HP Service Manager Exchange with SAP Solution Manager

Software Version: 1.10

Service Manager Version: 9.x

Installation and Administration Guide

Document Release Date: August 2013
Software Release Date: November 2012

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

This product includes software developed by the Apache Software Foundation (http://www.apache.org/). Portions Copyright © 1999-2003 The Apache Software Foundation. All rights reserved. SMSSMEX uses software from the Apache Jakarta Project including Apache Axis2, Apache Tomcat 5.5, Log4j Apache License, and Spring Framework.

Trademark Notices

Java and Oracle® are registered trademarks of Oracle and/or its affiliates.

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

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Intel®, Itanium® and Pentium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

SAP® is a registered trademark of SAP AG in Germany and in several other countries.

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 Online web site at:

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

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

HP Software 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 Introduction

This HP integration product implements HP Service Manager Exchange with SAP Solution Manager. This version only implements Service Manager Incident Exchange with SAP Solution Manager. Therefore, this document focuses on the HP Incident Exchange.

HP Incident Exchange

Businesses today increasingly rely on their mission-critical SAP applications. Disruptions in the SAP environment have a severe business impact. Keeping the system continuously available has never been more vital for success. In any SAP landscape, business process disruptions caused by an application or infrastructure incident must be proactively prevented. If disruptions do occur, they need to be quickly and efficiently resolved. HP and SAP have teamed up to solve this issue.

Incident management in enterprises today consists of disconnected incident management systems that often implement divergent processes. This situation diminishes collaboration within IT operations, lowers quality of service and productivity.

The integration of SAP Solution Manager Service Desk with HP Service Manager provides a cohesive Incident and Service Request Management solution for the entire enterprise, resulting in higher enterprise availability, improved service quality and reduced IT costs.

HP Incident Exchange builds a dynamic link between HP Service Manager Software and SAP Solution Manager Service Desk and improves the Incident and Service Request Management Process throughout the entire enterprise. HP Incident Exchange offers dynamic integration between HP Service Manager and SAP Solution Manager Service Desk for improved incident workflow.

The interface to exchange support messages between HP Service Manager and SAP Solution Manager Service Desk was designed and developed jointly by HP and SAP and is certified by SAP.

Existing Fragmented Incident Management Workflow

Performance monitoring of an SAP environment must include SAP and non-SAP applications.

SAP Solution Manager Service Desk

To monitor and manage SAP environments, IT operations management uses the SAP Solution Manager Service Desk to collect information about SAP systems and serves as an internal help desk for SAP installations. Users and administrators can create support messages from any SAP system. The messages are processed centrally in the Solution Manager Service Desk.

If the support message involves an SAP application, a solution may be available in the SAP Service Marketplace or from SAP Active Global Support or from the in-house SAP support team. But if the issue is not caused by the SAP application, the message will be forwarded to the administrators responsible for the non-SAP systems. The support call needs to be entered in a second or third service desk and tracked until resolved. In the meantime, the SAP Service Desk team waits for feedback before closing the call and informing the originator, who is temporarily left “in the dark”.

HP Service Manager

An incident can also be reported to the service desks monitoring non-SAP applications and infrastructure hardware and software. Many SAP customers have integrated these tasks in the HP Service Manager, which is able to support nearly all IT application and infrastructure components.

If a support call, for example, pertains to a “printing issue from an SAP application” and the HP Service Manager team detects no issue with the printer hardware or software, the call will be forwarded to the SAP service desk team to check whether it is related to the SAP application. Again the service call must be re-entered in a service desk, in this case in the SAP Solution Manager Service Desk. Additional information or attachments regarding the error or error resolution must be forwarded manually. The HP Service Manager team has to wait for feedback before informing the requesting user and closing the call.

In both cases the disconnected service desks and the fragmented incident management workflow impede the service desk team’s ability to resolve problems. Disadvantages of this non-integrated workflow are

- Only limited and often inconsistent information about the incident is available.
- It is difficult to monitor, track and report incidents or to work together toward resolution.
- Manual workarounds are required for the handover of incidents between the SAP and non-SAP service desks and for information updates.
- There is insufficient synchronization. The same incident may get reported, recorded and tracked in separate service desks, or the incidents may get lost or ‘dropped’.
- Expertise about the interrelationships of SAP applications with non-SAP applications and other IT components is lost.

This results in productivity loss and reduced quality of service.

Purpose of Document

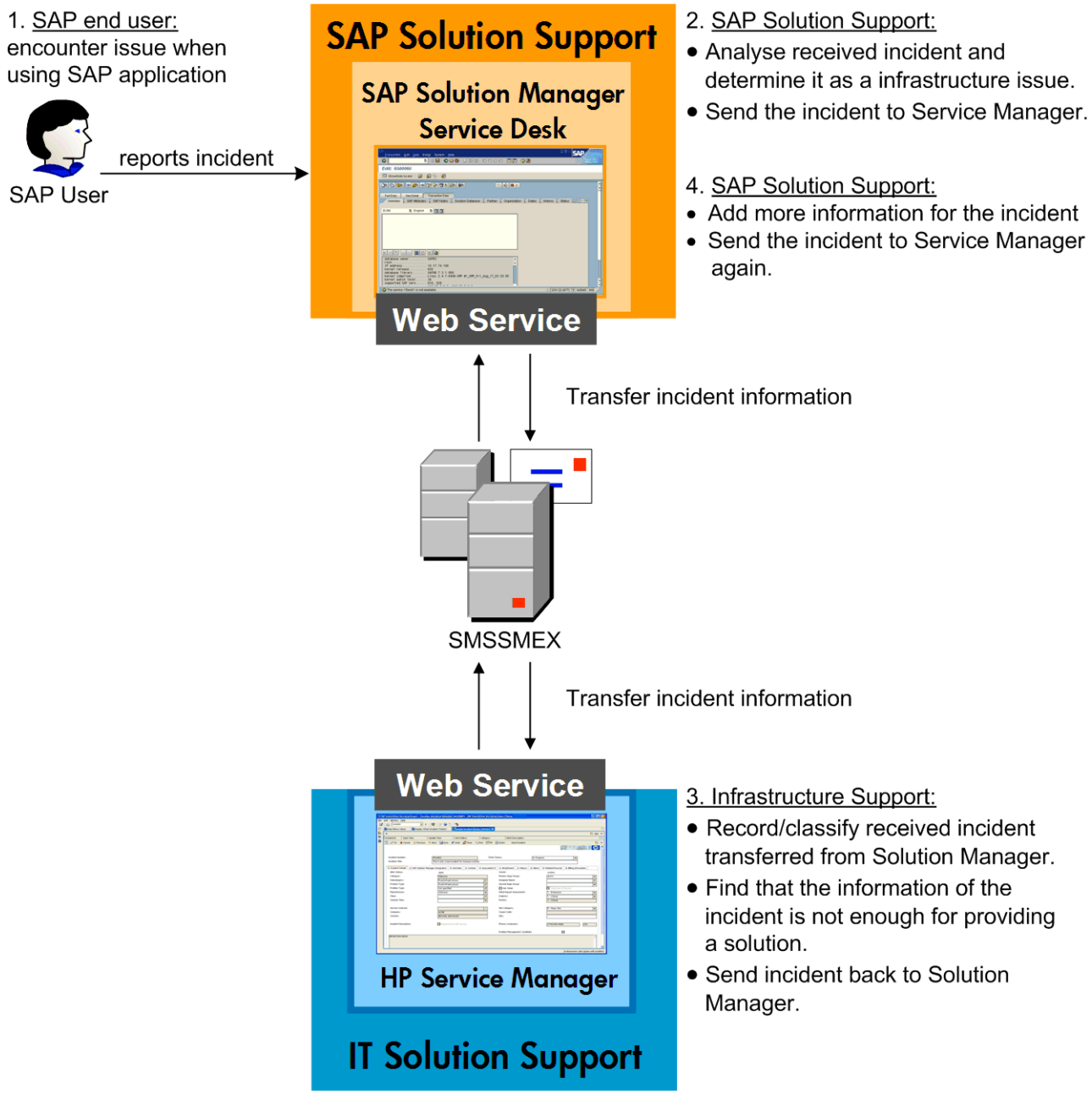
This document describes installation, configuration, administration and maintenance of HP Incident Exchange and the HP Incident Exchange web service. This guide is intended for use by HP consultants and application administrators that install and maintain HP Incident Exchange. This document is not an end user document.

Use Cases

This section discusses two use cases for HP Service Manager Exchange with SAP Solution Manager that demonstrate the integration scenarios.

Use Case 1: Incident Originates from Solution Manager

In this use case, a user reports an issue to SAP Solution Manager. The Solution Manager generates a new incident and sends the incident to Service Manager to request a solution for the issue.



9. SAP end user:
The issue is resolved

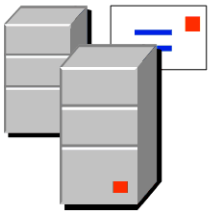
SAP User

info about
incident status and
resolution

6. SAP Solution Support:
- The solution provided by Service Manager does not solve the issue of user.
- Reject the Solution to Service Manager

8. SAP Solution Support:
- The solution solve user's issue
- Close the incident

Transfer incident information



SMSSMEX

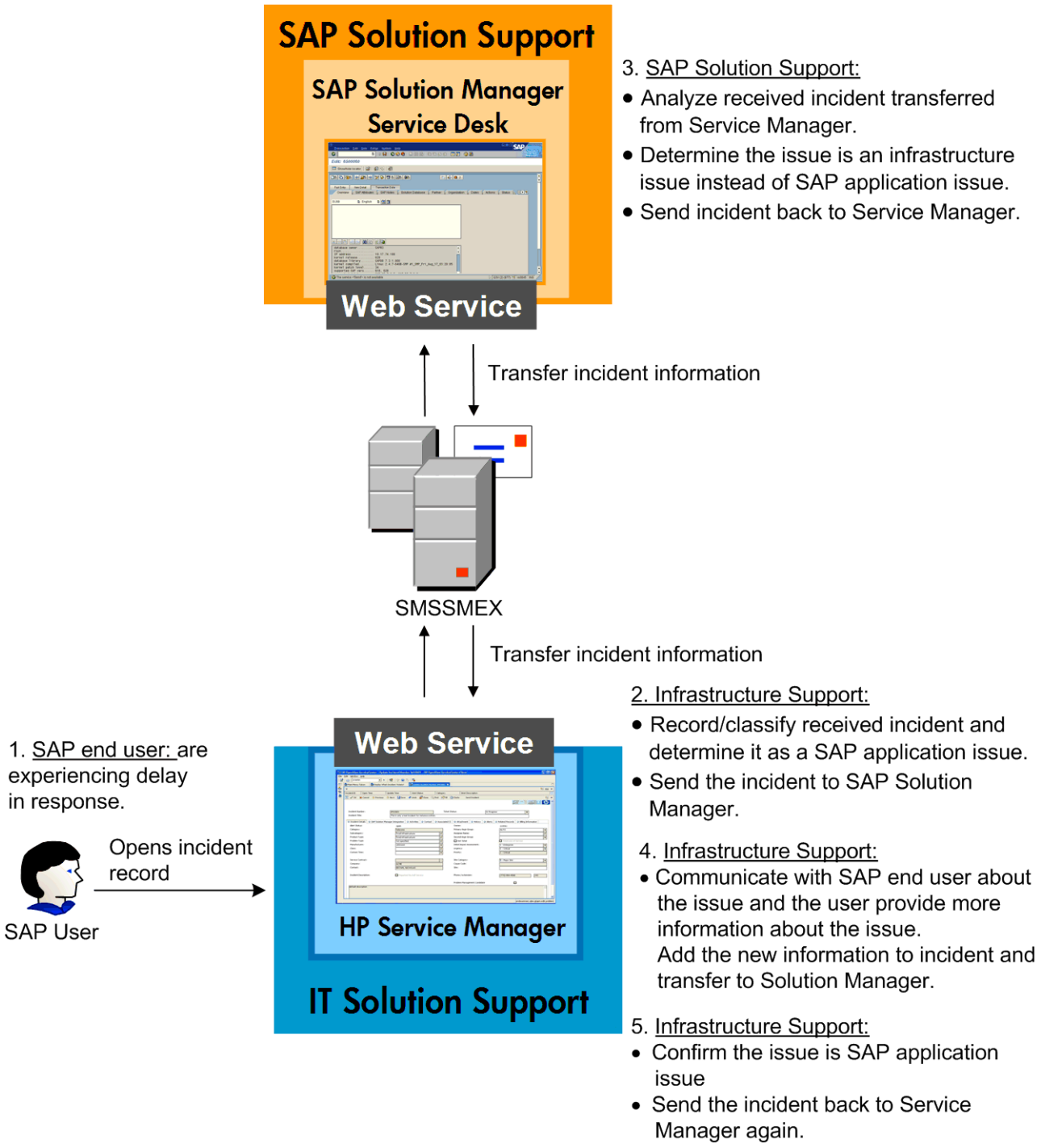
Transfer incident information

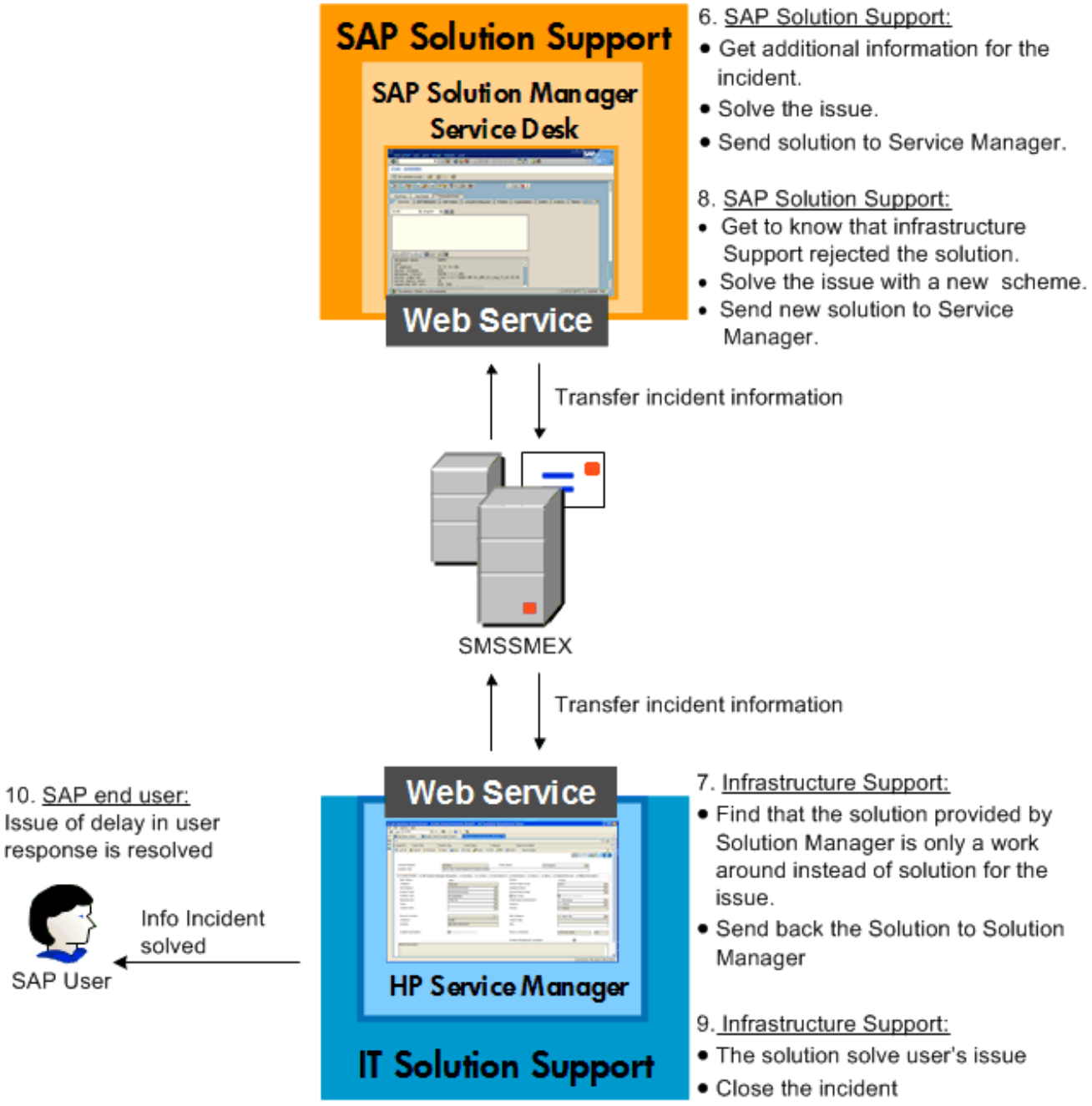
5. Infrastructure Support:
- Get additional information for the incident.
- Solve the issue.
- Send solution to SAP Solution Manager.

7. Infrastructure Support:
- Get to know that Solution Manager Support rejected the solution.
- Solve the issue with a new scheme.
- Send new solution to SAP Solution

Use Case 2: Incident Originates from Service Manager

In this use case, the user issue is captured and sent to Service Manager. An incident is generated in Service Manager and is sent to Solution Manager to request a solution for the issue.

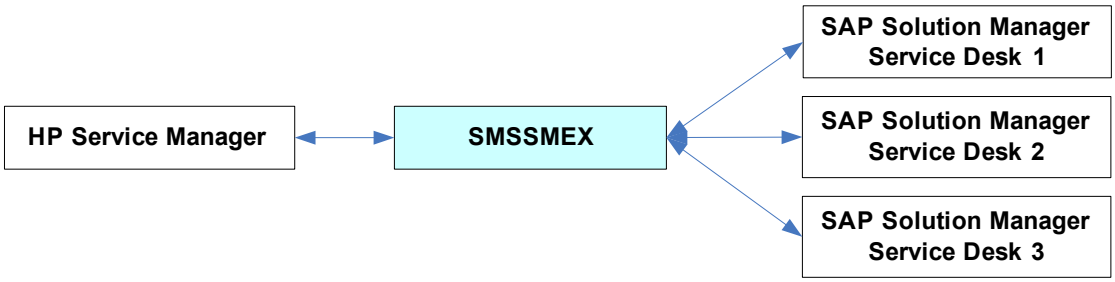



SAP Solution Support
SAP Solution Manager
Service Desk
Web Service
6. SAP Solution Support:
• Get additional information for the incident.
• Solve the issue.
• Send solution to Service Manager.
8. SAP Solution Support:
• Get to know that infrastructure Support rejected the solution.
• Solve the issue with a new scheme.
• Send new solution to Service Manager.
Transfer incident information
SMSSMEX
Transfer incident information
Web Service
HP Service Manager
IT Solution Support
7. Infrastructure Support:
• Find that the solution provided by Solution Manager is only a work around instead of solution for the issue.
• Send back the Solution to Solution Manager
9. Infrastructure Support:
• The solution solve user's issue
• Close the incident
10. SAP end user:
Issue of delay in user response is resolved
Info Incident solved
SAP User

2 Deployment Scenarios

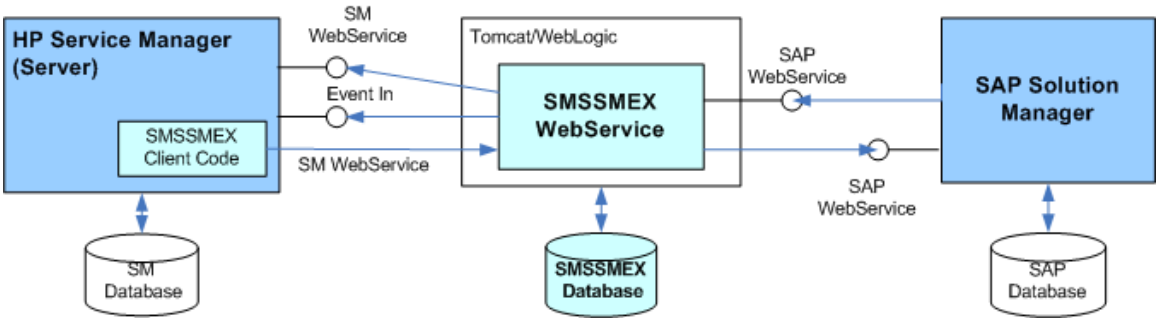
High Level Overview

SMSSMEX integrates a single Service Manager server with multiple external helpdesk systems.



Components

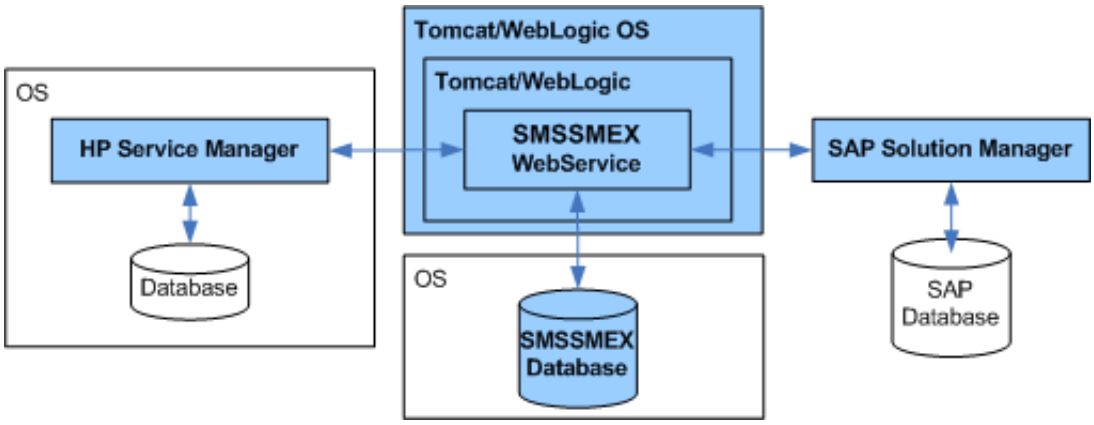
The following diagram shows the component details.



- HP Service Manager Server is the HP service desk system.
- Service Manager DB provides persistent storage for HP Service Manager.
- SMSSMEX Client Code consists of RAD and Java scripts, table definitions and GUI formats. The SMSSMEX webservices are called from this client code.
- WebServer is a Tomcat Web Application Server or WebLogic Application Server that hosts the SMSSMEX WebService (deployed as a `.war` file).
- SMSSMEX WebService exposes the incident webservice of HP Service Manager in the SAP format and transfers client requests to SAP Solution Manager webservices.
- SMSSMEX Database provides persistent storage for the SMSSMEX WebService.
- SAP Solution Manager is the Service Desk.

Support Matrix

The following diagram shows the supported components.



The following table shows the supported component versions.

Table 1 Supported component versions

Platform	Component	Versions
Service Manager / ServiceCenter	Service Manager	7.11, 9.20, 9.21, 9.30, 9.31 and 9.32
	ServiceCenter	6.2.2 or higher version
SMSSMEX OS	Windows Server	2003, 2003 R2 (32-bit) 2008, 2008 (32-bit)
	Linux	SUSE10
SMSSMEX Database	Microsoft ® SQL Server	2005, 2008
	Oracle Standard and Enterprise Edition	9.2, 10.2, 11
SAP Solution Manager	Solution Manager 7.0	>= SP12
	Solution Manager 7.1	
WebLogic Server		10.3.2

3 Installing and Configuring SMSSMEX on Tomcat

Installing SMSSMEX

The HP Service Manager Exchange with SAP Solution Manager product CD includes an autorun program for installation.

Prerequisites

- It is NOT recommended to install SMSSMEX and Service Manager/ServiceCenter on the same server.
- For installation on Unix, install HP OpenView AutoPass manually before the installation of HP Service Manager Exchange with SAP Solution Manager.

  The AutoPass installer is available from:

  `<SMSSMEX1.10 Release Package>\AutoPassInstaller\Linux\`

  — `HPOvLic-05.40.010-Linux2.6-release.rpm` for Linux

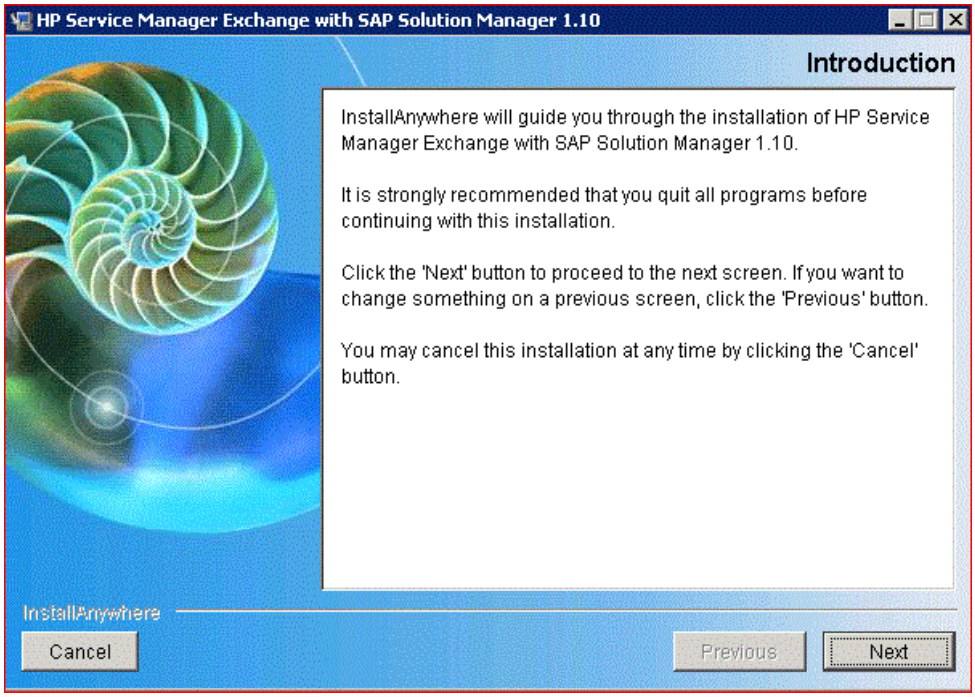
Install SMSSMEX

1. Log in to the operation system as a super user.
2. The installer is in:

— `<SMSSMEX1.10 Release Package>\InstData\Windows\VM\install.exe` (Windows 2003 Release 2 and Windows 2008)

— `<SMSSMEX1.10 Release Package>\InstData\Linux\VM\install.bin` (Linux)

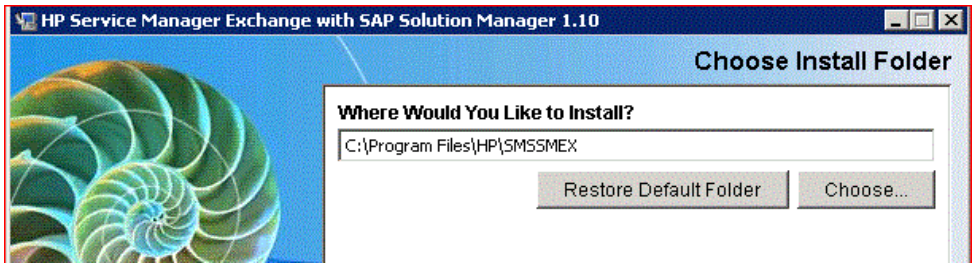
3 Run `install.bin` or `install.exe`. The Introduction dialog appears.



4 Click **Next**. The license agreement appears.

5 Select **I Accept the terms of License Agreement**.

6 Click **Next**. The Choose Install Folder page displays. For example, the default installation folder on Windows 2003 is `C:\Program Files\HP\SMSSMEX`.



7 Click **Next**. Review the summary information.

8 Click **Install**. The files are installed. For Windows installation, HP AutoPass Licensing installs the HP OpenView component automatically.

The Install Complete dialog appears.

9 Click **Done** to close the installer.

Uninstall SMSSMEX

To uninstall SMSSMEX on Windows, execute

<*SMSSMEX_installDir*>\Uninstall SMSSMEX\Uninstall SMSSMEX.exe

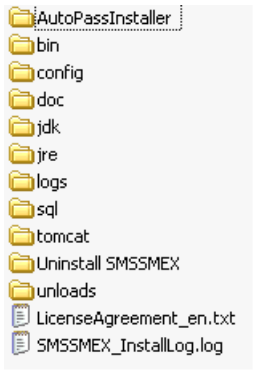
Or simply go to **Start** → **Programs** → **SMSSMEX** → **Uninstall SMSSMEX**.

To uninstall SMSSMEX on Unix, execute

<*SMSSMEX_installDir*>/Uninstall SMSSMEX/Uninstall_SMSSMEX

SMSSMEX Installed Files

After installation, the SMSSMEX folder has the following contents.



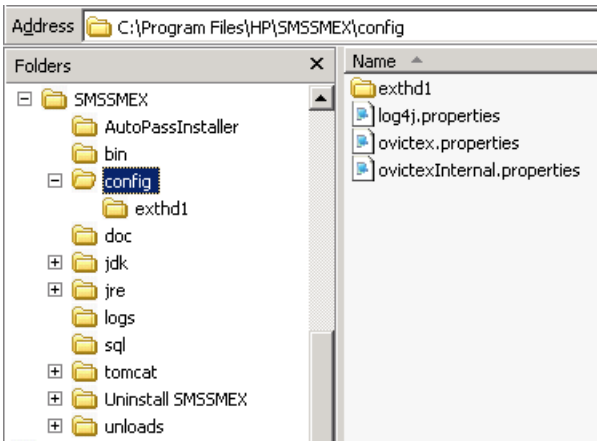


Table 2 Contents of \SMSSMEX

Directory	Content
bin	Executable commands and product description file
config	Web service configuration files
config\exthd1	Template for an external helpdesk configuration
sql	Database table creation/deletion scripts
unloads\SC6.2	ServiceCenter 6.2 customization unload files **Note:** SMSSMEX 1.10 unload files are available at <*SMSSMEX1.10 Release Package*>\unloads\ directory
unloads\SM9.2	Service Manager 9.2x customization unload files
unloads\SM9.3	Service Manager 9.3x customization unload files
logs	Log files
jdk	Internal JDK 5
tomcat	Tomcat 5.0.28
jre	Internal JRE by InstallAnywhere
AutoPassInstaller	HP AutoPass Licensing component installer (Windows only)
Uninstall SMSSMEX	Executable file for uninstallation

Configuring Tomcat

The connector for deploying the web service must be enabled. Uncomment the port specification in `<SMSSMEX_installDir>\tomcat\conf\server.xml`. For example:

```
<Connector port="8080"
  redirectPort="8443"
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
  enableLookups="false" acceptCount="100" debug="0"
  connectionTimeout="20000" disableUploadTimeout="true" />
```

You can modify the ports if necessary.

Setting up Database

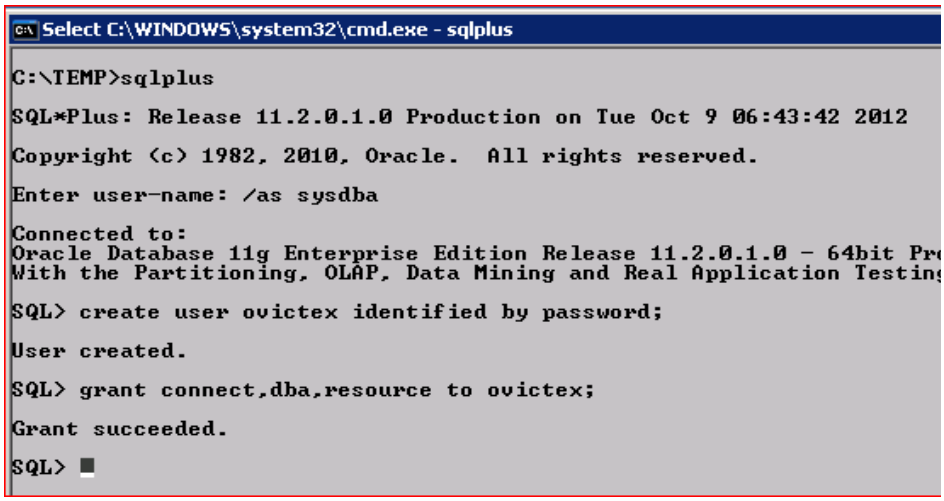
This section describes how to setup the database.

The SMSSMEX web service uses a database to store metadata. The SMSSMEX web service must be able to read table `v$database` (Oracle) or execute function `SERVERPROPERTY('ProductVersion')` (SQLServer). These system tables are queried when validating the database connections.

Oracle

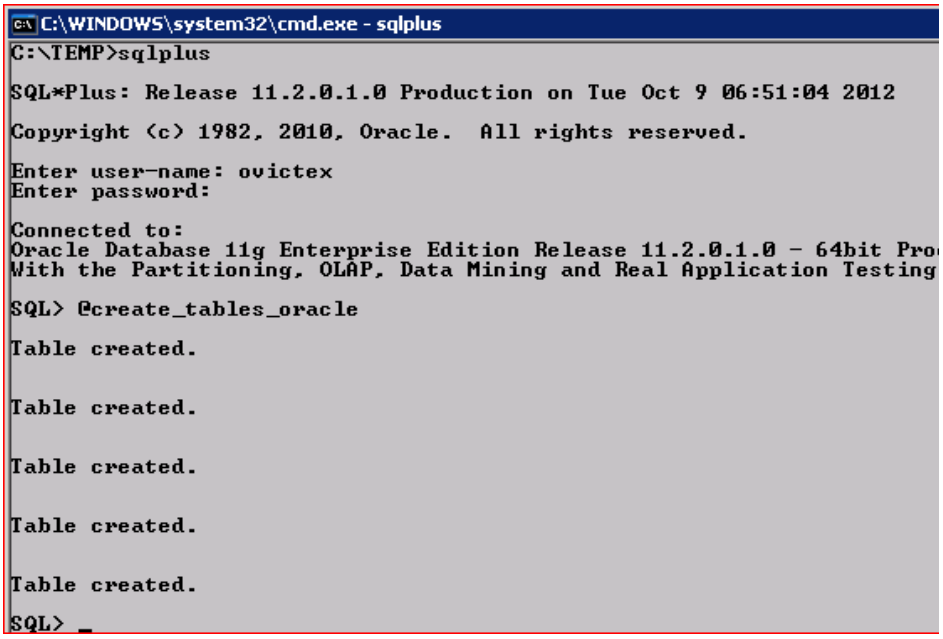
To setup the Oracle database do the following:

1 Create a user.



2 Give the user the right to do a select on table `v$database`. This system table is queried by the SMSSMEX web service to validate database connections.

3 Login as the user and run the script `create_tables_oracle.sql` (log in from path `<SMSSMEX_installDir>\sql` so that the script is found). This creates all required tables.

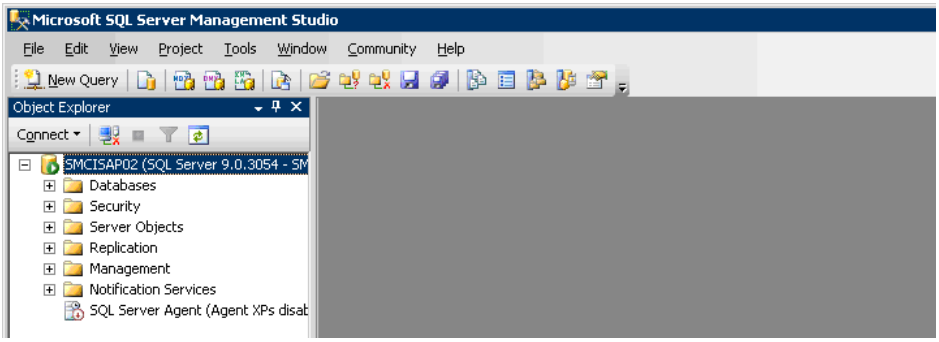


These tables are created within the schema of the database user (the tables are logically separated and do not interfere with each other).

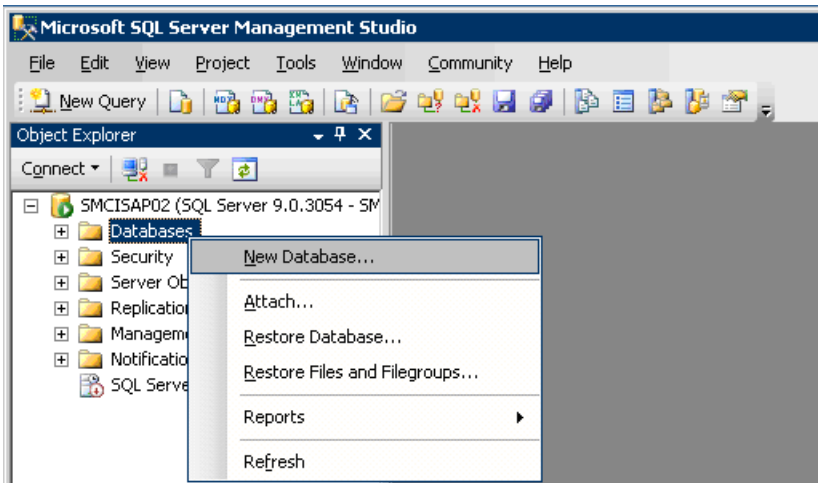
MS-SQL 2005

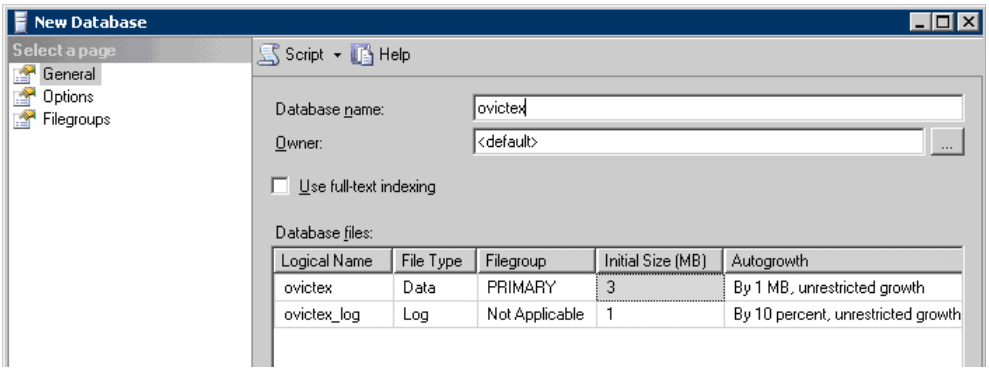
Do the following to create the required separate database for SMSSMEX tables:

1 Launch SQL Server Management Studio.

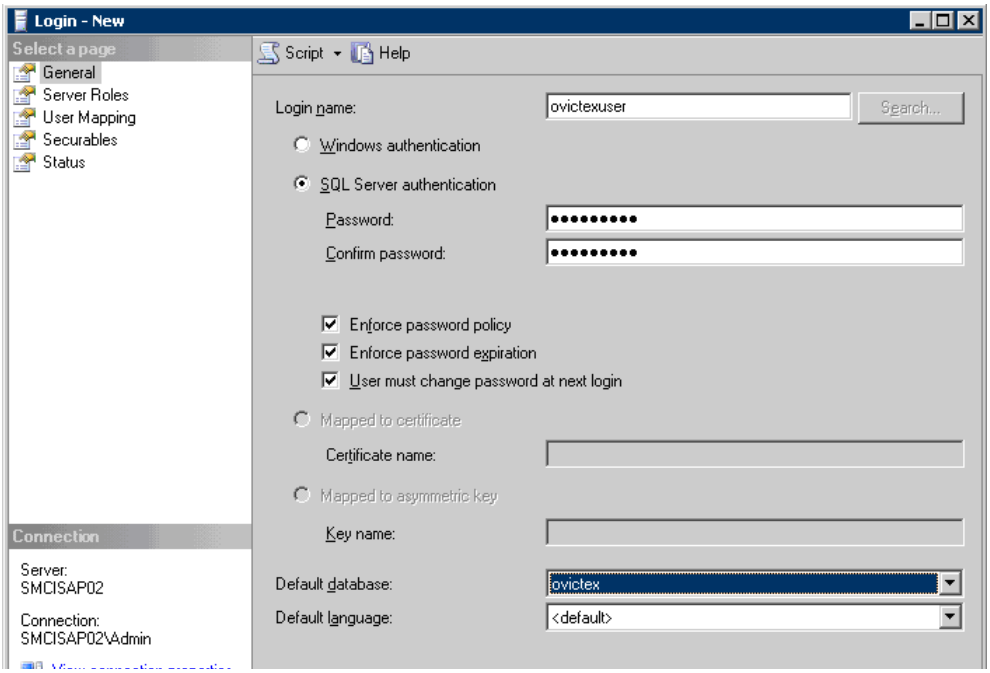


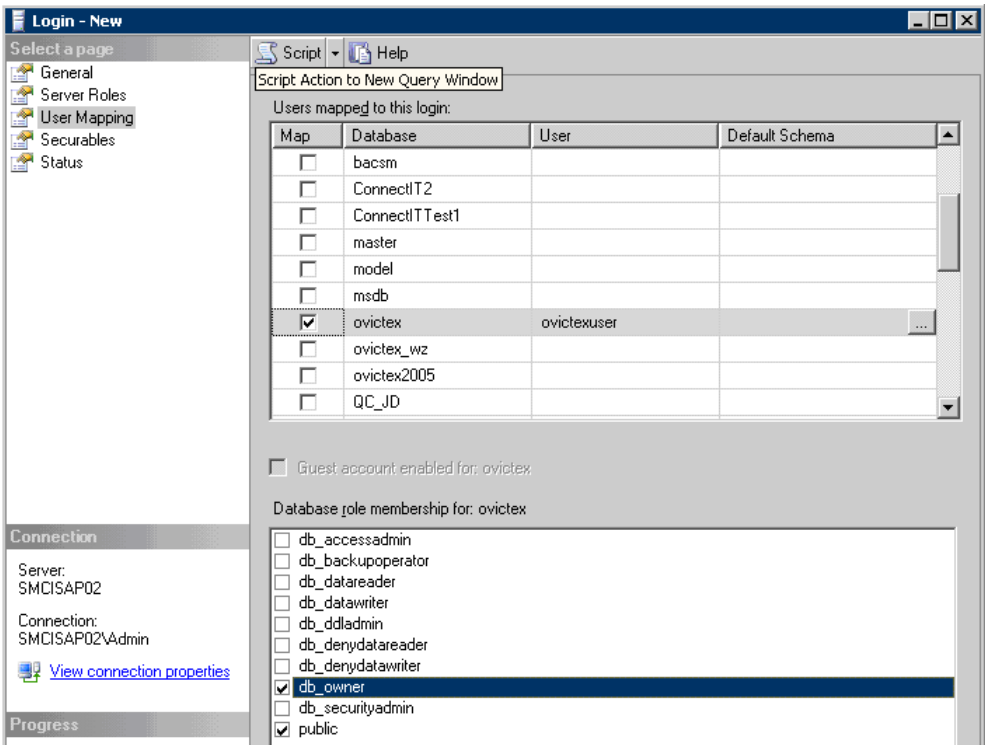
2 Create a new database (`ovictex`). Right-click on **Databases** and choose **New Database**.





3 Create a database user (`ovictexuser`) with permission for database `ovictex`. Right-click on **Security/Login** and select **New Login**.





4 Create the SMSSMEX tables.

a Click **New Query** on the toolbar and select database **`ovictex`**.

b Copy and execute the SQL scripts under folder

`<SMSSMEX_installDir>\sql\create_tables_sqlserver.sql`.

MS-SQL 2008

The DB setup for MS-SQL 2008 is similar to the MS-SQL 2005 setup. Refer to MS-SQL 2005 on page 21 for detail information.

Configuring ovictex.properties

File `<SMSSMEX_installDir>/config/ovictex.properties` must specify the local helpdesk installation. The file comments describe how to do this.

To configure the passwords, use command line application `<SMSSMEX_installDir>/bin/encryptPasswords.bat|sh` (do not enter the password directly in the file; passwords are stored in encrypted format). There are several sensitive fields that must be encrypted. These fields are discussed below. For more information about using `encryptPasswords.bat|sh`, see Tools on page 114.

The following parameters must be configured:

- Service Manager web service endpoint

— To connect to a Service Manager:

```
sc.webservice.endpoint = http://<ServiceManager host>:<Port>/
sc62server/PWS
```

— To connect to a ServiceCenter:

```
sc.webservice.endpoint = http://<ServiceCenter host>:<Port>/sc62server/ws
```

— The following are required parameters:

```
sc.user=<web service endpoint access user name>
sc.password=<encrypted password>
```

> `sc.password` must be filled by `encryptPasswords.bat|sh`. SMSSMEX supports SSL connections to Service Manager, but the parameter values are different than above and additional parameters must be set (see Security Between HP Service Manager and SMSSMEX on page 81).

- SMSSMEX database configuration information:

```
ovictex.db.type= <oracle | sqlserver>
ovictex.db.host=<database server address>
ovictex.db.port=<database server port number>
ovictex.db.instance=<sqlserver database server instance>
ovictex.db.name=<database name or oracle DB SID>
ovictex.db.user=<database user name>
ovictex.db.password=<database password>
```

> `ovictex.properties` contains examples. `ovictex.db.password` must be filled by `encryptPasswords.bat|sh`.

- One or more External Helpdesk instance names.
  - Parameters are **`exthd.instances.id.<number>`**, where *<number>* is a number {1,…,n}.
  - First number must be 1 and each number must be greater than the previous.
  - `ExtHdInstanceName` differentiates multiple External Helpdesks and is the name of the subfolder in *<SMSSMEX_installDir>*`/config` and the ExtHd configuration file.
- Property **`ovhd.incident.informationlog.entry.separator`** should be configured to a unique value that is not contained in messages exchanged between Helpdesks. By default it is configured to "**`----`**". Service Manager must be configured to use this separator to append information to the Journal. If this separator is contained in a message then duplicate information could be sent to the external Helpdesk (no data is lost).

Configuring File ovictexInternal.properties

The property file for internal configurations is in the *<SMSSMEX_installDir>*`/config` directory of the SMSSMEX installation. There is typically no need to configure this file.

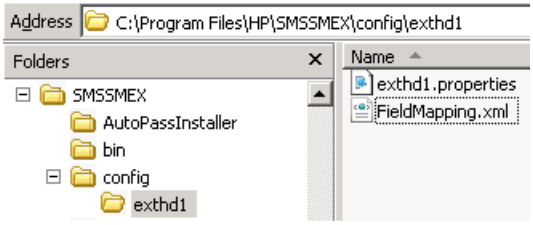
External Helpdesks

Main configuration file <*SMSSMEX_installDir*>/config/ovictex.properties must define all External Helpdesk Instances. For example:

- **exthd.instances.id.1 = exthd1**
- **exthd.instances.id.2 = SAP_exthd2**
- **exthd.instances.id.3 = NY200BM**

Each external helpdesk has the following configuration files:

- <*ExtHdInstanceName*>.properties
- FieldMapping.xml



The same names (such as exthd1, SAP_exthd2, NY200BM) must be used for the names of subfolders with specific configuration file names. The names must not contain spaces or special characters. The default configuration comes with a defined exthd1 sample External Helpdesk configuration.

To create a new instance:

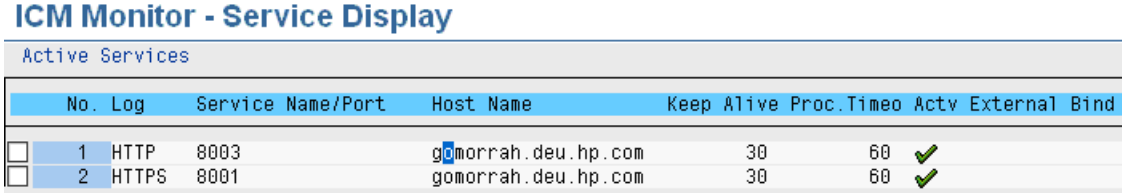
1. Add a new line in ovictex.properties for the new ExtHd.

   exthd.instances.id.2 = exthd2

2. Create the new subfolder <*SMSSMEX_installDir*>/config/exthd2.
3. Copy the configuration files for exthd1 to exthd2.
4. Rename <*SMSSMEX_installDir*>/config/exthd2/exthd1.properties to <*SMSSMEX_installDir*>/config/exthd2/exthd2.properties.
5. Make the required changes to the new files.
6. The following parameters must be configured in <*ExtHdInstanceName*>.properties:

   exthd.webservice.endpoint = http://<*SolutionManager host*>:<*Port*>/sap/bc/srt/rfc/sap/ICT_SERVICE_DESK_API?SAP-CLIENT=<*SAP client number*>
   exthd.webservice.authentication.scheme = BASIC
   exthd.webservice.authentication.username = <*SAP client user name*>
   exthd.webservice.authentication.password = <*encrypted SAP client user password*>

   IMG activity guides you to SAP transaction **/nsmicm**. Select the activity in menu **Goto** → **Services**.



This transaction shows the host and port for access to the SAP Solution Manager Service Desk web service. Specify the host/port in `<ExtHdInstanceName>.properties` as the endpoint entry.

➤ `exthd.webservice.authentication.password` must be filled by `encryptPasswords.bat|sh`.

Configuring FieldMapping.xml

The files `<SMSSMEX_installDir>/config/<ExtHdInstanceName>/FieldMapping.xml` must be adjusted to send/receive special/customized fields to/from the external Helpdesk. For detailed information see Field Mapping Configuration on page 115.

Verifying Configuration

Verify the configuration with the checker tool before trying to exchange incidents between Service Manager and SAP Solution Manager. The checker error messages are much more helpful for troubleshooting than Service Manager and Solution Manager error messages.

To execute the checker run

```
<SMSSMEX_installDir>/bin/checker.bat|sh
```

Checker checks the environment, database and HTTP connections and configuration of Service Manager. No incidents are exchanged. The following are the possible results:

- OK
- ERROR (partial failure; checks that the check depends on have failed)
- FAIL (with troubleshooting recommendations)

You can re-run a check by passing the number of the check to the executable. You can also examine the Incident Exchange log messages or run a trace. For more information about using checker.bat|sh, see Tools on page 114.

Starting/Stopping SMSSMEX

Starting from Windows:

```
<SMSSMEX_installDir>\bin\setup startup
```

Stopping from Windows:

```
<SMSSMEX_installDir>\bin\setup shutdown
```

Starting from Linux:

```
<SMSSMEX_installDir>/bin/setup.sh startup
```

Stopping from Linux:

```
<SMSSMEX_installDir>/bin/setup.sh shutdown
```

4 Installing and Configuring SMSSMEX on WebLogic

Installing SMSSMEX

See Installing SMSSMEX on page 17 for detailed instructions.

Setting up Database

See Setting up Database on page 20 for detailed instructions.

Configuring ovictex.properties

See Configuring ovictex.properties on page 23 for detailed instructions.

Configuring File ovictexInternal.properties

See Configuring File ovictexInternal.properties on page 24 for detailed instructions.

External Helpdesks

See External Helpdesks on page 25 for detailed instructions.

Configuring FieldMapping.xml

See Configuring FieldMapping.xml on page 26 for detailed instructions.

Verifying Configuration

See Verifying Configuration on page 26 for detailed instructions.

Deploying on WebLogic

1. Before starting the WebLogic server, set an environment variable named "SMSSMEX_HOME" to the pathname where this application is installed.

   For example, if the WebLogic server is installed in the `/opt/HP/SMSSMEX` directory, set the environment variable to the following:

   ```
   $ export SMSSMEX_HOME=/opt/HP/SMSSMEX
   ```

2. Start the WebLogic server and launch the WebLogic administration console.
3. Deploy the `ovictex.war` file in the `/opt/HP/SMSSMEX/war` directory. See the following steps for an example:

   a. Select **Domain Structure > Deployments** and click **Install**.

   b. Use the Install Application Assistant to locate the `ovictex.war` file.

   c. Select **Install this deployment as an application** and click **Next** until last step.

   d. Click **Finish** to exit the installation wizard.

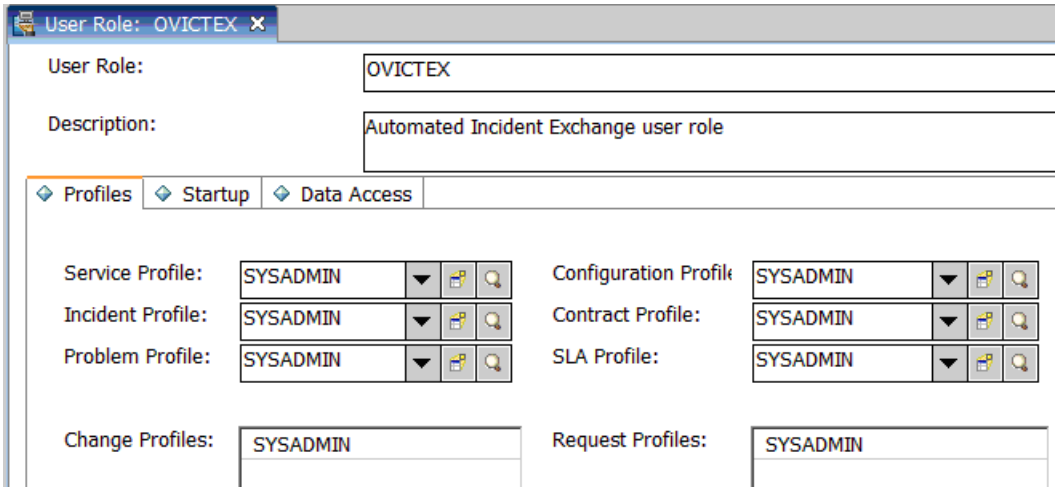
   For advanced configuration, refer to *BEA WebLogic Server Administration Console Online Help* for more information.

5 Customizing HP Service Manager

This chapter describes the customization required for HP Service Manager for the integration.

Creating a Service Manager User for Web Service

Incident Exchange uses one Service Manager user to connect to Service Manager web services. The user and the user role should be dedicated for the integration. The user requires the following permissions:



Do the following:

1. Log in to Service Manager with a System Administrator account.
2. Select **System Administration** → **Ongoing Maintenance** → **User Roles**.
3. Search for **`system administrator`** on Service Manager 9.x and above.

   In case your database is configured to case sensitive, try to use all lowercase search keyword instead of all UPPERCASE one, or vice versa.

4. Enter **`OVICTEX`** as the User Role.
5. Change Description to **`Automated Incident Exchange user role`**.
6. Click **Add**.
7. Select **System Administration** → **Ongoing Maintenance** → **User Quick Add Utility**.
8. Enter **`ovictex, INCIDENT EXCHANGE, Incident, Exchange, ovictex@hp.com`**.
9. Click **Next**.

10 For **User to clone** select **falcon**.

11 Click **Finish**.

12 Click **Save**.

13 Go to **System Administration** → **Ongoing Maintenance** → **Operators**, enter **`ovictex`** in the Login Name field, then click **Search**.

14 Change User Role to **`OVICTEX`**.

15 In the Security tab:

   a Enter the operator password for Password.

   b Uncheck Expire Password.

   c Check Never Expire Password.

16 Click **Save**.

Importing Customizations via Unload

This section describes how to configure Service Manager using unload. Additional customization of Service Manager is later required for the integration.

Core Unload

Unloads are used to transfer customizations from one Service Manager installation to another Service Manager installation. The Incident Exchange provides core unloads at the following paths:

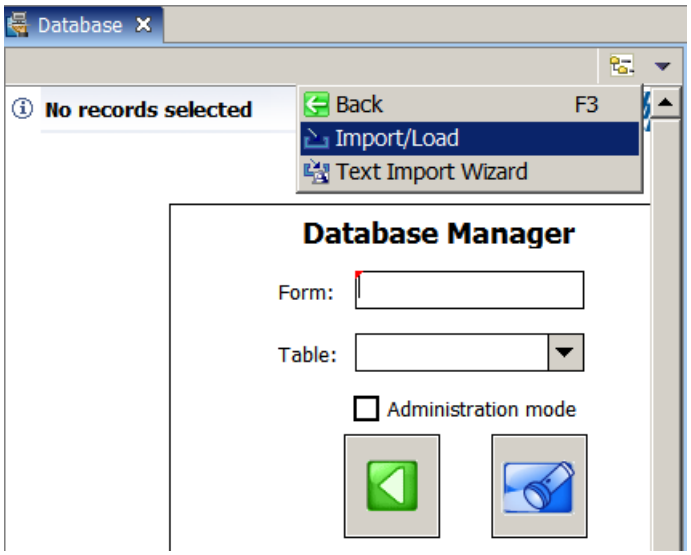
- `<SMSSMEX1.10 Release Package>\unloads\SM9.2\core_sm9.2.unl` for Service Manager 9.2x
- `<SMSSMEX1.10 Release Package>\unloads\SM9.3\core_sm9.3.unl` for Service Manager 9.3x

This unload contains new Service Manager records that are unique to Incident Exchange and do not override any existing Service Manager records.

To import the unload do the following:

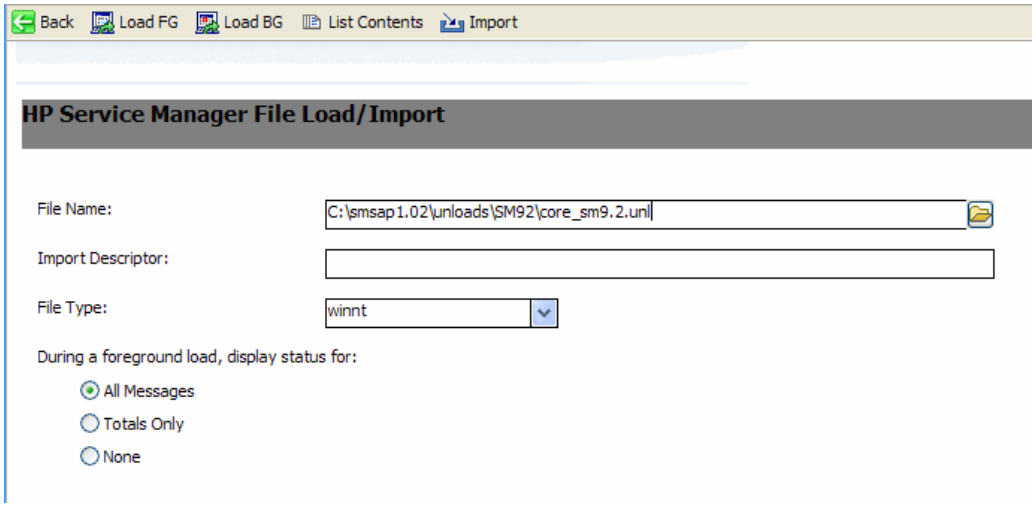
1 In the Service Manager client select **Tailoring** → **Database Manager**.

2 Select **Import/Load** from the menu.



3 Select `<SMSSMEX1.10 Release Package>\unloads\SM9.2\core_sm9.2.unl` for Service Manager 9.2x, or select `<SMSSMEX1.10 Release Package>\unloads\SM9.3\core_sm9.3.unl` for Service Manager 9.3x.

4 Click **Load FG** to start the import.



Demo Unload

The demo unload has all the customizations (for an uncustomized, default installation of Service Manager) required to set up a working Incident Exchange for a customer demonstration or evaluation.

Do not import the demo unload into existing development or production systems. The demo unload requires an uncustomized, default installation of Service Manager. The demo unload overrides many standard Service Manager records, and can not be removed or undone.

To import the unload do the following:

1 In the Service Manager client select **Tailoring** → **Database Manager**.

2 Select **Import/Load** from the menu.

3 Browse to the unload at

— `<SMSSMEX1.10 Release Package>\unloads\SM9.2\demo_sm9.2.unl` for Service Manager 9.2x

— `<SMSSMEX1.10 Release Package>\unloads\SM9.3\demo_sm9.3.unl` for Service Manager 9.3x

4 Press **Load FG** to start the import.

Customizing Demo Unload Manually

This section describes how to manually customize with the demo unload.

Adding Integration Name to the Info Table

For every SMIS integration, a field with the integration name should be created in the Info table.

1 Log in to Service Manager with a System Administrator account.

2 Select **System Definition** → **Tables** → **info** → **Tab Fields and Keys**.

3 Create the following field in SMIS structure.

Field Name	Data type
SMSAP	Logical

4 Click **Save**.

Incident Custom Fields and Web Service Exposure

Incident Exchange accesses Service Manager Incidents via the `probsummary` table. The factory-default exposure `IncidentManagement.wsdl` (service name `IncidentManagement` and object name `Incident`) is used, allowing Incident Exchange to function with other clients. Incident Exchange requires exposure of additional fields in the web service and creation of new fields.

1 Select **System Definition** → **Tables** → **probsummary** → **Tab Fields and Keys**.

2 Create the following additional fields in table `probsummary`.

Table 3 Incident custom fields and web service

Field name	Caption	Data type
custom.text.01	CustomText01	Character
custom.text.02	CustomText02	Character
...	...	...
custom.text.10	CustomText10	Character

Table 3 Incident custom fields and web service (cont'd)

sap.sid	Sap SID	Character
sap.client	Sap Client	Character
sap.installationnumber	Sap installation number	Character
hidden.meta.data	Hidden meta data for Incident Exchange	Character
is.incident.exchange	Flag for affiliation with Incident Exchange	Logical
exthd	External helpdesk for Incident Exchange	Character
sap.incident.type	Sap Incident type	Character

3 The history is written to an additional field in `probsummary` table:

- Add an Array `exchange.history`.
- Add structures that are also named `exchange.history`.
- Add structure fields `date.stamp` of type Date/time and `history.update` of type Text.

4 Incident Exchange is triggered asynchronously, requiring a handshaking mechanism to avoid triggering an action multiple times. To implement the mechanism, add a Boolean field named `is.ictex.action.blocked`. The fields are updated through Event Services and do not need to be exposed.

Field name	Type	Caption
exchange.history exchange.history date.stamp history.update	Array Structure Date/time Character	Log of Incident Exchange actions and events.
is.ictex.action.blocked	Logical	Flag that indicates if Incident Exchange is performing an exchange action.

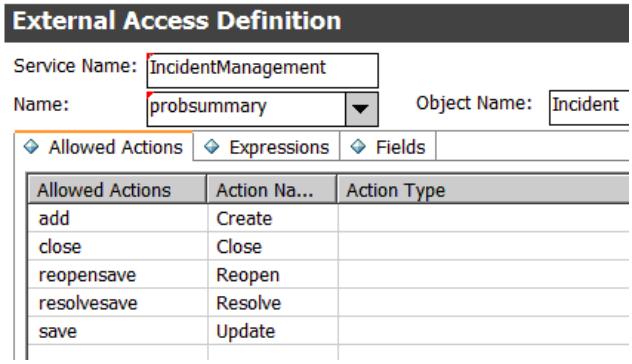
Expose Custom Fields in Web Service Interface IncidentManagement

To expose custom fields, do the following:

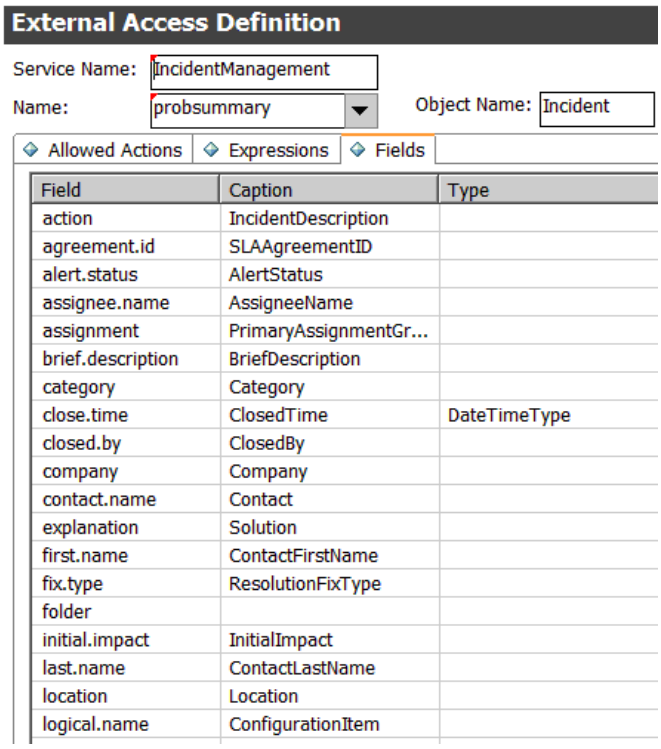
1 Select **Tailoring** → **Web Services** → **WSDL Configuration** on Service Manager 9.31 and earlier, or select **Tailoring** → **Web Services** → **Web Service Configuration** on Service Manager 9.32 and later.

2 Enter **`IncidentManagement`** in the `Service Name` field.

3 Click **Search**.



4 Select the **Fields** tab.



5 Add the custom fields in table `probsummary` that are exposed in web service interface IncidentManagement. If the field type is `Character`, then its not required to select the web service interface type for the field. The type of the field will be `StringType`.

Table 4 Custom fields in IncidentManagement

Field	Caption	Type
custom.text.01	CustomText01	
custom.text.02	CustomText02	
...	...	
custom.text.10	CustomText10	
sap.sid	SapSid	
sap.client	SapClient	

Table 4 Custom fields in IncidentManagement (cont'd)

sap.installationnumber	SapInstallationNumber	
hidden.meta.data	HiddenMetaData	
is.incident.exchange	IsIncidentExchange	BooleanType
exthd	Exthd	
priority.code	PriorityCode	
planned.end	PlannedEnd	DateTimeTyp e

6 Change Expressions from

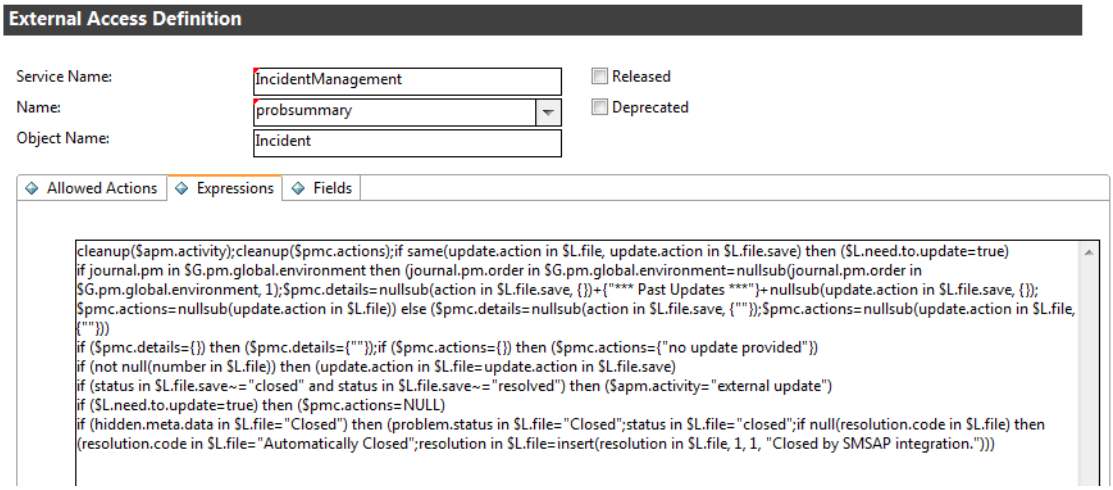
```
update.action in $L.file=update.action in $L.file.save
```

to (**[INCIDENTMANAGEMENT]** in `code_sm9.txt`)

```
if (not null(number in $L.file)) then (update.action in
$L.file=update.action in $L.file.save)
```

Then, add the following expression:

```
if (hidden.meta.data in $L.file="Closed") then (problem.status in
$L.file="Closed";status in $L.file="closed";if null(resolution.code in
$L.file) then (resolution.code in $L.file="Automatically
Closed";resolution in $L.file=insert(resolution in $L.file, 1, 1, "Closed
by SMSAP integration.")))
```



Contacts Web Service Exposure

To expose the contacts web service, do the following:

1 Select **Tailoring** → **Web Services** → **WSDL Configuration** on Service Manager 9.x.

2 Input **`contacts`** in the Name field.

3 Click **Search**.

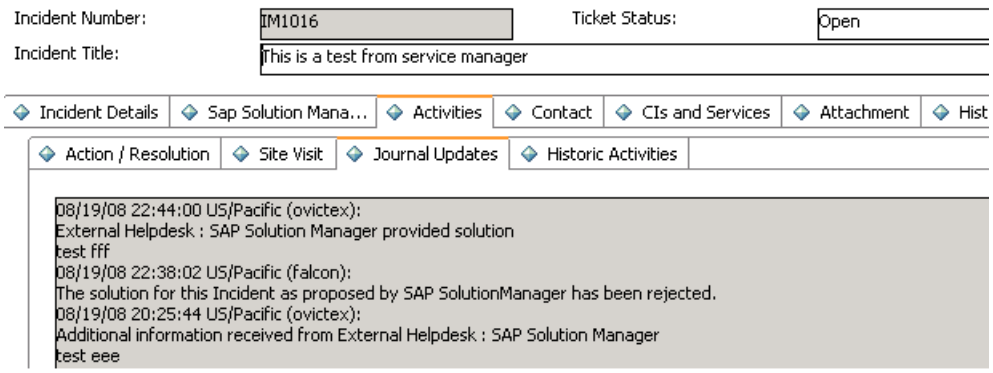
4 Select the **Fields** tab.

5 Add the following fields:

Field	Caption	Type
fax.phone	Fax	
operator.id	OperatorID	

Journal Separator Line Format

New entries are added to the Journal at the top. When an Incident is exchanged with SAP Solution Manager, only updates are exchanged to avoid duplication of journal entries.



The Incident Exchange separator string separates blocks in the Journal, allowing easy identification of new blocks that must be sent. The string is configured in the `ovictex.properties` file (property `sd.incident.informationlog.entry.separator`). The configured value must match the string in the customized Service Manager.

For this customization, all processes in the Document Engine related to Incident updates must be updated with the separator between Journal entries starting with the configured string (default is "`----`"). In the default Service Manager installation, the following processes are affected:

- `im.save`
- `im.close`
- `im.resolve`
- `im.reopen`
- `im.first`

The Incident Exchange core unload provides an additional process `im.exchange.incident` that already contains the modification. However, this must also be modified if the separator string deviates from the default.

1 Select **Tailoring** → **Document Engine** → **Processes**.

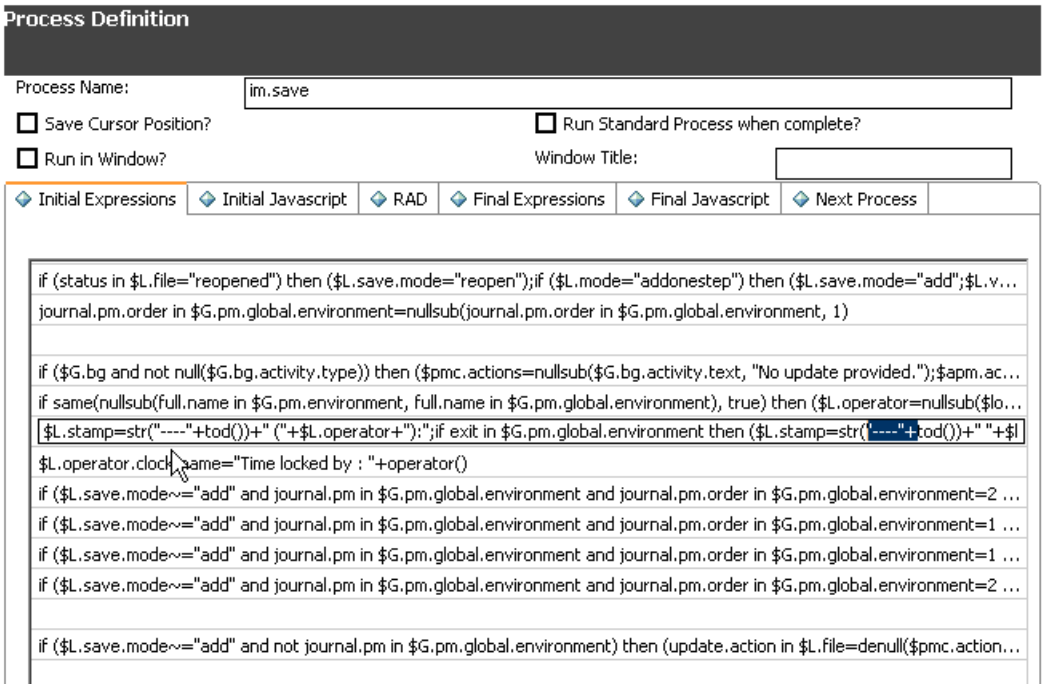
2 Search for all Processes starting with **`im.`**.

3 In Initial Expressions, look for modifications of the Journal timestamp separator (variable `$L.stamp`), and add the configured separator string to the beginning as shown in the following

```
$L.stamp=str("----"+tod())+" ("+$L.operator+"):".
```

And the final entire line (**[IM._JOURNAL]** in `code_sm9.txt`) is as follows:

```
$L.stamp=str("----"+tod())+" ("+$L.operator+"):";if exit in
$G.pm.global.environment then ($L.stamp=str("----"+tod())+"
"+$lo.time.zone+" ("+$L.operator+"):")
```



Template

When a new Incident is sent from SAP Solution Manager to Service Manager, Incident Exchange creates a new Incident with data for exchanged fields. The Incident management process inside Service Manager may require additional mandatory fields (such as `category`, `subcategory`, and `product type`) that must be filled out in order to submit the Incident. Values for these fields must be provided when the Incident is opened by Incident Exchange. In an uncustomized Service Manager, the Process `im.first` is invoked when an Incident is submitted.

1. Select **Tailoring** → **Document Engine** → **Processes**.
2. Search for `im.first`.
3. Add the following expressions to the Initial Expressions tab of `im.first`.

   *On Service Manager 9.x (***[IM.FIRST_INIT]** *in* `code_sm9.txt`*):*

   ```
   if (is.incident.exchange in $L.file=true) then (category in
   $L.file="incident";subcategory in $L.file="access";product.type in
   $L.file="authorization error";problem.type in
   $L.file="incident";assignment in $L.file="Application";if null(severity in
   $L.file) then (severity in $L.file="4");initial.impact in
   $L.file="1";site.category in $L.file="B";action in $L.file={"default
   description"});affected.item in $L.file="MyDevices"
   $L.comment="siehe JS"
   if same(nullsub(full.name in $G.pm.environment, full.name in
   $G.pm.global.environment), true) then ($L.operator=nullsub($lo.ufname,
   nullsub(operator(), "NULL"))) else ($L.operator=nullsub(operator(),
   ```

```
"NULL"))
$L.stamp=str("----"+tod())+" ("+$L.operator+"):";if exit in
$G.pm.global.environment then ($L.stamp=str("----"+tod())+"
"+$lo.time.zone+" ("+$L.operator+"):")
if (is.incident.exchange in $L.file=true) then (update.action in
$L.file={$L.stamp}+denull(update.action in $L.file))
```

4 In addition, if you want to set different default values for incident from different SAP client, you can update the code:

```
if (is.incident.exchange in $L.file=true) then (category in
$L.file="incident";subcategory in $L.file="access";product.type in
$L.file="authorization error";problem.type in
$L.file="incident";assignment in $L.file="Application";if null(severity in
$L.file) then (severity in $L.file="4");initial.impact in
$L.file="1";site.category in $L.file="B";action in $L.file={"default
description"});affected.item in $L.file="MyDevices"
to multiple line of
if (is.incident.exchange in $L.file=true and exthd in $L.file="exthd1")
then (category in $L.file="telecoms";subcategory in $L.file="fixed
infrastructure";product.type in $L.file="fixed
infrastructure";problem.type in $L.file="not specified";assignment in
$L.file="AUTO";severity in $L.file="1";initial.impact in
$L.file="1";site.category in $L.file="B";action in $L.file={"default
description"})
```

where `exthd1` refers to your SAP client exthd1.

The Logical field `is.incident.exchange` in `probsummary` is set by Incident Exchange and indicates if the Incident is opened by the Incident Exchange (or some other way). If multiple external helpdesks are connected to Service Manager via the Incident Exchange, the text field `exthd` can be compared with the configured external helpdesk IDs in order to set different values, depending on where the Incident originated from.

Incident Process

To control the visibility of SAP Solution manager according to SMSAP instance's status, the im.view.init and the im.open.setup incident processes needs to be customized:

1 Log in to Service Manager with a System Administrator account.

2 Select **Tailoring** → **Document Engine** → **Processes**.

3 Enter **`im.view.init`** in the Process Name field and click **Search**.

4 Add the following expression to the end of the Initial Expressions tab:

```
$SMSAP=nullsub(SMSAP in $G.system.info, "false")
```

5 Add the following scripts to the Initial Javascript tab (**[IM.VIEW_INIT]** in `code_sm9.txt`):

```
var configItem =
lib.smis_ConfigurationManager.getEnabledConfigItem("SMSAP");
if (configItem != null) {
    var solMans = configItem.getParametersByCategory("SolutionManager");

    var values = [];
```

```
    var names = [];

    for (var id in solMans) {
      values.push(id);
      names.push(solMans[id]);
    }
    system.vars.$G_solMans_values=values;
    system.vars.$G_solMans_names=names;
}
```

6 Click **Save** and **OK**.

7 Enter **`im.open.setup`** to the name field and click **Search**.

8 Add the following expression to the Initial Expressions tab:

```
$SMSAP=false
```

9 Click **Save** and **OK**.

Incident Form

Incident Exchange must be integrated into the incident management workflow. The operator working on the incident must be able to control and trigger Incident Exchange. If more than one external helpdesk is connected to Service Manager, then the target system must be selected.

Status and Hidden Metadata

The hidden metadata field stores the current Incident Exchange state and Service Manager role (`Requester` or `Provider`). This field determines which actions are currently valid for the Incident. The field is updated by Incident Exchange. Customizations can read but must not write this field.

The Incident exchange state model must be integrated into the Incident workflow. Updates to the hidden metadata field by Incident Exchange can change the Incident status. For example, when a solution has been proposed by SAP Solution Manager, the assigned Service Manager operator must be notified that a new solution is now available for processing. This can be done by inspecting the hidden metadata field and putting the Incident into a special queue if the status has been changed to **`SolutionProvided`**.

Exchange History

Incident Exchange keeps a log of all exchange actions and failures. Information from the log and hidden metadata field can be used to explain to the operator what kind of problem occurred.

The `probsummary` table contains the Array field `exchange.history` that contains a Structure of `date.stamp` and `history.update`. The table can be shown on the Incident Form as a table with two columns. The table can be placed anywhere. In the following example it is placed in a subform in a separate notebook tab named “SAP Solution Manager”. Additional elements (such as a combo box for selection of external helpdesks) can be placed on the notebook tab.

1 Open all Incident Forms that are parts of the Incident workflow.

On Service Manager 9.x:

`IM.open.incident`

`IM.update.incident`

`IM.close.incident`

`hp.sap.solution.sub`

2 Embed the created subform on the Incident form in a new Notebook tab (or in a new section on Service Manager 9.2x and above).

3 Add a Notebook tab or section to the following forms:

On Service Manager 9.x:

`IM.open.incident`

`IM.update.incident`

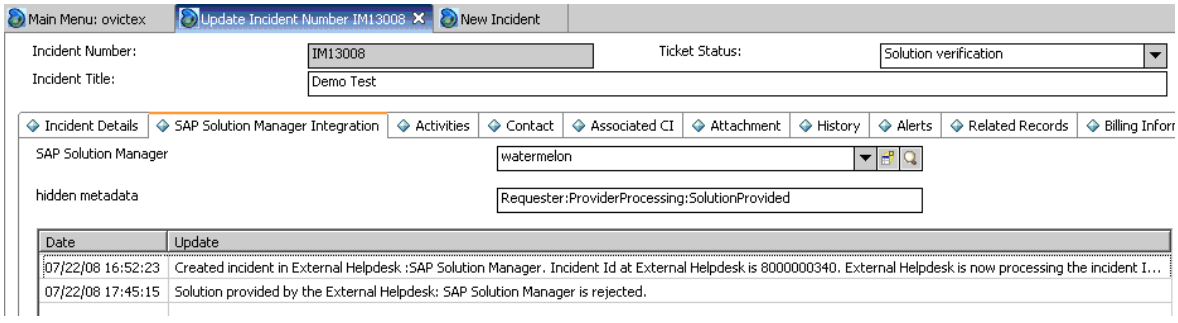
`IM.close.incident`

`hp.sap.solution.sub`

Property	**Value**
Caption	Sap Solution Manager
Visible condition	[$SMSAP]=true

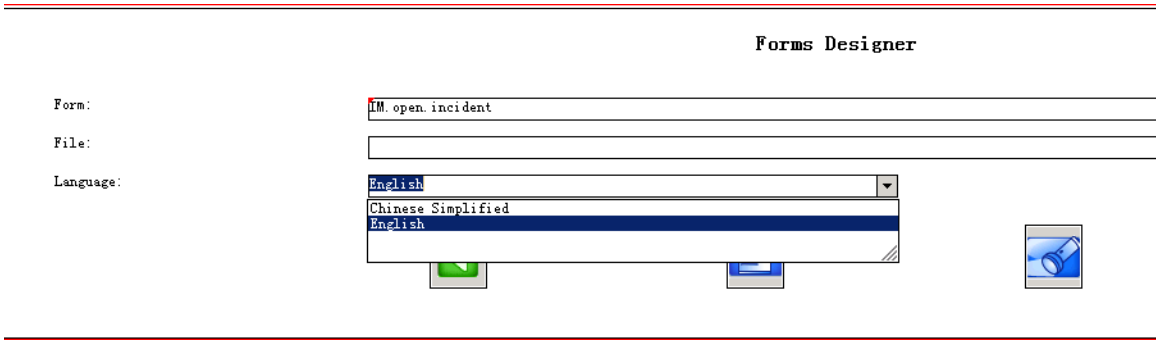
4 add a subform control into the SAP Solution Manager tab or section.

Property	**Value**
X	1
Y	0
Width	151
Height	28
Format	hp.sap.solution.sub



If multiple Language packs are applied to HP Service Manager, do the following to update the incident related forms for other languages.

1. Copy the `hp.sap.solution.sub` form from English to other languages and perform translation.
2. Repeat aforementioned step 3 and step 4 for `IM.open.incident`, `IM.update.incident` and `IM.close incident` for other languages.



Trigger Buttons

The Incident Exchange web service is triggered by an HTTP request from Service Manager. This request is submitted by a JavaScript function in the Script Library. The trigger transmits the

- Incident ID
- Action that triggered the status change
- ID of the external helpdesk

Before triggering an Incident exchange, the Incident must be saved and the `exthd` field set. An `im.exchange.incident` process performs the save operation and invokes the JavaScript trigger function.

A straightforward way is to allow operators to trigger Incident Exchange actions via additional buttons on the Incident form. Instead of pressing **New**, **Save** or **Close**, the operator selects to **Send Incident**, **Add Info** or **Refuse Solution**, and so on. The implementation must follow the Incident Exchange state diagram. Display options that enable or disable the trigger buttons must inspect the value of the hidden metadata do decide which trigger actions are currently available.

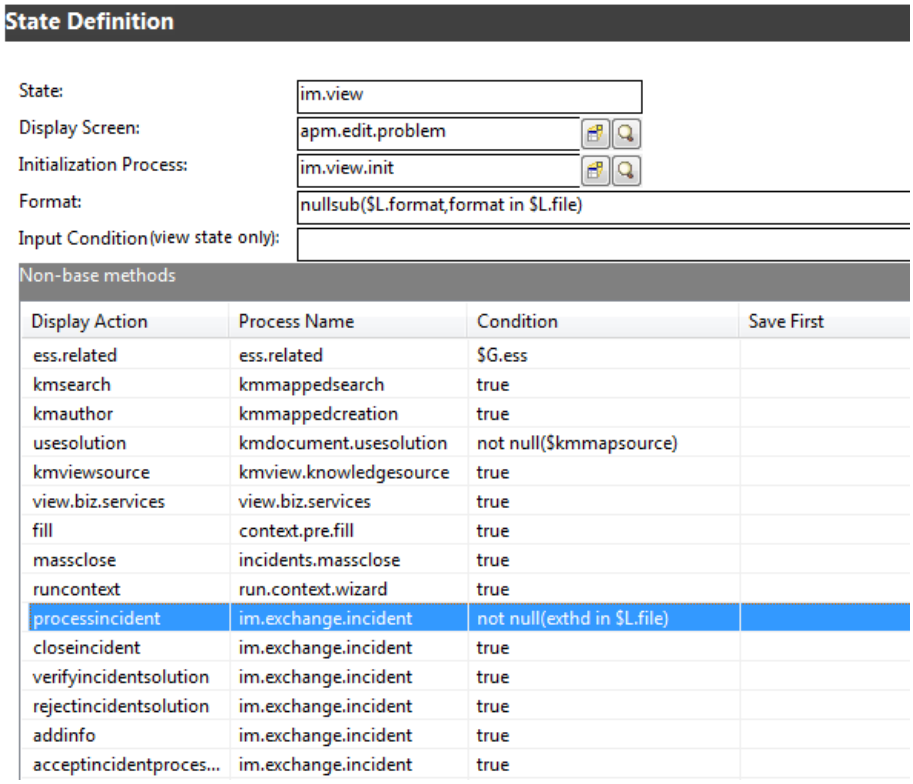
An action cannot be triggered multiple times, since the action request is sent asynchronously to the Incident Exchange. The exchange state of the Incident will only be updated (via Event In) during processing with the external helpdesk. Event In can only update the Incident that is not locked. This typically means that the operator has to abandon or refresh the Incident after invoking the Incident Exchange. An exception is the `Addinfo` action, which does not change the exchange state, but only synchronizes updates with the external helpdesk (and can thus be invoked multiple times without updating the Incident in Service Manager.) To block the action buttons after a button has been pressed (and trigger invoked) until the updated incident has been reloaded (including the updated exchange status modified via Event In), evaluate the field `is.ictex.action.blocked`. This field (initially NULL) is set by the trigger process (`im.exchange.incident`) and cleared via EventIn.

To set up buttons for the Incident Exchange:

1 Select **Tailoring** → **Document Engine** → **States** and search for `im.view`.

The display options are all created after the core unload is imported.

2 Connect the newly created Display Options with the provided Process `im.exchange.incident`.



Display Action	Process Name	Condition
processincident	im.exchange.incident	not null(exthd in $L.file)
closeincident	im.exchange.incident	true
verifyincidentsolution	im.exchange.incident	true
rejectincidentsolution	im.exchange.incident	true
addinfo	im.exchange.incident	true
acceptincidentprocessing	im.exchange.incident	true

To disable the duplicate button for exchange incidents:

1 Select **Tailoring** → **Tailoring Tools** → **Display Options**.

2 Enter **`apm.edit.problem_clone`** in the Unique ID field and click **Search**.

3 Add **`and hidden.meta.data in $L.filed=NULL`** to the end of the Condition field.

Selection of External Helpdesk System

If Service Manager is connection to multiple SAP Solution Manager helpdesks, then the helpdesk must be selected before initiating Incident Exchange. This could be implemented with new trigger buttons (“Send to SolMan1”, “Send to SolMan2”) or a Combo box on the Incident form. The helpdesk could be automatically selected based on the assigned operator or workgroup (or whatever the Incident workflow requires). If the connection is fixed between one SAP Solution Manager system and Service Manager, then the value can be hardcoded. In any case, the `exthd` field must be set before the Incident is exchanged.

SAP Configuration Item handling

Overview

From an SAP perspective, a Configuration Item (CI) is identified by three attributes:

- `Installation number`
- `SID`
- `Client`

Incident Exchange can send the SAP CI information that is attached to an Incident to SAP Solution Manager, and associate an SAP CI with an Incident based on the CI information provided by SAP Solution Manager. In Service Manager, SAP CI’s may be modeled and set up in any way, as long as the three identifying attributes are present.

Implementation

Incident Exchange stores SAP CI information in three fields in the `probsummary` table

- `sap.sid`
- `sap.client`
- `sap.installationnumber`

The Service Manager customization implements the bi-directional synchronization between the `Incident` fields and the Service Manager CIs, allowing the Incident Exchange to be adapted to any existing SAP CI configuration.

Example Implementation via New Device Type and fill.fc

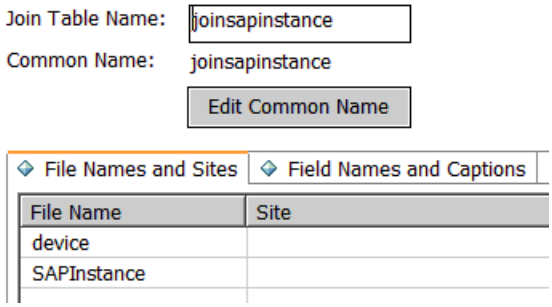
The following describes an example implementation of a new device type `SAPInstance` (created via **Configuration Management** → **Administration** → **Add New Device Type**). This device type needs fields for SAP SID, client and installation number. A new table SAPInstance should be created in advance with the following fields:

Table 5 New SAPInstance table fields

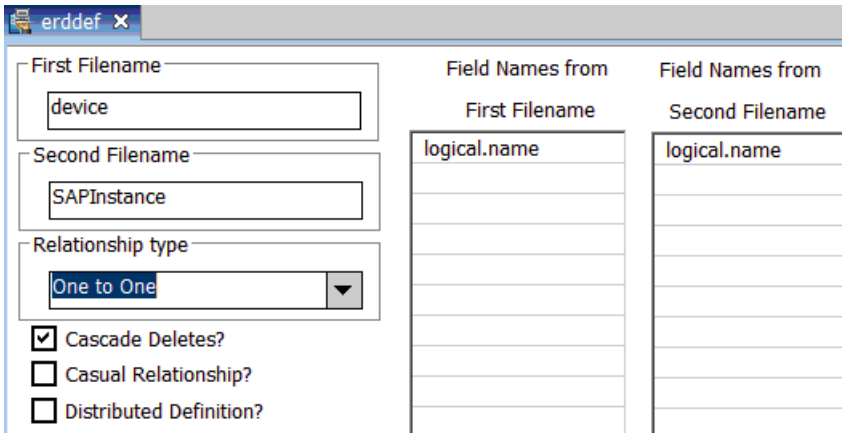
Field name	Type	Caption	Other Properties	
SID	Character	SID	Not null	Unique
client	Character	client	Not null	
installation.number	Character	installation.number	Not null	
logical.name	Character	logical.name		Unique

Do the following:

1 Generate a new join Def named `joinsapinstance` in **System Definition** → **Tables** → **joindefs** → **Forms** → **joindefs.g** → **Database Manager**.



2 Generate a new erddef in **System Definition** → **Tables** → **erddef** → **Forms** → **erddef.g** → **Database Manager**.



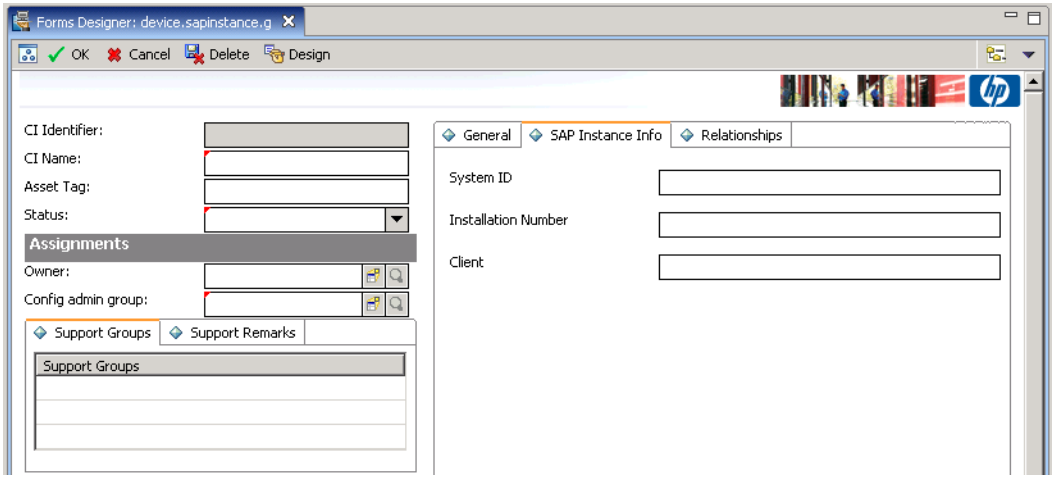
3 Create a form view for the device type you want to create in Service Manager.

To create form `device.sapinstance.g` for SAP Solution Manager device type, you can copy an existing form of device. For example,

— *On Service Manager 9.x*

Go to **Tailoring** → **Form Designer**, enter **`configurationItem`** in the Form field, then click **Search** to open the form in Forms Designer view, and click **Copy/Rename** in the pop-up menu to copy the form, and rename the newly copied form as `device.sapinstance.g`.

Then add a new tab **SAP Instance Info** in the form, and remove other tabs except for General and Relationships from the form:

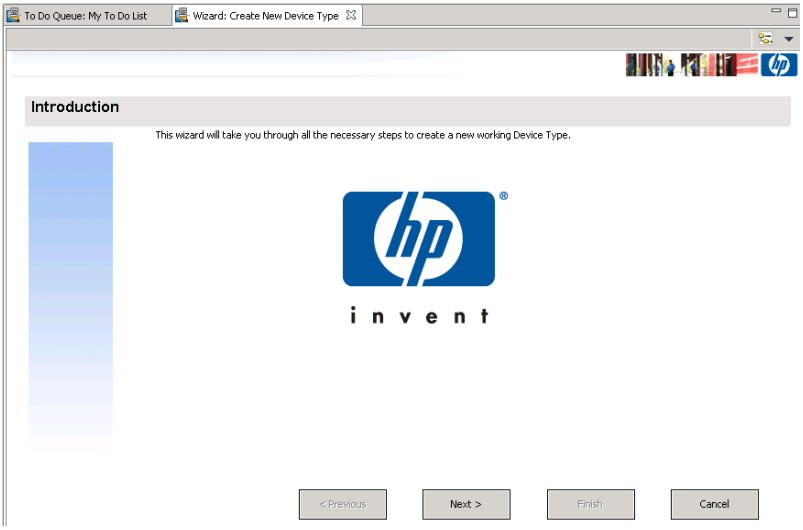


The SAP Instance Info tab form should include at least three fields: System ID, Installation Number, and Client.

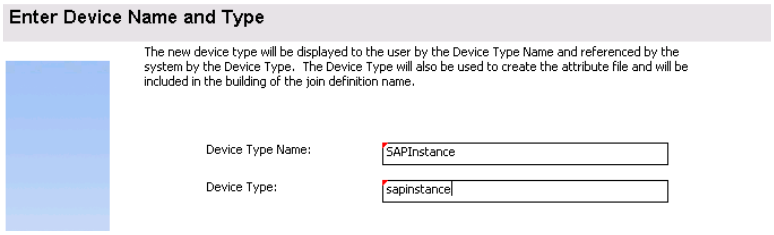
Control Component	Property	Value
System ID	input	SID
Installation Number	input	installation.number
Client	input	client

4 Add a new device type in Service Manager.

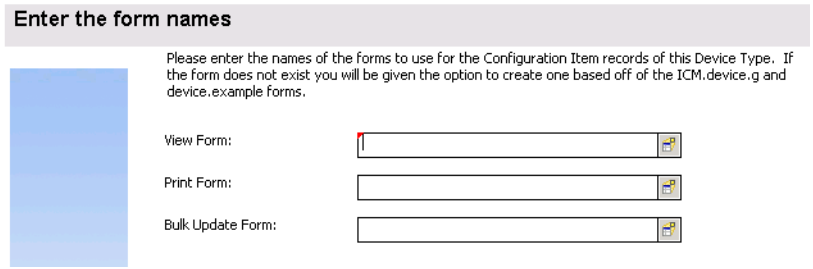
a Go to **Configuration Management** → **Administration** → **Add New Device Type**. The Introduction page of Wizard: Create New Device Type appears.

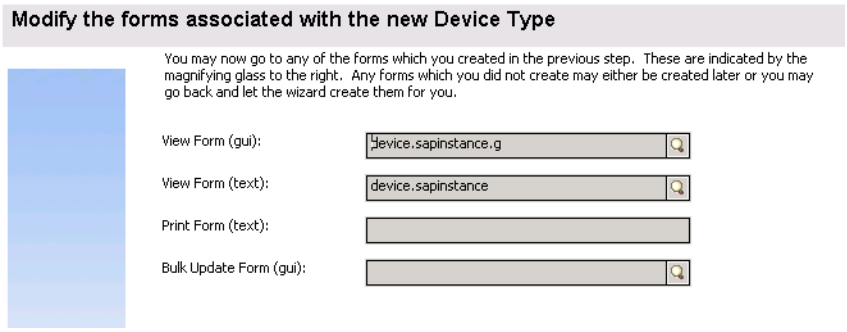


b Click **Next**. Follow the wizard to add a new device type.

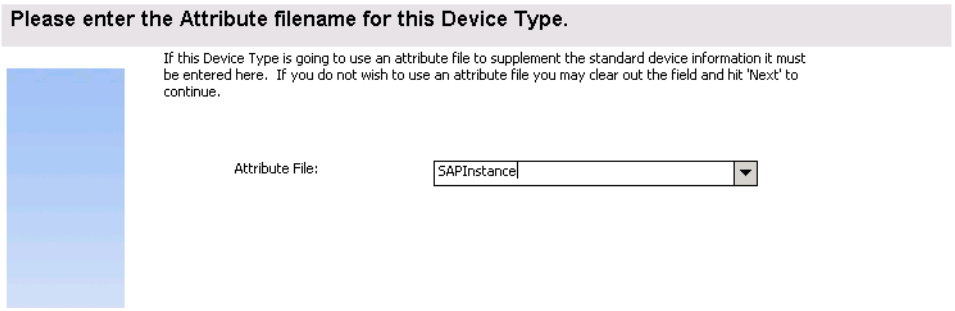


c Click **Next**. Enter **`device.sapinstance.g`** in the `View Form` field, and then click **Next**.

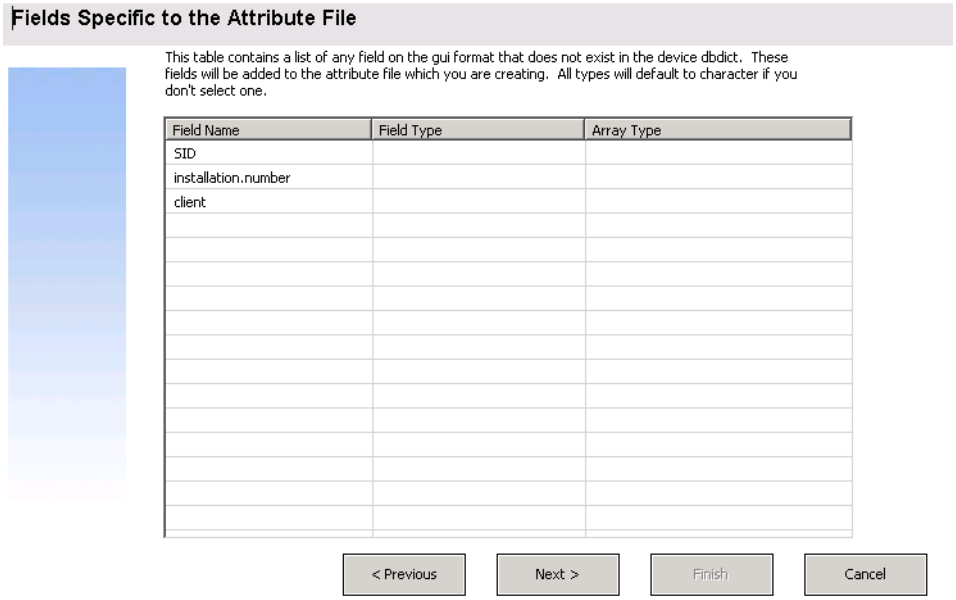




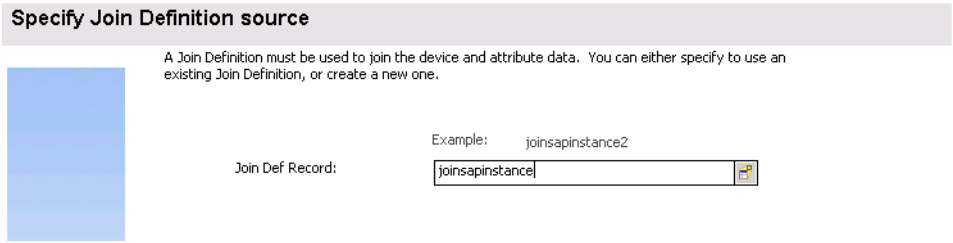
d Click **Next**. Enter **`SAPInstance`** in Attribute File field.



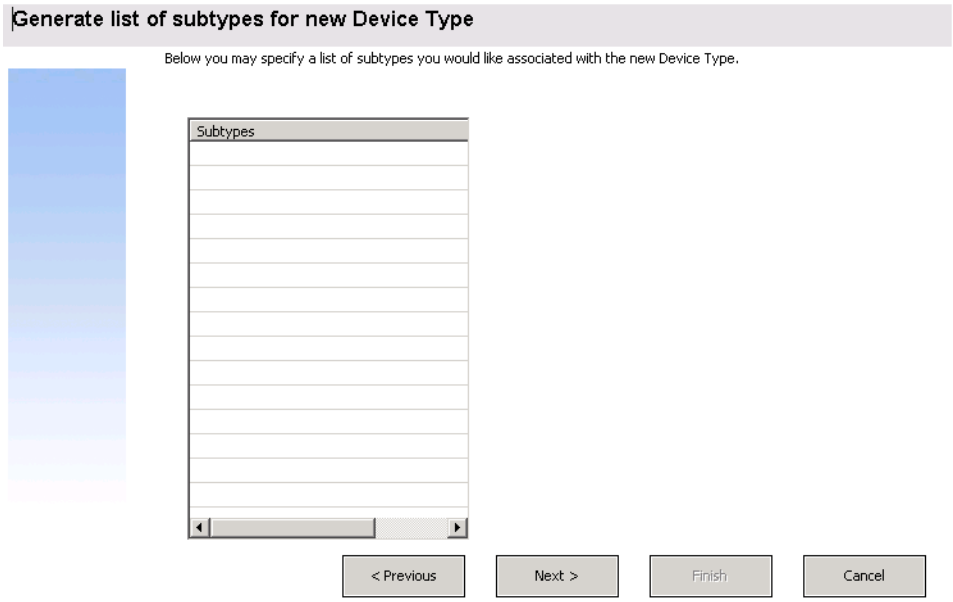
e Click **Next**. The Fields Specific to the Attribute File page appears.



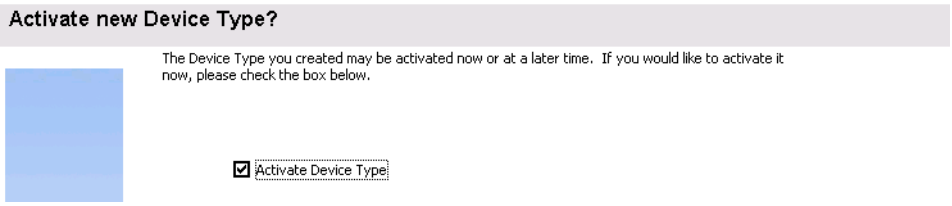
f Click **Next**. The Specify Join Definition source page appears. Enter join name **`joinsapinstance`**.



g Click **Next**. The Generate list of subtypes for new Device Type page appears.

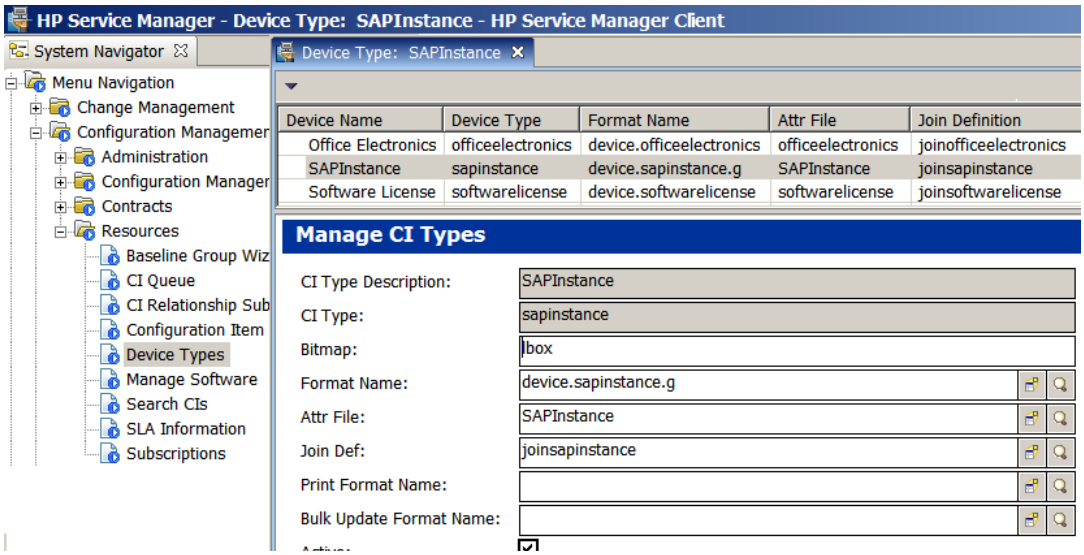


h Click **Next**. Check the checkbox to active the device type.



i Click **Next**. The device type is created.

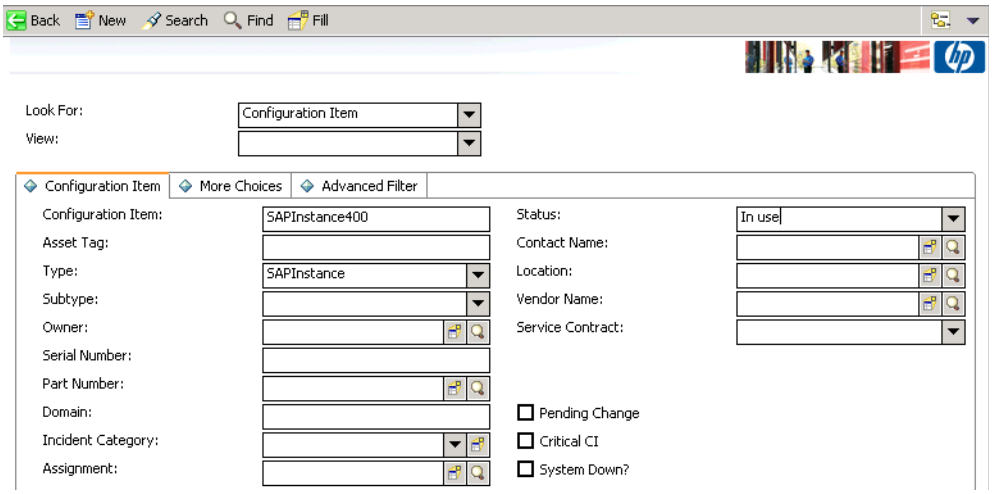
5 The newly generated device type is in **Configuration Management** → **Resources** → **Device Types**.



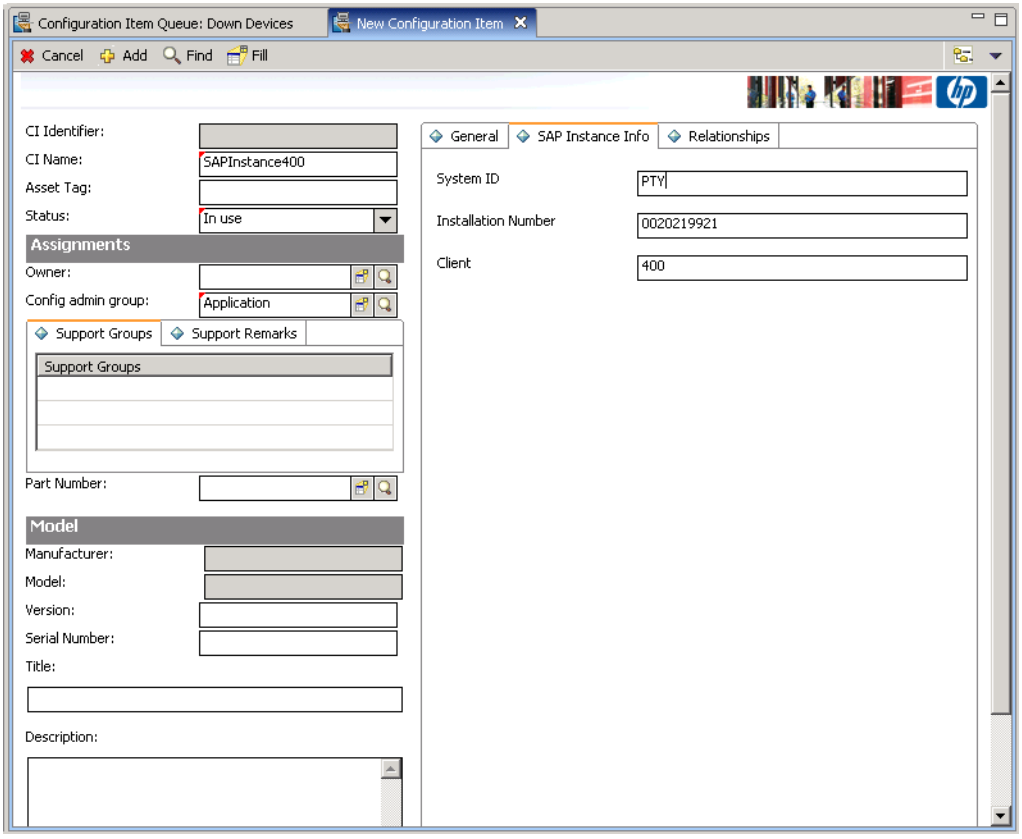
6 Log out and log in to the Windows client again.

7 Add a new SAP device item in Service Manager.

a Go to **Configuration Management** → **Resources** → **CI Queue** → **New**. Select **SAPInstance** in the Type dropdown list, and provide values for fields of your choices, such as what is shown below.

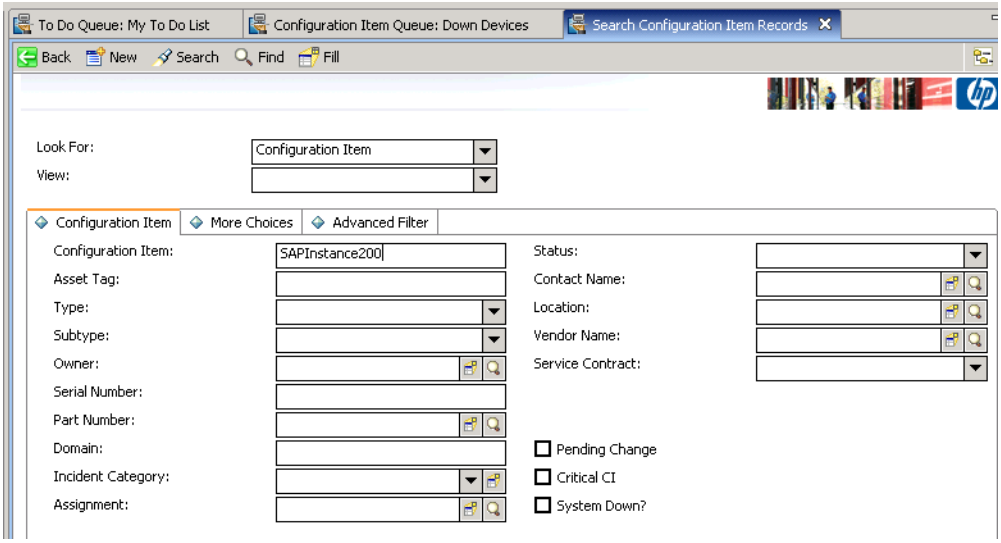


b Click **New**. Provide values for the fields needed, and then click **Add** to add the SAP device item in Service Manager.

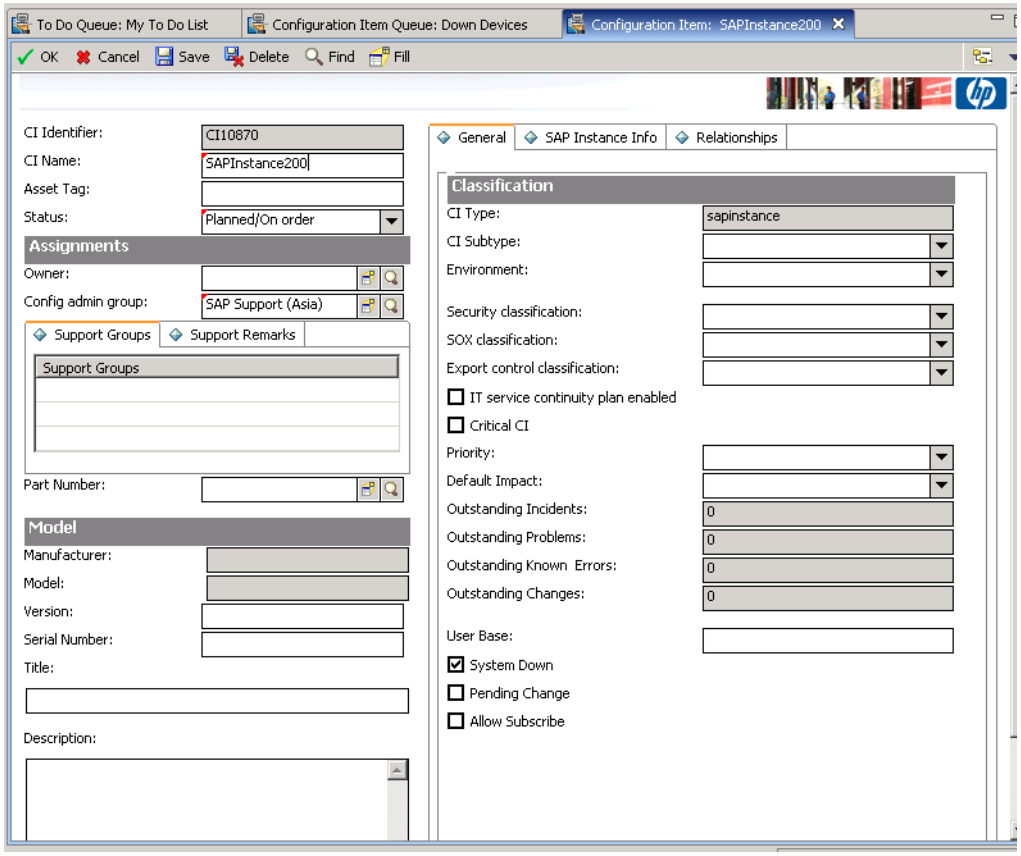


c Add relationship for the newly added SAP device item.

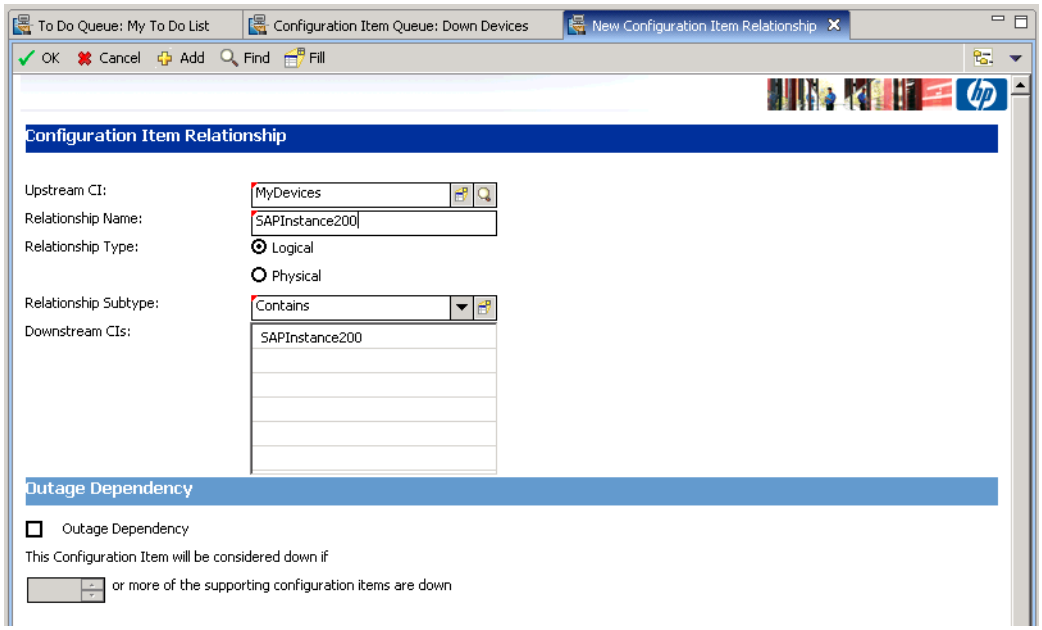
Go to **Configuration Management** → **Resources** → **CI Queue** → **New**. Enter the newly added SAP device item name in Configuration Item field, for example, **`SAPInstance200`**, and then click **Search**.



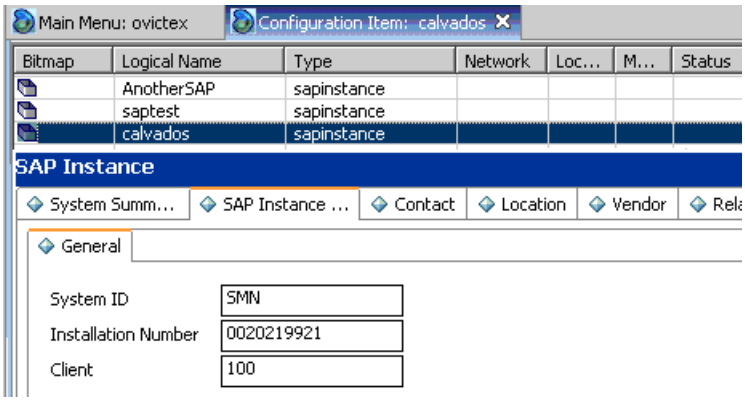
Select **Relationships** tab. The Relationships tab page appears.



Click **Add Upstream Relationship**. The Configuration Item Relationship page appears. Provide values as necessary, and then click **OK**. The relationship is added.

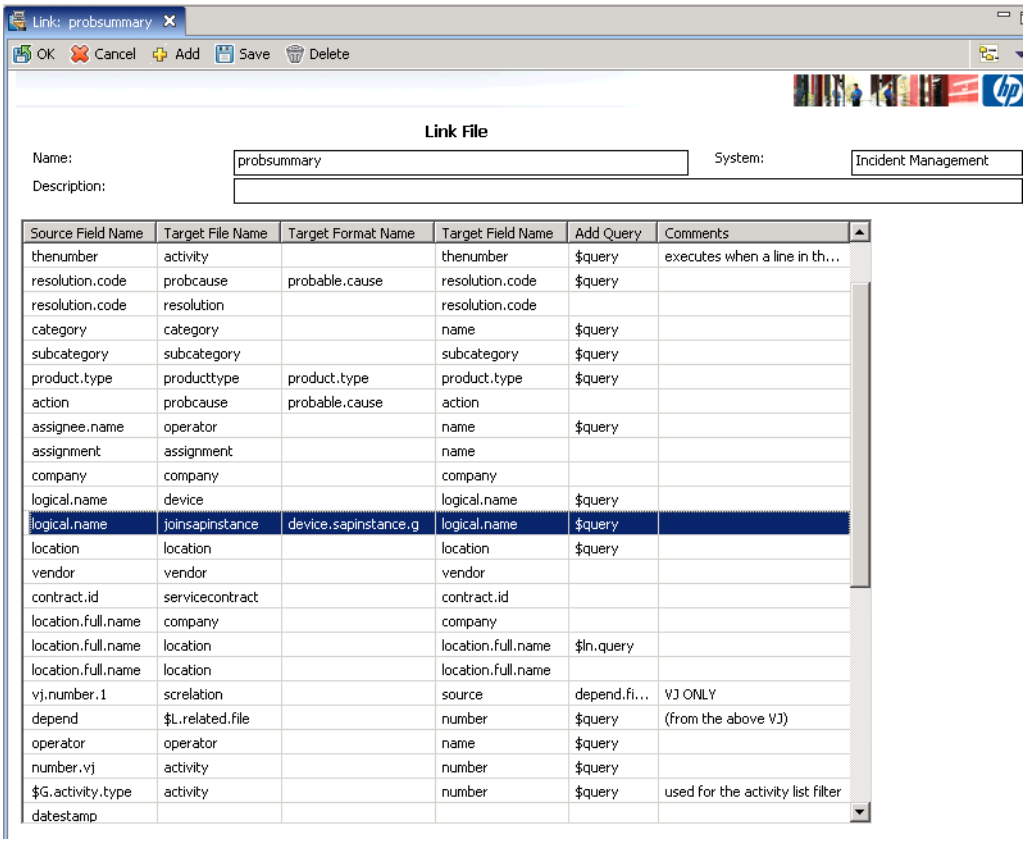


8 A custom form allows entry of the three attributes (and any other SAP-specific CI attributes) at **Configuration Management** → **Resources** → **CI Queue** → **New**. Select **SAPInstance** in the Type dropdown list.



9 The connection between the new device type `SAPInstance` and the fields in the `probsummary` table is made via a link definition. Select **Tailoring** → **Tailoring Tools** → **Links** and search for **`probsummary`**.

10 The existing CI lookup (links between `logical.name` and `device` have to be modified. Insert an additional link line that links `logical.name` with `joinsapinstance`.



11 The three attributes that define the SAP CI and the CI primary key must be linked.

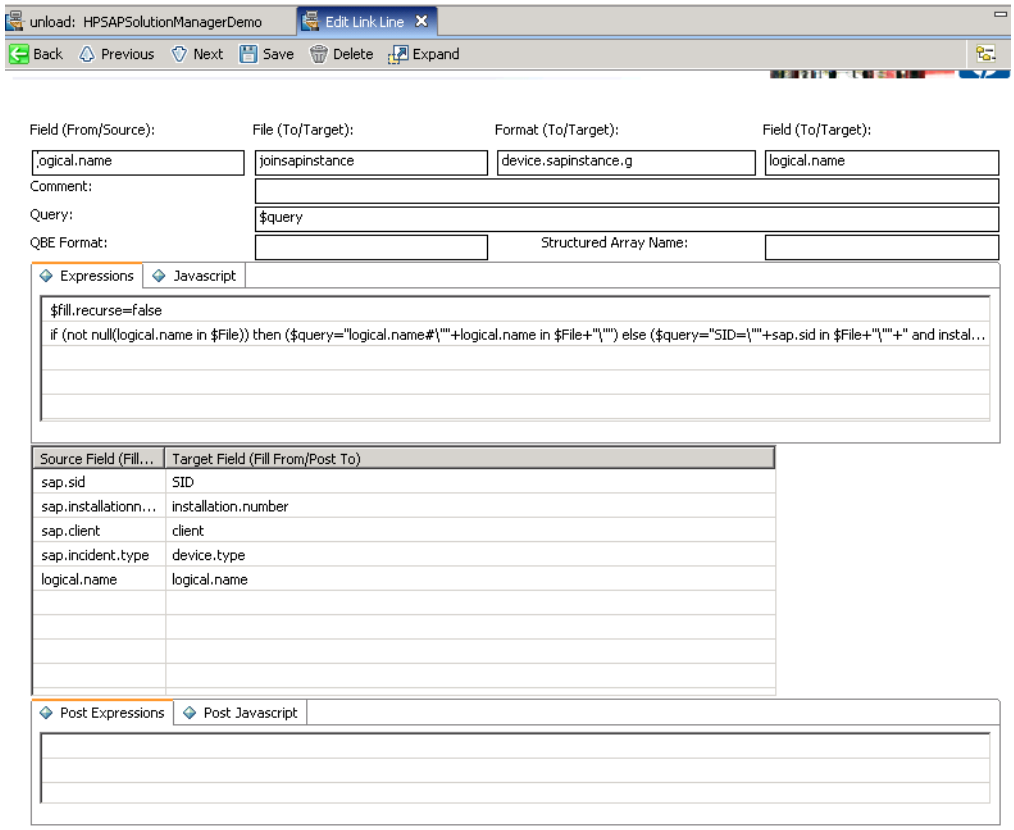
In Expressions tab, add the following two lines (**[LOGICAL.NAME_JOINSAPINSTANCE_EXPR]** in `code_sm9.txt`):

```
$fill.recurse=false
if (not null(logical.name in $File)) then
($query="logical.name#\""+logical.name in $File+"\"") else
($query="SID=\""+sap.sid in $File+"\""+" and
installation.number=\""+sap.installationnumber in $File+"\""+" and
client=\""+sap.client in $File+"\"")
```

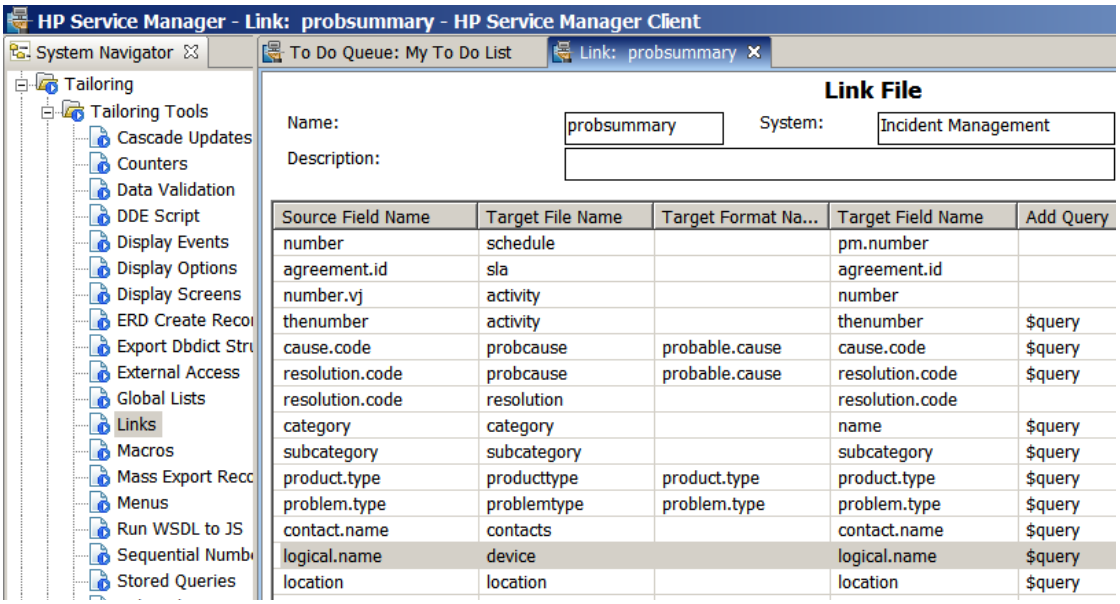
Field Name	Value
Field(From/Source)	logical.name
File(To/Target)	joinsapinstance
Format(To/Target)	device.sapinstance.g
Field(To/Target)	logical.name
Query	$query

Source Field(Fill To/Post From)	Target Field(Fill From/Post To)
sap.sid	SID
sap.installationnumber	installation.number
sap.client	client
sap.incident.type	device.type
logical.name	logical.name

On Service Manager 9.x:



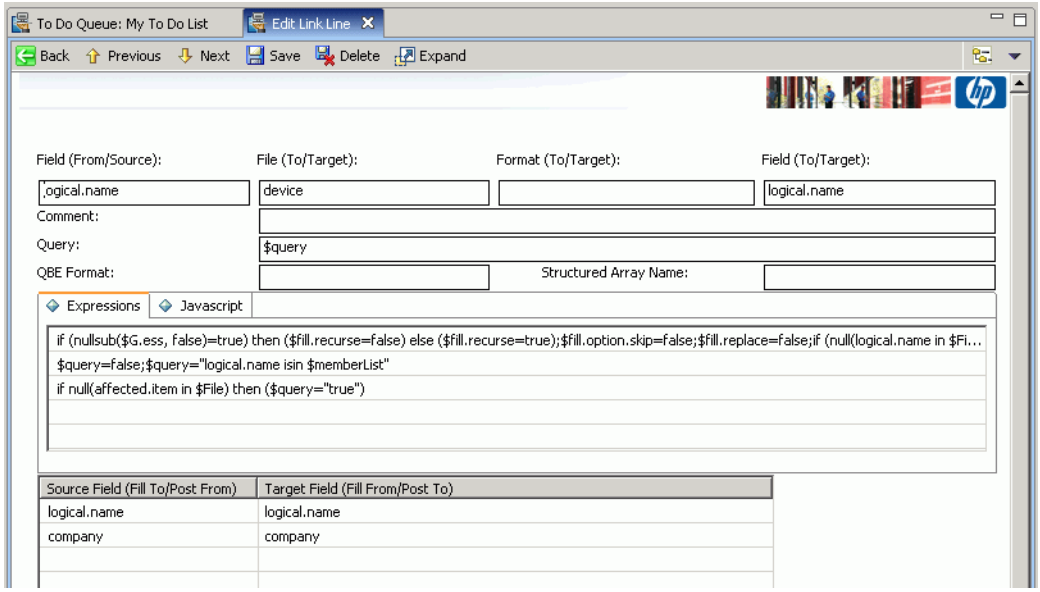
12 Click the line with `device` as Target File Name:



13 Insert the following expression on Service Manager 9.x (**[LOGICAL.NAME_DEVICE_EXPR]** in `code_sm9.txt`):

```
if (nullsub($G.ess, false)=true) then ($fill.recurse=false) else
($fill.recurse=true);$fill.option.skip=false;$fill.replace=false;if
(null(logical.name in $File) and not null(sap.sid in $File)) then
($fill.skip=true)
```

On Service Manager 9.x:

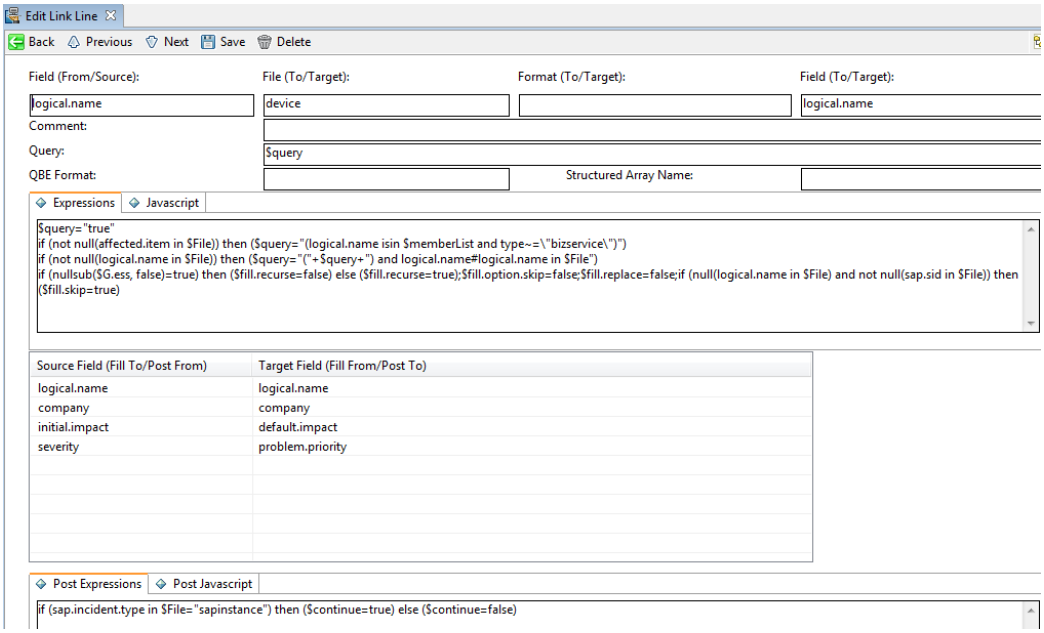


14 In the Post Expressions tab, add the following expression:

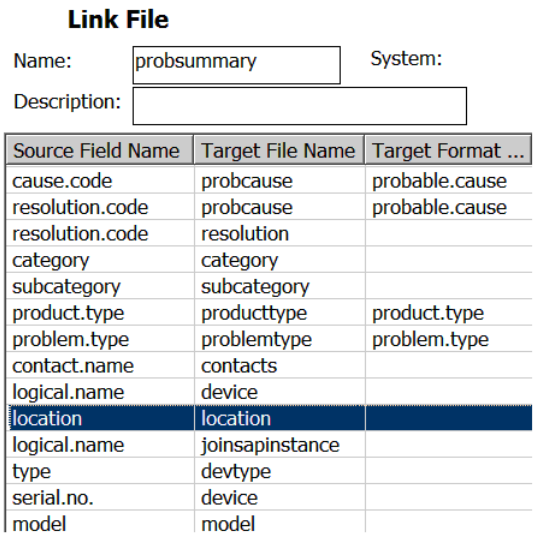
*On Service Manager 9.x (***[LOGICAL.NAME_DEVICE_POST]** in `code_sm9.txt`*):*

```
if (sap.incident.type in $File="sapinstance") then ($continue=true) else
($continue=false)
```

On Service Manager 9.x:



15 Click the line with `location` as Target File Name:

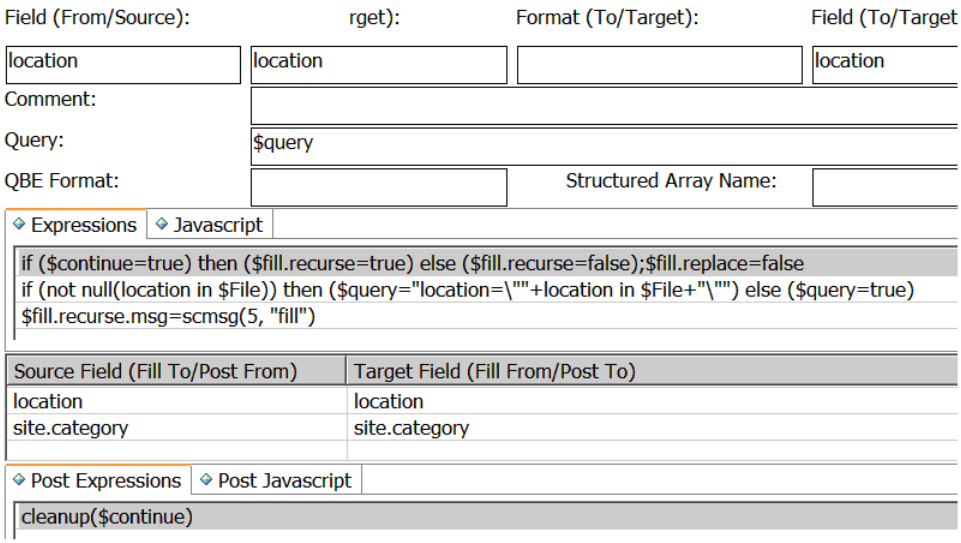


16 In the Expressions tab, replace the first line with the following expression (**[LOCATION_LOCATION_EXPR]** in `code_sm9.txt`):

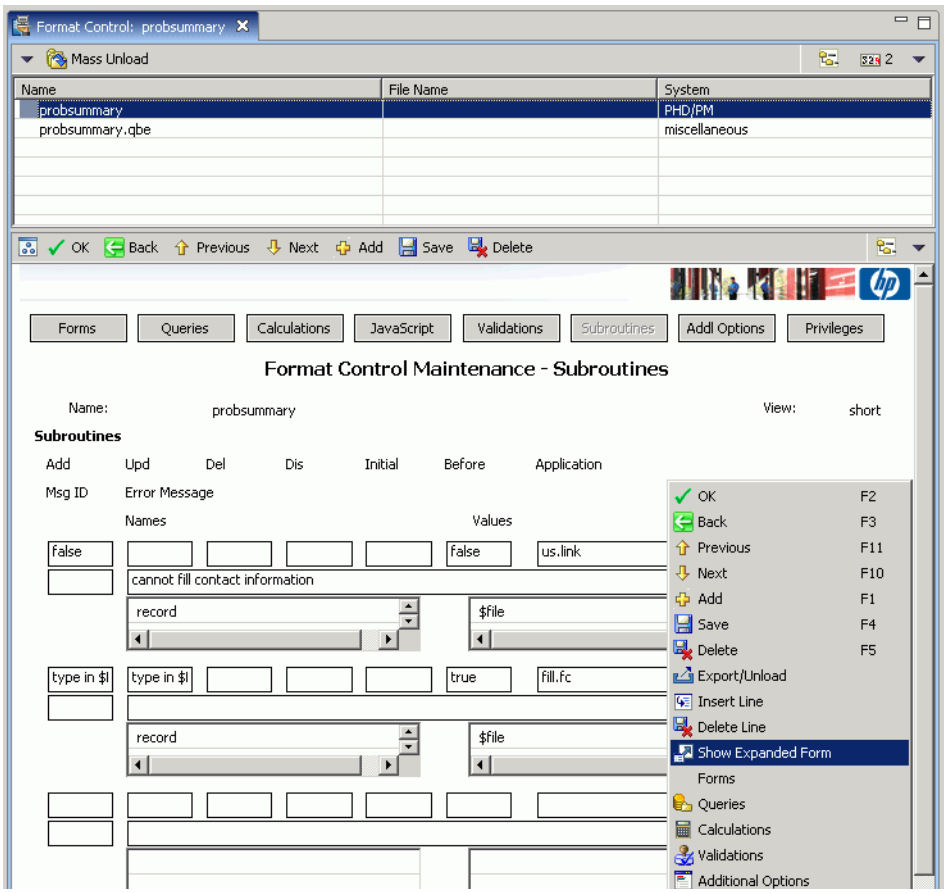
```
if ($continue=true) then ($fill.recurse=true) else
($fill.recurse=false);$fill.replace=false
```

17 In the Post Expressions tab add the following expression:

```
cleanup($continue)
```

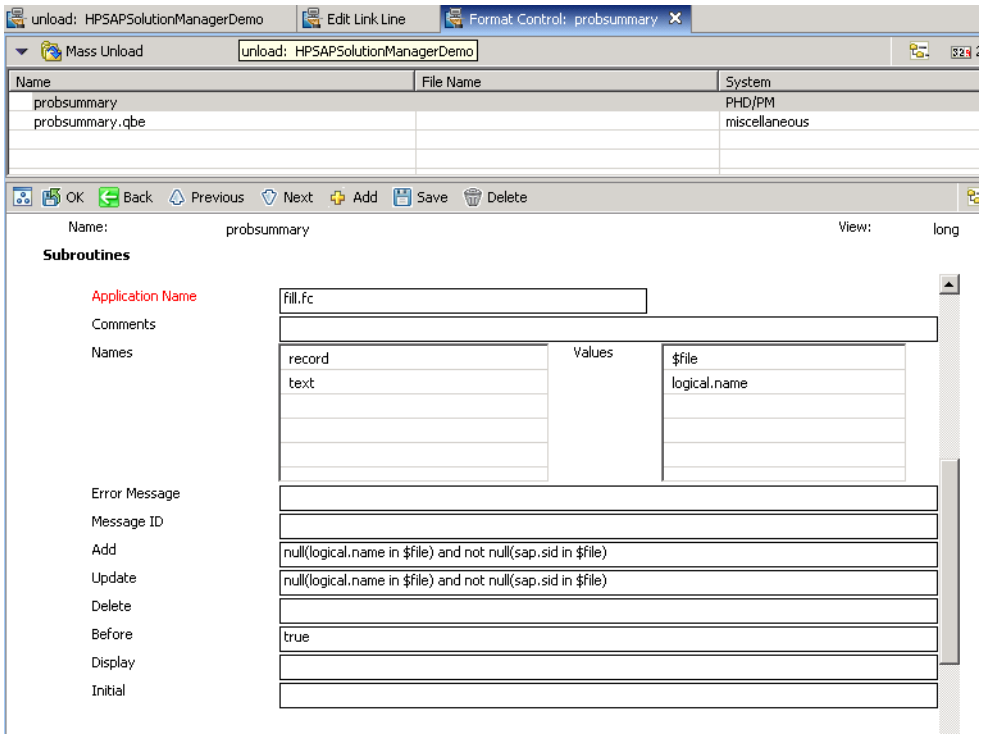


18 To automatically write the CI attributes into `probsummary` when a CI of type SAPInstance is attached to an Incident, the `fill.fc` application is invoked, select **Tailoring** → **Format Control** and search for **probsummary** → **Subroutines**, then right click and select **Show Expanded Form** from the pop-up menu.


Format Control: probsummary
Mass Unload
Name
File Name
System
probsummary
PHD/PM
probsummary.qbe
miscellaneous
OK
Back
Previous
Next
Add
Save
Delete
Forms
Queries
Calculations
JavaScript
Validations
Subroutines
Addl Options
Privileges
Format Control Maintenance - Subroutines
Name:
probsummary
View:
short
Subroutines
Add
Upd
Del
Dis
Initial
Before
Application
Msg ID
Error Message
Names
Values
false
false
us.link
cannot fill contact information
record
$file
type in $l
type in $l
true
fill.fc
record
$file
OK F2
Back F3
Previous F11
Next F10
Add F1
Save F4
Delete F5
Export/Unload
Insert Line
Delete Line
Show Expanded Form
Forms
Queries
Calculations
Validations
Additional Options

19 Add a `fill.fc` application, and fill the fields as shown in the screenshot below:

Field	Value
Application Name	fill.fc
Name	record
Value	$file
Name	text
Value	logical.name
Add ([FILL.FC] in `code_sm9.txt`)	null(logical.name in $file) and not null(sap.sid in $file)
Update ([FILL.FC] in `code_sm9.txt`)	null(logical.name in $file) and not null(sap.sid in $file)
Before	true



When an operator assigns a CI of type `sapinstance` to an Incident, the three attributes `SAP SID`, `client` and `installation number` are read from the SAP CI and put into the corresponding `Incident` fields (which are then exchanged with SAP Solution Manager). Similarly, when an Incident is submitted from SAP Solution Manager, the Incident created in Service Manager contains values in the fields `sap.sid`, `sap.client` and `sap.installationnumber`. These values are used to search for a corresponding CI of type sapinstance. If the CI exists, it is automatically attached to the Incident.

Implementation Alternatives and Enhancements

The above implementation assumes a simple CI setup. SAP CI's may be modeled in more complex ways.

For example, the three attributes `SAP SID`, `client` and `installation number` can be distributed over multiple CI's. A "parent" CI represents the entire SAP system, containing the attributes `SAP SID` and `installation number`, combined with "child" CI's that represent individual clients and contain the SAP `client` attribute. This model allows identification of problems affecting the entire SAP system or just a particular client.

The customization within Service Manager needs to be adapted to a particular SAP CI configuration. Incident Exchange directly interacts only with the `Incident` fields in the `probsummary` table. The synchronization with CI's inside Service Manager is the responsibility of the Service Manager customization.

Configuring WSDL Mapping

Configure the `IncidentManagement` WSDL Mapping table in WSDL Configuration of Service Manager as follow:

Field name	**Caption**
action	IncidentDescription
assignee.name	AssigneeName
brief.description	BriefDescription
initial.impact	InitialImpact
assignment	PrimaryAssignmentGroup
product.type	ProductType
resolution	Resolution
subcategory	SubCategory
severity	Urgency

Adding Instance in SMIS and Configuring Parameters

The SMSAP instance in SMIS integration is used to enable and disable the SMSAP integration. By configuring the SMSAP instance, you can slso specify the integration parameters such as SMSSMEX accessing URL and SAP clients. After the SMSAP instance is enabled, the customization to incident is visible and you can select a SAP Solution Manager client for incident information exchange. If the SMSAP instance is disabled or removed, the incident form customization for incident exchange with SAP Solution Manager will be invisible to the end users. Note that the incident information exchange from SAP SolMan to HP Service Manager will not be affected by SMSAP SMIS configuration. For more information about SMIS, refer to *Service Manager Web Help* > *Integrations* > *Integration Manager*.

The SM-SAP integration operates only when the SMSAP instance in SMIS is enabled. Do the following to add a new instance in SMIS and configure the parameters:

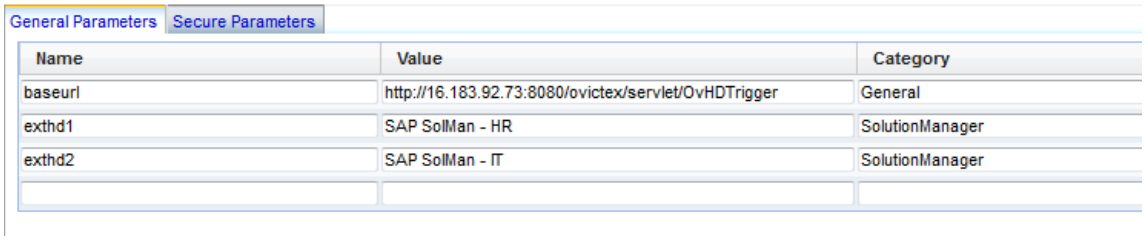
1. Log in to Service Manager with a System Administrator account.
2. Select **Tailoring** → **Integration Manager**. Integration Manager opens.
3. Click **Add** to open the wizard.
4. Select **SMSAP**, click **Next**.
5. Click **Next** to configure the parameters.
   a. Configure the baseurl for connection to SMSSMEX.

      Replace `<host>` and `<port>` according to the middleware installation.

   b. Configure SAP Solution Manager clients.

      Add general parameters for each of your SAP Solution Manager clients configured in SMSSMEX `ovictex.properties`. For example, `exthd.instances.id.1`, `exthd.instances.id.2` and so on.

   > Make sure to set your SAP Solution Manager client instance ID in the Name field, and **`SolutionManager`** in the Category field of each record. Value of each SAP Solution Manager client instance ID is the reference in HP Service Manager.



6. Click **Next** twice to the end of the wizard and click **Finish** to save the configurations.
7. Select the SMSAP instance and click **Enable** to enable the SM-SAP integration.

6 Configuring SAP Solution Manager

This chapter describes how to configure the SAP Solution Manager.

Prerequisites

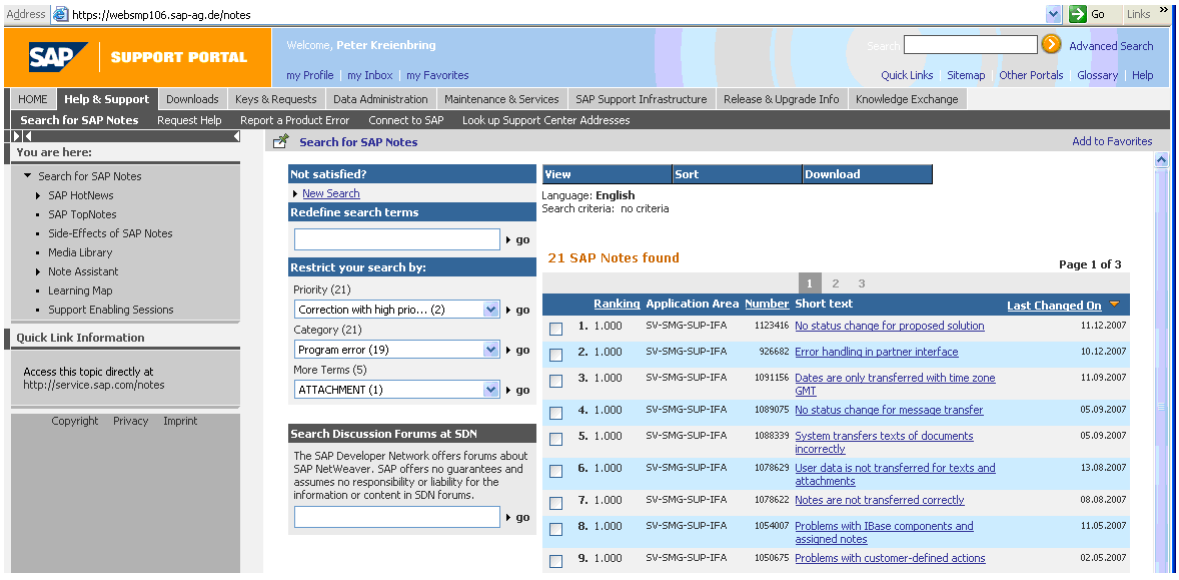
The prerequisites are:

- SAP Solution Manager 7.0 SP 12 (or higher) or SAP Solution Manager 7.1
- SAP Solution Manager SP12 if copying of business transaction SLFN for customization in a customer name space (for example ZLFN) is required
- SAP Solution Manager SP12. Required to copy a business transaction into a customer name space for customization (for example, to copy business transaction `SLFN` into customer name space `ZLFN`)
- Configured SAP Solution Manager Service Desk

Configured SAP Solution Manager Service Desk SSL encryption between SAP Solution Manager and Apache Tomcat requires:

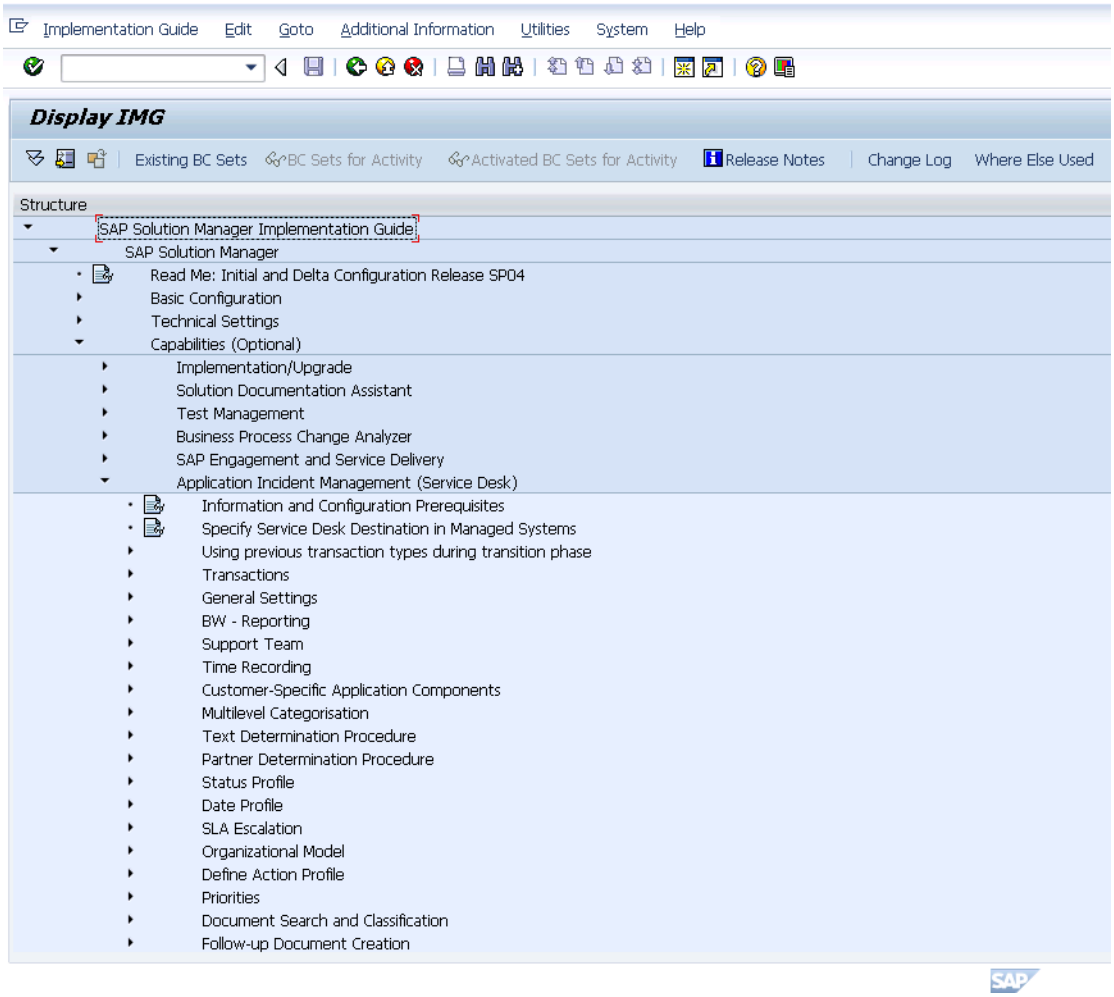
- Sapcryptolib 5.5.5C or higher
- SSL Server and SSL Client PSE
- SSL Server and SSL Client certificates trusted against a CA

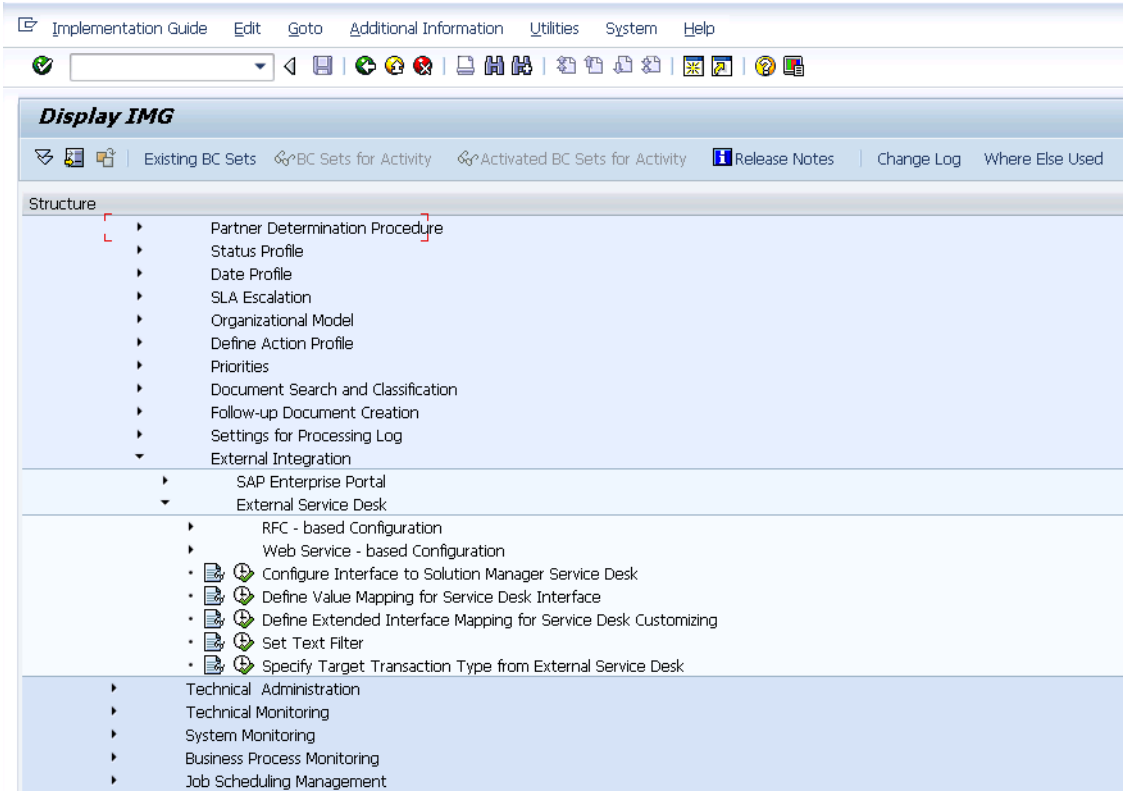
Integration with HP Service Manager requires implementation of the latest SAP notes (SAP application area SV-SMG-SUP-IFA) for the SP level stack of SAP Solution Manager. The following diagram shows the search results of SAP notes in the SAP Support Portal.



Configuring SAP Solution Manager External Service Desk Interface

SAP provides the Implementation Guide “External Service Desk” for configuring the external help interface. The Implementation Guide is located in SAP transaction `/nspro` under path `\SAP Solution Manager ImplementationGuide\SAP Solution Manager\Capabilities (Optional)\Application Incident Management (Service Desk)\External Integration\ External Service Desk`. The following diagrams show the Implementation Guide for configuring the connection to the external Service Desk.



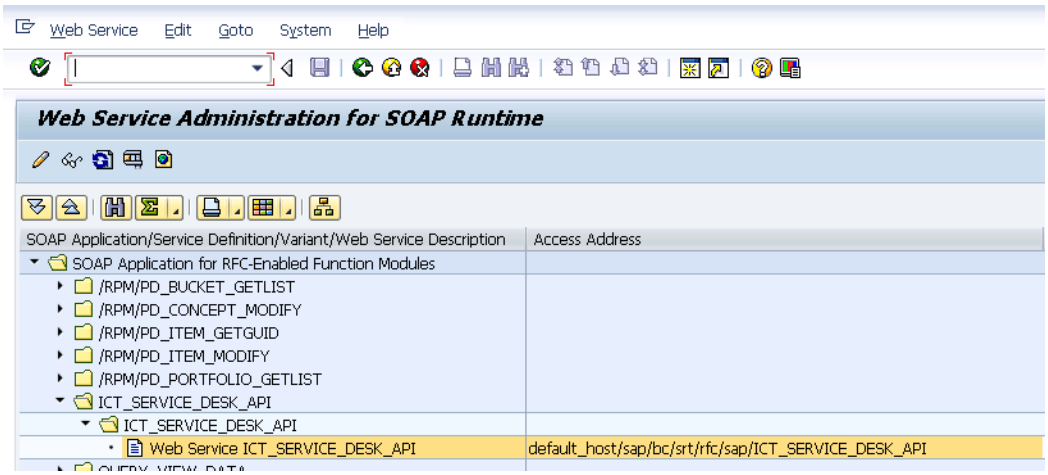


Click the leftmost text sign to view configuration steps. Click the clock sign to enter the corresponding transaction and edit the configuration.

Release Web Service

The Incident Exchange Web Service is deactivated by default. It is required to release the WEB-Service in the Internet Communication Manager Service tree.

1 The transaction to release a Web Service is **`/nwsconfig`**. After the Web Service is released the WS is in SAP transaction **`/nwsadmin`**.



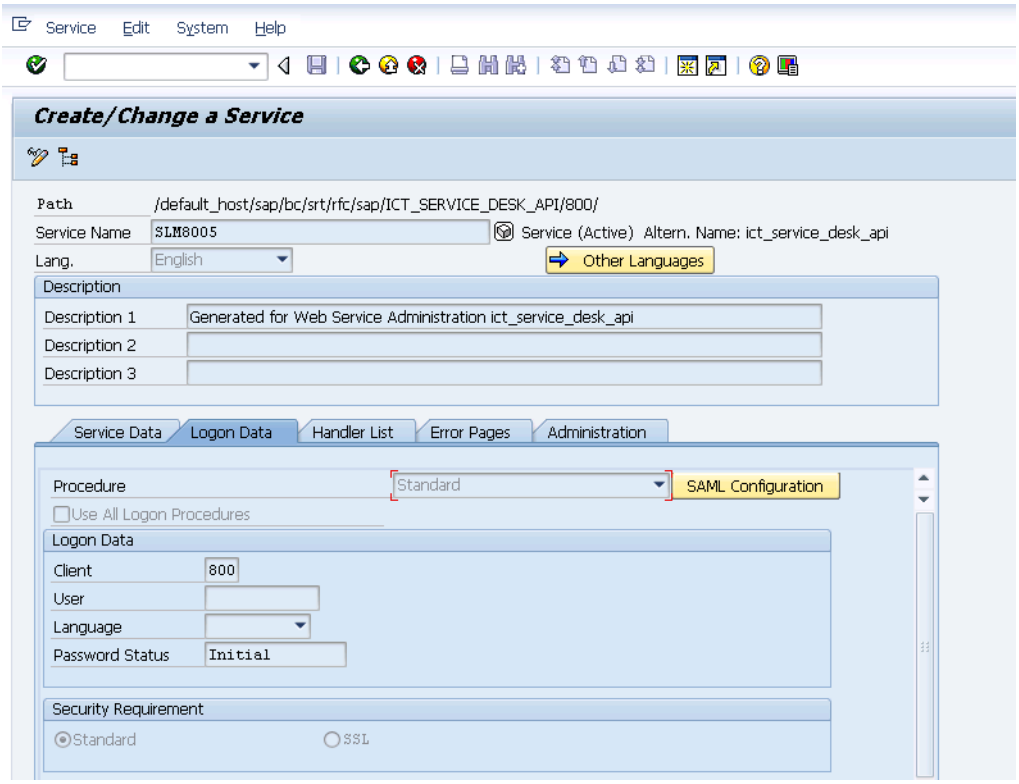
2 To determine the logon procedure of the Web Service for incoming requests, go to SAP transaction **/nsicf**.

3 Enter **ICT_SERVICE_DESK_API** as the service name.

4 Click **Execute** to execute the search.

5 Double-click the Service to edit or navigate to the path /default_host/sap/bc/srt/rfc/sap/ and select **ICT_SERVICE_DESK_API**.

6 In the Logon tab of Create/Change a Service dialog , select **Standard**.



The security section of this manual contains additional information for setting up SSL communications. Adding a user is not required. The Incident Exchange Web Service will use the user and password that is configured in the properties file for HTTP Basic authentication. This user must exist as an SAP user. It is not recommended to use a dialog user for this purpose.

Assign Roles to the Communication User

Configure an SAP user with permission to manage incidents in SAP Solution Manager Service Desk. Follow the instruction in the Implementation Guide and add the roles SAP_SUPPDESK_PROCESS and SAP_SUPPDESK_INTERFACE to the user. Exchanging a business partner with a default configuration interface requires the additional role SAP_CRM_BUSINESS_PARTNER.

To configure a user:

1 Select transaction **/nsu01**.

2 Input the name of the user.

3 Click **Display** . The user configuration transaction appears.

- A person who is assigned to an incident in HP Service Manager but does not exist in Solution Manager will be created as a Business Partner when the incident is forwarded to Solution Manager. Without the business partner role `SAP_CRM_BUSINESS_PARTNER` the incident can not be created or updated in Solution Manager and the error code 99 appears.
- A communication user is recommended, but not necessary.

Sending support messages to SAP AGS requires assigning an SAP Support Portal contact to Solution Manager users who will communicate with the SAP Support Portal via RFC connections. The contact maintained corresponds to the S-user in the SAP Support Portal without "S". See SAP Note 834534 and the SAP Solution Manager configuration guide for details of Solution Manager roles and authorizations.

Create HTTP Connection

Define the endpoint of the SMSSMEX Web-Service for communication between SAP Solution Manager and Apache Tomcat.

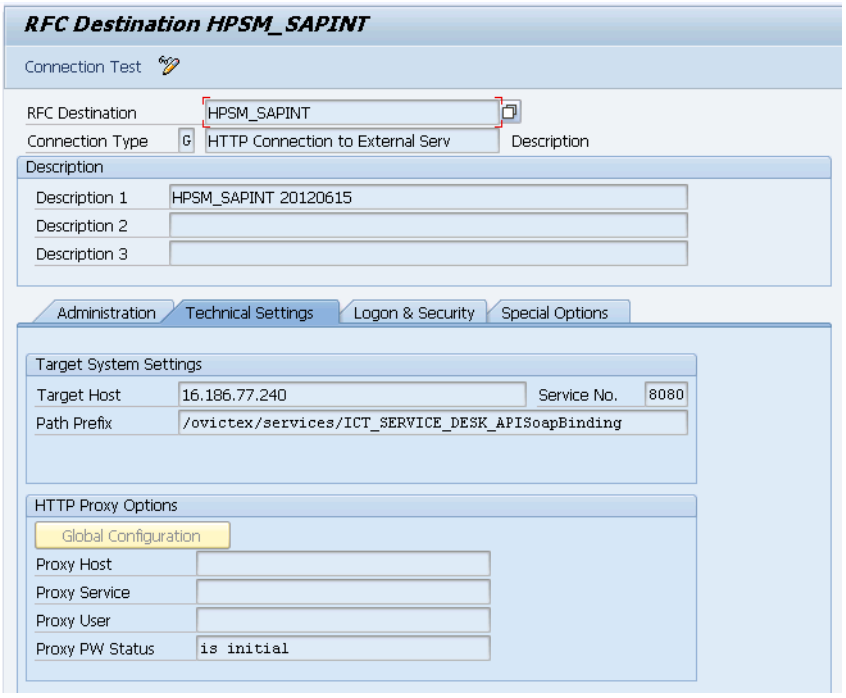
1 Select transaction **`/nsm59`**.

2 Create an RFC destination of type **`G`** (HTTP connection to external server).

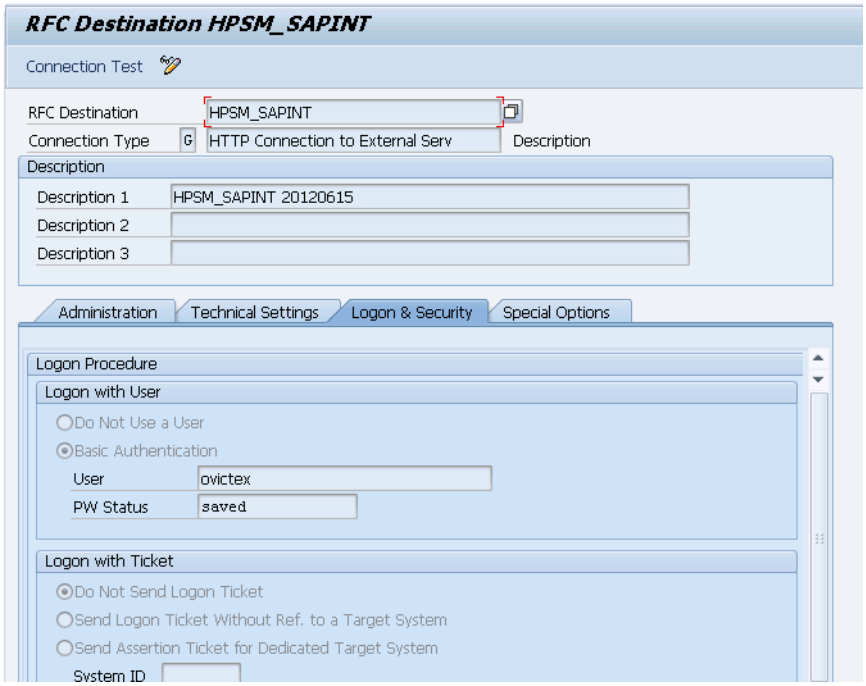
3 Go to the tab **Technical settings** and specify the endpoint of the SMSSMEX Web-Service. The default is:

```
Target Host: <host>
Service No: <port>
Path Prefix: /ovictex/services/ICT_SERVICE_DESK_APISoapBinding
```

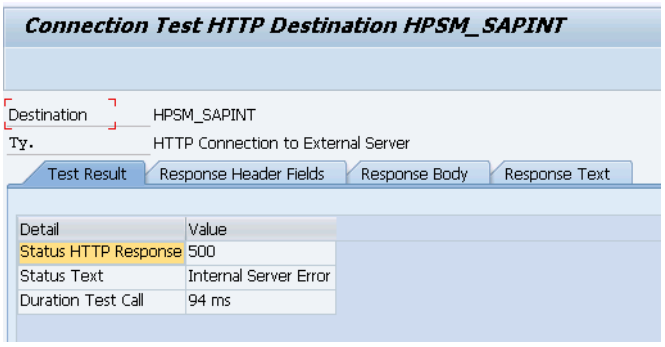
4 Add the endpoint in the RFC destination. Your network configuration may require specification of a proxy. The following example shows the RFC destination for host **`itsamqavm130`**.



5 In the Logon & Security tab define the security settings for outgoing requests. Select **Basic Authentication** for HTTP basic authentication. Add the user and password specified in **`ovictex.properties`** for HTTP basic authentication. The more secure SSL communication configuration is described in the security chapter of the manual. You can also select **No Logon** which is the default selection for "Logon&Security".



The following diagram shows the SMSSMEX Web service returning error 500. This result indicates the connection between SAP and SMSSMEX is established.



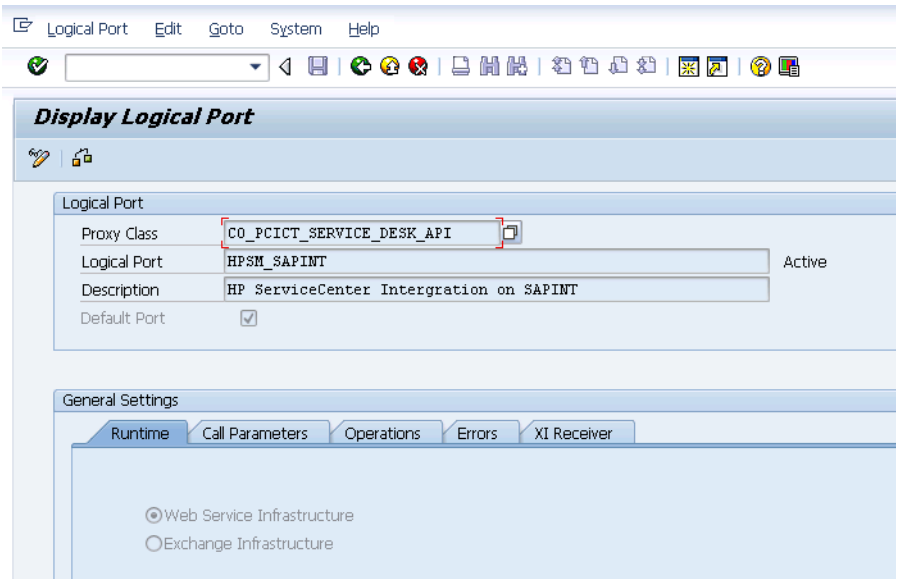
Create a Logical Port

The logical port is the container that encapsulates the outgoing requests. Define the logical port as specified in the Implementation Guide instructions.

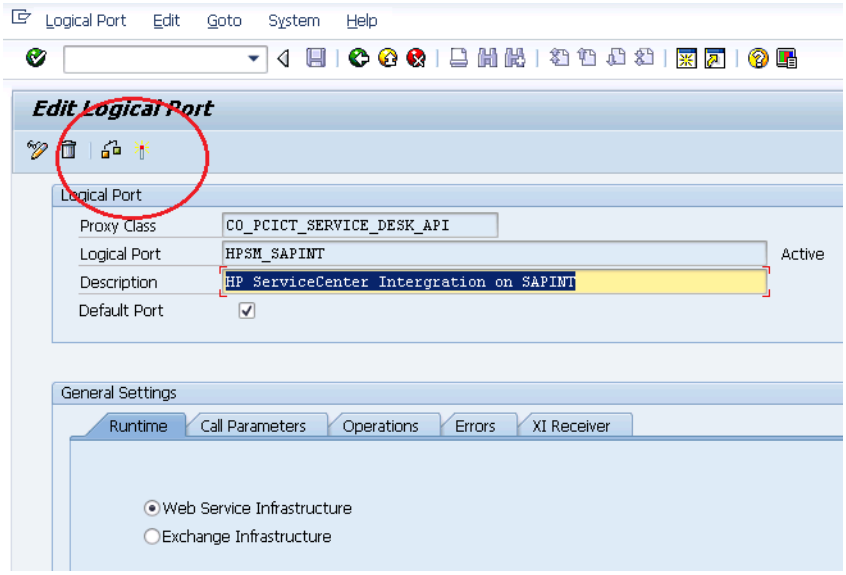
1 Go to transaction **`/nlpconfig`**.

2 Select **`CO_PCICT_SERVICE_DESK_API`** as the Proxy Class name.

3 In Call Parameters tab add the HTTP destination configured in the previous chapter.

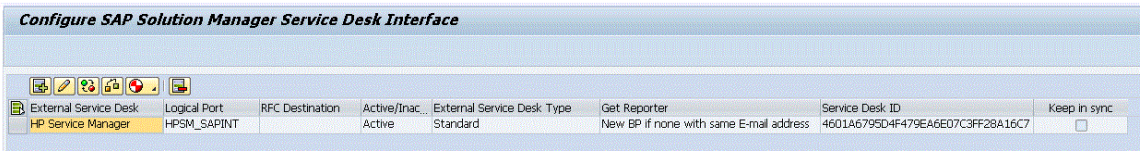


The port must be activated. Click **Active** to activate the logical port.



Configure Interface to SAP Solution Manager Service Desk

This activity configures the interface between the SAP Solution Manager service desk and the HP Service Manager. Follow the instructions in the Implementation Guide. The configuration requires that Apache Tomcat and the web service are configured and running. In this implementation step the SMSSMEX web service must deliver a unique Service Desk ID. If the Service Desk ID is changed, then the configuration must be repeated.



Use the **Check** button to verify the configuration. Any error message will be displayed in the output window. Use transaction **`/nictconf`** to jump to configuration transactions.

- Do not select the **Keep in sync** checkbox when configuring the interface to SAP Solution Manager service desk.
- If the check fails, try **Generate Default Mapping** → **Overwrite Old Values** and then run the check again. After configuration, click **Save** to save the configured interface.

Define Value Mapping for the Service Desk Interface

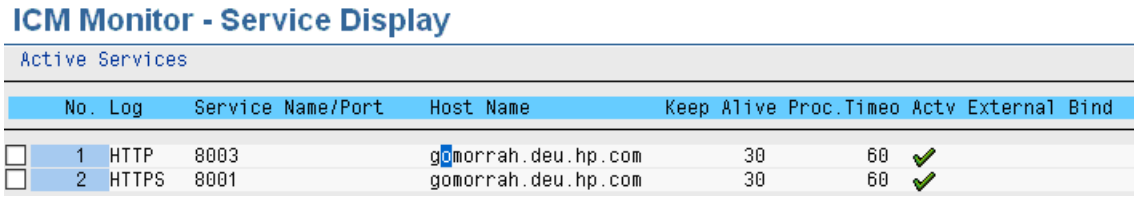
This IMG activity configures the value mapping between SAP Solution Manager Service Desk and SMSSMEX for ingoing and outgoing requests. Changing the default value mapping of the SAP Solution Manager is not required. If changes are necessary, use the field mapping file of the SMSSMEX configuration file. To change the default Mapping of the SAP Solution Manager, consult the instructions in the implementation guide.

Define Extended Interface Mapping for Service Desk

If SAP Solution Manager Service Desk is highly customized (not using standard SAP objects) then it might be necessary to change the interface mapping. The IMG activity instructions provide more information.

Get SAP Solution Manager Service Port

Go to SAP transaction **`/nsmicm`**. Select **Goto** → **Services**.



This transaction shows the host and the port required for access to the SAP Solution Manager Service Desk web service. Specify in `ovictex.properties` the host/port as the endpoint entry.

Solution Manager Tracing

SolutionManager is able to trace incoming and outgoing web-service XML messages. The messages can be downloaded and used for failure analysis.

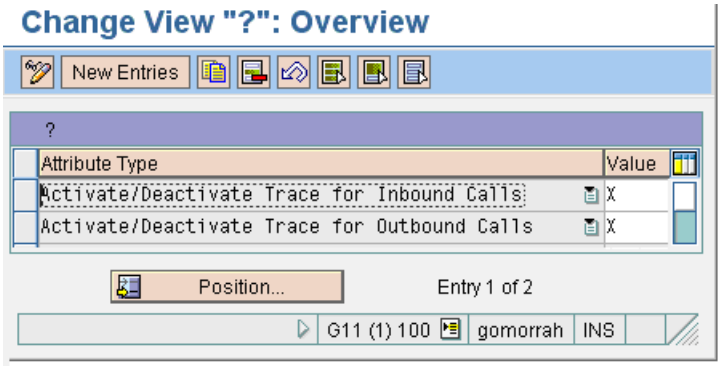
Enable tracing

To enable tracing, implement a SolutionManager Implementation Guide in transaction **`/nspro`**.

```
spro -> SAP Reference IMG ->
  SAP SolutionManager Implementation Guide ->
    SAP SolutionManager ->
      Configuration ->
        Scenario-Specific Settings ->
          Service Desk ->
            Connecting an External Service Desk ->
              Define Extended Interface Mapping for Service Desk Customizing
```

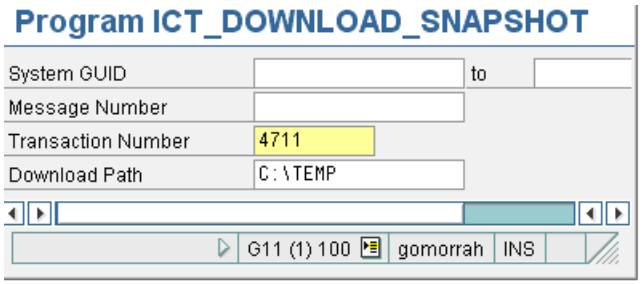
Add new entries to enable tracing for incoming and outgoing calls:

- `Activate/Deactivate Trace for Inbound Calls` = **`X`**
- `Activate/Deactivate Trace for Outbound Calls` = **`X`**



Download Trace File

To download the trace file, run **`ict_download_snapshot`** in transaction **`/nse38`**. Enter the SolutionManager incident id in the field `Transaction Number` and run the program (**`F8`**). The trace file will be downloaded to the local computer (for example, incident 4711 traces will be downloaded to C:/TEMP).



7 Configuring Security

This chapter describe the required security configuration settings.

Security between SAP Solution Manager and Tomcat

This section describes the security configuration between SAP Solution Manager and Tomcat.

Configure SAP Solution Manager for SSL

This section describes how to configure SAP Solution Manager for SSL.

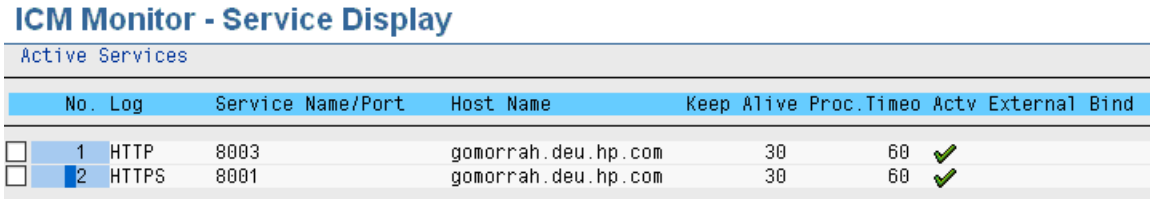
Checking SAP SSL Configuration

SAP WEB AS does not support or allow self-signed certificates for communication between Solution Manager and the SMSSMEX Web Service. All certificates must be trusted against a CA.

Before configuring SSL for the External Help Desk interface, check if the WEB AS that hosts the SAP Solution Manager is configured for using SSL.

ICM (Internet Communication Manager) HTTPS service is required for SSL communication. Check if SSL communication is possible in SAP transaction **`/nsmicm`** (select menu entry **GOTO** and select **Services** or press **`SHIFT+F1`**).

If SSL communication is possible then an active HTTPS service that is listening to a port is visible. In the example below, the HTTPS port is 8001. This port must be configured in the SMSSMEX web service properties file.

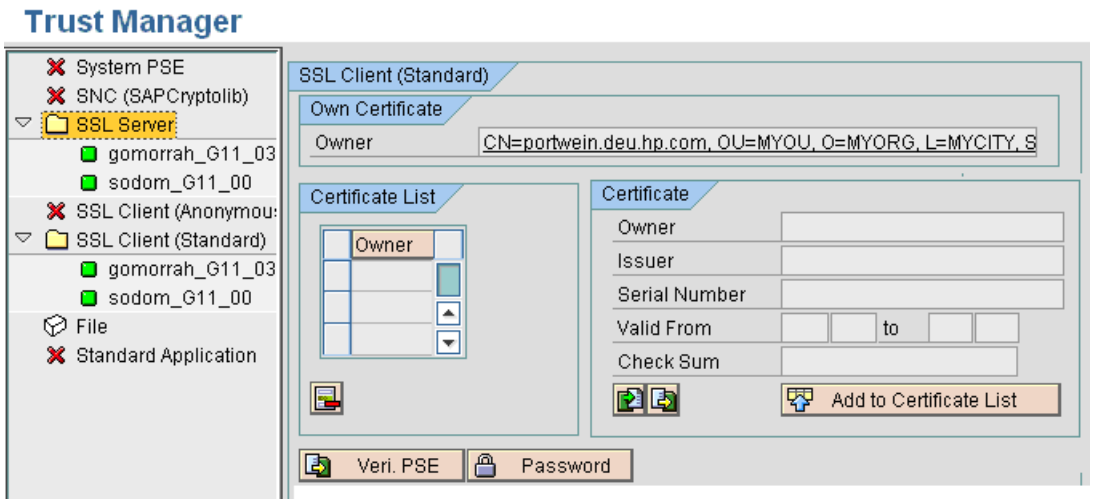


If an HTTPS service in the ICM monitor is not visible, then check the SSL Server configuration in Trust Manager. Start the Trust Manager with SAP transaction **`/nstrust`**.

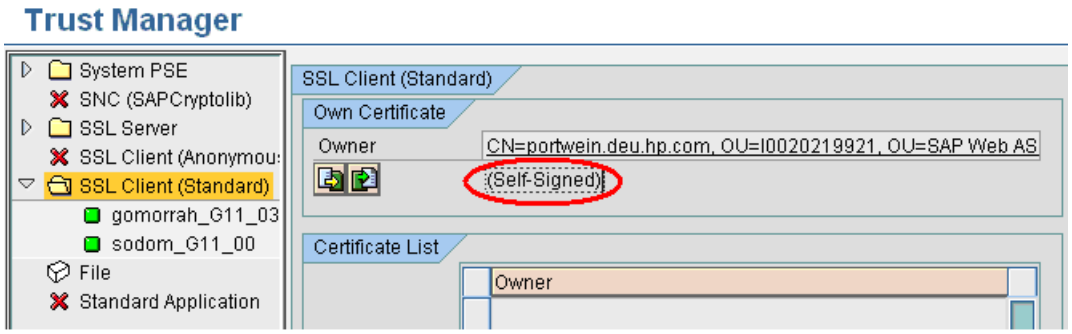
If the the PSE entries SSL Server and SSL Client (Standard) are not shown in the Trust Manager status section, then install and configure the SAP `sapcryptolib` library.

➤ Installing and configuring sapcryptolib requires a restart of the SAP WEB AS instance. The installation instructions are in the SAP online help. For more information, see Appendix B, Installing and Configuring SAPCRYPLIB.

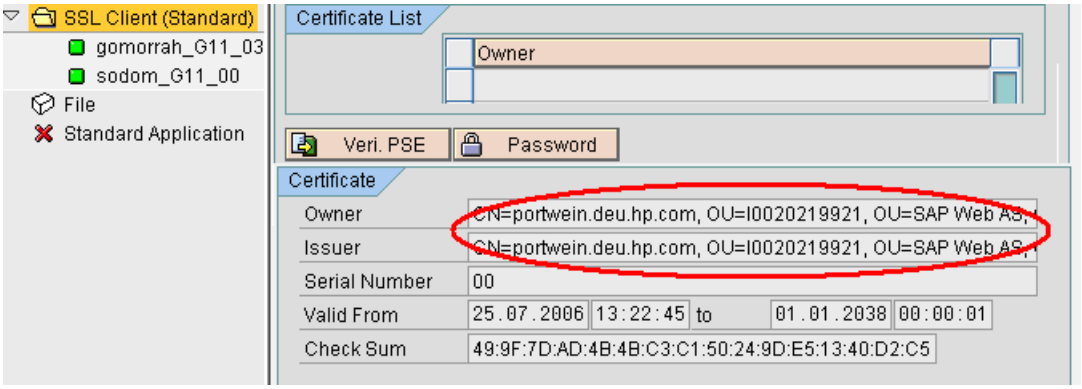
The following diagram shows the Trust Manager with the created PSE “SSL Server” and “SSL Client (Standard)”. The red X in front of the other PSE's indicates that the PSE’s have not been created. The PSE “SSL Server” and “SSL Client (Standard)” must be created.



In the next diagram the certificate of the PSE “SSL Client (Standard)” is “Self Signed”. Self-signed certificates are not supported for communication with Apache Tomcat (the certificate must be signed against a CA). If the certificate is signed the 'Self signed' certificate text will disappear.



Check the certificate by double-clicking the `Owner` attribute. The certificate details are shown in the Certificate section. If the Owner and Issuer have the same DN the certificate is self-signed.



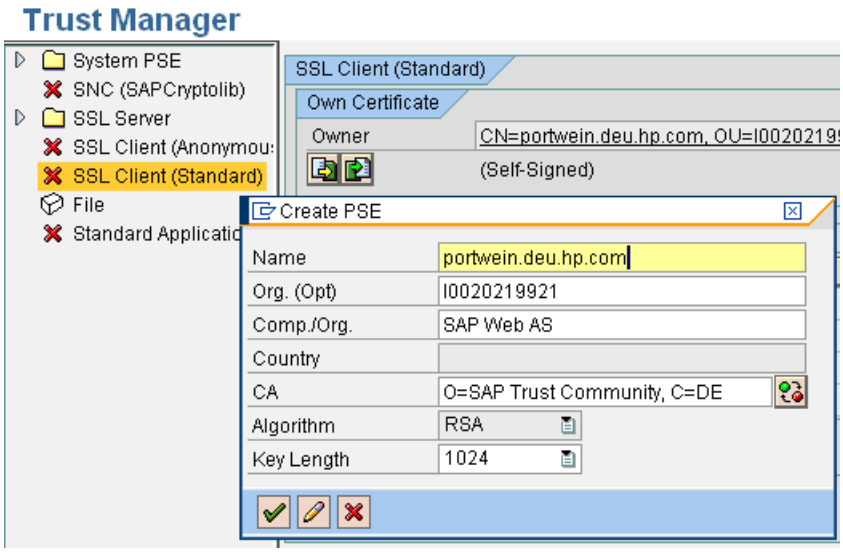
Creating a Client PSE in Trust Manager

To create a client PSE in Trust Manager, do the following:

1 Start the Trust Manager.

2 Select the **PSE SSL Client (Standard)** in the status section of the Trust Manager.

3 Click **Create**.

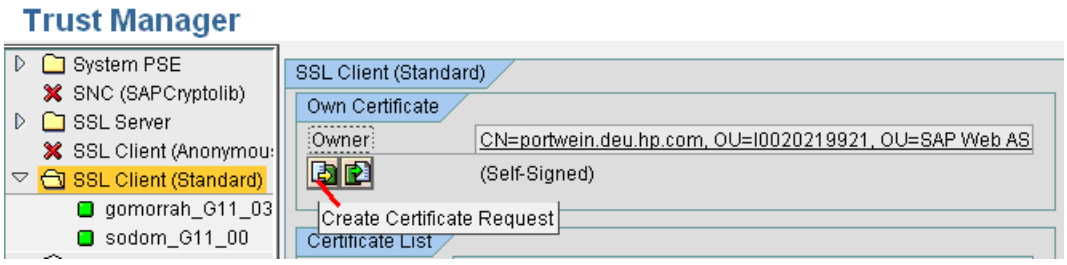


4 For the CN (Name) enter the fully qualified hostname of the SAP WEB AS system. All other entries must not be changed. The key length should be 1024.

5 Save the settings.

6 Double click **SSL Client (Standard)** in the status section. The `Own` certificate in the Own Certificate section is shown.

7 Click **Create Certificate Request**.

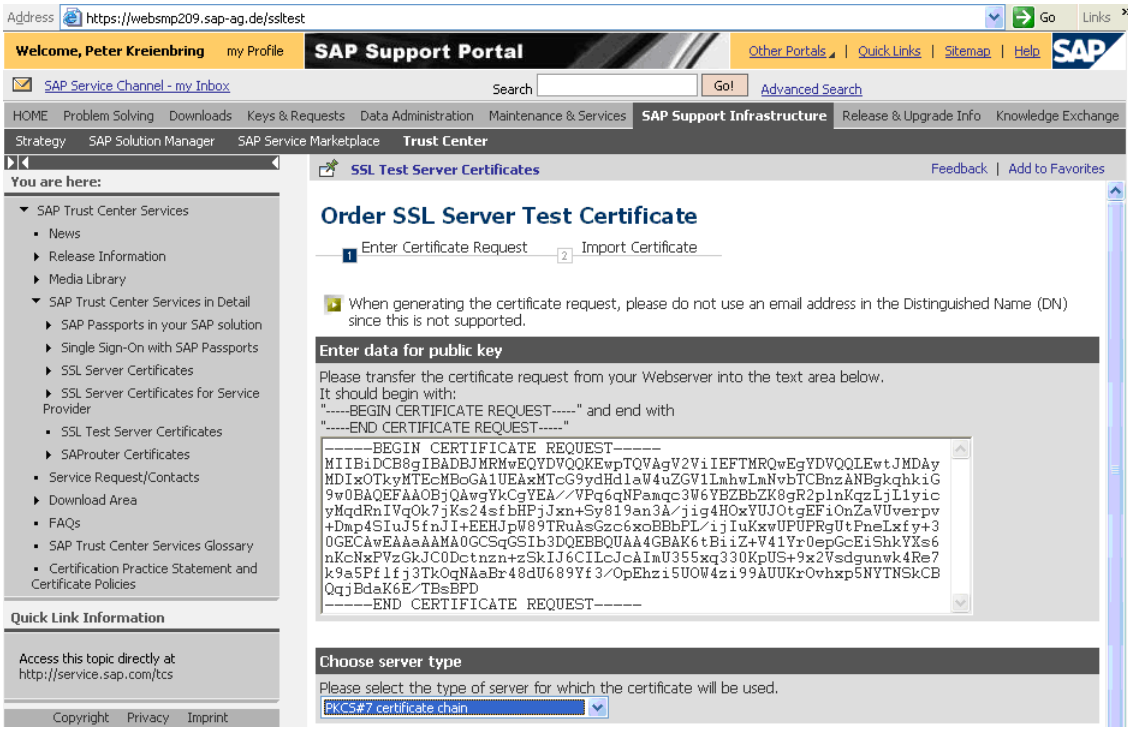


8 The Certification Request is shown. Copy the request to the Clipboard.

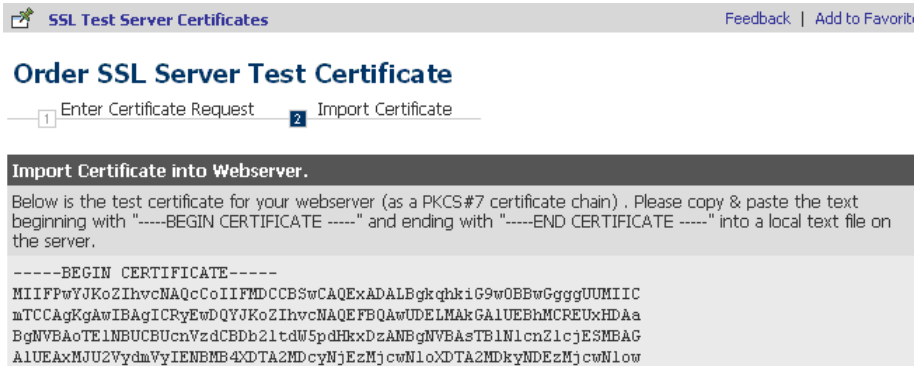
9 Certify the request with a CA.

SAP offers a two-month test period for signed certificates in the SAP Service Marketplace at **http://www.service.sap.com/ssltest**.

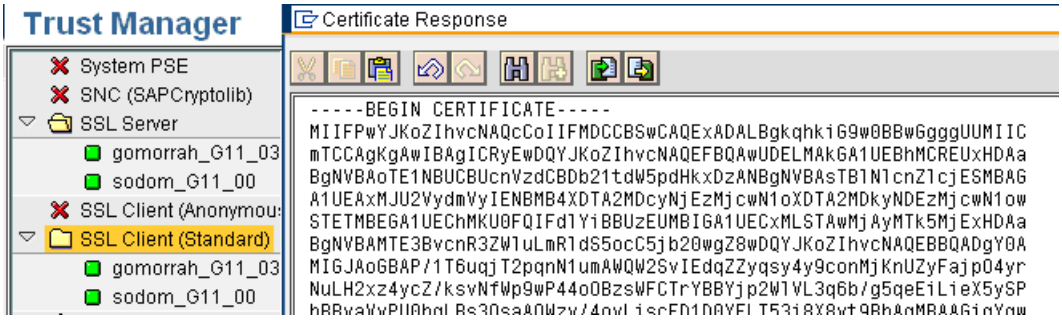
10 Request an SSL Server Test Certificate as shown in the following diagram (select the **PKCS#7** chain format).



11 Click **Continue**. The SSL Server Certificate is created.



12 Copy the Certificate response to your client PSE.



The certificate is now trusted against a CA. The required steps are different for other CA's. Contact the Trust Center for details. A certificate for the SSL Server PSE is also required.

Setting Up an Outgoing Connection in SAP Solution Manager

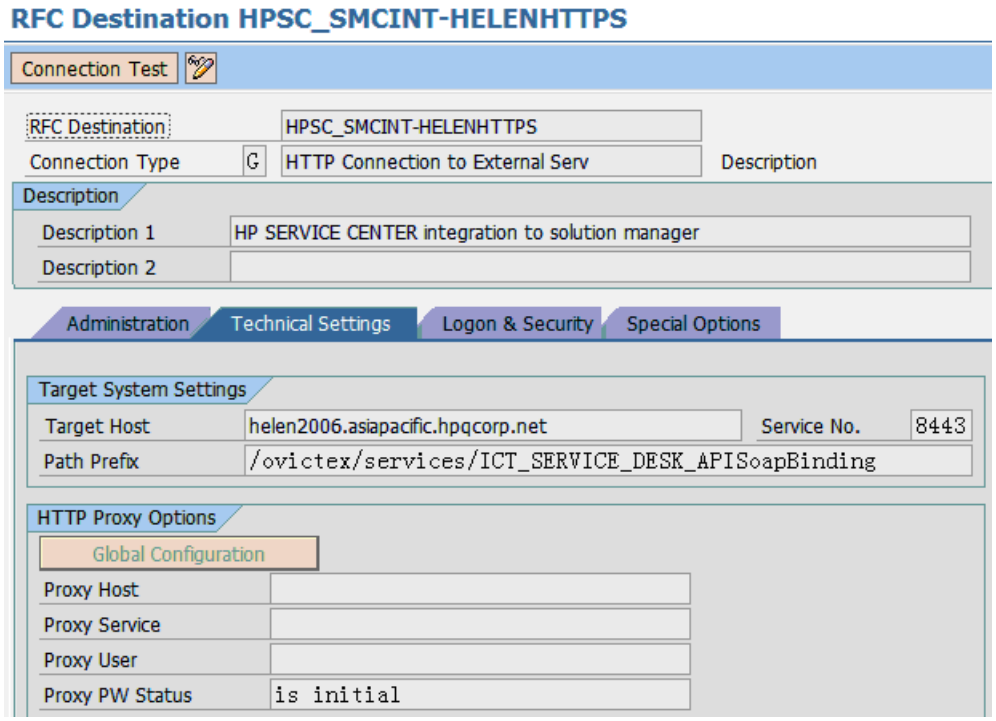
The outgoing connection from SAP Solution Manager to HP Apache Tomcat must be configured in SAP transaction **`/nsm59`**. Add a new or change an existing HTTP RFC destination with type G.

1 In SAP transaction **`/nsm59`** configure the HTTPS port of the Tomcat Server. A redirect from the HTTP port to the HTTPS port of Apache Tomcat will not work with the SAP WEB AS. The HTTPS port is defined in the `server.xml` configuration file of the Tomcat Server.

2 The SSL configuration of Apache Tomcat is switched off by default. Enable the configuration.

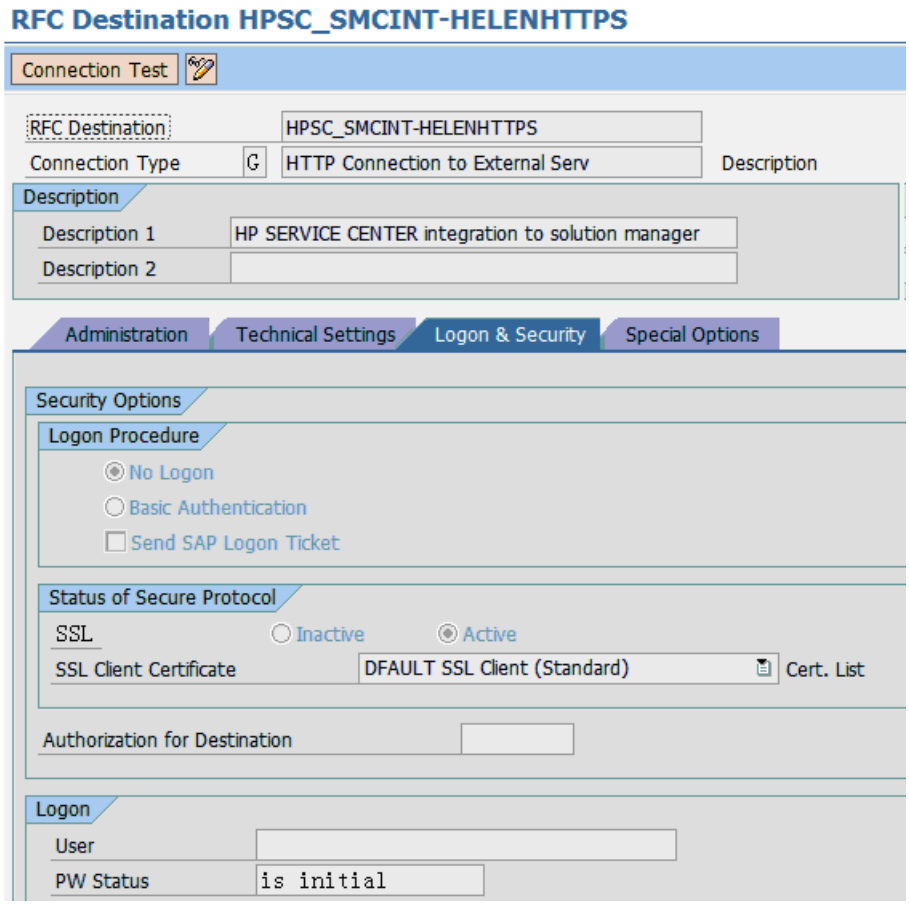
3 In the settings for the SSL HTTP connector, set the Tomcat default port for SSL communication to **`8443`**.

4 The diagram below shows the example configuration of the RFC Destination (in the `Target Host` field enter the server name (case sensitive) instead of the IP address).



5 In the Logon&Security tab of the RFC configuration define the logon procedure and the security protocol. Basic authorization with SSL communication and certificates is not supported by Apache Tomcat. Set the Logon Procedure to **`No Logon`**.

6 In the security protocol status enable SSL and select a PSE from the certification list. SAP provides PSE “ANONYM SSL Client” and “DFAULT SSL Client (Standard)”.



7 Check with the SAP Basis Administrator what client PSE should be used. In most cases this will be the PSE “SAP Client (Standard)”.

8 After assigning a client Certificate to the RFC destination, save the settings. The RFC destination is configured for using SSL with Apache Tomcat. A connection test will fail if the Server certificate in Apache Tomcat is not trusted against a CA.

9 Create a logical port (see Create a Logical Port on page 66).

10 Configure the interface between the SAP Solution Manager Service Desk and the HP Service Manager for the SSL outgoing connection (see Configure Interface to SAP Solution Manager Service Desk on page 68).

Set up an Incoming Connection in SAP Solution Manager

Configure the incoming connection in the ICF Service tree in SAP transaction `/nsicf`.

1 In SAP transaction **`/nsicf`** enter **`ICT_SERVICE_DESK_API`** as service name.

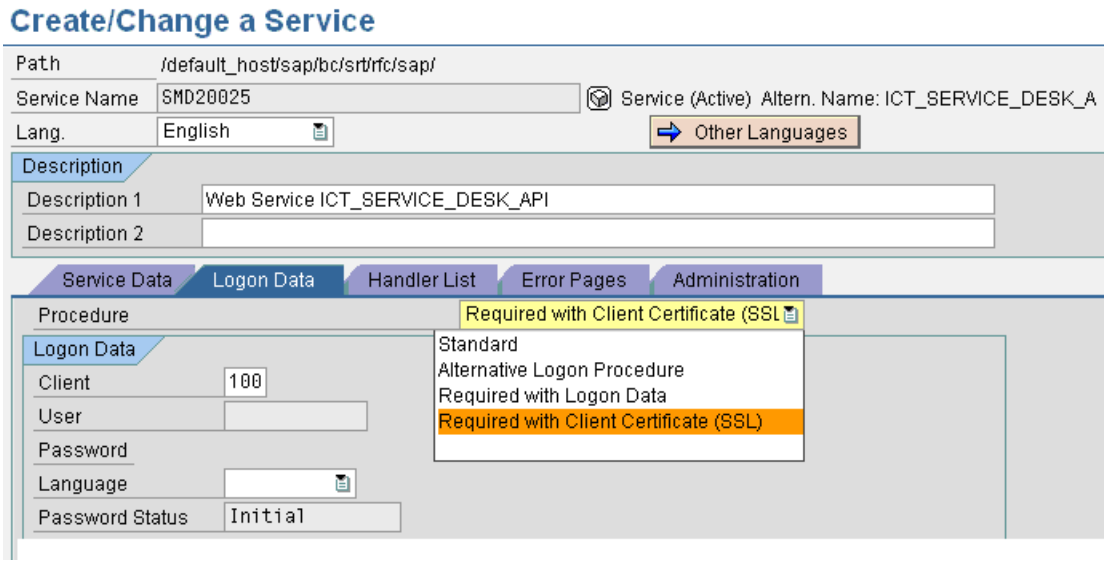
2 Execute the search of the service.

3 Double-click the Service to edit (or navigate to **`/default_host/sap/bc/srt/rfc/sap/`** and select **ICT_SERVICE_DESK_API**).

4 Open the **Create/Change a Service** dialog.

5 In the Logon tab select **Required with client Certificates (SSL)**.

6 Save the settings. Service **`ICT_SERVICE_DESK_API`** is configured for SSL connection only. In this procedure the lowest possible security level is specified. If "Required with Logon Data" is configured, then connecting via SSL and the client certificate is allowed.



For SSL communication, ensure that the ICM uses HTTPS.

Define the user mapping to the DN of the Certificate. The different ways of mapping are described in the SAP online help. Defining a user mapping to a DN is described below.

7 In SAP transaction **`/nse16`** open the view **`VUSREXTID`** (enter **`VUSREXTID`** in the table `Name` field).

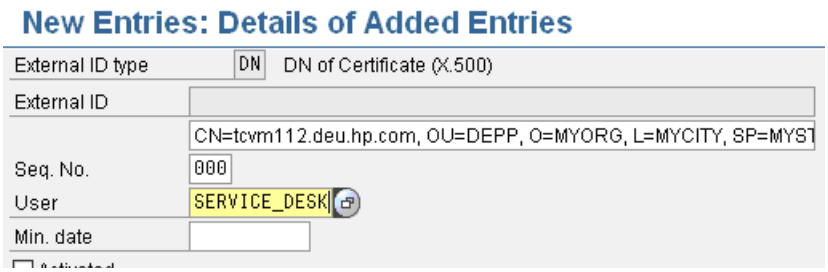
8 Select the Work Area **DN of Certificate X.500**.

9 In the user mapping dialog, as an external ID add the DN of the client certificate of Apache Tomcat (see Create Keystore and Truststore on page 78). Specify the exact DN of the certificate. For example:

```
CN=helen2006.asiapacific.hpqcorp.net, OU=TEST, O=GDCC, L=SH, SP=CN, C=CN
```

10 For **`Seq. No`** enter **`000, 001…`** (for internal use only).

11 Assign the SAP user for the Web Service. This user has all required permissions for managing incidents in SAP Solution Manager.



Set up SSL between SAP and SMSSMEX

This section describes how to setup SSL between SAP and SMSSMEX.

Create Keystore and Truststore

SMSSMEX requires

- Two separate stores that contain the certificates used to authenticate and encrypt communication.
- The following certificates
  - Signed certificate with the long hostname of the SMSSMEX server in the CN section (for example **`CN="server.hp.com"`**). This certificate must be mapped to an SAP user in SAP Solution Manager.
  - Certificate of the root CA used to sign the certificate of the SAP Solution Manager.
  - Certificate of the root CA used to sign the certificate of the SMSSMEX certificate.

The keystore must contain the following certificates:

- Root CA certificate used to sign the SMSSMEX certificate
- SMSSMEX certificate

The truststore must contain the root certificate used to sign the certificate of the SAP Solution Manager.

Any tool can be used to create and manage the key- and truststores. The following examples use the Java JDK tool `keytool` to create and import a signed certificate.

1. Create a self-signed certificate. The keypass and the storepass must be identical.

   ```
   keytool –genkey –alias <alias> -keyalg RSA –keystore <keystorefile>
   -storepass <password> -keypass <password> -dname "CN=<serverhost>,
   OU=<MYOU>, O=<MYORG>, L=<MYCITY>, ST=<MYSTATE>, C=<MY>"
   ```

   For example:

   ```
   keytool -genkey -alias ovictex -keyalg RSA -keystore "C:\Program
   Files\HP\SMSSMEX\config\certs\ovictex.keystore" -storepass ovictex
   -keypass ovictex -dname "CN=helen2006.asiapacific.hpqcorp.net, OU=TEST,
   O=GDCC, L=SH, ST=CN, C=CN"
   ```

2. Create a certificate request:

   ```
   keytool –certreq –keystore <keystorefile> -alias <alias> -storepass
   <password>
   ```

   For example:

   ```
   keytool -certreq -keystore "C:\Program
   Files\HP\SMSSMEX\config\certs\ovictex.keystore" -alias ovictex -storepass
   ovictex
   ```

3. Use the resulting certificate request to acquire a signed certificate from SAP Web (**https://websmp102.sap-ag.de/SSLTest**) with chain PKCS#7. Copy the signed response **`<filename>.p7b`** (for example, **`sap_rp.p7b`**).
4. Download the root certificate file for the following web site: **https://tcs.mysap.com/invoke/tc/getCert?SAPServerCA.der**.
5. Import the root certificate from the Certificate Authority (CA) into the keystore.

```
keytool –import –v –alias <alias2> –keystore <keystorefile> -storepass
<password> -file <rootcertificatefile>
```

For example:

```
keytool -import -v -alias saproot -keystore "C:\Program
Files\HP\SMSSMEX\config\certs\ovictex.keystore" -storepass ovictex -file
"C:\Program Files\HP\SMSSMEX\config\certs\getCert.cer"
```

6 Import the answer from the Certificate Authority into the keystore. Use the same keystore file and alias the request was created from.

```
keytool –import –v –alias <alias> –keystore <keystorefile> -storepass
<password> -file <certificatefile>
```

For example:

```
keytool -import -v -alias ovictex -keystore "C:\Program
Files\HP\SMSSMEX\config\certs\ovictex.keystore" -storepass ovictex -file
"C:\Program Files\HP\SMSSMEX\config\certs\sap.p7b"
```

To import the certificates into the truststore, use the same command as in the step above, but instead of **<keystorefile>** use the filename of the truststore (if it does not exist, it will be created automatically). For example:

```
keytool -import -v -alias saproot -keystore "C:\Program
Files\HP\SMSSMEX\config\certs\ovictex.truststore" -storepass ovictex -file
"C:\Program Files\HP\SMSSMEX\config\certs\getCert.cer"
```

Configure Tomcat SSL Use

To enable SSL with Tomcat, configure a new connector in the `server.xml` configuration file. The standard `server.xml` contains a connector definition that has been commented out. The following attributes are required:

```
port=<port>
scheme="https"
secure="true"
clientAuth="false"
sslProtocol = "TLS"
keystoreFile=<keystorefile>
keystorePass=<keystorepass>
truststoreFile=<truststorefile>
truststorePass=<truststorepass>
```

For example:

```
<Connector port="8443"
 maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
 enableLookups="false" disableUploadTimeout="true"
 acceptCount="100" debug="0" scheme="https" secure="true"
 clientAuth="false" sslProtocol="TLS"
 keystoreFile="C:/Program Files/HP/SMSSMEX/config/certs/ovictex.keystore"
 keystorePass="password"
 truststoreFile="C:/Program Files/HP/SMSSMEX/config/certs/ovictex.truststore"
 truststorePass="password"
/>
```

Configure Property Files

1 Modify `exthd.properties`.

a. In the SAP configuration files in property `exthd.webservice.endpoint` specify the new port (default is 8443) and use **`https://`** as the protocol. For example:

```
exthd.webservice.endpoint = https://watermelon.chn.hp.com:8001/sap/bc/
srt/rfc/sap/ICT_SERVICE_DESK_API?sap-client=300
```

b. Set the `exthd.webservice.authentication.scheme` to HTTPS.

2. Add the following configuration entries in `ovictex.properties`:

```
<saphostname>.keystore=C:/Program Files/HP/SMSSMEX/certs/ovictex.keystore
<saphostname>.keystore.password=~X1~H+7JAOrcX/R6kO5diPxV0w==
<saphostname>.truststore=C:/Program Files/HP/SMSSMEX/certs/
ovictex.truststore
<saphostname>.truststore.password=~X1~H+7JAOrcX/R6kO5diPxV0w==
```

For example:

```
watermelon.chn.hp.com.keystore= C:/Program Files/HP/SMSSMEX/certs/
ovictex.keystore
watermelon.chn.hp.com.keystore.password=~X1~H+7JAOrcX/R6kO5diPxV0w==
watermelon.chn.hp.com.truststore= C:/Program Files/HP/SMSSMEX/certs/
ovictex.truststore
watermelon.chn.hp.com.truststore.password=~X1~H+7JAOrcX/R6kO5diPxV0w==
```

Security Between HP Service Manager and SMSSMEX

This section describes how to configure security between HP Service Manager and SMSSMEX.

Configure HP Service Manager for SSL

This section describes how to configure HP Service Manager for SSL.

The prerequisite is OpenSSL version 9.7 or higher. For more information about downloading and installing OpenSSL, see **http://www.openssl.org**.

Service Manager provides in the `/RUN` directory an OpenSSL executable file that can be used to generate and sign most certificates. Implementing the trusted sign-on requires file `openssl.conf` in addition to the executable (the file is available after installing OpenSSL).

- The `*.pem` files must be different in at least one section when being prompted for distinguished name information. For Windows clients, this difference is the common name. If the Web Tier or Windows client is on the same system as the server then an additional differentiating entry, such as organization, must be made.
- When prompted, always use the fully qualified name (**`computer.domain.com`**) as the first/last name.

Generate a Private/Public Key Pair for Root Certificate Authority

1. Generate an RSA private key.

   ```
   openssl genrsa -des3 -out cakey.pem 2048
   ```

2. Create a self-signed root certificate for the Certificate Authority (CA).

   ```
   openssl req -new -key cakey.pem -x509 -days 1095 -out mycacert.pem -config
   openssl.conf
   ```

   To make a unique `.pem` file, give a unique Organization Name (for example `org1`). When asked for a Common Name, enter the fully qualified name of the Service Manager Server host.

3. Import the self-signed root certificate into a trust key store.

   ```
   keytool -import -keystore <trustkeystore> -trustcacerts -alias <alias>
   -file <certificate>
   ```

   For example:

   ```
   keytool -import -keystore cacerts -trustcacerts -alias scca -file
   mycacert.pem
   ```

Generate a Private/Public Key Pair for Service Manager Server

1. Generate a private/public key pair.

   ```
   keytool -genkey -alias <alias> -keystore <keystorefile>
   ```

   For example:

```
keytool -genkey -alias scserver -keystore scserver.keystore
```

When asked for organization name, enter a unique name (for example `org2`). When asked for first and last name, enter the fully qualified name of the Service Manager Server host.

2 Generate the request file.

```
keytool -certreq -alias <alias> -keystore <keystorefile> -file
<requestfile>
```

For example:

```
keytool -certreq -alias scserver -keystore scserver.keystore -file
scservercert_req.crs
```

3 Self-sign the request.

```
openssl x509 -req -days <validdays> -in <requestfile> -CA
<certificatefile> -CAkey <keystorefile> -CAcreateserial -out
<certificatefile>
```

For example:

```
openssl x509 -req -days 1095 -in scservercert_req.crs -CA mycacert.pem
-CAkey cakey.pem -CAcreateserial -out scservercert.pem
```

4 Import the root CA certificate into the server keystore.

```
keytool -import -trustcacerts -alias <alias> -keystore <keystorefile>
-file <certificatefile>
```

For example:

```
keytool -import -trustcacerts -alias scca -keystore scserver.keystore
-file mycacert.pem
```

5 Import the signed certificate into the keystore.

```
keytool -import -trustcacerts -alias <alias> -keystore <keystorefile> -
file <certificatefile>
```

For example:

```
keytool -import -trustcacerts -alias scserver -keystore scserver.keystore
-file scservercert.pem
```

Generate the Client Keystore for Service Manager Client

1 Generate the private/public key pair (with the first and last name and the case-sensitive fully qualified name of the machine).

```
keytool -genkey -alias <alias> -keystore <keystorefile>
```

For example:

```
keytool -genkey -alias scclient -keystore scclient.keystore
```

When asked for organization name, input a unique one (for example org3). When asked for the first and last name, enter the fully qualified name of the Service Manager client host.

2 Generate the request file.

```
keytool -certreq -alias <alias> -keystore <keystorefile> -file
<requestfile>
```

For example:

```
keytool -certreq -alias scclient -keystore scclient.keystore -file
scclientcert_req.crs
```

3 Self-sign the request.

```
openssl x509 -req -days <validdays> -in <requestfile> -CA
<certificatefile> -CAkey <keystorefile> -CAcreateserial -out
<certificatefile>
```

For example:

```
openssl x509 -req -days 365 -in scclientcert_req.crs -CA mycacert.pem
-CAkey cakey.pem -CAcreateserial -out scclientcert.pem
```

4 Import the root CA certificate into the client keystore.

```
keytool -import -trustcacerts -alias <alias> -keystore <keystorefile>
-file <certificatefile>
```

For example:

```
keytool -import -trustcacerts -alias scca -keystore scclient.keystore
-file mycacert.pem
```

5 Import the self-signed certificate into the client keystore.

```
keytool -import -trustcacerts -alias <alias> -keystore <keystorefile>
-file <certificatefile>
```

For example:

```
keytool -import -trustcacerts -alias scclient -keystore scclient.keystore
-file scclientcert.pem
```

Generate the Client Keystore for SMSSMEX

1 Generate the private/public key pair.

```
keytool -genkey -alias <alias> -keystore <keystorefile>
```

For example:

```
keytool -genkey -alias ovictex -keystore ovictex.keystore
```

When asked for organization name, enter a unique name (for example `org4`). When asked for the first and last name, enter the fully qualified name of the incident exchange middleware host.

2 Generate the request file.

```
keytool -certreq -alias <alias> -keystore <keystorefile> -file
<requestfile>
```

For example:

```
keytool -certreq -alias ovictex -keystore ovictex.keystore -file
ovictexcert_req.crs
```

3 Self-sign the request.

```
openssl x509 -req -days <validdays> -in <requestfile> -CA
<certificatefile> -CAkey <keystorefile> -CAcreateserial -out
<certificatefile>
```

For example:

```
openssl x509 -req -days 365 -in ovictexcert_req.crs -CA mycacert.pem
-CAkey cakey.pem -CAcreateserial -out ovictexcert.pem
```

4 Import the root CA certificate into the client keystore.

```
keytool -import -trustcacerts -alias <alias> -keystore <keystorefile>
-file <certificatefile>
```

For example:

```
keytool -import -trustcacerts -alias scca -keystore ovictex.keystore -file
mycacert.pem
```

5 Import the self-signed certificate into the client keystore.

```
keytool -import -trustcacerts -alias <alias> -keystore <keystorefile>
-file <certificatefile>
```

For example:

```
keytool -import -trustcacerts -alias ovictex -keystore ovictex.keystore
-file ovictexcert.pem
```

Generate the Trust-List Keystore for Service Manager Server

1 Export the certificate file.

```
keytool -export -alias <alias> -keystore <keystorefile> -file
<certificatefile>
```

For example:

```
keytool -export -alias scclient -keystore scclient.keystore -file
scclientpubkey.crt
keytool -export -alias ovictex -keystore ovictex.keystore -file
ovictexpubkey.crt
```

2 Import the certificate file.

```
keytool -import -alias <alias> -file <certificatefile> -keystore <jksfile>
```

For example:

```
keytool -import -alias scclient -file scclientpubkey.crt -keystore
trustedclients.jks
keytool -import -alias ovictex -file ovictexpubkey.crt -keystore
trustedclients.jks
```

SSL Configuration in Service Manager Server

1 Import the root certificate of the SAP Certificate Authority into the trust key store.

```
keytool -import -keystore <trustkeystore> -trustcacerts -alias <alias>
-file <certificate>
```

For example:

```
keytool -import -keystore cacerts -trustcacerts -alias sapca -file
sapca.cert
```

2 Copy the generated files `cacerts`, `scserver.keystore`, and `trustedclients.jks` into `<Service Manager installation path>\Server\RUN`.

3 Add the following entries to `sm.ini`:

```
#
# SSL configuration
#
ssl:1
ssl_reqClientAuth:1

#
# Certificates
#
truststoreFile:cacerts
truststorePass:password
keystoreFile:scserver.keystore
keystorePass:password
ssl_trustedClientsJKS:trustedclients.jks
ssl_trustedClientsPwd:password
```

4 Open the Service Manager Client.

5 Go to **Tailoring** → **Script Library**.

6 Search for name **`HPSAPTrigger`**.

7 Change the following javascript code.

```
var url = "http://<smssmex full host name>:<port>/ovictex/servlet/
OvHDTrigger?parameters="
+ encodeURIComponent(action) + ";" + encodeURIComponent(incidentId) + ";"
+ encodeURIComponent(extHdId);
```

to

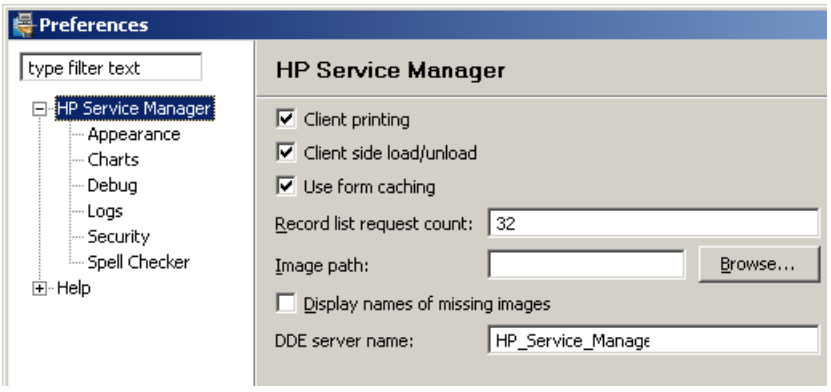
```
var url = "https://<smssmex full host name>:<ssl port>/ovictex/servlet/
OvHDTrigger?parameters="
+ encodeURIComponent(action) + ";" + encodeURIComponent(incidentId) + ";"
+ encodeURIComponent(extHdId);
```

SSL Configuration in Service Manager Client

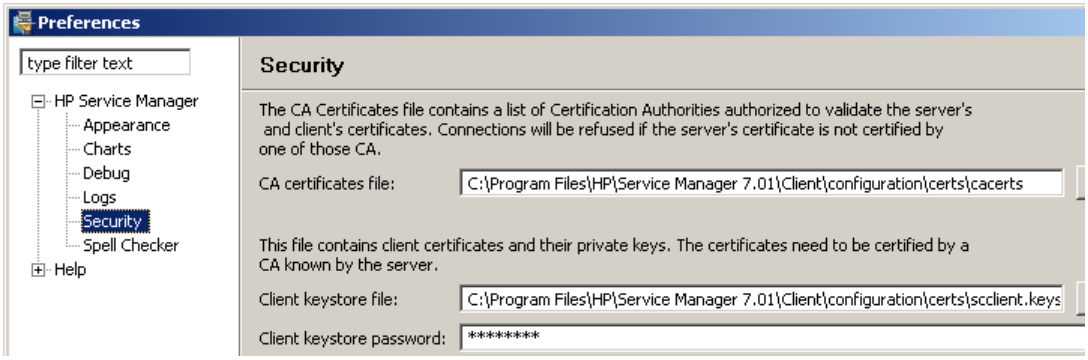
To configure SSL in Service Manager client, do the following:

1 Open the Service Manager Client.

2 From the menu select **Window** → **Preferences...** to open the Preferences dialog.



3 Expand the HP Service Manager node in the left menu tree. Select **Security** to open the client security dialog.



4 Click **Browse...**.

5 Specify the CA certificates file and Client keystore file.

6 Input the password of the client keystore in the `Client keystore password` field.

7 Click **OK** to save the Security configuration.

8 Restart Service Manager Client to enable the newly configured Security information.

9 In the Connections dialog, the value of field Server host name must be the fully qualified name of the Service Manager server.

10 In the Advanced tab, make sure that **Use SSL Encryption** is checked.

Service Manager Web Client SSL Configuration

To configure SSL in Service Manager web client, do the following:

1 Copy the trust keystore and client keystore files to the WEB-INF folder of the Service Manager Web Application Server.

2 Open the Web configuration file `web.xml` in a text editor.

3 Modify the following configuration entry.

```
<init-param>
  <param-name>serverHost</param-name>
  <param-value>servername.domainname.com</param-value>
</init-param>
```

For example:

```
<init-param>
  <param-name>serverHost</param-name>
  <param-value>SMCI02.chn.hp.com</param-value>
</init-param>
<init-param>
  <param-name>serverPort</param-name>
  <param-value>serverPort</param-value>
</init-param>
```

For example:

```
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

```
<init-param>
  <param-name>ssl</param-name>
  <param-value>true</param-value>
</init-param>
<init-param>
  <param-name>cacerts</param-name>
  <param-value>trustKeystore</param-value>
</init-param>
```

For example:

```
<init-param>
  <param-name>cacerts</param-name>
  <param-value>/WEB-INF/cacerts</param-value>
</init-param>
<init-param>
  <param-name>keystore</param-name>
  <param-value>clientKeystore</param-value>
</init-param>
```

For example:

```
<init-param>
  <param-name>keystore</param-name>
  <param-value>/WEB-INF/scclient.keystore</param-value>
</init-param>
<init-param>
  <param-name>keystorePassword</param-name>
  <param-value>clientKeystorePassword</param-value>
</init-param>
```

For example:

```
<init-param>
  <param-name>keystorePassword</param-name>
  <param-value>sm9client</param-value>
</init-param>
```

4 Open `WEB-INF/classes/application-context.xml` in a text editor. Change

```
/**=httpSessionContextIntegrationFilter,anonymousProcessingFilter
```

to

```
/
**=httpSessionContextIntegrationFilter,preAuthenticationFilter,anonymousP
rocessingFilter
```

Configure SMSSMEX for SSL Communication with Service Manager

To configure SMSSMEX for SSL communications with Service Manager, do the following:

1 Import the root CA into the trust keystore.

```
keytool -import -trustcacerts -alias <alias> -keystore <keystorefile>
-file <certificatefile>
```

For example:

```
keytool -import -trustcacerts -alias scca -keystore ovictex.truststore
-file mycacert.pem
```

2 Configure `ovictex.properties`.

a Set `sc.webservice.endpoint`.

```
sc.webservice.endpoint = http://<smhostname>:<port>/sc62server/PWS
```

b Add the following configuration entries in `ovictex.properties`.

```
<smhostname>.keystore=<ovictex keystore file>
<smhostname>.keystore.password=<keystore password>
<smhostname>.truststore=<ovictex truststore file>
<smhostname>.truststore.password=<truststore password>
```

For example:

```
sc.webservice.endpoint = http://SMCI02.chn.hp.com:13080/sc62server/PWS
……
SMCI02.chn.hp.com.keystore=C:/Program Files/HP/SMSSMEX/config/certs/
ovictex.keystore
SMCI02.chn.hp.com.keystore.password=~X1~eD+6cy6OMNxdK9tcCQVBww==
SMCI02.chn.hp.com.truststore=C:/Program Files/HP/SMSSMEX/config/certs/
ovictex.truststore
SMCI02.chn.hp.com.truststore.password=~X1~eD+6cy6OMNxdK9tcCQVBww==
```

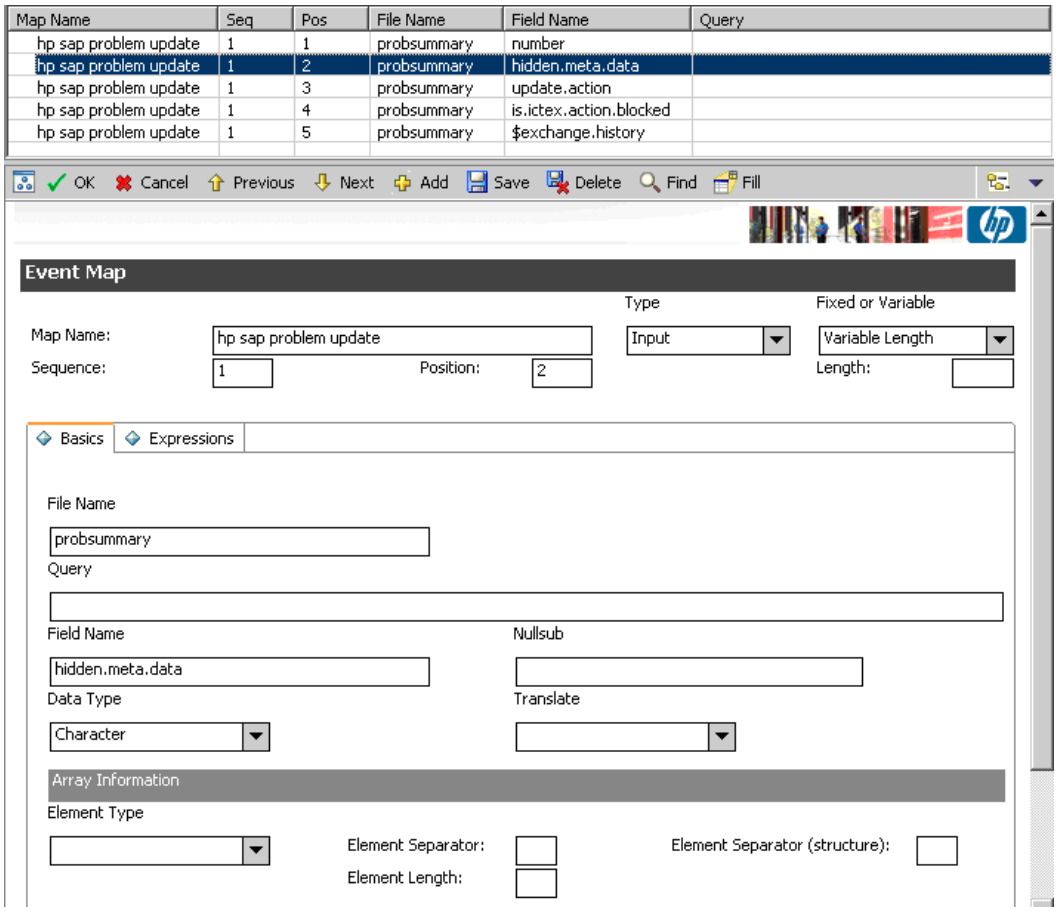
The `keystore.password` and `truststore.password` should use `<SMSSMEX_installDir>/bin/encryptPasswords.bat` to encrypt. For usage of `encryptPasswords.bat`, refer to Tools on page 114.

8 Upgrading SMSSMEX

Upgrading SMSSMEX from V1.00 to V1.01

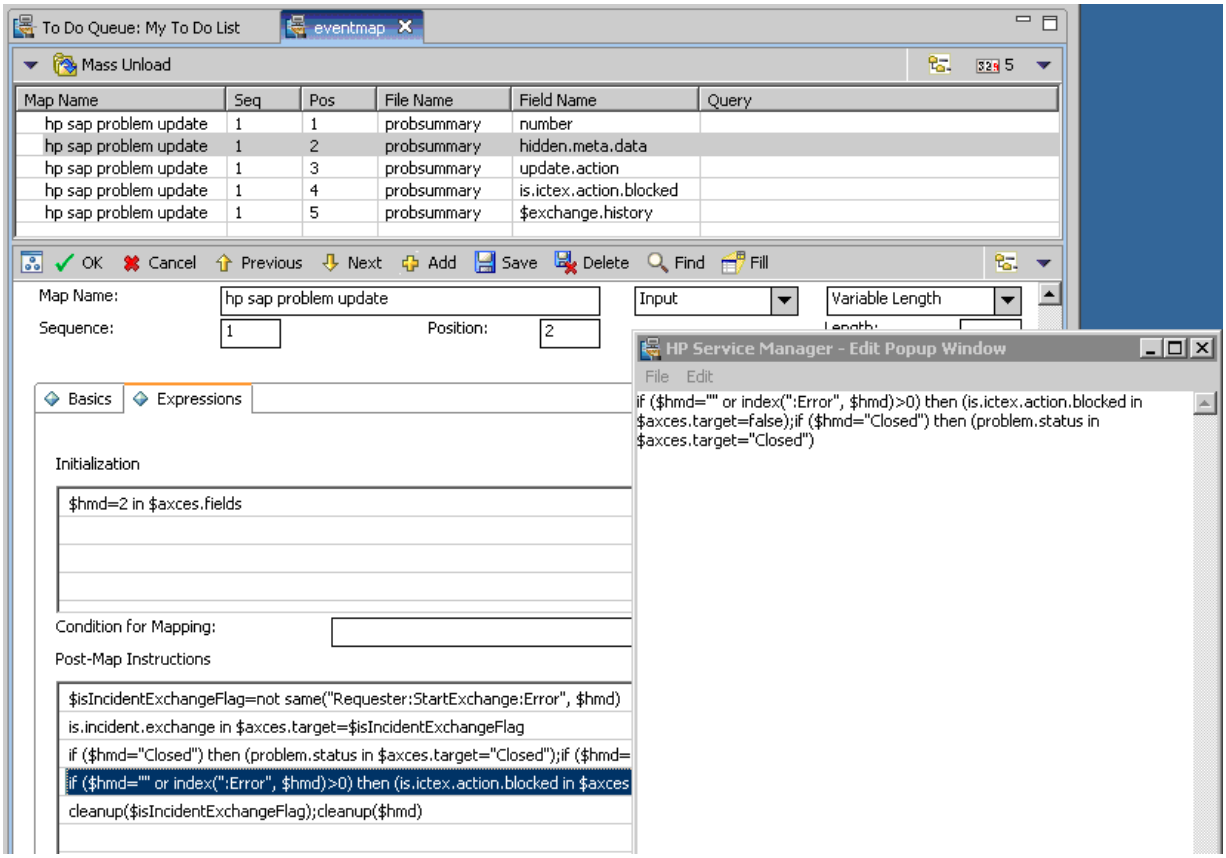
To upgrade SMSSMEX from v1.00 to v1.01, perform the following steps:

- Open maps **hp sap problem update**, and update **hidden.meta.data** Post-Map instruction.
  a. Go to **Tailoring** → **Event Services** → **Maps** on Service Manager, or **Utilities** → **Event Services** → **Administration** → **Maps** on Service Center, the Event Map page appears.
  b. Enter **`hp sap problem update`** in Map Name field, and click **Search**.

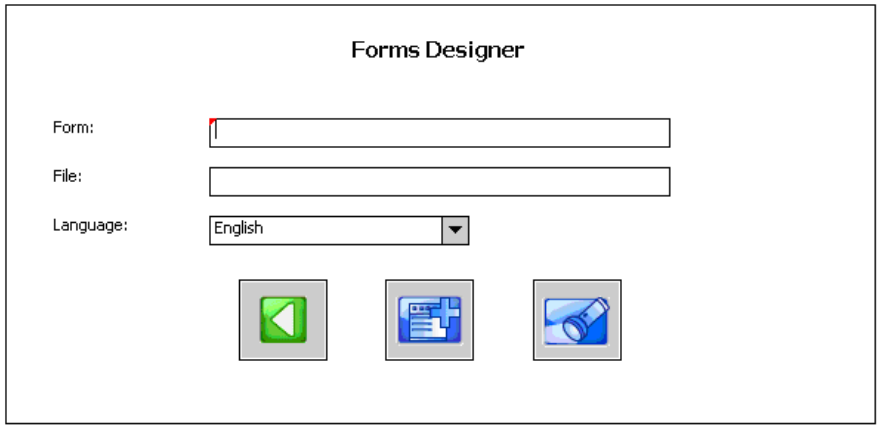


  c. Select the entry with Field Name as `hidden.meta.data`, and go to **Expressions** tab. Insert the following line (**[HIDDEN.META.DATA_UPGRADE]** in `code_sm9.txt`) in Post-Map Instructions before the `cleanup($isIncidentExchangeFlag);cleanup($hmd)` line:

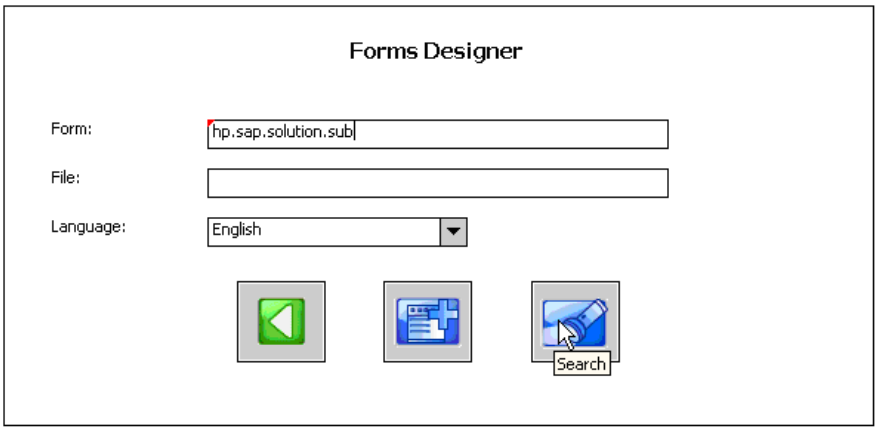
```
if ($hmd="" or index(":Error", $hmd)>0) then (is.ictex.action.blocked
in $axces.target=false);if ($hmd="Closed") then (problem.status in
$axces.target="Closed"; status in $axces.target="closed")
```



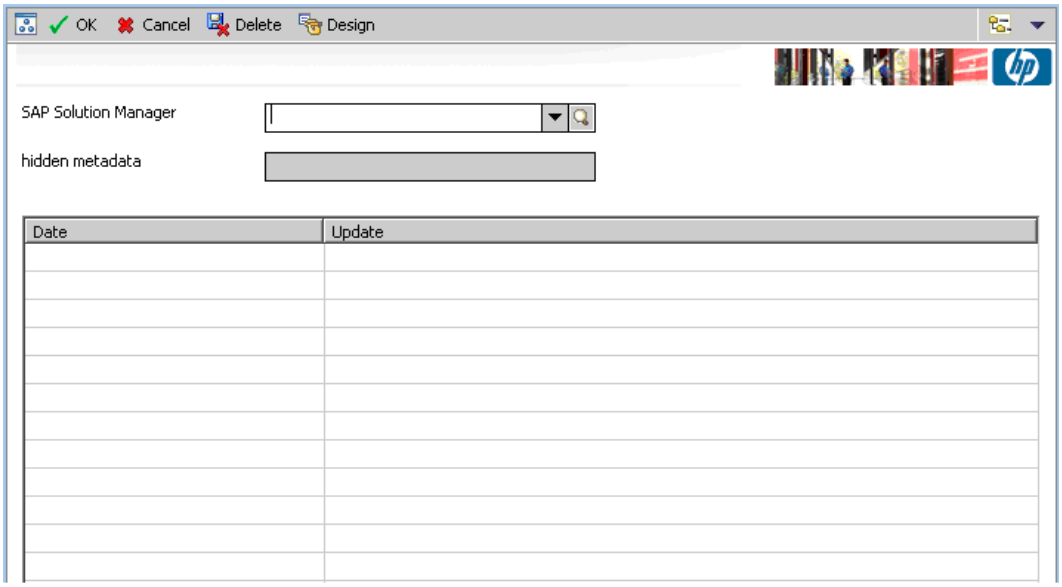
- Modify `hp.sap.solution.sub` form.
  a. Go to **Tailoring** → **Forms Designer** on Service Manager, or **Toolkit** → **Forms Designer** on Service Center. The Forms Designer page appears.



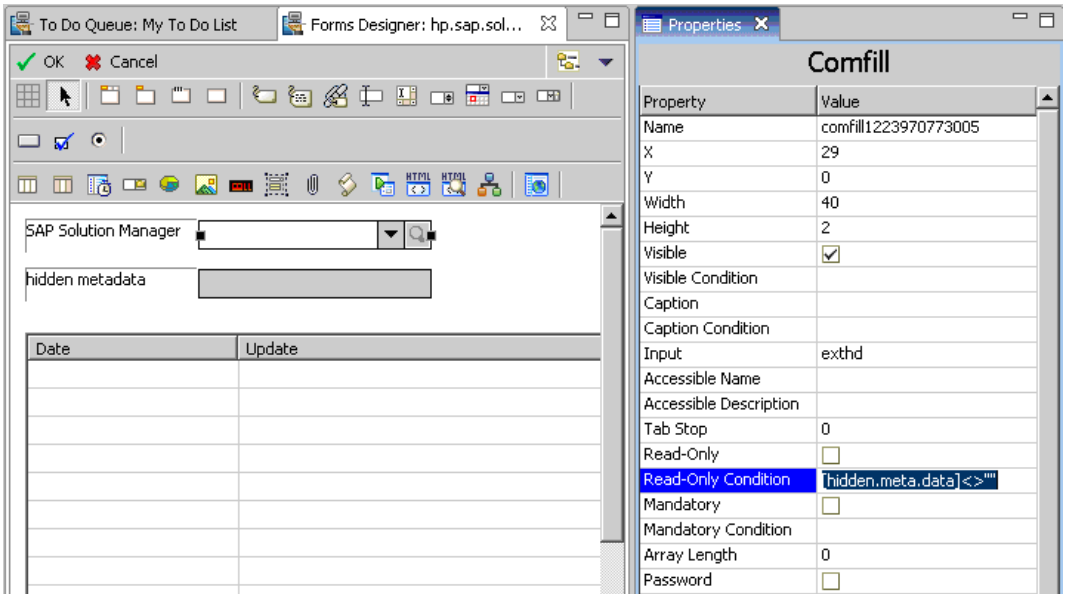
b Enter **`hp.sap.solution.sub`**, and then click **Search**.



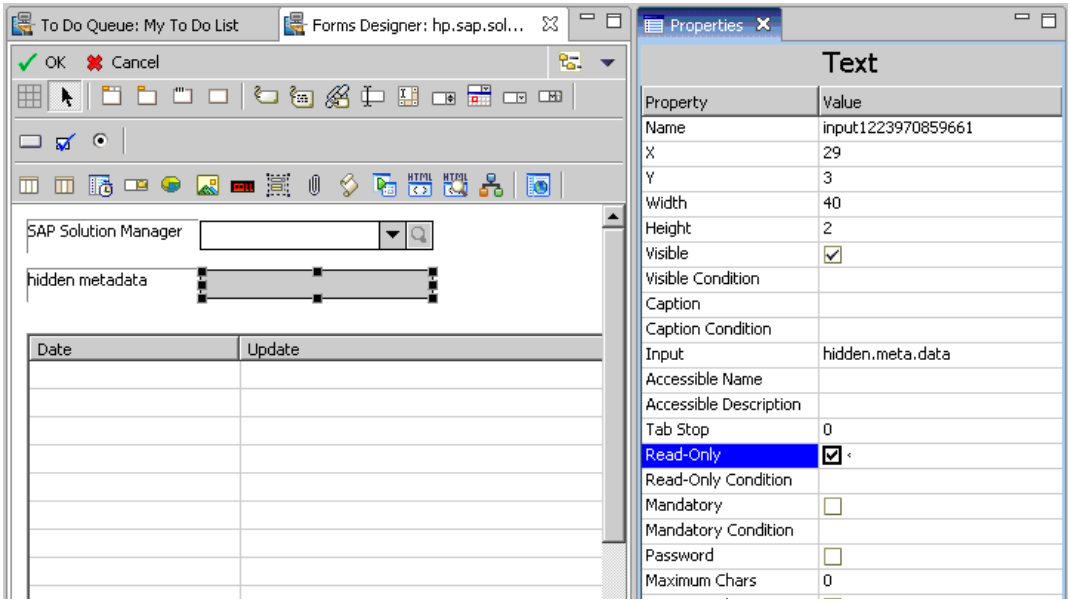
c The Design window appears. Click **Design** to enable design mode for the `hp.sap.solution.sub` form.



d Select **comfill** control from the dropdown list for **SAP Solution Manager** field, then in the Properties pane displayed, enter **`[hidden.meta.data]<>""`** as the value for Read-Only Condition property, and uncheck **Fill Button Visible** to hide the fill button.



e Set **hidden metadata** field to read-only mode by checking `Read-Only` property in the Properties pane.



- Add `hp.sap.solution.sub` sub-form into the following forms:

  `IM.default.open.g`
  `IM.default.update.g`

  — On Service Manager, see step 1 to step 4 on page 40, Exchange History section in Chapter 5 for detailed instructions.

- Add a link record for field `exthd` of `probsummary` table:
- Add the following expression.

a Go to **External Access** page.

- On Service Manager 7.0x, go to **Tailoring** → **Tailoring Tools** → **External Access**
- On Service Center 6.2, go to **Utilities** → **Tools** → **Web Services** → **External Access**

b Enter **`IncidentManagement`** in the Service Name field, click **Search**, and then add the following expression in Expressions.

```
if (hidden.meta.data in $L.file="Closed") then (problem.status in
$L.file="Closed")
```

Upgrading SMSSMEX from V1.01 to V1.02

The SAP System Landscape Directory Registration is a new feature in SMSSMEX v1.02. However, this feature is optional. If you do not deploy the SAP System Landscape Directory, the functionality of SMSSMEX v1.02 will not be affected.

For detailed SAP System Landscape Directory registration instructions, refer to *Appendix E, SAP System Landscape Directory Registration*.

Upgrading SMSSMEX from V1.02 to V1.10

To upgrade SMSSMEX from v1.02 to v1.10, perform the following tasks:

Task 1: Updating the probsummary table

Do the following:

1 Select **System Definition** → **Tables** → **probsummary** → **Tab Fields and Keys**.

2 Create the following additional field in table `probsummary`.

Field name	Caption	Data type
sap.incident.type	Sap Incident type	Character

Task 2: Setting default closure code and resolution when closing incidents

When closing incidents from SAP Solution Manager, change the incidents' status and set default closure code and resolution if empty. Do the following:

1 Click **Tailoring** → **Web Services** → **WSDL configuration**.

2 Enter **`IncidentManagement`** in the Service Name field and click **Search**.

3 In the Expression tab, replace the last code line with the following:

```
if (hidden.meta.data in $L.file="Closed") then (problem.status in
$L.file="Closed";status in $L.file="closed";if null(resolution.code in
$L.file) then (resolution.code in $L.file="Automatically
Closed");resolution in $L.file=insert(resolution in $L.file, 1, 1, "Closed
by SMSAP integration."))
```

4 In the Fields tab, update the caption for severity to Urgency.

5 Click **Save**.

Task 3: Configuring WSDL Mapping

See *Configuring WSDL Mapping* on page 58.

Task 4: Setting default closure code and resolution when updating from event service

Do the following:

1 Click **Tailoring** → **Event Service** → **Maps**.

2 Enter **`hp sap problem update`** in the Map Name field. Enter **`2`** in the Position field and click **Search**.

3 In the Expressions tab, locate the code line starting with `if ($hmd="Closed")`. Replace this line with the following:

```
if ($hmd="Closed") then (problem.status in $axces.target="Closed";status
in $axces.target="closed";if null(resolution.code in $axces.target) then
(resolution.code in $axces.target="Automatically Closed");resolution in
$axces.target=insert(resolution in $axces.target, 1, 1, "Closed by SMSAP
integration."))
```

4 Click **Save**.

Task 5: Updating the exchange process

Do the following:

1 Click **Tailoring** → **Document Engine** → **Processes**.

2 Enter **`im.exchange.incident`** in the Process Name field and click **Search**.

3 In the Initial Expressions tab, append the following scripts to the end of the codes:

```
if ($L.action="processincident") then (update.action in
$L.file=insert(update.action in $L.file, 1, 1, scmsg(3, "SMSAP", {number
in $L.file}));update.action in $L.file=insert(update.action in $L.file, 1,
1, $L.stamp))

if ($L.action="acceptincidentprocessing") then (update.action in
$L.file=insert(update.action in $L.file, 1, 1, scmsg(7,
"SMSAP"));update.action in $L.file=insert(update.action in $L.file, 1, 1,
$L.stamp))

if ($L.action="rejectincidentsolution") then (update.action in
$L.file=insert(update.action in $L.file, 1, 1, scmsg(8,
"SMSAP"));update.action in $L.file=insert(update.action in $L.file, 1, 1,
$L.stamp))
```

4 Click **Save**.

Task 6: Updating the HPSAPTrigger script

Do the following:

1 Click **Tailoring** → **Script Library**.

2 Enter **`HPSAPTrigger`** in the Name field and click **Search**.

3 Replace the content with the **[HPSAPTrigger]** section in `code_sm9.txt`.

4 Click **Save**.

Task 7: Updating link line in probsummary

Do the following:

1. Click **Tailoring** → **Tailoring Tools** → **Links**.
2. Enter **`probsummary`** in the Name field and click **Search**.
3. Locate the line with `device` as Target File Name. Click **More** → **Select Line**.
4. Change the scripts in the Post Expressions to the following:

   ```
   if (sap.incident.type in $File="sapinstance") then ($continue=true) else
   ($continue=false)
   ```

5. Click **Save**.
6. Locate the line with `joinsapinstance` as Target File Name. Click **More** → **Select Line**.
7. In Source Field (Fill To/Post From), change **`type`** to **`sap.incident.type`**.
8. Click **Save**.

Task 8: Updating format control - probsummary

Do the following:

1. select **Tailoring** → **Format Control** and search for **probsummary** → **Subroutines**, then right-click and select **Show Expanded Form** from the pop-up menu.
2. Update the `Add` field and the `Update` field in the `fill.fc` application as shown below:

Field	Value
Add (**[FILL.FC]** in `code_sm9.txt`)	null(logical.name in $file) and not null(sap.sid in $file)
Update (**[FILL.FC]** in `code_sm9.txt`)	null(logical.name in $file) and not null(sap.sid in $file)

Task 9: Creating system messages

Do the following:

1. Enter **`scmsg`** in the command field and click **Execute Command**. The Search Message Records page opens.
2. Enter each of the following messages, and click **Add**:

Language Code	Class	Message Number	Severity	Text
en	SMSAP	1	1	SAP Solution Manager has received Incident %S from Service Manager. This incident is "%S".
en	SMSAP	2	1	Select a SAP Solution Manager before sending the incident.
en	SMSAP	3	1	Incident ID at external helpdesk is %S
en	SMSAP	4	1	Incident %S's sending is failed: %S

Language Code	Class	Message Number	Severity	Text
en	SMSAP	5	1	%S: (trigger #%S: Incident %S;%S;%S
en	SMSAP	6	1	----Automatically send to SAP:
en	SMSAP	7	1	Send Back from External Service Desk.
en	SMSAP	8	1	Refuse Solution from External Service Desk.

Task 10: Adding SMSAP SMIS template

Before managing the SM-SAP integration in SMIS, do the following to create a template for the SM-SAP integration:

1. Click **Tailoring** → **Script Library**.
2. Enter **`SMSAPManager`** in the Name field and then click **Add**.
3. Copy and paste the following scripts for SMSAPManager, and then click **Compile** to verify the syntax.

```
//SMSSM1.10 SM9.3
var Class = lib.smis_Prototype.getClass();

var SMSAPManagerClass = Class.create(lib.smis_Manager.getClass(), {

  isScheduleBased: function() {
    return false;
  }
});
function getClass() {return SMSAPManagerClass;}
```

4. Click **Save** and **OK**.
5. Click **Tailoring** → **Database Manager**.
6. Enter **`SMISRegistry`** in the Table field and then click **Search**.
7. Enter the following values to the fields in the General tab:

Field	Value
Name	SMSAP
Version	1.10
Manager Class Name	SMSAPManager
SM Adapter	smis_DummyAdapter
Endpoint Adapter	smis_DummyAdapter

Field	Value
Instance Count	1
Category	UI-based
Description	Service Manager integrate with SAP Solution Manager

8 Add the following parameter in the Parameter tab:

Name	Value	Category
baseurl	http://<host>:<port>/ovictex/servlet/OvHDTrigger	General

9 Click **Add** to save the template.

Task 11: Adding the SMSAP field in info table

For every SMIS integration, a field with the integration name should be created in info table.

1 Select **System Definition** → **Tables** → **info** → **Tab Fields and Keys**.

2 Create a field in SMIS structure. The Field Name is **`SMSAP`** and the Data Type is **`Logical`**.

3 Click **Save**.

Task 12: Customizing incident processes

To control the visibility of SAP Solution Manager according to the SMSAP instance's status, do the following to customize the incident processes for `im.view.init` and `im.open.setup`:

1 Click **Tailoring** → **Document Engine** → **Processes**.

2 Enter **`im.view.init`** and click **Search**.

3 In the Initial Expressions tab, append the following scripts to the end of the codes:

```
$SMSAP=nullsub(SMSAP in $G.system.info, "false")
```

4 In the Initial Javascript tab, copy and paste the following scripts:

```
var configItem =
lib.smis_ConfigurationManager.getEnabledConfigItem("SMSAP");

if (configItem != null) {

        var solMans =
configItem.getParametersByCategory("SolutionManager");


        var values = [];

        var names = [];


        for (var id in solMans) {

                values.push(id);
```

```
                names.push(solMans[id]);
            }

            system.vars.$G_solMans_values=values;
            system.vars.$G_solMans_names=names;
    }
```

5 Click **Save** and **OK**.

6 Enter **`im.open.setup`** and click **Search**.

7 Add the following code to the Initial Expressions tab:

```
$SMSAP=false
```

8 Click **Save** and **OK**.

Task 13: Updating display options

Check SMIS enabled/disabled status for all exchange buttons. Do the following to add a statement to the condition of each display option:

1 Click **Tailoring** → **Tailoring Tools** → **Display Options**.

2 Enter **`processincident`** and click **Search**.

3 In the Condition field, append the following script to the end of the codes:

```
and nullsub(SMSAP in $G.system.info, "false")
```

4 Click **Save**.

5 Repeat step 3 and step 4 for other display options, which include **`closeincident`**, **`verifyincidentsolution`**, **`rejectincidentsolution`**, **`addinfo`** and **`acceptincidentprocessing`**.

Task 14: Adding condition in formats

Do the following to add SMIS status condition of each format:

1 Click **Tailoring** → **Forms Designer**.

2 Enter **`IM.open.incident`** in the Form field and click **Search**.

3 Click **Design**.

4 Select the SAP Solution Manager section, and enter **`[$SMSAP]=true`** in the Visible Condition field of the Proterties tab.

5 Click **OK**.

6 Click **OK**.

7 Repeat step 2 to step 6 for other display formats, which include **`IM.update.incident`** and **`IM.close.incident`**.

Task 15: Updating the SAP Solution Manager subform

The SAP Solution Manager clients are configured in SMIS. Do the following to update the SAP Solution Manager subform:

1 Click **Tailoring** → **Forms Designer**.

2 Enter **`hp.sap.solution.sub`** in the Form field and click **Search**.

3 Click **Design**.

4 Click **Comfill**.

5 In the Property tab, clear the checkbox for the Third Button Visible field.

6 Remove the value in the Value List Condition field and the Display List Condition field.

7 Type **`$G.solMans.values`** in the Value List field.

8 Type **`$G.solMans.names`** in the Value List field.

9 Click the **Hidden Metadata** label and update the caption to **`Exchange Status`**.

10 Click **OK**.

11 Click **OK**.

Task 16: Disabling the duplicate button for exchange incidents

Do the following to disable the duplicate button for exchange incidents:

1 Click **Tailoring** → **Tailoring Tools** → **Display Options**.

2 Enter **`apm.edit.problem_clone`** in the Unique ID field and click **Search**.

3 In the Condition field, append the following script to the end of the codes:

```
and hidden.meta.data in $L.filed=NULL
```

4 Click **Save** and **OK**.

Task 17: Renaming labels in Service Manager

Do the following to rename Reject Solution to Refuse Solution:

1 Click **Tailoring** → **Tailoring Tools** → **Display Options**.

2 Enter **`apm.edit.problem_rejectincidentsolution`** in the Unique ID field and click **Search**.

3 In the Default Label field, rename **`Reject Solution`** to **`Refuse Solution`**.

Do the following to rename Close Incident to Close SAP Incident:

1 Click **Tailoring** → **Tailoring Tools** → **Display Options**.

2 Enter **`apm.edit.problem_closeincident`** in the Unique ID field and click **Search**.

3 In the Default Label field, rename **`Close Incident`** to **`Close SAP Incident`**.

Task 18: Renaming icons

Do the following:

1 Rename `tclose_i.gif` to `tclose_s.gif`.

2 Rename `treject.gif` to `trefuse.gif`.

Refer to *Appendix D, Deploying Button Icons* for more information about where the buttons are stored.

Task 19: Checking if SAP Solution Manager is selected when clicking Send Incident

Do the following:

1 Click **Tailoring** → **Tailoring Tools** → **Display Options**.

2 Enter **`apm.edit.problem_processincident`** in the Unique ID field and click **Search**.

3 In the Pre JavaScript tab, add the following script:

```
if (system.vars.$L_file.exthd==null || system.vars.$L_file.exthd=="")
system.functions.msg(system.functions.scmsg( 2, "SMSAP" ),3);
```

4 Click **Save**.

5 Click **Tailoring** → **Document Engine** → **Processes**.

6 Enter **`im.view`** and click **Search**.

7 In the condition of processincident Display Action, add the following script:

```
not null(exthd in $L.file)
```

8 Click **Save**.

Task 20: Upgrading SMSSMEX midware on Tomcat

Do the following:

1 Stop SMSSMEX V1.02.

See *Starting/Stopping SMSSMEX* on page 26 to stop SMSSMEX V1.02.

2 Backup the configuration files.

a Backup all files in the `<SMSSMEX_installDir>\config` folder.

b Backup the `<SMSSMEX_installDir>\tomcat\conf\server.xml` file.

c Backup other configuration files which have been customized.

3 Uninstall SMSSMEX V1.02.

See *Uninstall SMSSMEX* on page 18 to uninstall SMSSMEX V1.02.

4 Install SMSSMEX V1.10.

See *Install SMSSMEX* on page 17 to install SMSSMEX V1.10.

5 Configure SMSSMEX V1.10.

a See *Configuring Tomcat* on page 20 to configure Tomcat.

b See *Configuring ovictex.properties* on page 23 to configure `ovictex.properties`.

c See *Configuring File ovictexInternal.properties* on page 24 to configure `ovictexInternal.properties`.

d See *External Helpdesks* on page 25 to configure external Helpdesks.

e See *Configuring FieldMapping.xml* on page 26 to configure `FieldMapping.xml`.

Or you can copy parameter values from the backup configuration files to configure SMSSMEX V1.10. Do not just copy and replace `ovictex.properties` and `FieldMapping.xml` because these files have been updated in SMSSMEX V1.10.

6 Start SMSSMEX V1.10.

See *Starting/Stopping SMSSMEX* on page 26 to start SMSSMEX V1.10.

Task 21: Upgrading SMSSMEX midware on Weblogic

Do the following:

1 Stop SMSSMEX V1.02 and the WebLogic server.

2 Backup the configuration files.

a. Backup all files in the `<SMSSMEX_installDir>\config` folder.

b. Backup other configuration files which have been customized.

3. Uninstall SMSSMEX V1.02.

   a. See *Installing SMSSMEX* on page 27 to uninstall SMSSMEX V1.02.

   b. Remove `ovictex.war` and other unzipped files from the `% SMSSMEX_HOME%/war` folder.

4. Install SMSSMEX V1.10.

   See *Installing SMSSMEX* on page 27 to install SMSSMEX V1.10.

5. Configure SMSSMEX V1.10.

   a. See *Configuring ovictex.properties* on page 27 to configure `ovictex.properties`.

   b. See *Configuring File ovictexInternal.properties* on page 27 to configure `ovictexInternal.properties`.

   c. See *External Helpdesks* on page 27 to configure external Helpdesks.

   d. See *Configuring FieldMapping.xml* on page 27 to configure `FieldMapping.xml`.

   Or you can copy parameter values from the backup configuration files to configure SMSSMEX V1.10. Do not just copy and replace `ovictex.properties` and `FieldMapping.xml` because these files have been updated in SMSSMEX V1.10.

6. Deploy the `ovictex.war` file on WebLogic.

   See *Deploying on WebLogic* on page 28 to deploy the `ovictex.war` file on WebLogic.

7. Start SMSSMEX V1.10 and the WebLogic server.

Task 22: Adding instance in SMIS and configuring parameters

See *Adding Instance in SMIS and Configuring Parameters* on page 59 for more information.

9 Licensing

This chapter describes licensing.

License Types

The following license types are available:

- InstantOn license provides full access to all features for 60 days.
- Permanent license is node-locked (restricted to a range of IP addresses).

Autopass License Management

Autopass License Management is a tool for license management of HP products.

1. Start Autopass License Management.
   a. For Windows (Autopass is by default installed under `C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC`):

      `"<OvLIC_Install_Path>\demo\hpovliccli.bat" -gui "%SMSSMEX_HOME%\bin\SMSSMEX_pdf.txt"`

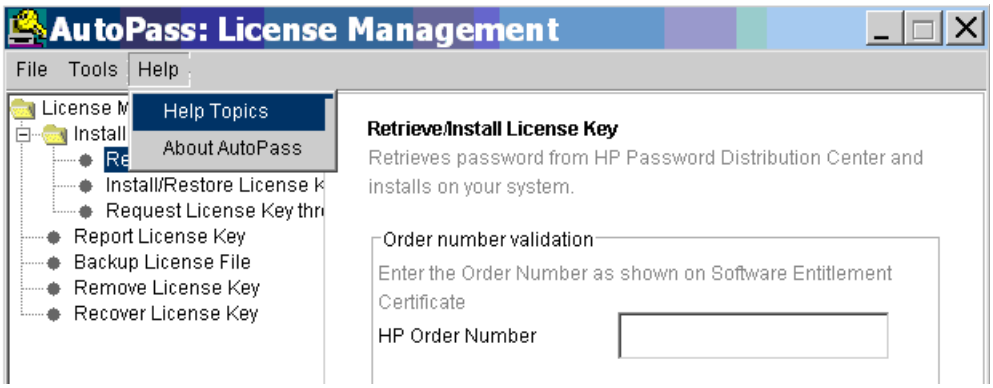
   b. For HP-UX:

      `/bin/sh/opt/OV/HPOvLIC/demo/hpovliccli.sh -gui /opt/HP/SMSSMEX/bin/SMSSMEX_pdf.txt`

   c. For Linux:

      `/opt/OV/HPOvLIC/demo/hpovliccli.sh -gui /opt/HP/SMSSMEX/bin/SMSSMEX_pdf.txt`

2 To install or remove the license, refer to the Autopass help (click **Help** on the toolbar of Autopass License Management and click **Help Topics** from the menu).

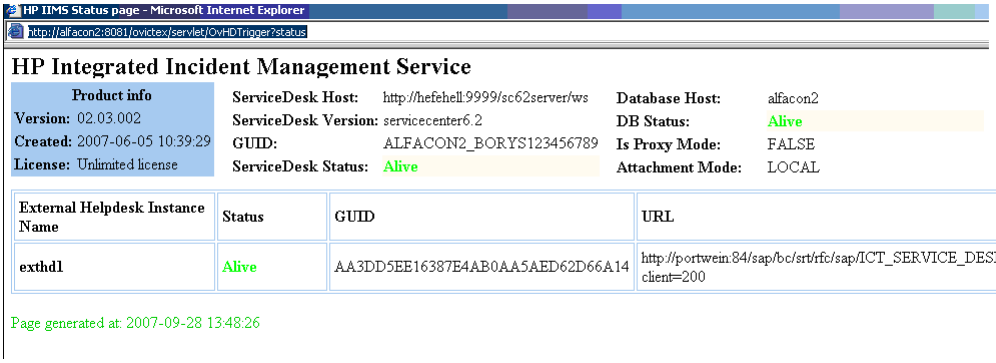


10 Status Page

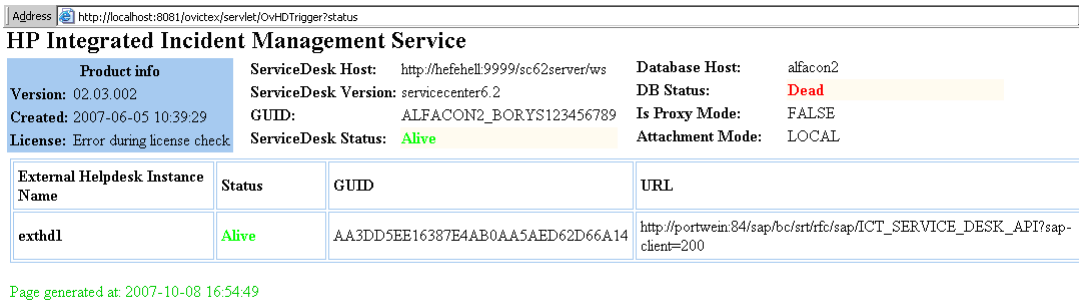
The HP Incident Management Service provides a comprehensive overview of the status of the incident exchange systems and services and provides extensive information for troubleshooting. The URL of the status page is

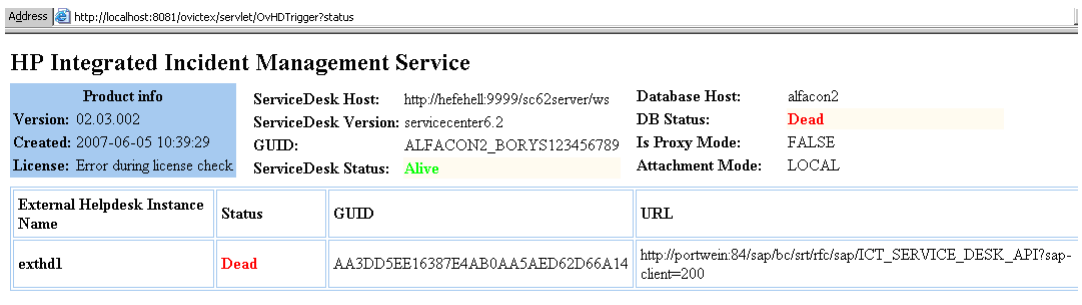
`http://<hostname>:<port>/ovictex/servlet/OvHDTrigger?status`

The following is an example status page.



When a service becomes unavailable, the status changes from `Alive` to `Dead`.





11 Troubleshooting

This chapter describes how to troubleshoot common problems. The checker tool (see Verifying Configuration on page 26) is a good aid for troubleshooting.

checker.bat and encryptPasswords.bat Fail

Problem

The exception “Class not found” appears in the console when running **`checker.bat`** or **`encryptPasswords.bat`**.

Cause

The library files that checker requires were not extracted to the required Tomcat.

Solution

1. Run **`setup startup`**. Tomcat extracts `ovictex.war` and a copies the required jar files.
2. Restart Tomcat.

Incident not Sent to SAP AGS

Problem

Incident is not sent to SAP AGS when using a newly configured priority in Solution Manager.

Cause

Incidents that have set new priorities in Solution Manager can not be sent to SAP AGS (only default priorities can be sent).

Solution

`fieldMapping.xml` maps to default priorities.

java.lang.OutOfMemoryError

Cause

Too many incidents with big attachments are exchanged simultaneously.

Solution

Increase the Java Virtual Machine heap size in `catalina.bat` (Tomcat).

```
set JAVA_OPTS=-Xms512m -Xmx1024m
```

Record in EventIn is not Executed

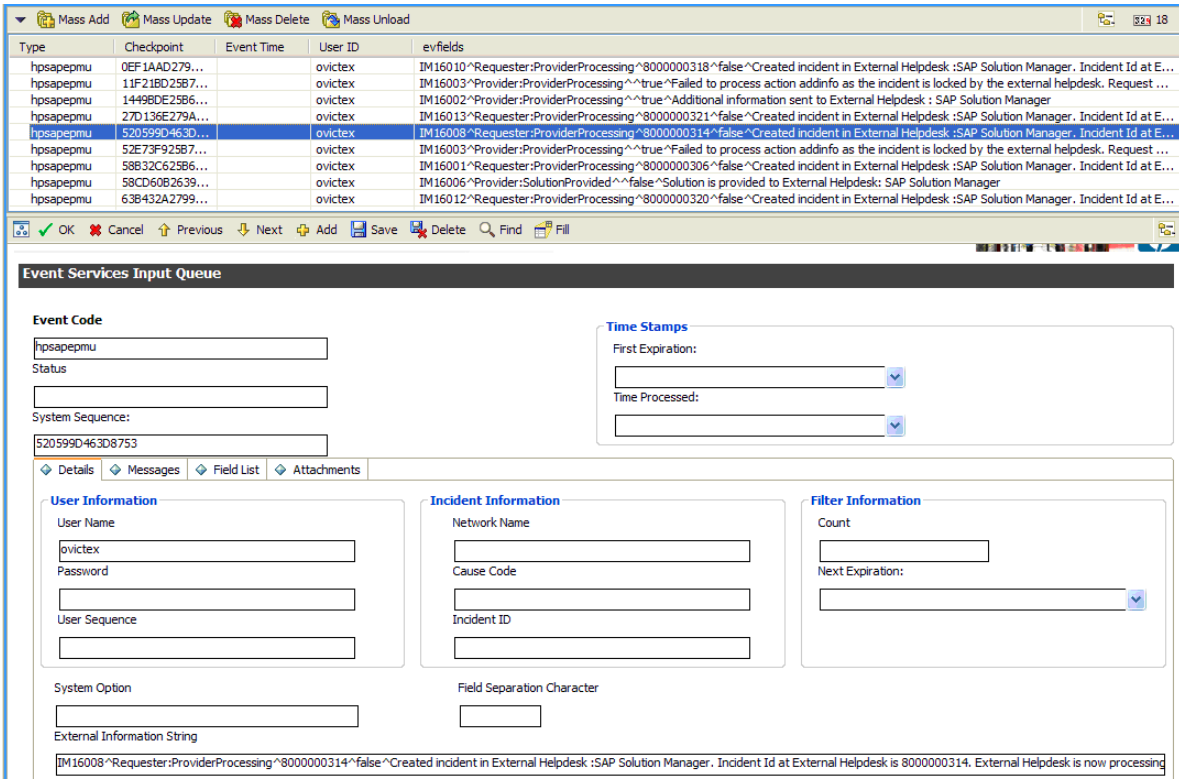
Problem

The record in table `EvenIn` is not executed. After Service Manager sends the incident to Solution Manager, the process is finished, but the following problems occur:

- Integration buttons for the incident are not shown correctly.
- Field `hidden.meta.info` is not updated.

Cause

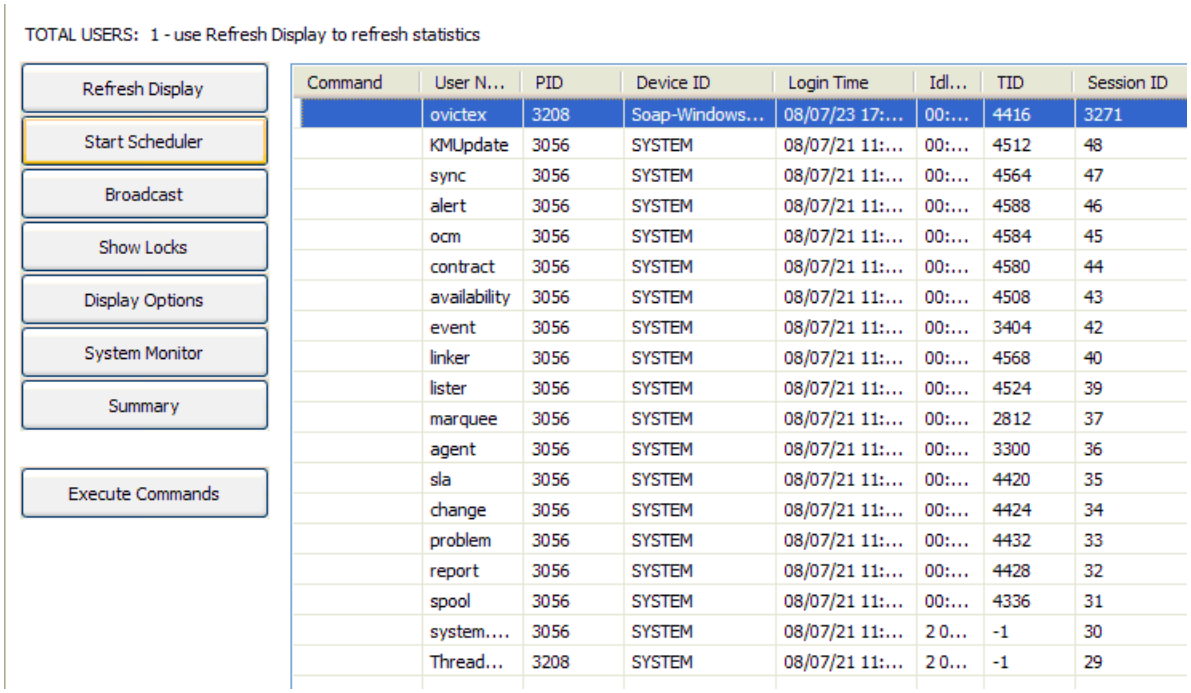
The Event In process threads are not started when the Service Manager server starts, so in the Input Events window (**Tailoring** → **Event Services** → **Input Events**) the input events are not handled (as shown in the following diagram).



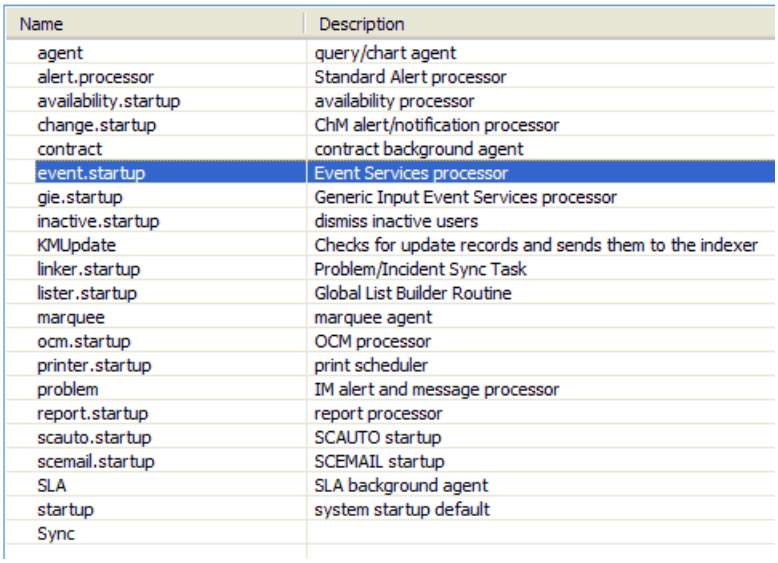
Solution

To handle the input events, start the event in process threads.

1 Go to **System Status**



2 Click **Start Scheduler.**



3 Start `event.startup` and dependent process threads.

Incident Update or Process Action Fails

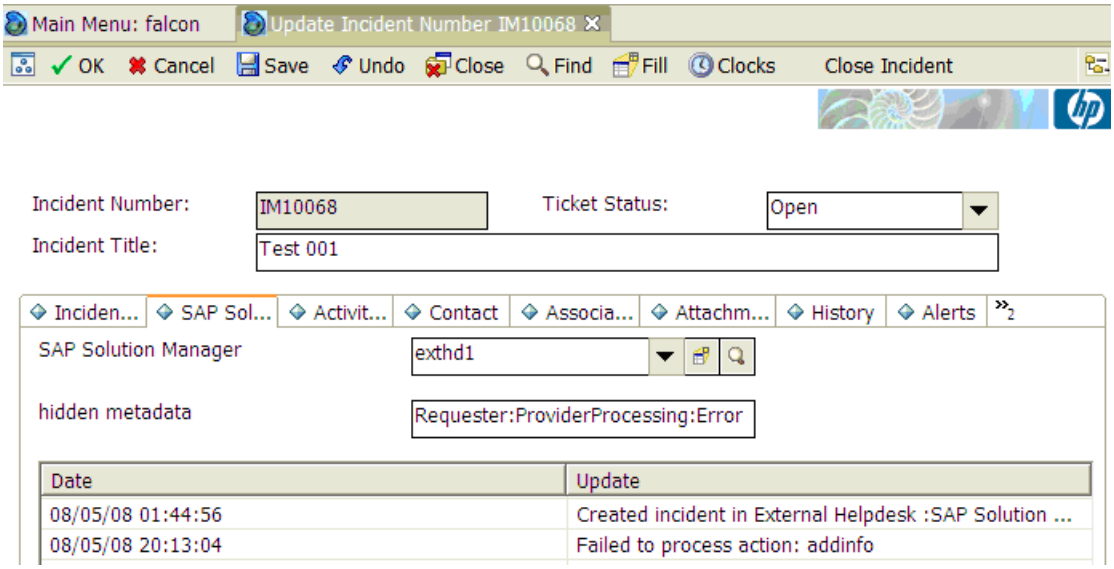
Problem

Some incidents are not exchanged between Service Manager and SAP Solution Manager. The log file or console message of SMSSMEX displays WARN or FATAL level information as described below.

1 Service Manager sends an incident to SAP Solution Manager:

```
WARN  com.hp.ov.ictex  - Failed to process action addinfo as the incident
is locked by the external helpdesk. Request will be sent again later.
```

The following alert may appear in Service Manager:



2 SAP Solution Manager updates the incident to Service Manager.

```
DEBUG com.hp.ov.ictex  - Failed to update incident. id:IM10068
DEBUG com.hp.ov.ictex  - Response code = 3. Probably an Incident: IM10068
is locked.
FATAL com.hp.ov.ictex  - Saving of incident failed. Received Message from
ServiceCenter:  Resource Unavailable
null
FATAL com.hp.ov.ictex  - An error occured while processing incident ID
IM10068. Message: Resource Unavailable
null
DEBUG com.hp.ov.ictex  - An error occured while processing incident ID
IM10068. Message: Resource Unavailable
null
com.hp.ov.ictex.ovhdaccess.OvHDException: Resource Unavailable
null
at com.hp.ov.ictex.ovhdaccess.servicecenter.Incident.save(Unknown Source)
at
com.hp.ov.ictex.exthdrequesthandler.OvictexServer.updateIncident(Unknown
Source)
...
```

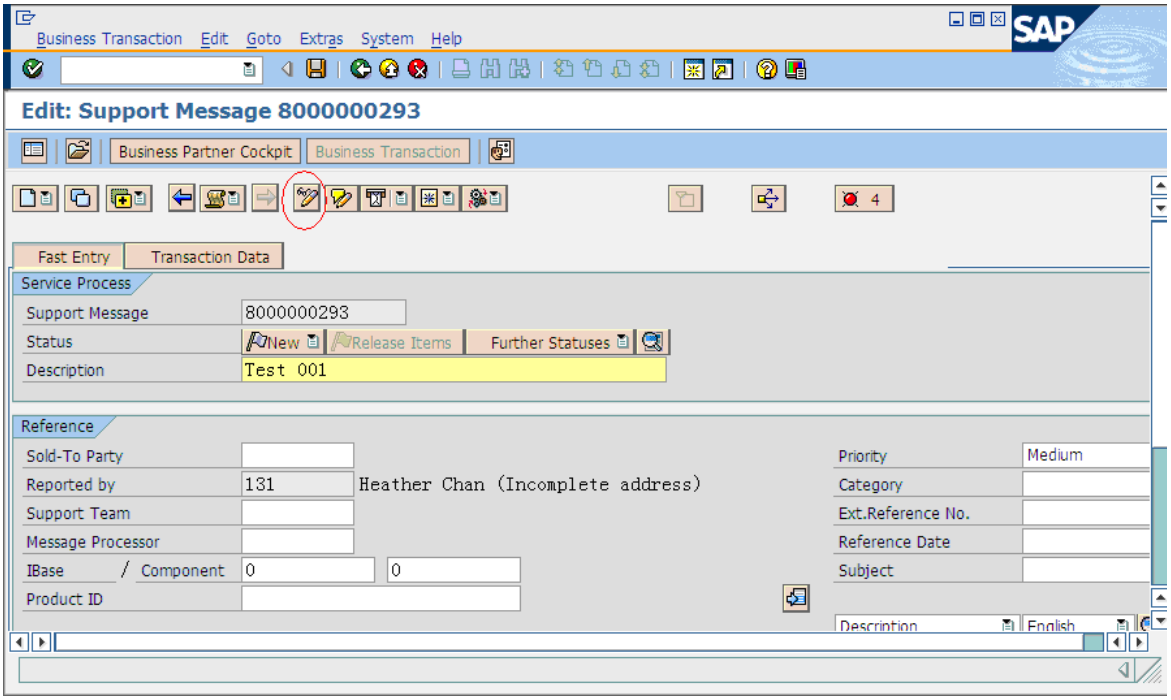
Cause

The incident in HP Service Manager or SAP Solution Manager is locked:

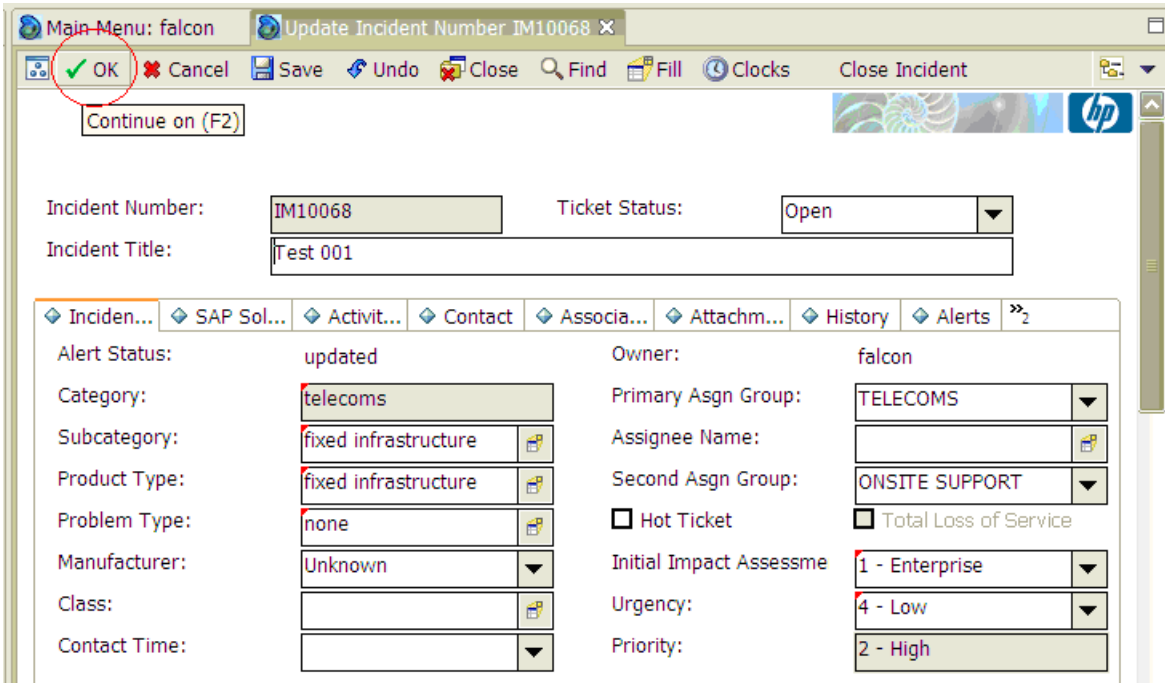
1. In SAP Solution Manager, if the user does not click the button **Display/Change Trans.** to release an incident write lock, HP Service Manager can not update or send a message to SAP.
2. In HP Service Manager, if the user does not click **OK** to release an incident write lock in time, the incident maintains the “Updating” status and no message from SAP can be accepted (until the status changes).

Solution

In SAP Solution Manager always click **Display/Change Trans.** after finishing or updating an activity.



In HP Service Manager click **OK** after finishing or updating an activity.



Information is not Updated in SAP Solution Manager

Problem

An open support message is not changed after synchronization from Service Manager to SAP.

Cause

SAP solution manager does not refresh the support message automatically.

Solution

In SAP GUI, exit from the current transaction and execute transaction **`crmd_order`**.

A Incident Exchange Details

Database Tables

The database tables required to operate the exchange service are created with the SQL scripts `create_tables_oracle.sql` or `create_tables_sqlserver.sql`.

Table 6 Database tables required for exchange service

Table	Field	Description
systemguid	systemguid	Unique system web service GUID
tasklist	ovhdid	ID of incident that triggered the action
	action	Action for the incident (state transitions of status diagram). Can be ProcessIncident, AddInfo, AcceptIncidentProcessing, RejectIncidentSolution, VerifyIncidentSolution, or CloseIncident.
	startTimestamp	Creation timestamp of entry.
	enqueueTimestamp	Timestamp for ordering of tasks. Initial value is startTimestamp.
	earliestReadyTimestamp	Timestamp that specifies the earliest time when this entry can be processed. Empty means immediately. Task becomes ready only after this time.
	state	Task state. Can be 1=READY or 2=INPROCESS (task is processed already).
	tries	Number of attempts to complete this task.
	guid	GUID of the task to delete the correct entry in database.
	sapid	Name of external help desk instance that incident is exchanged with.

Table 6 Database tables required for exchange service (cont'd)

runtimedata	incidentguid	GUID of exchanged incident
	ovhdincidentid	ID of incident in helpdesk managed by web service
	exthdincidentid	ID of incident in the external helpdesk
	requesterguid	System GUID of requester helpdesk for that incident
	providerguid	System GUID of provider helpdesk for that incident
	metadata	Incident state in statement diagram and role the ServiceDesk has for this incident (Requester or Provider). Stored in the same format used for the `Hidden_Meta_Data` field in ServiceDesk (such as `Requester:RequesterProcessing`, `Provider:SolutionProvided`).
	infologid	Reference to multiple entries in runtimedata_infolog.
	attachmentid	Reference to multiple entries in runtimedata_attachments.
	lastchange	Timestamp of last change of entry.
runtimedata_infolog	infologid	Key referenced from runtimedata.
	infologblock	Number of infolog block sent already.
runtimedata_attachments	attachmentid	Key referenced from runtimedata.
	filename	Filename of an attachment for incident.
	attachmentguid	GUID for attachment (also known by external helpdesk) to delete attachment.

Tools

There are several configuration tools in the installation \bin directory. Tool scripts are available for Windows (.bat) and Unix (.sh):

- `encryptPasswords` encrypts the passwords in the configuration file. All properties ending with `.password` must be configured with this tool. Use `-global` or <*instance key*> as a parameter.
  - `global`
    Encrypt a password in the global properties file (`ovictex.properties`). For example:

    ```
    encryptPasswords.bat -global
    ```

  - <*instance key*>
    Encrypt a password in the configuration file of a specific instance. For example:

    ```
    encryptPasswords.bat exthd
    ```

- `setup` is the setup script for Tomcat start/stop. For Tomcat

- Start: `Setup startup`
- Shutdown: `Setup shutdown`
- Start with debug mode: `Setup debug startup`

- `checker` checks the configuration in `ovictex.properties` and Service Manager configuration (see Verifying Configuration on page 26 for more information) .

Field Mapping Configuration

Incident exchange web service exchanges incident data as XML documents between Service Manager and the external HelpDesk SAP Solution Manager. Incident exchange transforms the incident data in Service Manager to an XML message for SAP Solution Manager, and transforms Solution Manager data to an XML message for Service Manager. The transformation maps the field name in Service Manager to XML elements in Solution Manager while taking into account the following:

- Field names in Service Manager are usually different from the message element name.
- Service Manager field data type can differ from the message element data type.
- Not all message elements have corresponding data fields in Service Manager. Such fields are usually combined into a single log field called `Journal`.
- Some fields also require value mapping. For example, the possible values for the `Priority` field in Service Manager are **1 - Critical**, **2 - High**, **3 - Average**, **4 - Low**. The Solution Manager Priority can be **5**, **4**, **3**, **2**, **1**. These values must be specified in the `FieldValueMapping` configuration.
- Service Manager can assign customized fields to an Incident. These fields can be mapped to message elements.

A declarative field mapping file defines the mapping outlined above and

- Enables the exchange of incident data between two helpdesks with reduced code size (the same code can handle any number of fields)
- Improves flexibility (mapping can be changed without changing code)
- Improves extensibility and customizability (a deployment-specific mapping can be added without changing code)
- Used to map incident data with an external helpdesk other than Solution Manager

Types of Mapping

The mapping file supports field mapping and field value mapping. Field mapping is simple (XML message element is a single value) or composite (multiple values such as an array).

Structure of FieldMapping XML file

The field mapping configuration is related to the `ICT_SERVICE_DESK_API` WSDL scheme defined by SAP Solution Manager. The mapping consists of field mapping and value mapping.

Field mapping includes:

- `IctHead`

- IctIncidentAttachment
- IctIncidentSapNotes
- IctIncidentSolutions
- IctIncidentUrls
- IctIncidentStatement
- IctIncidentAdditionalInfo

The following is a mapping file example:

```
<FieldMapping ExtHDField="IctHead/AgentId" >
  <OutOvHDField>AssigneeName</OutOvHDField>
  <OutDataType>Person</OutDataType>
  <InOvHDField>AssigneeName</InOvHDField>
  <InDataType>Person</InDataType>
</FieldMapping>
```

In the above example:

- Element IctHead/AgentId of SAP Solution Manager (sub-element AgentId of top level element IctHead) maps to the field AssigneeName exposed by the Service Manager IncidentManagement Web Service.
- Data types for the IN and OUT exchange modes are specified in the InDataType and OutDataType tags.
- Person type indicates that the Exchange must convert incoming data (to/from the Service Manager) to/from an internal Person type that corresponds with the IctIncidentPerson type of the SAP SolutionManager web service.
- InDataType and OutDataType tags declare types on the Service Manager side.

Composite Field Mapping

Composite field mapping maps a message element to a OvHD field depending upon the value of a sub-element (key) of the element ExtHDKeyField (OvHD and ExtHD are old terms; in this document, OvHD correspond to HP Service Manager and ExtHD correspond to SAP Solution Manager). A different value for the key defines mapping to a different Service Manager field. The following is a composite field mapping example.

```
<CompositeFieldMapping  ExtHDField="IctIncidentStatement"
  ExtHDKeyField="IctIncidentStatement/TextType">
<!-- For exchanging information log -->
<FieldMapping ExtHDField="IctIncidentStatement/Text" >
   <InDataType>InformationLog</InDataType>
   <OutDataType>InformationLog</OutDataType>
   <KeyFieldOutVal>SU99</KeyFieldOutVal>
   <KeyFieldInVal>SU99</KeyFieldInVal>
</FieldMapping>
<!-- for exchanging Solution Provided   -->
<FieldMapping ExtHDField="IctIncidentStatement/Text" >
  <InOvHDField>Resolution</InOvHDField>
  <OutOvHDField>Resolution</OutOvHDField>
  <KeyFieldOutVal>SU99</KeyFieldOutVal>
  <KeyFieldInVal>SU01</KeyFieldInVal>
</FieldMapping>
<!-- for exchanging CustomText01 (as example)  -->
```

```
<FieldMapping ExtHDField="IctIncidentStatement/Text" >
  <InOvHDField>CustomText01</InOvHDField>
  <OutOvHDField>CustomText01</OutOvHDField>
  <KeyFieldOutVal>SU99</KeyFieldOutVal>
  <KeyFieldInVal>SU77</KeyFieldInVal>
</FieldMapping>
<!-- For sending custom fields from OVHD to external HD create an entry as
the example below. Replace the place holder strings as per your
configuration  -->
<!--
  <FieldMapping ExtHDField="IctIncidentStatement/Text" >
     <OutOvHDField>USER_VISIBLE_FIELDNAME_FOR_THAT_CUSTOM_FIELD
     </OutOvHDField>
     <KeyFieldOutVal>TEXT_TYPE_AS_DEFINED_BY_USER_FOR_THIS_FIELD
     </KeyFieldOutVal>
  </FieldMapping>
-->
</CompositeFieldMapping>
```

Element `IctIncidentStatement/Text` is mapped to the information log if the key element `IctIncidentStatement/TextType` is **`SU99`** or to `Resolution` field if the key element is **`SU01`** (for an incoming message).

This is used when a message has multiple occurrences of the same element that have different sub-element values. The sub-element is referred to as the key field. In the example above the `IctIncidentStatement/TextType` element is the key field. For a composite field mapping, every instance of `FieldMapping` has a unique `KeyFieldInVal`.

Field Value Mapping

Field value mapping maps the values of a message element to the corresponding value of an OvHD field. The following is an example.

```
<FieldValueMapping Id="IctHead/Priority">
  <ValueMapping OvHDValue="4" ExtHDValue="5"/>
  <ValueMapping OvHDValue="4" ExtHDValue="4"/>
  <ValueMapping OvHDValue="3" ExtHDValue="3"/>
  <ValueMapping OvHDValue="2" ExtHDValue="2"/>
  <ValueMapping OvHDValue="1" ExtHDValue="1"/>
</FieldValueMapping>
```

Since both helpdesks priority lists can be configured, check the actual values in the field value mapping.

Field Mapping Schema

The RelaxNG Compact Schema of the mapping file is shown below.

```
default namespace =
  "http://schemas.hp.com/openview/incidentExchange/mapping"
start =
  element IncidentExchMapping {
    attribute targetNamespace { xsd:anyURI },
    element FieldMappings {
      (FieldMapping
```

```
        | element CompositeFieldMapping {
            attribute ExtHDField { string },
            attribute ExtHDKeyField { string },
            FieldMapping+
          })+
      } &
      element ValueMappings {
        element FieldValueMapping {
          attribute Id { string },
          element ValueMapping {
            attribute ExtHDValue { string },
            attribute OvHDValue { string }
          }+
        }
      }
    }
  FieldMapping =
    element FieldMapping {
    ## field accessor in XML document using XPath like notation. Example:
    ## ExtHDField="IctHead/AgentId"
      attribute ExtHDField { string },
      attribute ValueMappingId { string }?,
      (element InOvHDField { string } &
      (element DefaultOutOvHDField { string }
       | element OutOvHDField { string })? &
      element InDataType { "InformationLog" | "Priority" | "Date" |
        "Attachment"  | "OvCISearchKey" }? &
      element OutDataType { "Person" | "Priority" | "Date" | "Attachment" |
        "OvCISearchKey" }? &
      element KeyFieldOutVal { string }? &
      element KeyFieldInVal { string }? )
    }
```

The schema elements are described in the following table.

Table 7 Schema element functionality

Schema element	Function
IncidentExchMapping	Top-level element of the mapping schema.
FieldMappings	Container element for all `FieldMapping` and `CompositeFieldMapping` elements.
ValueMappings	Container element for `FieldValueMapping` elements.
FieldMapping	Maps a message element to an `OvHD` field and includes type information for storing to and loading from OvHD. Optionally contains a reference to a `FieldValueMapping` element through attribute `ValueMappingId`. The value of this attribute must match the value of attribute `Id` in a `FieldValueMapping` element. When this reference is present, the information in the `FieldValueMapping` must be used to map field value.

Table 7 Schema element functionality (cont'd)

Schema element	Function
CompositeFieldMapping	Maps a message element to a `OvHD` field depending upon the value of a sub-element (key) of the element. A different key value defines mapping to a different `OvHD` field. The `keyFieldInVal` must be unique for each individual field mapping within a composite field mapping.
ValueMapping	Maps the OvHD value of a message element to an ExtHD value.
ExtHDField	Field accessor in XML message document in XPATH like expression that identifies a specific field of exchanged incident information.
InOvHDField	Indicates the `OvHD` field name where information received from the external helpdesk is written for a specific `ExtHDField`.
OutOvHDField	Indicates the `OvHD` field name whose value is sent to the external helpdesk for a specific `ExtHDField`.
DefaultOutOvHDField	If this element appears in a field mapping then the value of this element is taken as the default value sent to the external helpdesk for a specific `ExtHDField`. For example, if a mapping `DefaultOutOvHDField` is specified as `DefaultUserId` and `OutDataType` is specified as `Person`, the default user ID will be sent to the external helpdesk for a specific `ExtHDField`.
InDataType	Datatype for storing the field value to `OvHD`.
OutDataType	Datatype for loading the field value from `OvHD`.
InDataType and OutDataType	Specifies the method to call for reading/writing information from/ to the incident using the OvHDAccess layer. InDataType and `OutDataType` are optional elements. If not specified, then the field types are assumed to be String. Otherwise the following data types can be specified: • Priority: Priority of an incident • Date: A date field • Attachment: Refers to an attachment • Person: Indicates that the information is a person detail.
OvCISearchKey	Indicates the information is used as a search key for CI in OvHD.
InformationLog	Applicable only for InDataType. Indicates the information should be appended to the Information Log.
KeyFieldInVal	Value stored in OvHD for the element used as the key field.
KeyFieldOutVal	Value sent to ExtHD for the element used as the key field.

Default Field Mapping File and Customization

Prerequisites

SMSSMEX operates with Service Manager based on the extended IncidentManagement Web Service and supports only the fields listed below (exposed in the Service Manager IncidentManagement WS).

Table 8 SMSSMEX supported fields

Field	Type	Field	Type
IncidentID	Text	Subcategory	Text
Category	Text	SLAAgreementID	Decimal
OpenTime	Datetime	PlannedEnd	Datetime
OpenedBy	Text	SiteCategory	Text
PriorityCode	Text	ProductType	Text
Severity	Text	ProblemType	Text
UpdatedTime	Datetime	ResolutionFixType	Text
PrimaryAssignment Group	Text	UserPriority	Text
ClosedTime	Datetime	Solution	Text
ClosedBy	Text	InitialImpact	Text
ClosureCode	Text	CustomText01	Text
ConfigurationItem	Text	CustomText02	Text
Location	Text	CustomText03	Text
IncidentDescription		CustomText04	Text
Resolution	Resolution	CustomText05	Text
AssigneeName	Text(OperatorID)	CustomText06	Text
Contact	Text(ContactID)	CustomText07	Text
JournalUpdates		CustomText08	Text
AlertStatus	Text	CustomText09	Text
ContactLastName	Text	CustomText10	Text
ContactFirstName	Text	SapSid	Text
Company	Text	SapClient	Text
BriefDescription	Text	SapInstallationNum ber	Text

Table 8 SMSSMEX supported fields (cont'd)

TicketOwner	Text	HiddenMetaData	Text
UpdatedBy	Text	IsIncidentExchange	Boolean
IMTicketStatus	Text	attachments	Attachments

Adding Fields to fieldMapping.xml

The default field mapping file (provided with the incident exchange web service) does not include all fields from the web service and can be extended. Any additional field mapping can be included in section `IctIncidentStatement`. The following is an example:

```
<FieldMapping ExtHDField="IctIncidentStatement/Text">
  <OutOvHDField>SC_WS_FIELDNAME</OutOvHDField>
  <KeyFieldOutVal>SOLMAN_FIELD_TYPE </KeyFieldOutVal>
</FieldMapping>
<FieldMapping ExtHDField="IctIncidentStatement/Text">
  <OutOvHDField>CustomText09</OutOvHDField>
  <KeyFieldOutVal>SU99 </KeyFieldOutVal>
</FieldMapping>
```

In the above example, the custom field defined in Service Manager is sent to the external HD, so `KeyFieldOutVal` is defined at the external helpdesk. No `InOvHDField` or `KeyFieldInVal` is specified since the example only sends to the external helpdesk.

IN/OUT data exchange requires definition of `IN` and `OUT`:

```
<FieldMapping ExtHDField="IctIncidentStatement/Text">
  <OutOvHDField>SC_WS_FIELDNAME1</OutOvHDField>
  <InOvHDField>SC_WS_FIELDNAME2</InOvHDField>
  <KeyFieldOutVal> SOLMAN_FIELD_TYPE1 </KeyFieldOutVal>
  <KeyFieldInVal> SOLMAN_FIELD_TYPE2 </KeyFieldInVal>
</FieldMapping>
```

In this example if the values of `SC_WS_FIELDNAME1` and `C_WS_FIELDNAME2` are the same, then the `OvHD` field is overwritten when information is sent from external helpdesk (1:1 field synchronization). For example:

```
<FieldMapping ExtHDField="IctIncidentStatement/Text">
  <OutOvHDField>CustomText09</OutOvHDField>
  <InOvHDField>CustomText09</InOvHDField>
  <KeyFieldOutVal SU01</KeyFieldOutVal>
  <KeyFieldInVal>SU01</KeyFieldInVal>
</FieldMapping>
```

In the following example, `CustomText08` updates field `ZZ08` in SAP Solution Manager, but `ZZ08` updates `CustomText09` in Service Manager (does not overwrite CustomText08).

```
<FieldMapping ExtHDField="IctIncidentStatement/Text">
  <OutOvHDField>CustomText08</OutOvHDField>
  <InOvHDField>CustomText09</InOvHDField>
  <KeyFieldOutVal ZZ08</KeyFieldOutVal>
  <KeyFieldInVal>ZZ08</KeyFieldInVal>
</FieldMapping>
```

Additional Information

Section `IctIncidentAdditionalInfo` defines synchronization of CIs between SAP Solution Manager and Service Manager and defines the method for sending SAP Attributes from SAP Solution Manager.

The first part of the mapping describes CI mapping handling and must not be changed.

```
<FieldMapping ExtHDField="IctIncidentAdditionalInfo/AddInfoValue" >
  <OutDataType>OvCISearchKey</OutDataType>
  <InDataType>OvCISearchKey</InDataType>
  <KeyFieldOutVal>SAPSystemID</KeyFieldOutVal>
  <KeyFieldInVal>SAPSystemID</KeyFieldInVal>
</FieldMapping>
<FieldMapping ExtHDField="IctIncidentAdditionalInfo/AddInfoValue" >
  <OutDataType>OvCISearchKey</OutDataType>
  <InDataType>OvCISearchKey</InDataType>
  <KeyFieldOutVal>SAPSystemClient</KeyFieldOutVal>
  <KeyFieldInVal>SAPSystemClient</KeyFieldInVal>
</FieldMapping>
<FieldMapping ExtHDField="IctIncidentAdditionalInfo/AddInfoValue" >
  <OutDataType>OvCISearchKey</OutDataType>
  <InDataType>OvCISearchKey</InDataType>
  <KeyFieldOutVal>SAPInstNo</KeyFieldOutVal>
  <KeyFieldInVal>SAPInstNo</KeyFieldInVal>
</FieldMapping>
```

The following two attributes are used only when Solution Manager forwards an Incident to SAP Solution Manager.

```
<FieldMapping ExtHDField="IctIncidentAdditionalInfo/AddInfoValue" >
  <InDataType>InformationLog</InDataType>
  <KeyFieldInVal>SAPIncidentID</KeyFieldInVal>
</FieldMapping>
<FieldMapping ExtHDField="IctIncidentAdditionalInfo/AddInfoValue"
   ValueMappingId="IctIncidentAdditionalInfo/AddInfoValue
   /SAPIncidentStatus" >
  <InDataType>InformationLog</InDataType>
  <KeyFieldInVal>SAPIncidentStatus</KeyFieldInVal>
</FieldMapping>
```

The only attributes that do not have read-only status in the SAP Solution Manager are CI attributes, allowing IN-mode mapping (from SAP Solution Manager to Service Manager). The following table defines the available attributes:

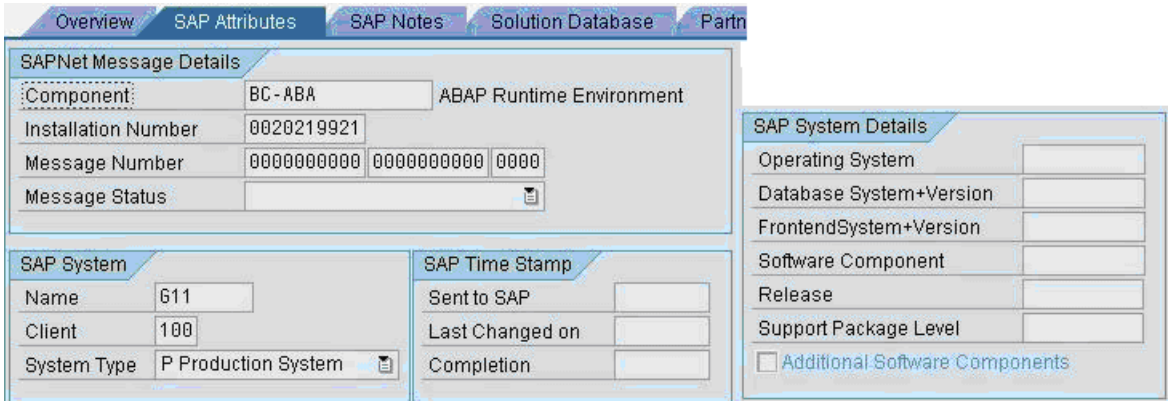
Table 9 Attribute Types of the SAP Solution Manager

AttributeType	Description
SAPComponent	SAP Component (e.g. SV-SMG-SUP)
SAPSystemID	SAP System ID
SAPSystemClient	Client of SAP System
SAPCategory	Category of the Incident
SAPSystemType	SAP System Type

Table 9 Attribute Types of the SAP Solution Manager

SAPInstNo	SAP Installation Number
SAPSubject	Subject of the Incident
SAPOperatingSystem	Operating System of SAP System
SAPDatabase	Database of SAP System
SAPFrontend	Frontendsystem and Version
SAPSoftwareComponent	Software Component
SAPSoftwareComponentRelease	Software Component Release
SAPSoftwareComponentPatch	Software Component Patch
SAPIncidentID	ID of the Incident at SAP (when forwarded to SAP)
SAPIncidentStatus	Status of the Incident at SAP (when forwarded to SAP)

In the SAP GUI most attributes are in the SAP Attributes tab.



The following example writes all incoming additional values of type `SAPDatabase` to the Journal in Service Manager:

```
<FieldMapping ExtHDField="IctIncidentAdditionalInfo/AddInfoValue" >
  <InDataType>InformationLog</InDataType>
  <KeyFieldInVal>SAPDatabase</KeyFieldInVal>
</FieldMapping>
```

The following example updates field `CustomText03`.

```
<FieldMapping ExtHDField="IctIncidentAdditionalInfo/AddInfoValue" >
  <InOvHDField>CustomText03</InOvHDField>
  <KeyFieldInVal>SAPDatabase</KeyFieldInVal>
</FieldMapping>
```

Changeable Mappings

The following mappings can be modfied.

Table 10 Changeable mappings

Mapping	Description
IctHead/AgentID IctHead/ReporterID IctIncidentAttachment/PersonId IctIncidentStatement/PersonId	`OutOvHDField/InOvHDField` field name can be modified if the replacement field contains the ID of a Contact joined with a contact table that is exposed via ConfigurationManagement Web Service (defined in the default configuration). The `AssigneeName` field contains the `operator` name of Service Manager instead of the `contacts` name.
IctHead/ShortDescription	`OutOvHDField/InOvHDField` can be modified with any text field from Service Manager.
IctHead/RequestedEnd	Can be modified with any datetime field in the Service Manager.

Required Mappings: The following mappings are required and must not be changed.

- `IctHead/Priority` (the value mapping for this field mapping can be changed)
- `IctIncidentSapNotes/item`
- `IctIncidentSolutions/item`
- `IctIncidentUrls/item`
- `IctIncidentAdditionalInfo/AddInfo`Value (first 3 mappings)

Person Synchronization Details

SAP Solution Manager to Service Manager

Persons sent from SAP Service Manager can be mapped to person fields in Service Manager. When Person details are received, the corresponding contact record is found in Service Manager by querying the Configuration Management Web Service. The resolved contact ID must be set in the mapped field. The exchange web service describes persons with the following fields:

- Sex
- First name
- Last name
- Telephone
- Mobile phone
- Fax
- Email

Fields that are used to find persons in Service Manager:

- Email
- First name
- Last name

Persons are searched by all three fields. If no matching person is found in Service Manager or duplicates are found, then a notification is added to the Journal. For example, an empty email causes the following message in response to Journal updates:

```
Warning! Contact can not be found. Firstname,Lastname,Email fields should
not be empty. Invalid contact: FirstName: "Nicholas" LastName: "Brown"
Phone number: "(770) 954-4588" Fax number: "(770) 954-4590" …
```

SMSSMEX does not create Persons or Contacts. An operator-type lookup is enforced only for the `AssigneeName` field.

Mapping from Service Manager to SAP Solution Manager is performed in the same way. The ID of the `Person` field in the Service Manager is used to make an additional call to Configuration Management WS to get all details about the Person. The collected data is forwarded to the Solution Manager. In SAP Solution Manager the ID of the Person is checked. If the ID is

- Known: Solution Manager assigns an existing record to the Incident.
- Not known: Solution Manager tries to resolve a Person via the email field. If this is not possible, a new Person is created.

SMSSMEX Version

To find out the version of the SMSSMEX service in Tomcat, do one of the following:

- Open `<`*`SMSSMEX_installDir`*`>\tomcat\webapps\ovictex\WEB-INF\lib\ovictex.war` with a zip tool. The war file `MANIFEST.MF` file contains the version information.
- Go to the Status page.

B Installing and Configuring SAPCRYPLIB

To install SAPCRYPLIB (see **https://service.sap.com/sap/support/notes/510007**) do the following:

1. Download `SAPCRYLIB` from the website "SAP Download Area - SAP Cryptographic Software" at **https://websmp101.sap-ag.de/~form/handler?_APP=00200682500000000917&_EVENT=DISPLAY**.
2. Use `sapcar.exe` to extract the SAR file:

   ```
   sapcar -xvf sarfile_name
   ```

3. Copy the extra files to `\usr\sap\[Instance folder]\DVEBMGS00\exe`.
4. In transaction **`/nrz10`** in the `Profile` field, select the profile with `prof.type` of "Instance profile".
5. Select **Extended maintenance** in Edit Profile.
6. Click **Change**.
7. Add the following parameters:

   ```
   ssf/name          = SAPSECULIB
   ssf/ssfapi_lib    = $(DIR_EXECUTABLE)\sapcrypto.dll
   ```

8. Restart the system.
9. Go to transaction **`/nsmicm`**.
10. Select the menu entry **GOTO** and select **Services** or press **`SHIFT+F1`**.
11. If the HTTPS port is not listed, then configure the profile. Add or change the following parameter:

    ```
    icm/server_port_2 PROT=HTTPS,PORT=[SSL Port]
    ```

12. In transaction **`/nsmicm`** select from the **Administration** → **ICM** → **Restart** → **Yes** to restart ICM.

C Logging

The following describes the location of log files.

- Windows: If you start SMSSMEX from
  - **`setup -startup`**

    `%SMSSMEX_HOME%/logs/smssmex.log.<date>`
  - Tomcat

    `%SMSSMEX_HOME%/tomcat/logs/smssmex.log.<date>`
- Unix: If you start SMSSMEX from
  - **`setup.sh -startup`**

    `%SMSSMEX_HOME%/logs/smssmex.log.<date>`
  - Tomcat

    `%SMSSMEX_HOME%/tomcat/logs/smssmex.log.<date>`

D Deploying Button Icons

SMSSMEX enhances the functionality of Service Manager by adding some buttons in incident form to trigger message exchange related actions. The icons for the buttons are provided additionally in the release package (under `<SMSSMEX1.10 Release Package>\icons` folder). You can deploy them to the Service Manager Client manually.

Service Manager has two client applications: Windows Client and Web Client. For each of the clients, the icons should be deployed separately.

Windows Client

- SC 6.2.x

  Copy button icons from `<SMSSMEX1.10 Release Package>\icons` folder to `<Client_Home>\plugins\com.peregrine.eclipse.user_6.2.x.x\icons\obj16`.

  For example,

  `C:\Program Files\Peregrine Systems\ServiceCenter 6.2\Client\plugins\com.peregrine.eclipse.user_6.2.7.0\icons\obj16`.

  For more information, see page 110 of *ServiceCenter 6.2 Installation Guide*.

- Service Manager 9.2x / 9.3x

  Copy button icons from `<SMSSMEX1.10 Release Package>\icons` folder to `<Client_Home>\plugins\com.hp.ov.sm.client.eclipse.user_9.xx\src\resources\icons\obj16`

  For more information, see *Service Manager 9.20 Installation Guide*.

Web Client

Copy the button icons from `<SMSSMEX1.10 Release Package>\icons` folder to the following locations:

- On Service Center 6.2x, `<WebApps_Root>\sc\images\obj16` directory. For example, `C:\apache-tomcat-5.0.28\webapps\sc\images\obj16`.
- On Service Manager 9.2x and 9.3x, `<WebApps_Root>\webtier-9.20\images\obj16` directory.

E SAP System Landscape Directory Registration

System Landscape Directory is the central information repository for your system landscape (Software Catalogue). It contains information about all installable and installed components in a system landscape. This section decribes how to register this integration into System Landscape Directory.

Prerequisites

Service Landscape Directory is running.

Registering System Landscape Directory

1. Browse to the <*SMSSMEX1.10 Release Package*> and copy the `SLDReg` folder to your computer.
2. Open the `SLDReg` folder. Modify the `HPSMISystem.properties` file according to the parameter descriptions in the file. For example, update the `ComputerName` variable to the host name which is running SMSSMEX.

   ComputerName = <your computer name>
3. Run the following command to compile XML file:

   ```
   java -cp SLDReg.jar com.hp.sm.sld.XMLGenerator
   ```

   After execution, HPSMI.xml is generated.
4. Run the following command to register System Landscape Directory:

   ```
   java -cp SLDReg.jar com.hp.sm.sld.Register <SLD_HOST> <SLD_HTTPPORT>
   <UserName> <Password>
   ```

   In this command:

   - `<SLD_HOST>` is the host name of the Service Landscape Directory server.
   - `<SLD_HTTPPORT>` is the http port of the service landscape directory service.
   - `<UserName>` is the name that you use to log in to the server.
   - `<Password>` is the password that you use to log in to the server.