcCatRequestOnBehalf` capability word is part of your user profile.

Mass Update

As an administrator, you can apply new capability words to a large group of users in a single operation. Follow the steps in Service Manager documentation that describes the mass update feature to apply a capability word to the user profile of a group of users.

Task 3: Synchronize the Timeout Value

Service Manager administrators should make sure Service Request Catalog and Service Manager timeout settings match.

1. On the Service Manager server, if Service Manager uses a default installation path, open this file with a text editor:

   Windows: `C:\Program Files\HP\Service Manager n.nn\Server\RUN\sm.ini`

   Unix: `/opt/..../HP/Service Manager n.nn/Server/RUN/sm.ini`

2. On the Service Request Catalog server, open this file with a text editor:

   Windows: `C:\...\src\WEB-INF\web.xml`

   Unix: `/opt/..../src/WEB-INF/web.xml`

3. Verify the `webservices_sessiontimeout` value in the `sm.ini` file matches the `session-timeout` value configured in the `web.xml` file.

   — The `sm.ini` file should have a setting for `webservices_sessiontimeout`. Example:

   ```
   webservices_sessiontimeout:1800
   ```

   This value is expressed in seconds. (1800 seconds = 30 minutes)

   — The `web.xml` file has the `<session-timeout>` parameter set to 30 minutes by default.

   ```
   <session-config>
      <session-timeout>30</session-timeout>
   </session-config>
   ```

4. If you change the value in either file, save the file.
5. Close both files.

Refine the Service Manager Catalog

Complete Task 1: Remove Empty Categories to maximize performance and Task 2: Remove Unsupported Characters to improve the user experience.

Task 1: 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.

Task 2: Remove Unsupported Characters

Service Manager has a rich text editor that enables the Service Catalog administrator to embellish item descriptions with formatting, embedded links, and other readability enhancements. All of these formatting attributes do not render successfully when viewed through the Service Request Catalog interface. HP recommends that the Service Catalog administrator adjust these descriptions to limit formatting of the detailed description field to the following:

- Links (<a>) to relative internal paths or absolute external URLs
- Bold (or <b>) text
- Line breaks (
)
- These font (<font>) attributes:
  - color (specified in hexadecimal values only)
  - face (including a comma-delimited list of fonts)
  - size (specified in pixels or relative points, such as +2 or -4)
  - letterspacing
  - kerning (0 or 1)
- These image (<img>) attributes:
  - src (required)
  - width (in pixels)
  - height (in pixels)
- Italic (<i>) text
- Leading spaces in category and item names
- These paragraph (<p>) attributes:
  - align (left, right, justify, center)
  - class
- The class span () attribute

  Multiple properties in a single span tag, such as Hello, are supported. Nested span tags are not supported.

- These text formatting (<textformat>) attributes:
  - indent
  - blockindent
  - leftmargin (in points)
  - rightmargin margin (in points)
  - leading (in pixels)
  - tabstops (in a comma-delimited list)
- Underlined (<u>) text
- Simple lists. Simple lists render as unordered (bulleted) lists. Numbered lists are not supported. Example:

```
<ul>
      <li>Item</li>
      <li>Item</li>
      <li>Item</li>
      <li>Item</li>
</ul>
```

  Nested lists are not supported.

Caution: Unsupported formatting is ignored in the Service Request Catalog application.

Task 3: 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 maximum size 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.

Task 4: Multi-Company Mode

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.

The recommended tasks in Service Manager are:

1. Enable Multi-Company mode in the System Information Company record.

   (System Administration > Base System Configuration > Miscellaneous > System Information Record > General)

2. Verify that you have a complete company record with Multi-Company enabled.

   (**Tailoring > Database Manager > Table** (company) > **Search** > company > **Search** > Show Company in Multi-Company Lists)

3. Enable Mandanten security by specifying a filtering condition for the svcCatalog and svcDisplay tables. Use the Mandanten Field Restriction form to specify **`MS.company`** in the **Mandant Field Name** field. Create two records: one for **File Name** svcCatalog and one for **File Name** svcDisplay.

   (**System Administration > Ongoing Maintenance> Mandanten > Mandanten Field Restrictions**)

4. Create a Mandanten security group. The **Security ID** must be a an upper case value. Example: SRC. Type this value in the first row of the **Include Value List.** Click **Add** to save the security group record.

   (**System Administration > Ongoing Maintenance> Mandanten > Mandanten Security Groups**)

5. 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.

6. Assign the company to each category, sub-category, package, and item. This property does not cascade. If you are an advanced user, you can use these RAD Debugger commands to accomplish this task:

   — `d MS.company in $L.file`

   — `x MS.company in $L.file="company name"`

7. 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.

Task 5: Contact Lookup Functionality

All Service Request Catalog users must have complete contact information specified in their Service Manager operator and contact records for the contact lookup 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 lookup.

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 lookup 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)

Support Other Locales

When you deploy the complete language packs, Service Request Catalog provides complete out-of-box support for these languages:

- Brazilian Portuguese
- Dutch
- French
- German
- Italian
- Japanese
- Simplified Chinese
- Spanish
- Czech
- Hungarian
- Korean
- Polish
- Russian

Service Request Catalog also supports the currency code and symbol for each of these languages. However, you may have a global community of users who each want to view the contents in their preferred language. If this is the case, complete the steps in the following sections to enable as many locales as you wish. 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.

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 on page 52 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 Follow step 1 on page 51 through step 9 on page 52 to verify the target language is configured in Service Manager.

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

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.

1. In the System Navigator pane, click **Service Catalog** > **Manage Catalog** to locate the record to be localized.
2. Click **Search**. Service Manager displays a list of the item records in the catalog.
3. Scroll to find the target record in the list.
4. Click the record in the list to display all the record information in the Category Definition form.
5. When Service Manager displays the record, right-click anywhere in the form and select **Localize this record** from the context menu.

   Service Manager displays the *Create Localized Data* wizard.
6. Choose the language from the drop-down list. Example: select German.
7. Replace the English **Name**, **Short Description**, and **Long Description** with equivalent values in the language you selected in step 6.
8. Click **Finish**. Service Manager displays a confirmation message above the Category Definition form.
9. Repeat step 3 through step 8 to create localized versions of each catalog record.
10. When you are finished, click **OK**.
11. 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-1.30\resources\client\`

Unix: `/opt/..../src-1.30/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-1.30\resources\client\hu\`

Unix: `/opt/..../src-1.30/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 this folder:

Windows: `C:\...\src-1.30\resources\client\`

Unix: `/opt/..../src-1.30/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-1.30\resources\client\ko\CurrencySymbol_ko.properties`

Unix: `/opt/..../src-1.30/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.

Customize the Interface

Service Request Catalog enables you to change the branding information in the heading area of the user interface. The heading area is 300 pixels wide and 35 pixels high. Within this area, you can customize the interface with your logo and any other branding elements.

To prepare the branding image file:

1. Create a branding image that is 300 pixels wide and 35 pixels high.
2. Save the image with a `.png` file type.
3. Rename this image `logo.png`.

If you have an existing image with a different name and in a different format, such as `myCompany.jpg`, you can convert the image to the `.png` format first (`myCompany.png`) and then rename the file `logo.png`. Service Request Catalog looks for that exact file name when it renders the user interface. The final image must conform to the recommended size or the results may be unpredictable.

To post the new image file:

1. Store the new images in this folder:

   Windows: `C:\apache-tomcat-7.0.x\webapps\src-1.30\images\branding\logo`

   Unix: `/opt/..../apache-tomcat-7.0.x/webapps/src-1.30/images/branding/logo`
2. Restart Tomcat.
3. Log into the Service Request Catalog application to view the new heading.

Start Service Request Catalog

When you complete all basic configuration tasks, you are ready to start Service Request Catalog. Make sure the Service Manager server is running before you start Service Request Catalog.

1. **Windows:** Go to the **Control Panel** > **Administrative Tools** > **Services** dialog to start the Apache Tomcat service.

   Unix: Run `/opt/...../apache/apache-tomcat-7.0.x/bin/startup.sh`
2. Type this URL into a supported browser window:

   `http://hostname.domainName:nnnn/src/`

   where *`hostname.domainName`* is the fully qualified domain name of the tomcat server, and *`:nnnn`* is the tomcat port number on that server. The exact format of the URL depends on how you configured the `path` parameter in step 6 on page 14.
3. Log in with your Service Manager credentials.

4 Service Manager Administration Tasks

There are administrative tasks that must be completed in the Service Manager Windows Client to enable corresponding functionality in Service Request Catalog.

Read these sections to learn how to enable these features.

Create Support Catalog Categories and Items

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.

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.

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**.

11 Service Manager displays the Category Definition interface where you can refine the category by choosing an Owner or even attaching an image.

12 To create a Support category, select the **Non-cart category** checkbox.

13 Select the **Support category** checkbox.

14 Click the **Access** tab.

15 Click the first empty row in the **Available to** table.

16 Select the Service Catalog group capability word that you want to associate with this new category.

   Example: svcCatEmployeeRequester.

17 Add as many capability word groups to the list as necessary.

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

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**.

Edit a Support 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.

5 Browse the list to find the category or item that you want to edit.

Example: Repair software

6 Change any information about the category, or the item.

Example: You may not want to delete the category or item, but clear the **Active** checkbox to make the category or item unavailable temporarily.

7 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.

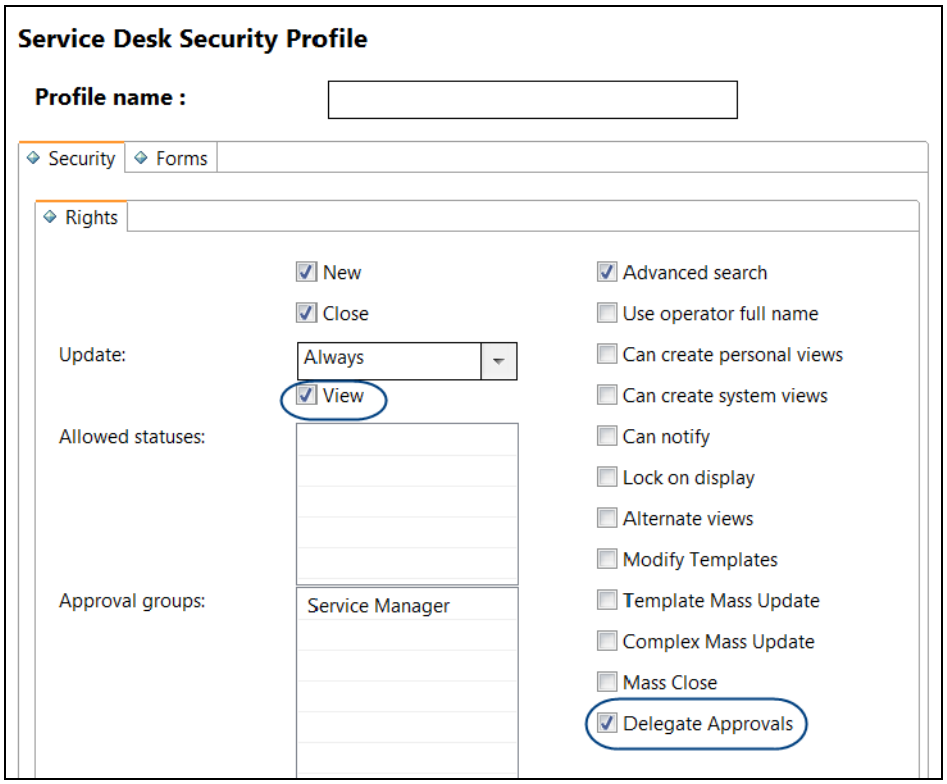
Example: Repair software

6 Click **Delete**.

Delegate Approval Authority

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.



11. If necessary, select **View**.
12. Click **Save**.

Tailor Service Request Catalog Custom Fields

Service Manager enables administrators to customize Enterprise Self Service (ESS) forms. These forms are used to submit a catalog or support request directly using a Service Manager interface. If you are using Service Request Catalog as the interface, you want to see some customization supported to meet your specific business requirements. This chapter describes how you can use a simple Service Manager wizard to define the necessary fields and labels that will appear on the Service Request Catalog panels when you submit a request.

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.

Example: 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
- Dates

Service Manager Functionality

Through a Service Manager tailoring wizard, you can flag any custom field as mandatory. 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. 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.

Service Request Catalog Functionality

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.

When you check the status of a request that contains custom fields, the fields appear as read-only information. If a custom field was optional, and the user did not provide a response or a value, the custom field does not appear in the status view.

A request with custom fields is eligible to be resubmitted when it is in a pending state, but the values in the custom fields cannot be edited and resubmitted.

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 you use to create the custom fields. For more information, see step 3 on page 15.

General Directions

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 9.30 server. The client can be an earlier version, but the server must be a 9.30 server.
2. Expand the left Navigation pane.
3. Click **Tailoring** > **SRC Tailoring**. Service Manager displays an SRC Tailoring wizard. The first page of the wizard lists existing configurations. A *configuration* is the description of one or more new sections with custom fields that will appear in a Service Request Catalog checkout panel. There can be a maximum of three configurations that add new sections with custom fields to the Support catalog, Services catalog, or General Support checkout panel. If your Service Manager server runs in multi-company mode, you can create a maximum of three configurations for each company.

How Do I Add a Configuration?

Follow these steps to create a new Service Manager configuration. Use the Example suggestions to create a sample configuration of your own.

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

   Example: Type **`Support Custom Fields`**.
6. Choose the **Checkout Panel** where the new fields will appear. 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 field, or you can create a new one. Click **Select a Label** or **Add a Label**. If you choose Select a Label, you can select any label that has already been added as a custom field label. If you choose Add a Label, you are creating a new label.

   Example: Click **Add a label** and type **`FAX`** for the name of the new label.
10. Click **Next**.
11. Click the **Field Name** drop-down to choose the field that you want populate with new information.

    Example: Choose `Fax` from the drop-down list.

12 In the **Modifiability** field, you can set the field permission to one of the following:

- **Editable during submission**, which means the user can provide input when submitting a request. This permission is only valid during initial submission. No new input is available if the request is resubmitted.
- **Always read-only**, which means no user changes are allowed or required when submitting a request.

Example: Select **Editable during submission** from the drop-down list.

13 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.

- If you want the user to provide some information, choose **Text**.
- If you want the user to provide a lot of information (like a description), choose **MultiText**.
- If you want the user to make a simple choice, or specify a true/false condition, choose **Checkbox**.
- If you want the user to choose from a pre-defined list, choose **Pick List**.

Example: Choose the **Text** option.

14 Click **Next**.

15 Optional. If you want the user to select a value from a field an existing Service Manager table, then you must identify the table in the **Lookup Table** field.

Example: You can leave this blank if you want to create a plain text field. In this case, we want a text box, so you can skip to step 16.

Note: If you use a lookup table here, then you must choose the lookup field from that table in step .

Optional. If you selected a lookup table in the last step, choose the **Lookup Field** in that table.

Note: If you specify a lookup table field here, make sure that you chose the lookup associated lookup table in step 15.

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

Example: Click the **Mandatory** checkbox.

17 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).

18 Click **Next**.

19 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.**

20 Repeat step 9 through step 18 to add another custom field.

Example: Add another custom field for a Phone Number field, add a checkbox, or add a text box that collects user comments.

> **Note:** You can add a specific field only once in a configuration. For example, if you added a lookup for a certain table and field, you cannot add it to the same configuration again, even if the field type is different.

21 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 return to the wizard to make changes.
- Click **Remove Custom Field** to delete that field.

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

How Do I Add a Section to a Configuration?

Sections are new groups of fields that request more information from the user when they submit a request. For example, you can add a new section to obtain an alternate delivery address from the user. Follow these steps to add a new section to a configuration.

1 From the SRC Tailoring Wizard home page, click **Configure Sections**.

2 Click **Add Section**.

3 Repeat step 9 on page 63 through step 18 on page 64 to complete the new section.

4 When the new section is complete, click **Finish**.

Other Section Tasks

There are other editorial changes that you can make 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**.

How Do I Copy or Edit a Configuration?

Service Manager supports up to three existing configurations for each company. You cannot create more than three for any company, but you can create a new configuration from an existing configuration for another company, or you can create a second or third configuration for a single company using this shortcut.

1 From the Service Manager Navigation pane, click **Tailoring** > **Database Manager**, or type **db** in the **Execute Command** text box.

2 Type **`svcSrcConfig`** in the **Table** field.

3 Click the **Search** icon.

4 From the displayed list, select `svcSrcConfig` and click **Search** again. Select a configuration from the list.

5 Change the **Name**, **Company**, or **Checkout Panel**.

6 Click **Add** if you created a new configuration. Click **Save** if you edited an existing configuration.

You can view and edit any new configuration using the SRC Tailoring Wizard.

How Do I Delete a Configuration?

Follow these steps to delete a custom configuration.

1 From the SRC Tailoring Wizard home page, select an existing configuration.

2 Click **Remove Configuration**.

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**
Create a configuration with custom fields	Labels and error messages will not have localized equivalents unless you configure them. To localize labels for sections or fields in Service Request Catalog checkout panels, search the `scmessage` table for message entries that have a class of `srcconfig`, and then add a localized version for each of such entries by using the language code of the target language.
Add new fields to an existing configuration.	Existing requests will not display those fields when you check their status.

If I Do This	This Is the Result
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.
Rename a field, or change the field type, in a Service Manager table that is used by a configuration.	The configuration does not update automatically with the renamed field and the original field is no longer available. You must edit the configuration to reference the field by its new name. If you change a field type, you must edit the configuration to delete the field reference and then add it again. This action causes the new data type to be referenced correctly.
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 does resubmit the request, the new mandatory field is added to it. The problem is that the user is unable to edit any custom fields during the resubmission, including the new one that requires data. Form validation fails. **Best Practice:** New mandatory fields should have a default value set to ensure they validate in a similar scenario.

Localization

Service Manager stores custom field labels in the `scmessage` table. To localize these labels, search the `scmessage` table for message entries that have a class of `srcconfig`, and then add a localized version for each entry by using the language code of the target language. For more information, see the Service Manager help topics about accessing and localizing message records.

5 Performance

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

These changes remedy the problem that occurs when the Service Manager server is unable to fully translate Service Manager queries to the SQL server when fields appearing in the query are mapped to large object (LOB) type fields. The recommended changes prevent inefficient scans by the Service Manager server when it runs queries against tables that reference LOB type fields.

If you already mapped any repeating group in one of the referenced tables to 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, the detailed instructions in Task 1 assume you do not 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.

Task 1. 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 sql mapping utility.

To edit the dbdict for the Approval file

1. In the Fields pane, scroll down to and 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 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.

12 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 **APPROVALA1** table was created.

13 Note whether any SQL error messages appear.

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

Task 2. 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 sql mapping utility.

To edit the capability file to an Array table

1 Select the **parent.tree** type array.

2 Click the **edit** field.

3 Set SQLTable to **a1**.

4 Click **Next**. You should be on the **parent.tree** type character.

5 Change the SQL type to **VARCHAR(50)**.

6 Change SQLTable to **a1** and click **OK**.

7 Go to SQLTables.

8 Add **a1 CAPABILITYA1 oracle10**.

9 Click **OK**. A pop-up dialog displays the DDL to create the new table.

Important: Do **not modify** the DDL.

10 Copy this DDL to the clipboard for future reference.

11 Click **User Alters.** Do ***not*** click SM Alters.

12 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.

13 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 **CAPABILITYA1** table was created.

14 Note whether or not any SQL error messages are displayed.

15 Use Oracle sql developer or another database management tool to verify that the **CAPABILITYA1** table was created.

Task 3. Remap the acces.list Field

The next step is to remap the `acces.list` field in the svcCatalog file to an Array table. Use the dbdict utility, not the system definition utility or the sql mapping utility.

To edit the svcCatalog file to an Array table

1 Select the **access.list** type array.

2 Click the **edit** field.

3 Set SQLTable to **a2**.

4 Click **Next**. You should be on the **access.list** type character.

5 Change the SQL type to **VARCHAR(50)**.

6 Change SQLTable to **a2** and click **OK**.

7 Go to SQLTables.

8 Add **a2 SVCCATALOGA2 oracle10**.

9 Click **OK**. A pop-up dialog displays the DDL to create the new table.

Important: Do **not modify** the DDL.

10 Copy this DDL to the clipboard for future reference.

11 Click **User Alters.** Do ***not*** click SM Alters.

12 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.

Task 4. 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 sql mapping utility.

To edit the kmgroup file to an Array table

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

   Important: Do **not modify** the DDL.
10. Copy this DDL to the clipboard for future reference.
11. Click **User Alters.** Do ***not*** click SM Alters.
12. 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.
13. 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.
14. Note whether or not any SQL error messages are displayed.
15. Use Oracle sql developer or another database management tool to verify that the **KMGROUPA1** table was created.

Task 5. Remap the members Field

The next step is to remap the members field in the cm3groups file to an Array table.

To edit the cm3groups file to an Array table

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

   Important: Do **not modify** the DDL.
10. Copy this DDL to the clipboard for future reference.
11. Click **User Alters.** Do ***not*** click SM Alters.
12. 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.
13. 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.
14. Note whether or not any SQL error messages are displayed.
15. Use Oracle sql developer or another database management tool to verify that the **CM3GROUPSA1** table was created.