HP Service Manager Exchange with SAP Solution Manager

Software Version: 1.10

Service Manager Version: 7.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

Introduction 9

HP Incident Exchange 9

Existing Fragmented Incident Management Workflow 9

SAP Solution Manager Service Desk 9

HP Service Manager 10

Purpose of Document 10

Use Cases 11

Use Case 1: Incident Originates from Solution Manager 11

Use Case 2: Incident Originates from Service Manager 13

Deployment Scenarios 15

High Level Overview 15

Components 15

Support Matrix 16

Installing and Configuring SMSSMEX on Tomcat 17

Installing SMSSMEX 17

Prerequisites 17

Install SMSSMEX 17

Uninstall SMSSMEX 19

SMSSMEX Installed Files 19

Configuring Tomcat 20

Setting up Database 20

Oracle 21

MS-SQL 2005 22

MS-SQL 2008 24

Configuring ovictex.properties 24

Configuring File ovictexInternal.properties 25

External Helpdesks 25

Configuring FieldMapping.xml 26

Verifying Configuration 26

Starting/Stopping SMSSMEX 27

Installing and Configuring SMSSMEX on WebLogic 29

Installing SMSSMEX 29

Setting up Database 29

Configuring ovictex.properties 29

Configuring File ovictexInternal.properties 29

External Helpdesks 29

Configuring FieldMapping.xml 29
Verifying Configuration 29
Deploying on WebLogic 30
Customizing HP Service Manager 31
Creating a Service Manager User for Web Service 31
Importing Customizations via Unload 32
Core Unload 32
Demo Unload. 34
Customizing Demo Unload Manually. 35
Trigger URL 35
Incident Custom Fields and Web Service Exposure 35
Expose Custom Fields in Web Service Interface IncidentManagement 36
Contacts Web Service Exposure. 38
Journal Separator Line Format 39
Template 41
Incident Form 43
Status and Hidden Metadata 43
Exchange History 43
Trigger Buttons 47
Selection of External Helpdesk System 48
SAP Configuration Item handling 49
Overview 49
Implementation 49
Example Implementation via New Device Type and fill.fc 49
Implementation Alternatives and Enhancements. 65
Creating HPSAPSolutionManager Table 65
Configuring WSDL Mapping. 66
Adding New Client for SAP Solution Manager 66
Configuring SAP Solution Manager 69
Prerequisites 69
Configuring SAP Solution Manager External Service Desk Interface 70
Release Web Service 71
Assign Roles to the Communication User 72
Create HTTP Connection 73
Create a Logical Port 74
Configure Interface to SAP Solution Manager Service Desk 76
Define Value Mapping for the Service Desk Interface 76
Define Extended Interface Mapping for Service Desk 76
Get SAP Solution Manager Service Port 76
Solution Manager Tracing. 77
Enable tracing. 77
Download Trace File. 77
Configuring Security 79
Security between SAP Solution Manager and Tomcat 79
Configure SAP Solution Manager for SSL 79

Checking SAP SSL Configuration 79
Creating a Client PSE in Trust Manager. 81
Setting Up an Outgoing Connection in SAP Solution Manager 83
Set up an Incoming Connection in SAP Solution Manager. 84
Set up SSL between SAP and SMSSMEX. 86
Create Keystore and Truststore 86
Configure Tomcat SSL Use 87
Configure Property Files 87
Security Between HP Service Manager and SMSSMEX. 89
Configure HP Service Manager for SSL 89
Generate a Private/Public Key Pair for Root Certificate Authority 89
Generate a Private/Public Key Pair for Service Manager Server. 89
Generate the Client Keystore for Service Manager Client 90
Generate the Client Keystore for SMSSMEX 91
Generate the Trust-List Keystore for Service Manager Server 92
SSL Configuration in Service Manager Server 92
SSL Configuration in Service Manager Client 93
Service Manager Web Client SSL Configuration. 94
Configure SMSSMEX for SSL Communication with Service Manager. 95

Upgrading SMSSMEX. 97
Upgrading SMSSMEX from V1.00 to V1.01. 97
Upgrading SMSSMEX from V1.01 to V1.02. 101
Upgrading SMSSMEX from V1.02 to V1.10. 101

Licensing 107
License Types 107
Autopass License Management. 107

Status Page 109

Troubleshooting 111
checker.bat and encryptPasswords.bat Fail. 111
Incident not Sent to SAP AGS 111
java.lang.OutOfMemoryError 111
Record in EventIn is not Executed 112
Incident Update or Process Action Fails 114
Information is not Updated in SAP Solution Manager 116

Incident Exchange Details 117
Database Tables 117
Tools 118
Field Mapping Configuration 119
Types of Mapping 119
Structure of FieldMapping XML file 119
Composite Field Mapping. 120
Field Value Mapping. 121
Field Mapping Schema 121
Default Field Mapping File and Customization 124

Prerequisites . . . 124
Adding Fields to fieldMapping.xml. . . . 125
Additional Information . . . 126
Changeable Mappings . . . 128
Person Synchronization Details . . . 128
SAP Solution Manager to Service Manager . . . 128
SMSSMEX Version . . . 129

Installing and Configuring SAPCRYPLIB . . . 131

Customizing HP ServiceCenter . . . 133
Creating a ServiceCenter User . . . 133
Importing Customizations via Unload . . . 134
Core Unload . . . 134
Demo Unload. . . . 135
Customizing Demo Unload Manually. . . . 135
Incident Custom Fields and Web Service Exposure . . . 136
Contacts Web Service Exposure. . . . 138
Event Services Web Service Exposure. . . . 138
Journal Separator Line Format . . . 139
Template . . . 141
Incident form. . . . 142
Status and Hidden Metadata . . . 142
Exchange History . . . 142
Trigger Buttons . . . 146
Selection of External Helpdesk System . . . 152
SAP Configuration Item Handling . . . 152
Overview . . . 152
Implementation . . . 152
Example Implementation via New Device Type and fill.fc . . . 153
Implementation Alternatives and Enhancements. . . . 159
HPSAPSolutionManager . . . 160

Logging . . . 161

Configuring HP ServiceCenter for SSL. . . . 163
Setting Up ServiceCenter for SSL Communication with SMSSMEX. . . . 163
Setting Up SMSSMEX for SSL Communication with ServiceCenter. . . . 169

Deploying Button Icons . . . 171
Windows Client . . . 171
Web Client . . . 171

SAP System Landscape Directory Registration . . . 173
Prerequisites . . . 173
Registering System Landscape Directory . . . 173

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

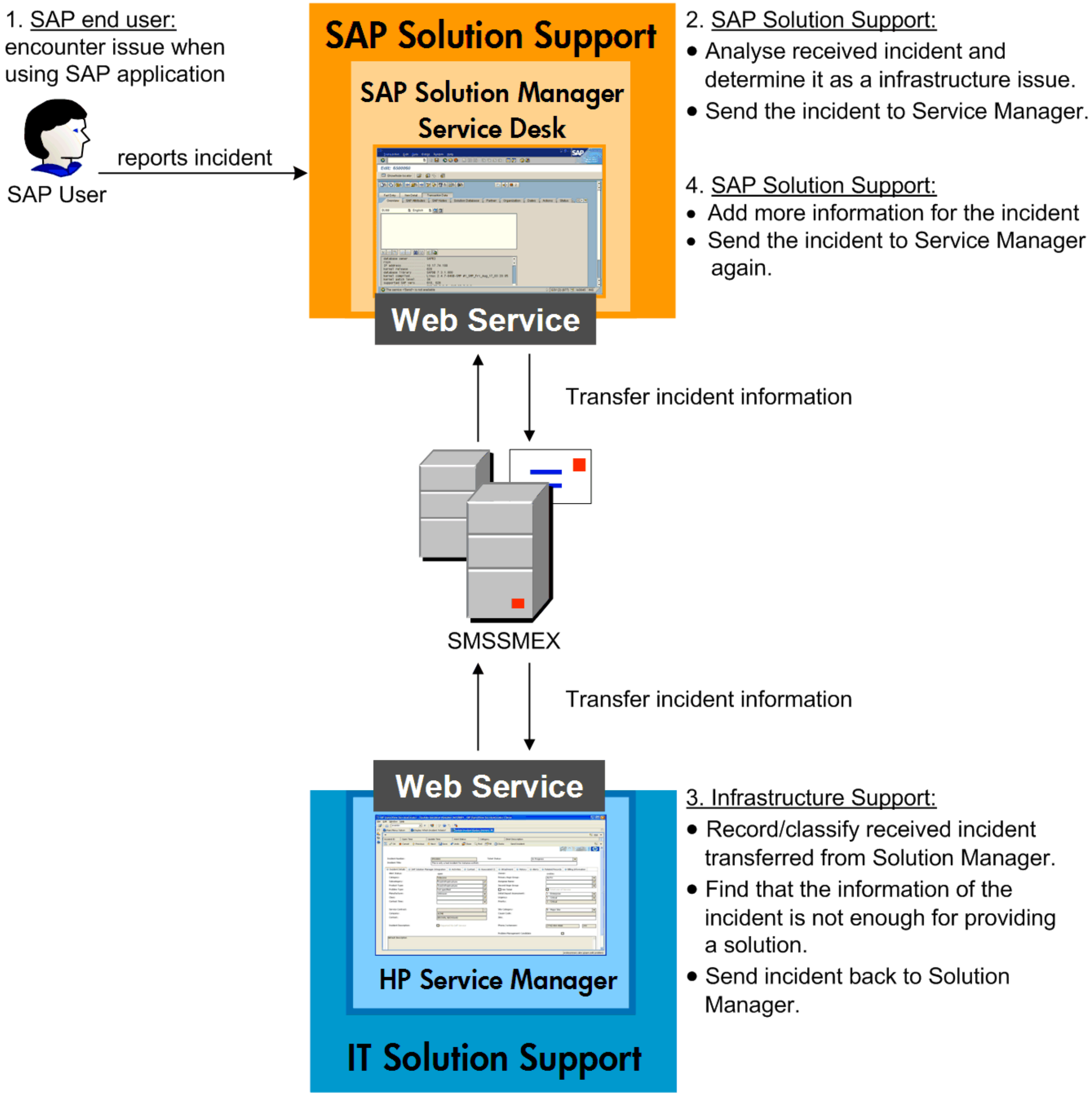
This document describes installation, configuration, administration and maintenance of HP Incident Exchange and the HP Incident Exchange web service. This guide is intended for use by HP consultants and application administrators that install and maintain HP Incident Exchange. This document is not an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

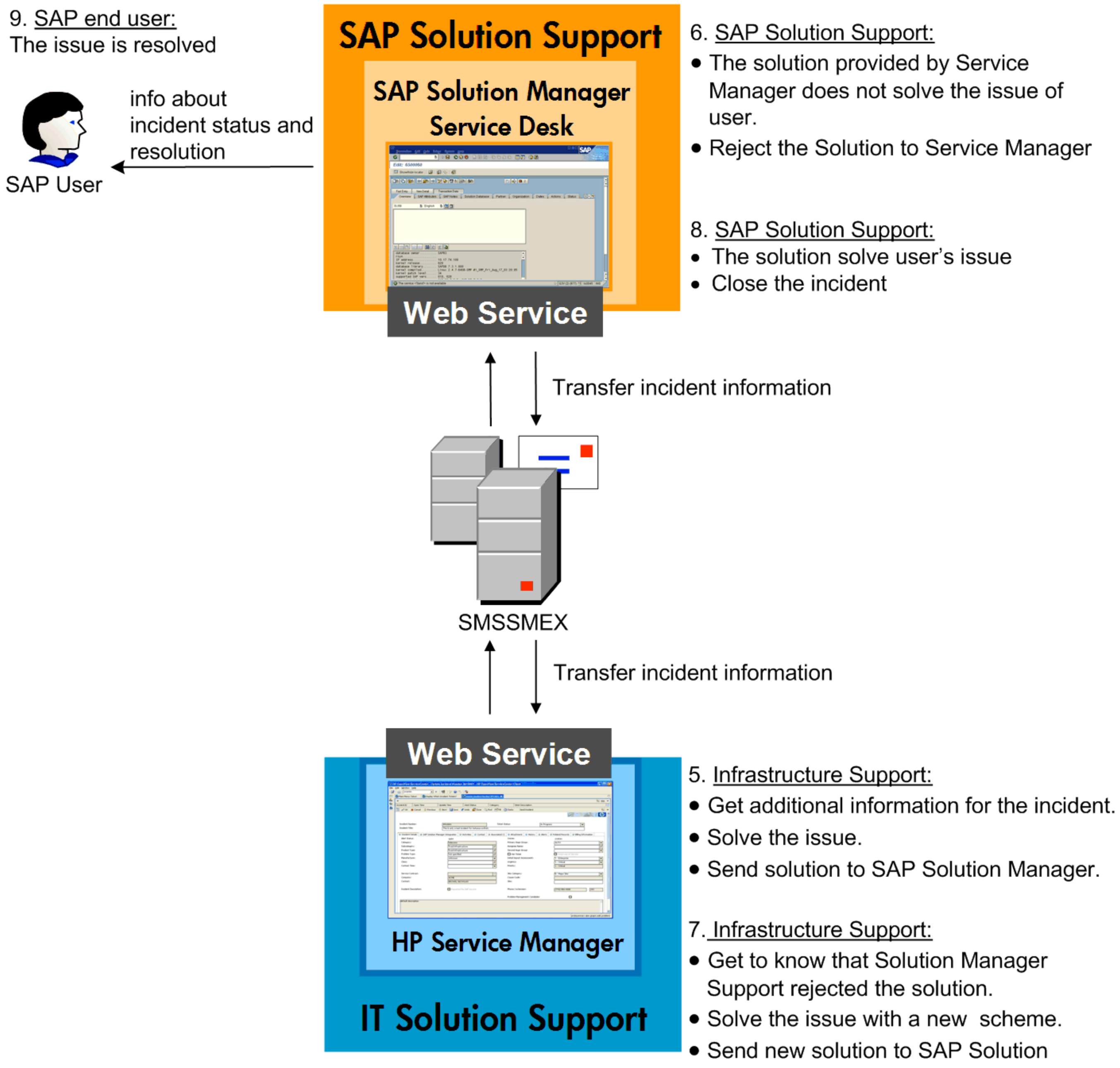


1. SAP end user:
encounter issue when using SAP application

2. SAP Solution Support:
- Analyse received incident and determine it as a infrastructure issue.
- Send the incident to Service Manager.

4. SAP Solution Support:
- Add more information for the incident
- Send the incident to Service Manager again.

3. Infrastructure Support:
- Record/classify received incident transferred from Solution Manager.
- Find that the information of the incident is not enough for providing a solution.
- Send incident back to Solution Manager.


SAP Solution Support
SAP Solution Manager Service Desk
Web Service
info about incident status and resolution
SAP User
Transfer incident information
SMSSMEX
Transfer incident information
Web Service
HP Service Manager
IT Solution Support

9. SAP end user:
The issue is resolved

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

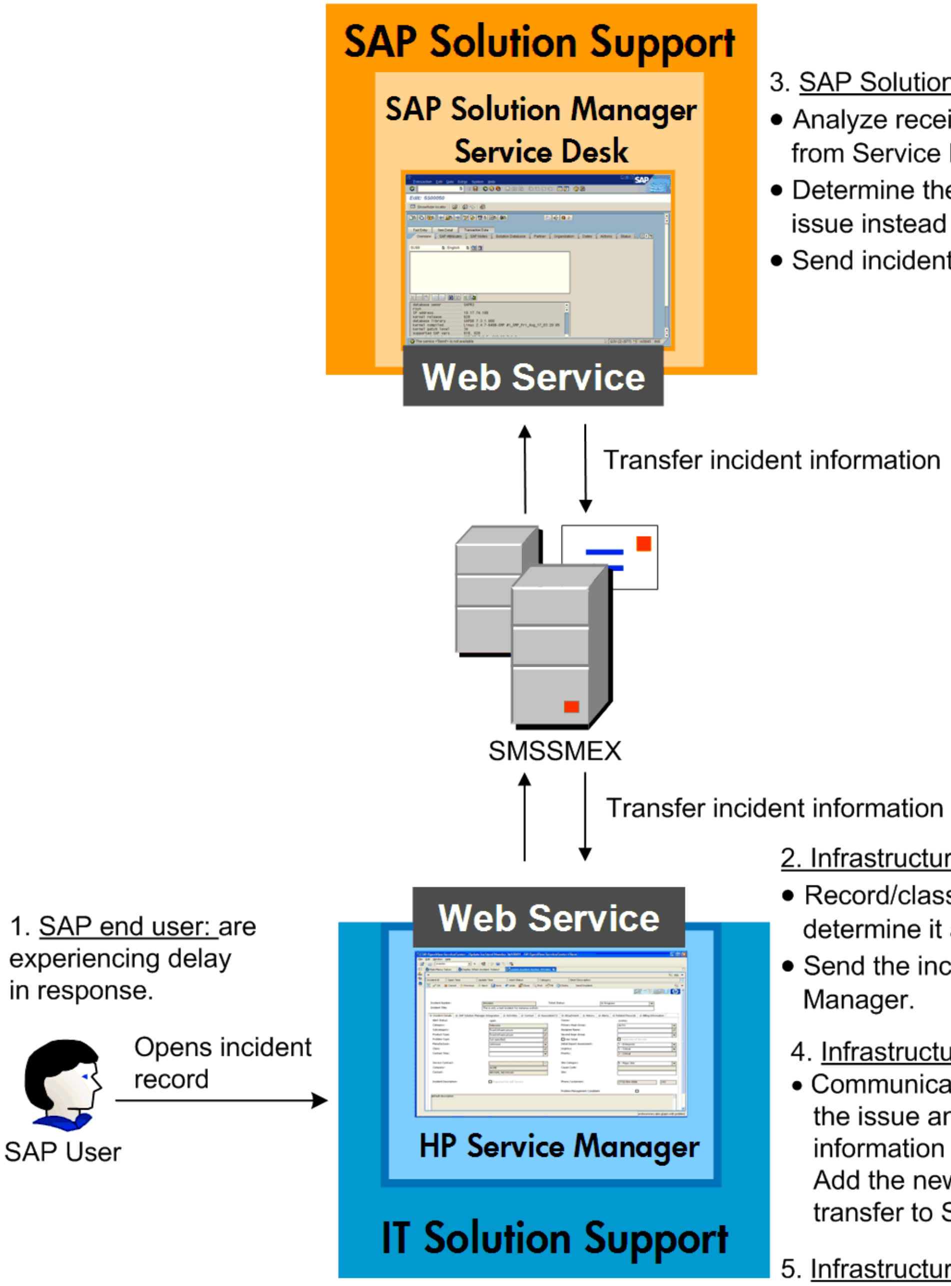
8. SAP Solution Support:
- The solution solve user's issue
- Close the incident

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.



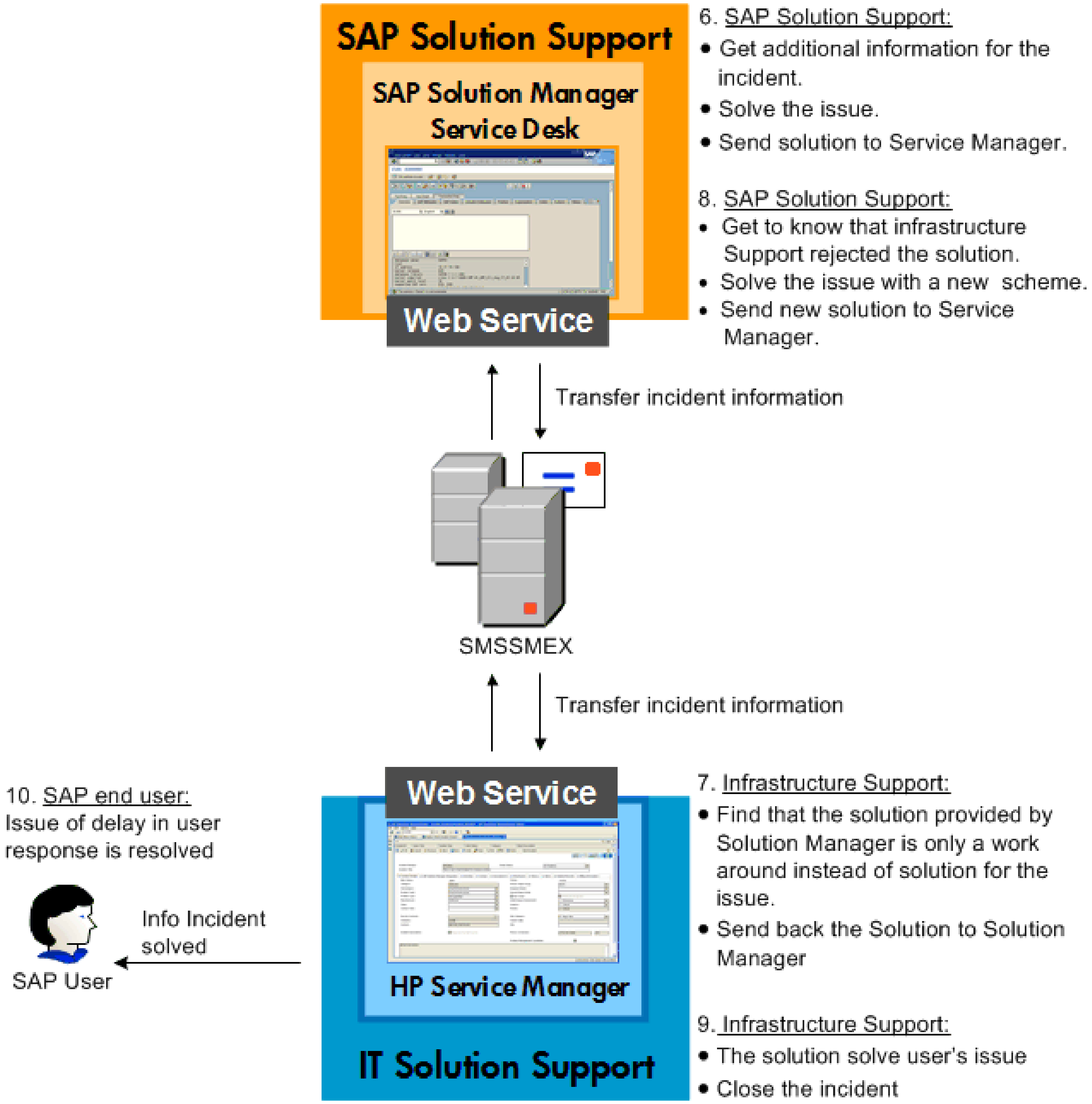
1. SAP end user: are experiencing delay in response.

2. Infrastructure Support:
- Record/classify received incident and determine it as a SAP application issue.
- Send the incident to SAP Solution Manager.

3. SAP Solution Support:
- Analyze received incident transferred from Service Manager.
- Determine the issue is an infrastructure issue instead of SAP application issue.
- Send incident back to Service Manager.

4. Infrastructure Support:
- Communicate with SAP end user about the issue and the user provide more information about the issue. Add the new information to incident and transfer to Solution Manager.

5. Infrastructure Support:
- Confirm the issue is SAP application issue
- Send the incident back to Service Manager again.



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.

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:

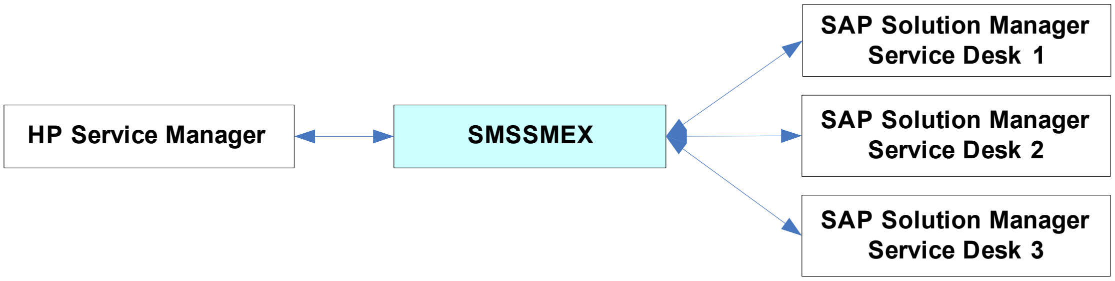
- The solution solve user's issue
- Close the incident

10. SAP end user:
Issue of delay in user response is resolved

2 Deployment Scenarios

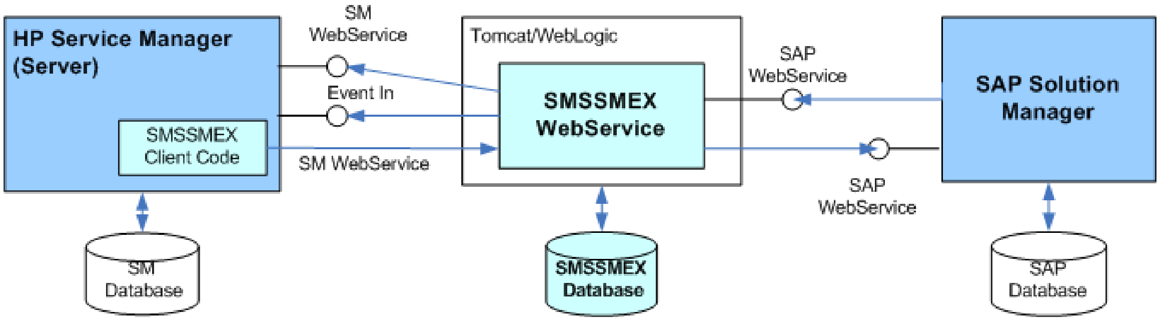
High Level Overview

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


HP Service Manager
SMSSMEX
SAP Solution Manager Service Desk 1
SAP Solution Manager Service Desk 2
SAP Solution Manager Service Desk 3

Components

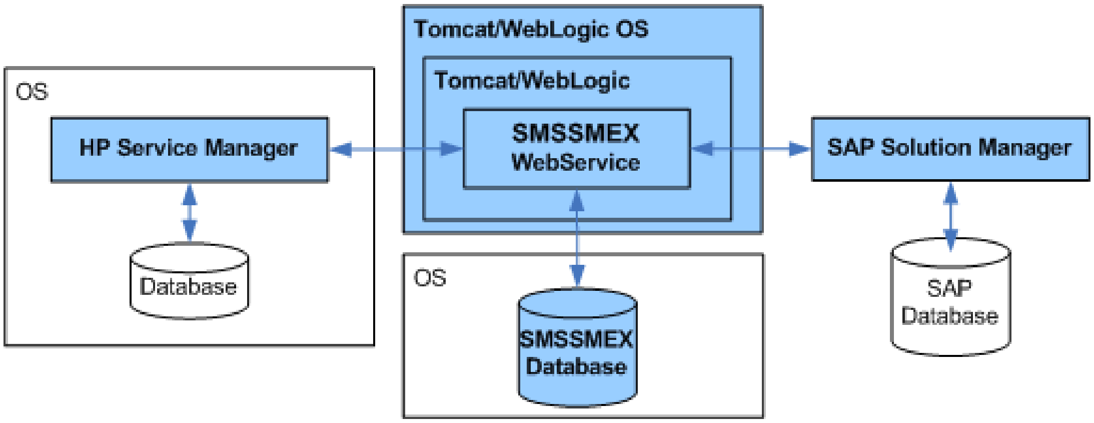
The following diagram shows the component details.



- 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

  Or

  `<`*`SM7.01 DVD`*`>\Server\Unix\autopass\`

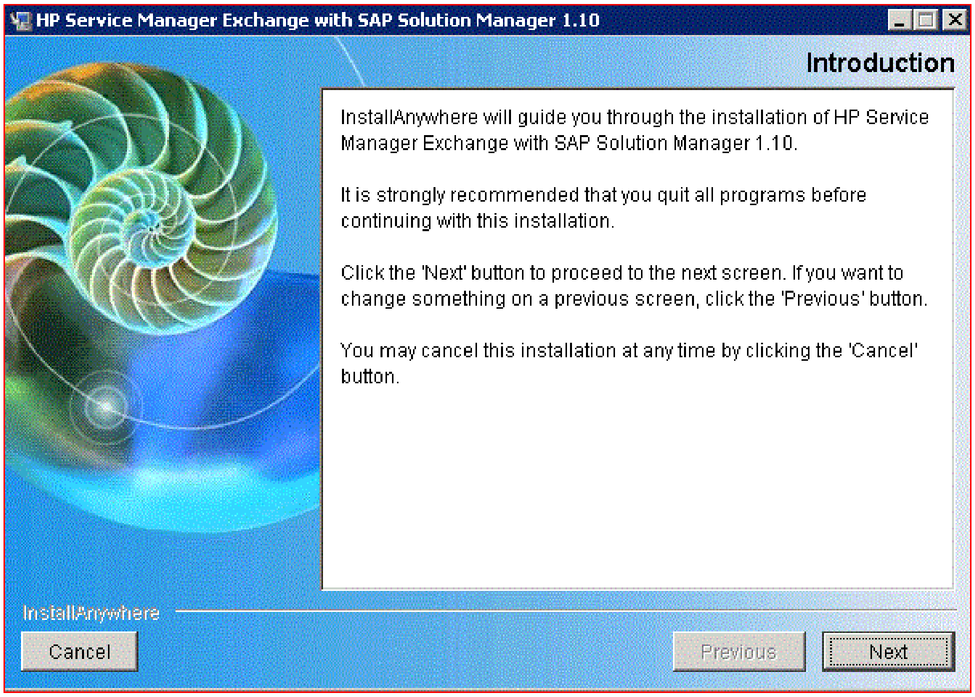
  — `HPOvLic-05.40.010-Linux2.6-release.rpm` for Linux

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.



HP Service Manager Exchange with SAP Solution Manager 1.10

Introduction

InstallAnywhere will guide you through the installation of HP Service Manager Exchange with SAP Solution Manager 1.10.

It is strongly recommended that you quit all programs before continuing with this installation.

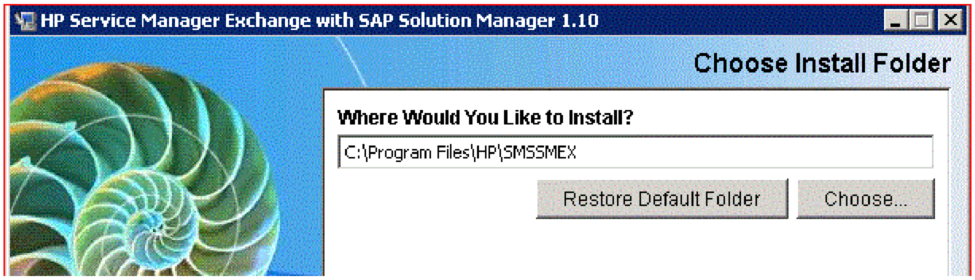
Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

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`.



HP Service Manager Exchange with SAP Solution Manager 1.10

Choose Install Folder

Where Would You Like to Install?

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
```

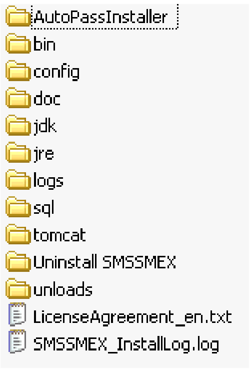
Or simply go to **Start** → **Programs** → **SMSSMEX** → **Uninstall SMSSMEX**.

To uninstall SMSSMEX on Unix, execute

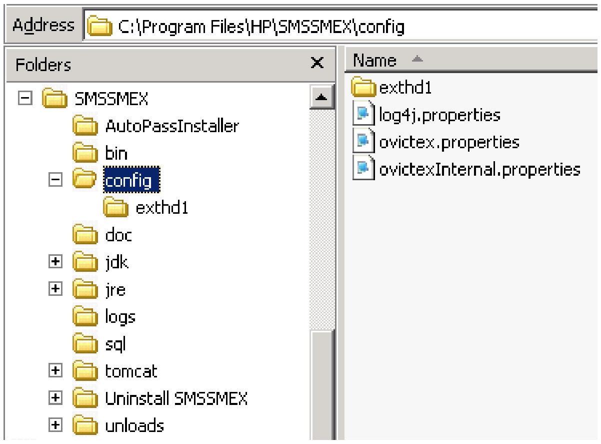
```
<SMSSMEX_installDir>/Uninstall SMSSMEX/Uninstall_SMSSMEX
```

SMSSMEX Installed Files

After installation, the SMSSMEX folder has the following contents.



AutoPassInstaller
bin
config
doc
jdk
jre
logs
sql
tomcat
Uninstall SMSSMEX
unloads
LicenseAgreement_en.txt
SMSSMEX_InstallLog.log



Address C:\Program Files\HP\SMSSMEX\config

Folders

SMSSMEX
AutoPassInstaller
bin
config
exthd1
doc
jdk
jre
logs
sql
tomcat
Uninstall SMSSMEX
unloads

Name

exthd1
log4j.properties
ovictex.properties
ovictexInternal.properties

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\SM7.0	Service Manager 7 customization unload files **Note:** SMSSMEX 1.10 unload files are available at `<`*`SMSSMEX1.10 Release Package`*`>\unloads\` directory
unloads\SM7.1	Service Manager 7.10 customization unload files
logs	Log files
jdk	Internal JDK 5
tomcat	Tomcat 5.0.28

Table 2 Contents of \SMSSMEX

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

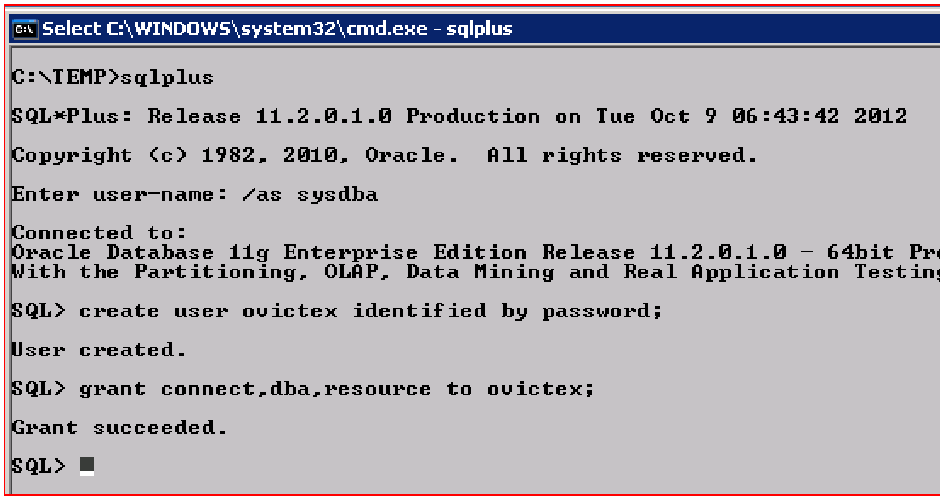
This section describes how to setup the database.

➤ 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.

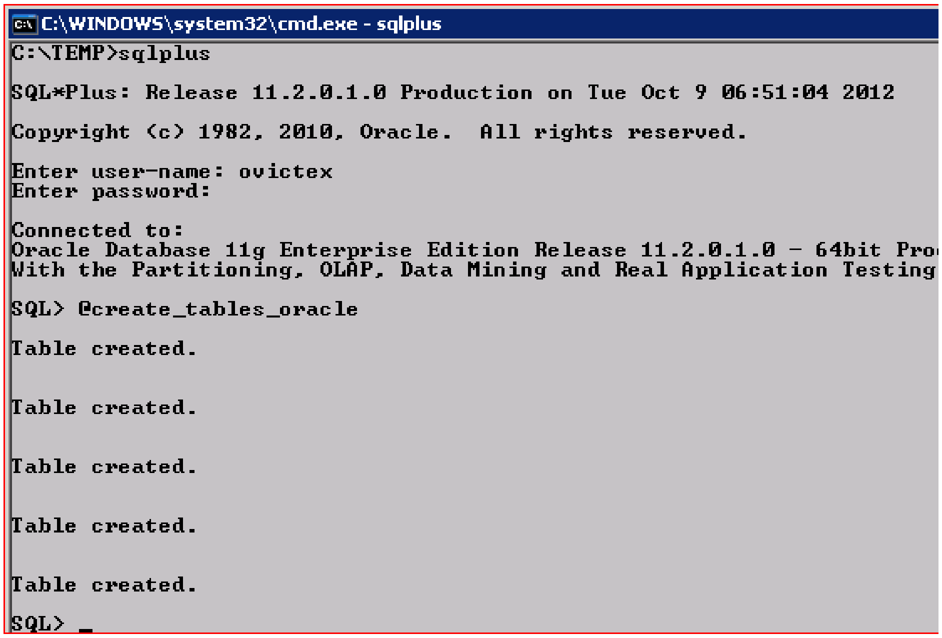


Select C:\WINDOWS\system32\cmd.exe - sqlplus

```
C:\TEMP>sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 9 06:43:42 2012

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: /as sysdba

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Pr
With the Partitioning, OLAP, Data Mining and Real Application Testin

SQL> create user ovictex identified by password;

User created.

SQL> grant connect,dba,resource to ovictex;

Grant succeeded.

SQL> 
```

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.



C:\WINDOWS\system32\cmd.exe - sqlplus

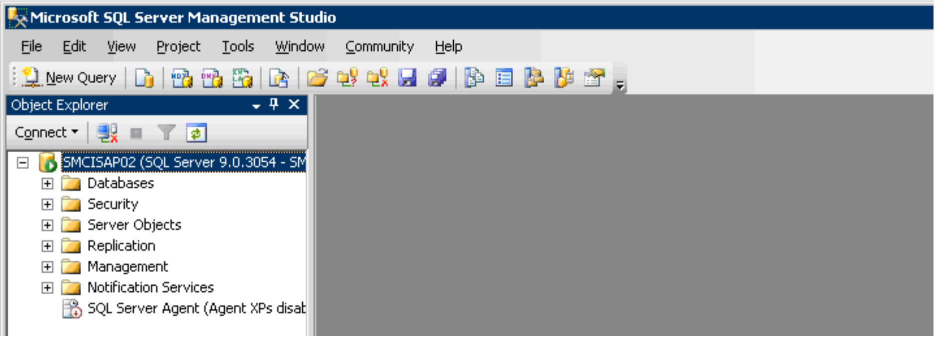
```
C:\TEMP>sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 9 06:51:04 2012

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: ovictex
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Pro
With the Partitioning, OLAP, Data Mining and Real Application Testing

SQL> @create_tables_oracle

Table created.


Table created.


Table created.


Table created.


Table created.

SQL> _
```

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

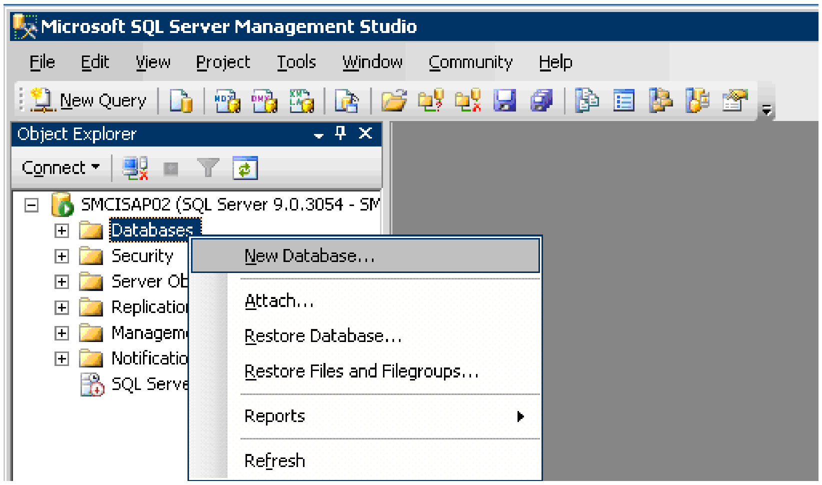
MS-SQL 2005

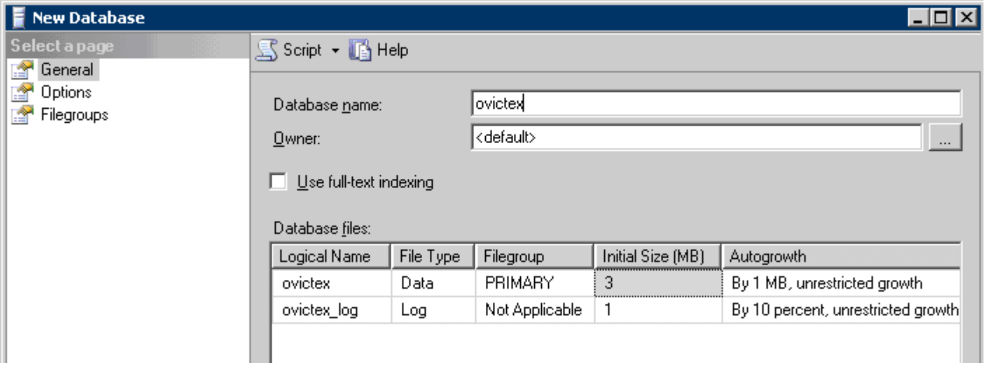
Do the following to create the required separate database for SMSSMEX tables:

1 Launch SQL Server Management Studio.


Microsoft SQL Server Management Studio
File Edit View Project Tools Window Community Help
New Query
Object Explorer
Connect
SMCISAP02 (SQL Server 9.0.3054 - SM
Databases
Security
Server Objects
Replication
Management
Notification Services
SQL Server Agent (Agent XPs disab

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


Microsoft SQL Server Management Studio
File Edit View Project Tools Window Community Help
New Query
Object Explorer
Connect
SMCISAP02 (SQL Server 9.0.3054 - SM
Databases
Security
Server Ob
Replicatio
Managem
Notificatio
SQL Serve
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Reports
Refresh



Database name: ovictex

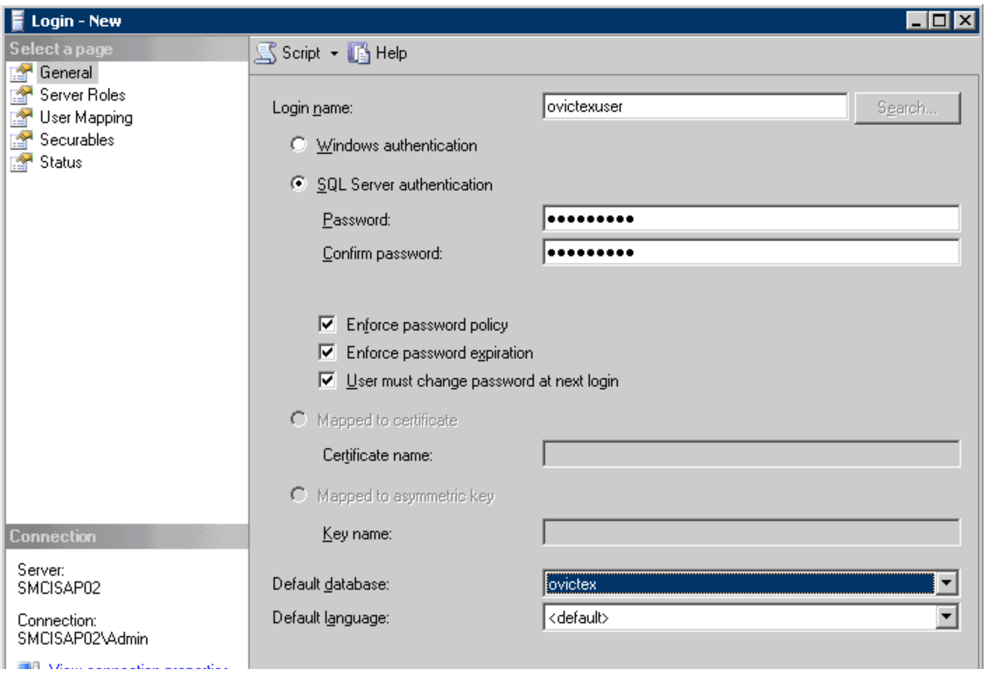
Owner: <default>

☐ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
ovictex	Data	PRIMARY	3	By 1 MB, unrestricted growth
ovictex_log	Log	Not Applicable	1	By 10 percent, unrestricted growth

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



Login name: ovictexuser

○ Windows authentication

● SQL Server authentication

Password: ●●●●●●●●●

Confirm password: ●●●●●●●●●

☑ Enforce password policy

☑ Enforce password expiration

☑ User must change password at next login

○ Mapped to certificate

Certificate name:

○ Mapped to asymmetric key

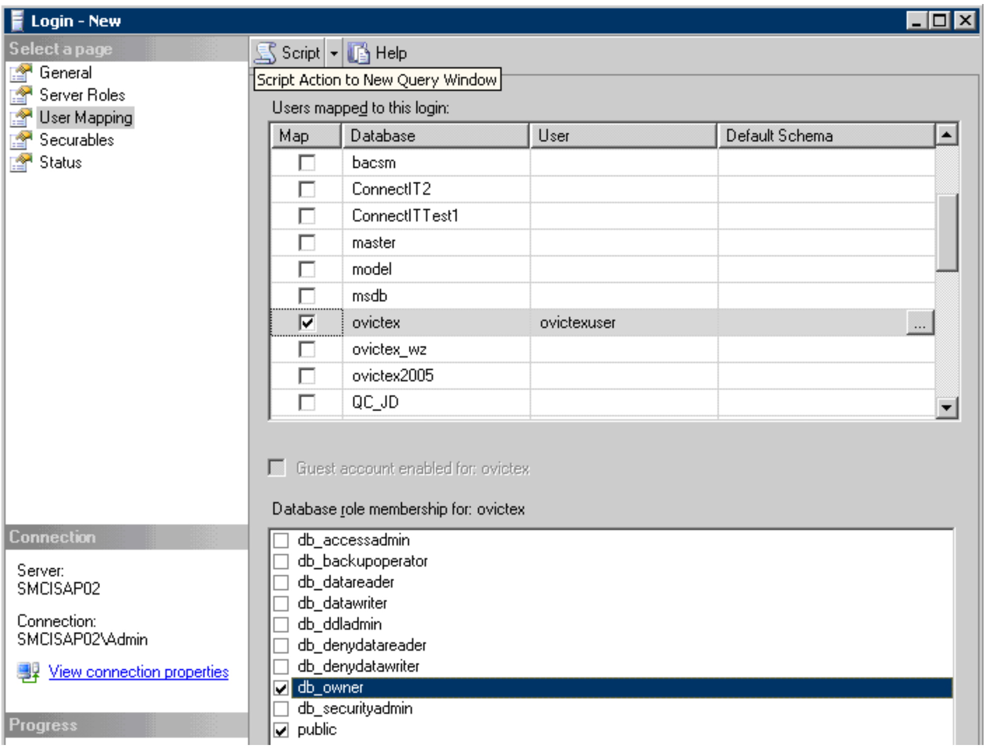
Key name:

Default database: ovictex

Default language: <default>

Server:
SMCISAP02

Connection:
SMCISAP02\Admin



Users mapped to this login:

Map	Database	User	Default Schema
□	bacsm		
□	ConnectIT2		
□	ConnectITTest1		
□	master		
□	model		
□	msdb		
☑	ovictex	ovictexuser	...
□	ovictex_wz		
□	ovictex2005		
□	QC_JD		

□ Guest account enabled for: ovictex

Database role membership for: ovictex

□ db_accessadmin
□ db_backupoperator
□ db_datareader
□ db_datawriter
□ db_ddladmin
□ db_denydatareader
□ db_denydatawriter
☑ db_owner
□ db_securityadmin
☑ public

Server:
SMCISAP02

Connection:
SMCISAP02\Admin

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 22 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 118.

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 89).

- 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.

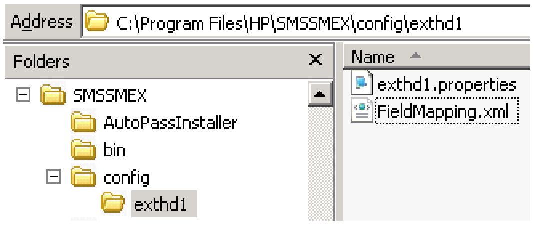
External Helpdesks

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`.

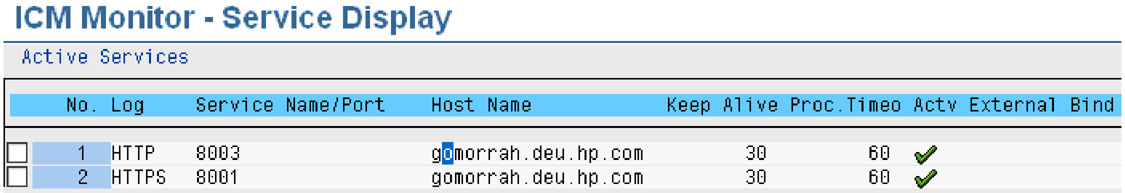
4 Rename `<`*`SMSSMEX_installDir`*`>/config/exthd2/exthd1.properties` to `<`*`SMSSMEX_installDir`*`>/config/exthd2/exthd2.properties`.

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**.



ICM Monitor - Service Display

	No.	Log	Service Name/Port	Host Name	Keep Alive	Proc.Timeo	Actv	External Bind
□	1	HTTP	8003	gomorrah.deu.hp.com	30	60	✓	
□	2	HTTPS	8001	gomorrah.deu.hp.com	30	60	✓	

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 119.

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 118.

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 24 for detailed instructions.

Configuring File ovictexInternal.properties

See Configuring File ovictexInternal.properties on page 25 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.

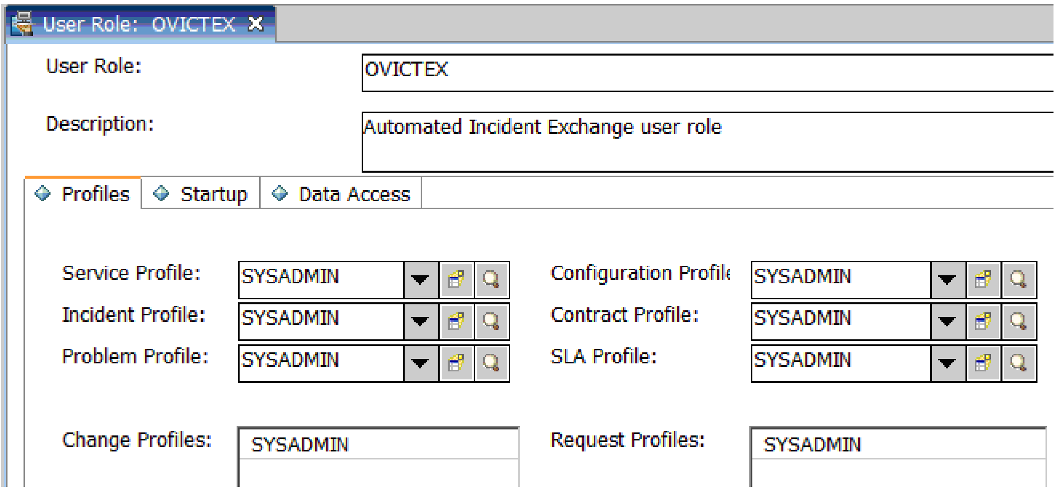
5 Customizing HP Service Manager

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

For information on ServiceCenter customization, see Chapter C, Customizing HP ServiceCenter.

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:



User Role: OVICTEX

Description: Automated Incident Exchange user role

Service Profile: SYSADMIN

Incident Profile: SYSADMIN

Problem Profile: SYSADMIN

Configuration Profile: SYSADMIN

Contract Profile: SYSADMIN

SLA Profile: SYSADMIN

Change Profiles: SYSADMIN

Request Profiles: SYSADMIN

Do the following:

1 Select **System Administration** → **Ongoing Maintenance** → **User Roles**.

2 Search for **`SYSTEM ADMINISTRATOR`** on Service Manager 7.0x, or **`system administrator`** on Service Manager 7.10 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.

3 Enter **`OVICTEX`** as the User Role.

4 Change Description to **`Automated Incident Exchange user role`**.

5 Click **Add**.

6 Select **System Administration** → **Ongoing Maintenance** → **User Quick Add Utility**.

7 Enter **`ovictex, INCIDENT EXCHANGE, Incident, Exchange, ovictex@hp.com`**.

8. Click **Next**.
9. For **User to clone** select **falcon**.
10. Click **Finish**.
11. Click **Save**.
12. Go to **System Administration** → **Ongoing Maintenance** → **Operators**, enter **`ovictex`** in the Login Name field, then click **Search**.
13. Change User Role to **`OVICTEX`**.
14. In the Security tab:
    - a. Enter the operator password for Password.
    - b. Uncheck Expire Password.
    - c. Check Never Expire Password.
15. 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 a core unload at the following path:

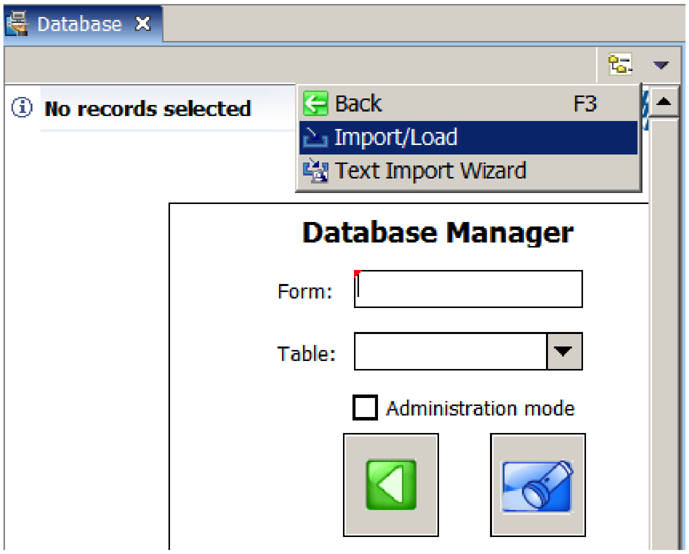
- `<SMSSMEX1.10 Release Package>\unloads\SM7.0\core_sm7.0.unl` for Service Manager 7.0x
- `<SMSSMEX1.10 Release Package>\unloads\SM7.1\core_sm7.1.unl` for Service Manager 7.1x

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.



No records selected

Database Manager

Form:

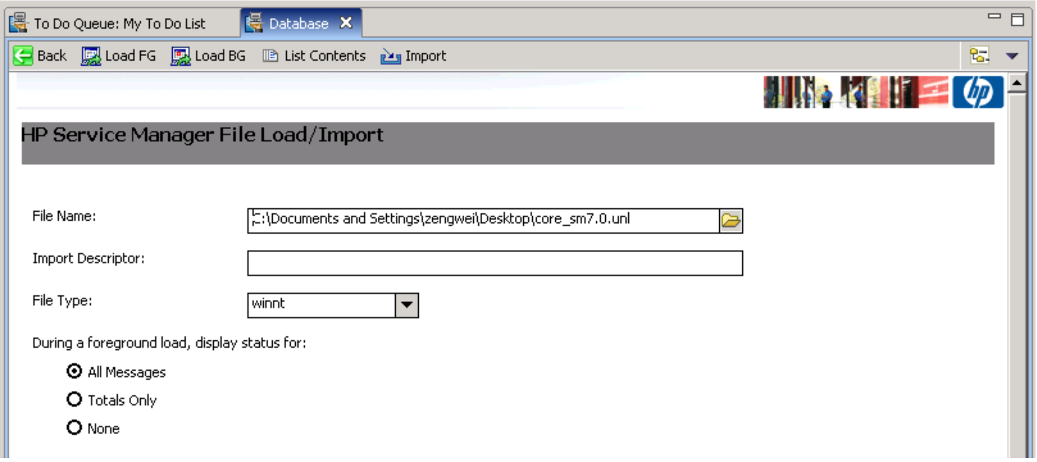
Table:

□ Administration mode

3 Select <*SMSSMEX1.10 Release Package*>`\unloads\SM70x\core_sm7.0.unl` on Service Manager 7.0x, or <*SMSSMEX1.10 Release Package*>`\unloads\SM71\core_sm7.1.unl` on Service Manager 7.1x.

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

On Service Manager 7.0x:



HP Service Manager File Load/Import

File Name: C:\Documents and Settings\zengwei\Desktop\core_sm7.0.unl

Import Descriptor:

File Type: winnt

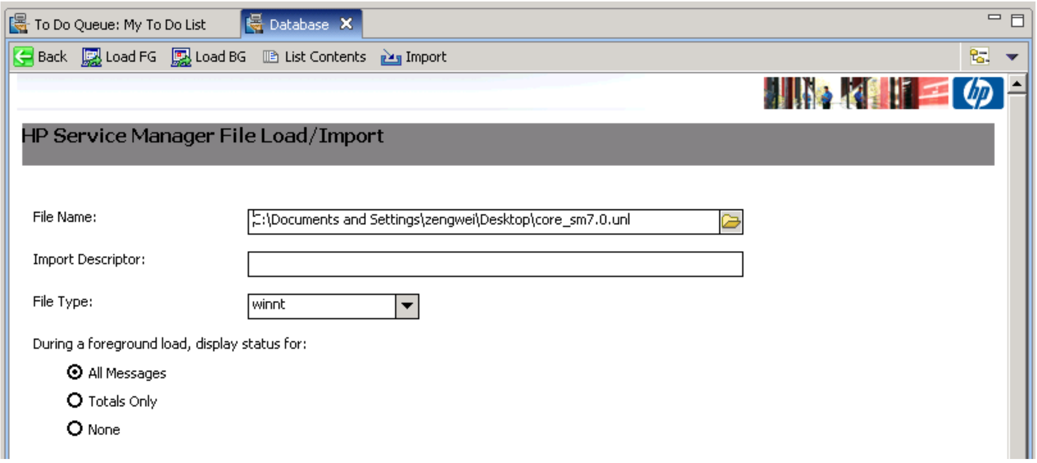
During a foreground load, display status for:

● All Messages

○ Totals Only

○ None

On Service Manager 7.1x:



HP Service Manager File Load/Import

File Name: C:\Documents and Settings\zengwei\Desktop\core_sm7.0.unl

Import Descriptor:

File Type: winnt

During a foreground load, display status for:

● All Messages

○ Totals Only

○ None

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\SM70x\demo_sm7.0.unl` on Service Manager 7.0x, or
   - `<`*`SMSSMEX1.10 Release Package`*`>\unloads\SM71\demo_sm7.1.unl` on Service Manager 7.1x.
4. Press **Load FG** to start the import.

Customizing Demo Unload Manually

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

Trigger URL

Incident exchanges and updates from ServiceCenter to SAP Solution Manager are sent to the Incident Exchange web service using an HTTP trigger. The trigger URL must be configured inside the ServiceCenter Script Library.

1. Select **Toolkit** → **Script Library**.
2. Enter **`HPSAPTrigger`** in the Name field, and click **Search**.
3. Specify the Tomcat hostname and port to replace the localhost and 8080 defined in the URL of the scripts.
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
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

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.

5 Include the following `probsummary` table fields in the web service API.

Field name	**Type**	**Caption**	**Field name in API**	**API Field type**
priority.code	Character	Priority Code	PriorityCode	StringType
planned.end	Date/time	Planned End	PlannedEnd	DateTimeType

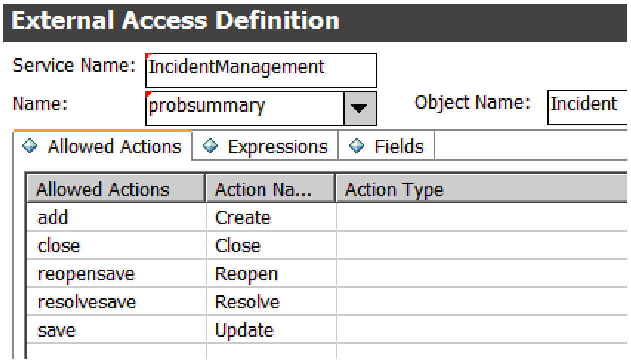
Expose Custom Fields in Web Service Interface IncidentManagement

To expose custom fields, do the following:

1 Select **Tailoring** → **Tailoring Tools** → **External Access** on Service Manager 7.0x, or **Tailoring** → **Web Services** → **WSDL Configuration** on Service Manager 7.10 and above.

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

3 Click **Search**.



External Access Definition

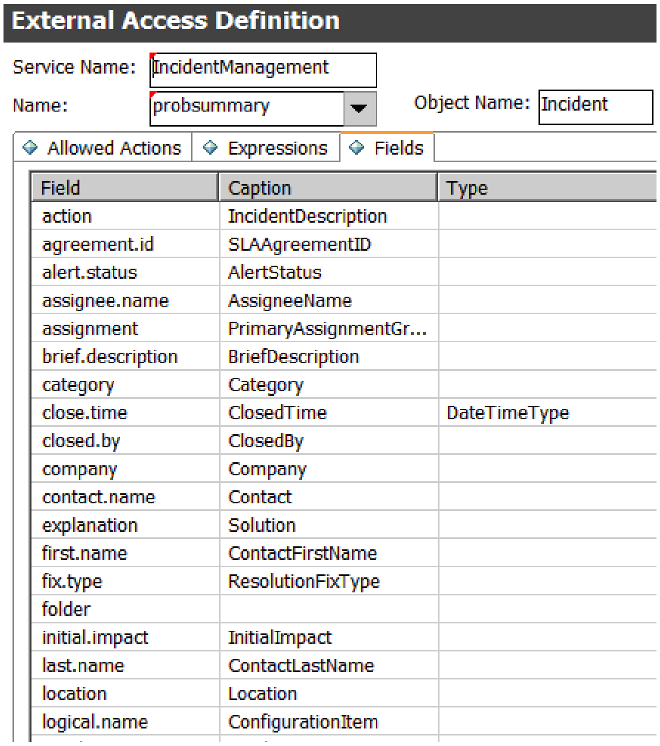
Service Name: IncidentManagement

Name: probsummary

Object Name: Incident

Allowed Actions	Action Na...	Action Type
add	Create	
close	Close	
reopensave	Reopen	
resolvesave	Resolve	
save	Update	

4 Select the **Fields** tab.



External Access Definition

Service Name: IncidentManagement

Name: probsummary

Object Name: Incident

Field	Caption	Type
action	IncidentDescription	
agreement.id	SLAAgreementID	
alert.status	AlertStatus	
assignee.name	AssigneeName	
assignment	PrimaryAssignmentGr...	
brief.description	BriefDescription	
category	Category	
close.time	ClosedTime	DateTimeType
closed.by	ClosedBy	
company	Company	
contact.name	Contact	
explanation	Solution	
first.name	ContactFirstName	
fix.type	ResolutionFixType	
folder		
initial.impact	InitialImpact	
last.name	ContactLastName	
location	Location	
logical.name	ConfigurationItem	

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

6 Change Expressions from

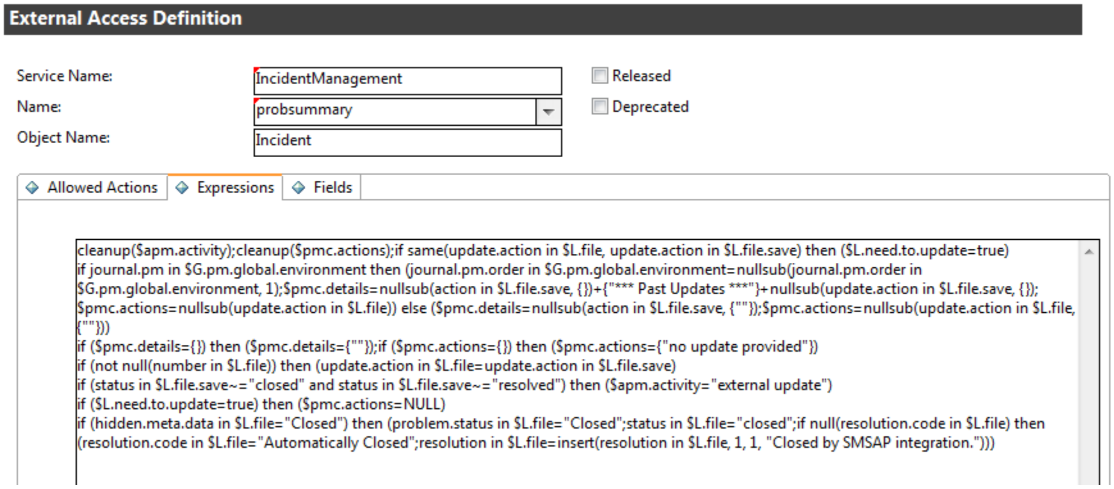
```
update.action in $L.file=update.action in $L.file.save
```

to (**[INCIDENTMANAGEMENT]** in `code_sm7.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.")))
```



Service Name: IncidentManagement

Name: probsummary

Object Name: Incident

□ Released

□ Deprecated

```
cleanup($apm.activity);cleanup($pmc.actions);if same(update.action in $L.file, update.action in $L.file.save) then ($L.need.to.update=true)
if journal.pm in $G.pm.global.environment then (journal.pm.order in $G.pm.global.environment=nullsub(journal.pm.order in
$G.pm.global.environment, 1);$pmc.details=nullsub(action in $L.file.save, {})+{"*** Past Updates ***"}+nullsub(update.action in $L.file.save, {});
$pmc.actions=nullsub(update.action in $L.file)) else ($pmc.details=nullsub(action in $L.file.save, {""});$pmc.actions=nullsub(update.action in $L.file,
{""}))
if ($pmc.details={}) then ($pmc.details={""});if ($pmc.actions={}) then ($pmc.actions={"no update provided"})
if (not null(number in $L.file)) then (update.action in $L.file=update.action in $L.file.save)
if (status in $L.file.save~="closed" and status in $L.file.save~="resolved") then ($apm.activity="external update")
if ($L.need.to.update=true) then ($pmc.actions=NULL)
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

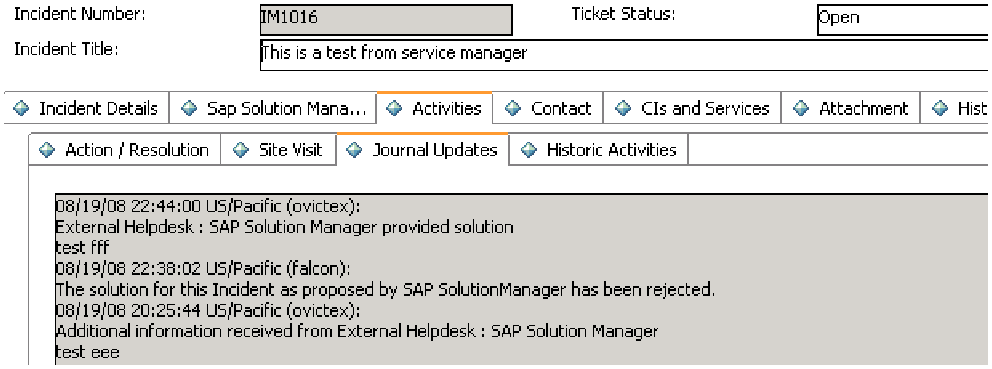
To expose the contacts web service, do the following:

1. Select **Tailoring** → **Tailoring Tools** → **External Access** on Service Manager 7.0x, or **Tailoring** → **Web Services** → **WSDL Configuration** on Service Manager 7.10 and above.
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.



Incident Number: IM1016

Ticket Status: Open

Incident Title: This is a test from service manager

```
08/19/08 22:44:00 US/Pacific (ovictex):
External Helpdesk : SAP Solution Manager provided solution
test fff
08/19/08 22:38:02 US/Pacific (falcon):
The solution for this Incident as proposed by SAP SolutionManager has been rejected.
08/19/08 20:25:44 US/Pacific (ovictex):
Additional information received from External Helpdesk : SAP Solution Manager
test eee
```

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`

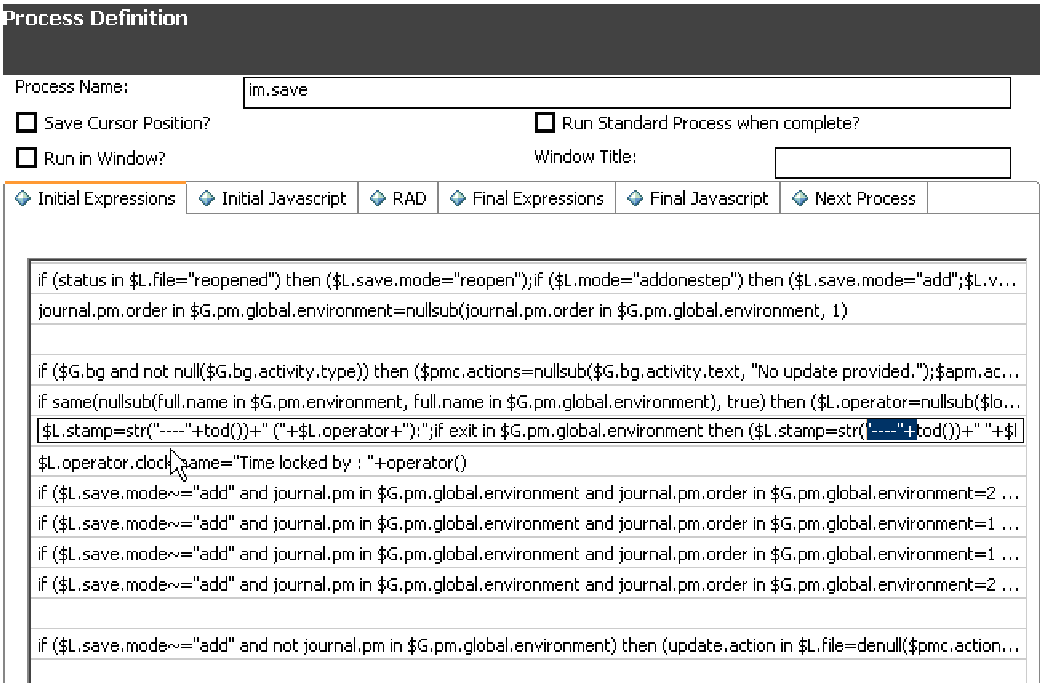
The Incident Exchange core unload provides an additional process `im.exchange.incident` that already contains the modification. However, this must also be modified if the separator string deviates from the default.

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_sm7.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+"):")
```



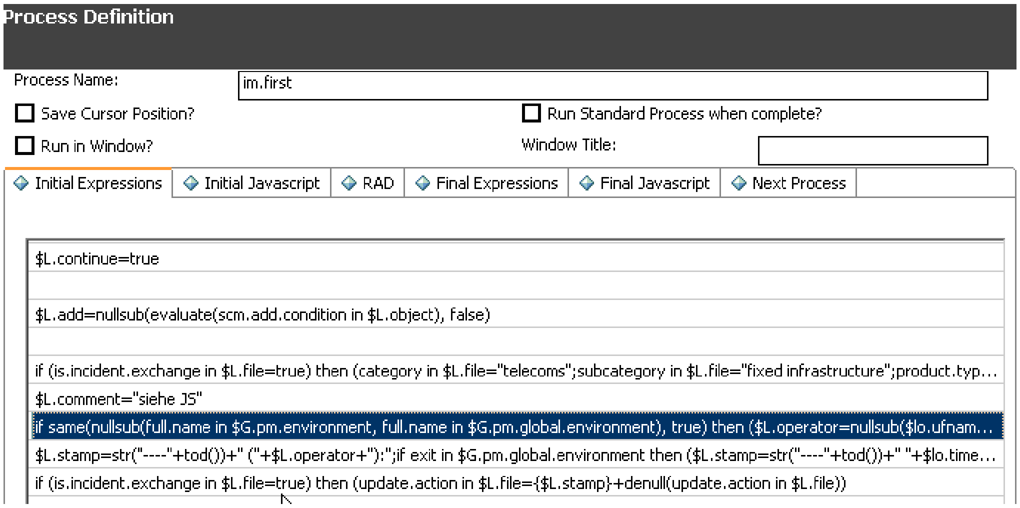
Process Name: im.save

□ Save Cursor Position? □ Run Standard Process when complete?

□ Run in Window? Window Title:

```
if (status in $L.file="reopened") then ($L.save.mode="reopen");if ($L.mode="addonestep") then ($L.save.mode="add";$L.v...
journal.pm.order in $G.pm.global.environment=nullsub(journal.pm.order in $G.pm.global.environment, 1)

if ($G.bg and not null($G.bg.activity.type)) then ($pmc.actions=nullsub($G.bg.activity.text, "No update provided.");$apm.ac...
if same(nullsub(full.name in $G.pm.environment, full.name in $G.pm.global.environment), true) then ($L.operator=nullsub($lo...
$L.stamp=str("----"+tod())+" ("+$L.operator+"):";if exit in $G.pm.global.environment then ($L.stamp=str("----"+tod())+" "+$l
$L.operator.clock.name="Time locked by : "+operator()
if ($L.save.mode~="add" and journal.pm in $G.pm.global.environment and journal.pm.order in $G.pm.global.environment=2 ...
if ($L.save.mode~="add" and journal.pm in $G.pm.global.environment and journal.pm.order in $G.pm.global.environment=1 ...
if ($L.save.mode~="add" and journal.pm in $G.pm.global.environment and journal.pm.order in $G.pm.global.environment=1 ...
if ($L.save.mode~="add" and journal.pm in $G.pm.global.environment and journal.pm.order in $G.pm.global.environment=2 ...

if ($L.save.mode~="add" and not journal.pm in $G.pm.global.environment) then (update.action in $L.file=denull($pmc.action...
```

4 Normally the first Journal entry is only entered *after* the Incident has been created. But Incident Exchange adds Journal entries (the external helpdesk Incident ID) during creation of an Incident. To ensure that the initial Journal entries also contain a separator (required for block detection by Incident Exchange), add the following statements in im.first to insert a separator.



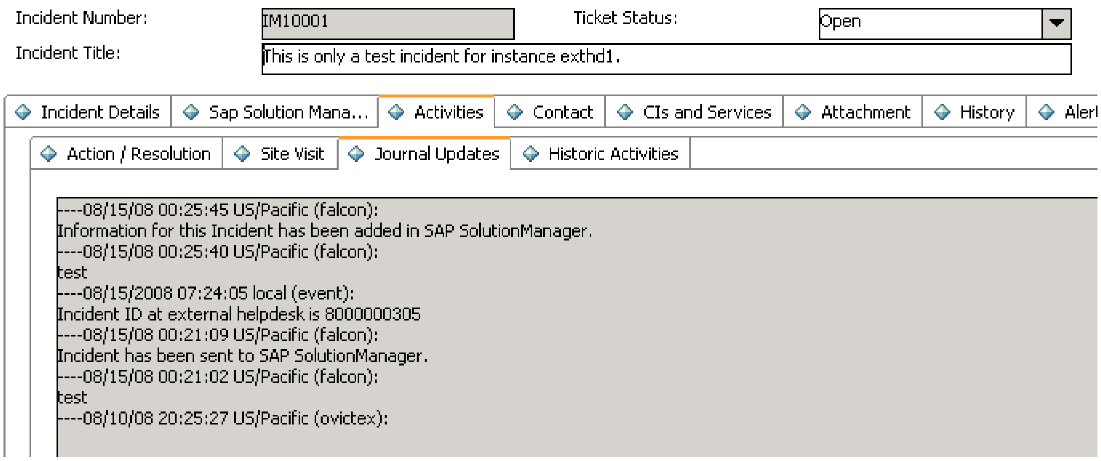
Process Name: im.first

□ Save Cursor Position? □ Run Standard Process when complete?

□ Run in Window? Window Title:

```
$L.continue=true

$L.add=nullsub(evaluate(scm.add.condition in $L.object), false)

if (is.incident.exchange in $L.file=true) then (category in $L.file="telecoms";subcategory in $L.file="fixed infrastructure";product.typ...
$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.ufnam...
$L.stamp=str("----"+tod())+" ("+$L.operator+"):";if exit in $G.pm.global.environment then ($L.stamp=str("----"+tod())+" "+$lo.time...
if (is.incident.exchange in $L.file=true) then (update.action in $L.file={$L.stamp}+denull(update.action in $L.file))
```

After the modification, all Journal updates should contain the separator as shown in the following diagram.



Incident Number: IM10001
Ticket Status: Open
Incident Title: This is only a test incident for instance exthd1.

```
----08/15/08 00:25:45 US/Pacific (falcon):
Information for this Incident has been added in SAP SolutionManager.
----08/15/08 00:25:40 US/Pacific (falcon):
test
----08/15/2008 07:24:05 local (event):
Incident ID at external helpdesk is 8000000305
----08/15/08 00:21:09 US/Pacific (falcon):
Incident has been sent to SAP SolutionManager.
----08/15/08 00:21:02 US/Pacific (falcon):
test
----08/10/08 20:25:27 US/Pacific (ovictex):
```

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**.

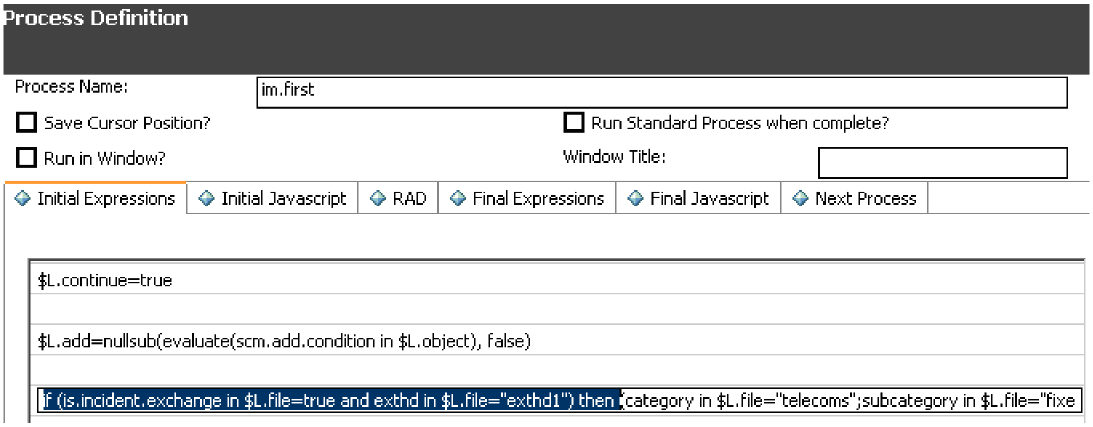
2 Search for `im.first`.

3 Add an Initial Expression that sets all required fields in `probsummary` that are not yet set by Incident Exchange.

> 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.

The following is an example expression:

```
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"})
```



Process Definition

Process Name: im.first

□ Save Cursor Position?

□ Run Standard Process when complete?

□ Run in Window?

Window Title:

```
$L.continue=true

$L.add=nullsub(evaluate(scm.add.condition in $L.object), false)

if (is.incident.exchange in $L.file=true and exthd in $L.file="exthd1") then (category in $L.file="telecoms";subcategory in $L.file="fixe
```

4 Add the following expressions to the Initial Expressions tab of `im.first`.

*On Service Manager 7.0x (***[IM.FIRST_INIT]** *in* `code_sm7.txt`*):*

```
if (is.incident.exchange in $L.file=true) 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";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"})
$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))
```

*On Service Manager 7.10 and above (***[IM.FIRST_INIT_7.10]** *in* `code_sm7.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))
```

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 part of the Incident workflow.

   On Service Manager 7.0x:

   `IM.template.open.g`

   `IM.template.update.g`

   `apm.quick.g`

   `IM.default.open.g`

   `IM.default.update.g`

   On Service Manager 7.10 and above:

   `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.

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

On Service Manager 7.0x:

`IM.template.open.g`

`IM.template.update.g`

`apm.quick.g`

`IM.default.open.g`

`IM.default.update.g`

On Service Manager 7.10 and above:

`IM.open.incident`

`IM.update.incident`

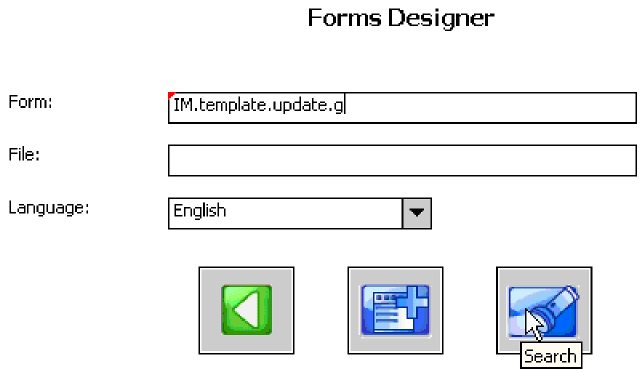
`IM.close.incident`

`hp.sap.solution.sub`

Property	**Value**
Caption	Sap Solution Manager

4 Add a Subform Control into the Sap Solution Manager tab or section.

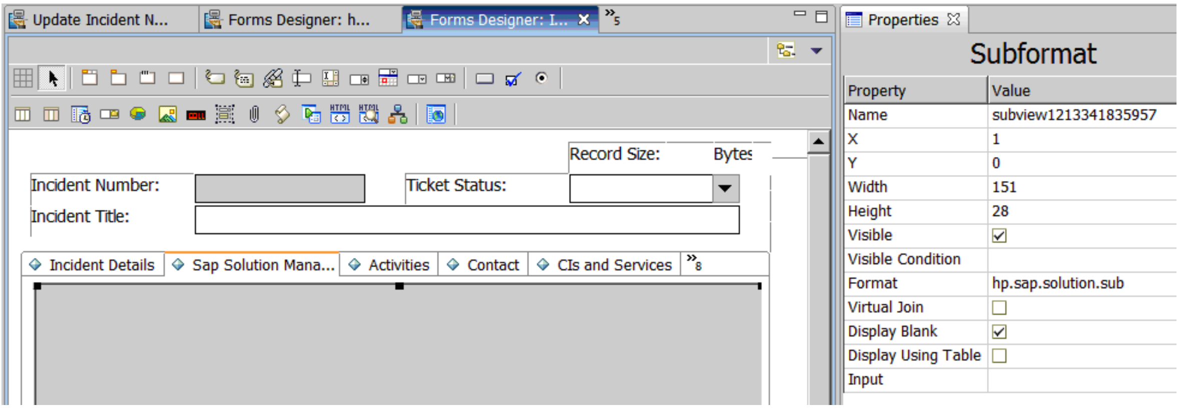
Property	**Value**
X	1
Y	0
Width	151
Height	28
Format	hp.sap.solution.sub


Forms Designer

Form: IM.template.update.g

File:

Language: English



Record Size: Bytes

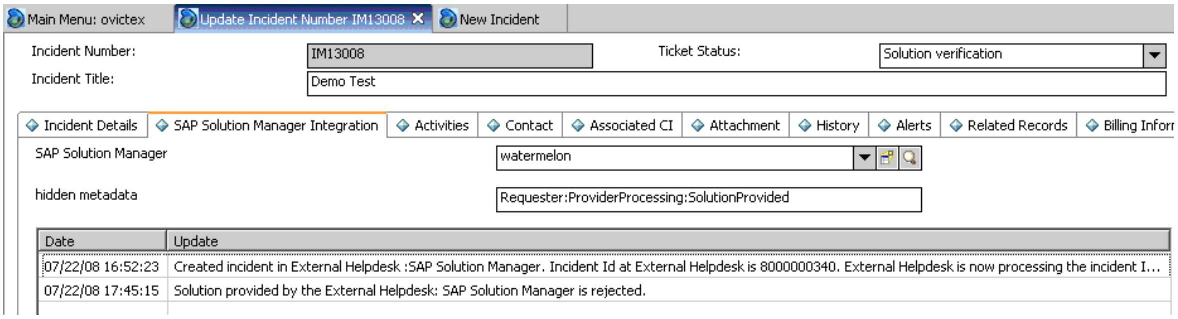
Incident Number:

Ticket Status:

Incident Title:

Subformat

Property	Value
Name	subview1213341835957
X	1
Y	0
Width	151
Height	28
Visible	☑
Visible Condition	
Format	hp.sap.solution.sub
Virtual Join	□
Display Blank	☑
Display Using Table	□
Input	



Incident Number: IM13008

Ticket Status: Solution verification

Incident Title: Demo Test

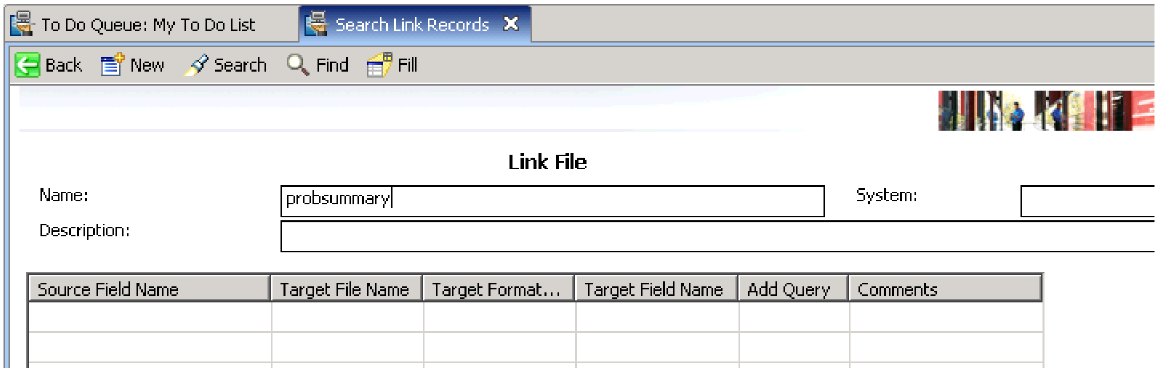
SAP Solution Manager: watermelon

hidden metadata: Requester:ProviderProcessing:SolutionProvided

Date	Update
07/22/08 16:52:23	Created incident in External Helpdesk :SAP Solution Manager. Incident Id at External Helpdesk is 8000000340. External Helpdesk is now processing the incident I...
07/22/08 17:45:15	Solution provided by the External Helpdesk: SAP Solution Manager is rejected.

5 Add a link record for field `exthd` of `probsummary` table.

a Go to **Tailoring** → **Tailoring Tools** → **Links**. Enter **`probsummary`** in Name field, then click **Search**.



Link File

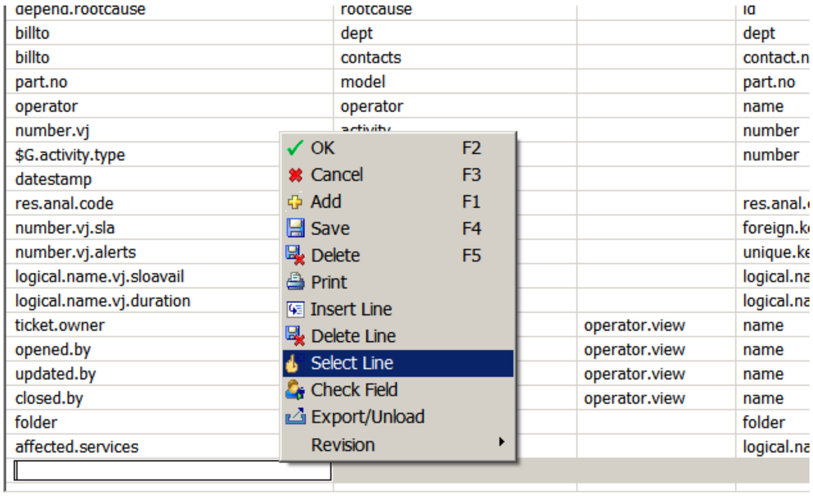
Name: probsummary

System:

Description:

Source Field Name	Target File Name	Target Format...	Target Field Name	Add Query	Comments

b Right click on an empty line in the table, select **Select Line** from the pop-up menu. The Edit Link page appears.



depend.rootcause	rootcause		id
billto	dept		dept
billto	contacts		contact.n
part.no	model		part.no
operator	operator		name
number.vj	activity		number
$G.activity.type			number
datestamp			
res.anal.code			res.anal.(
number.vj.sla			foreign.k
number.vj.alerts			unique.ke
logical.name.vj.sloavail			logical.na
logical.name.vj.duration			logical.na
ticket.owner		operator.view	name
opened.by		operator.view	name
updated.by		operator.view	name
closed.by		operator.view	name
folder			folder
affected.services			logical.na

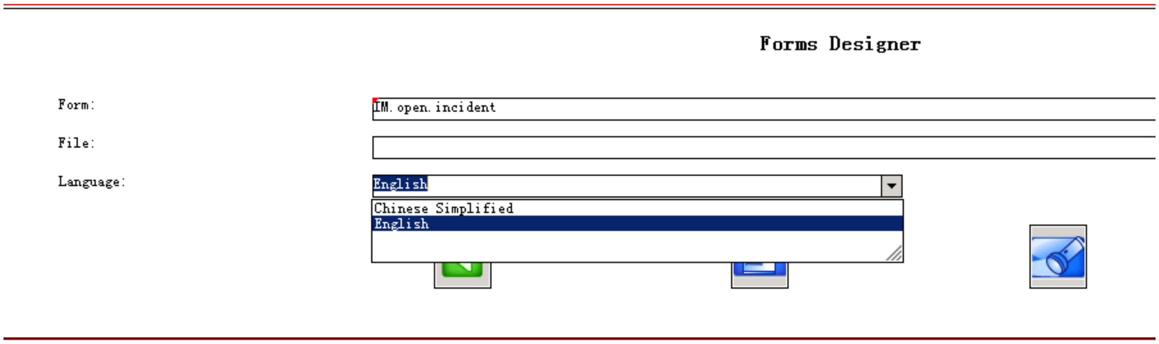
c Add the following values, and then click **Save**.

Field	Value
Field(From/Source)	exthd
File(To/Target)	HPSAPSolutionManager
Format(To/Target)	HPSAPSolutionManager
Field(To/Target)	id
Source Field (Fill To/Post From)	exthd
Target Field(Fill From/Post To)	id

➤ If multiple Language packs are applied to HP Service Manager 7.10 and above, 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.



Form: IM.open.incident

File:

Language: English

Chinese Simplified
English

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.

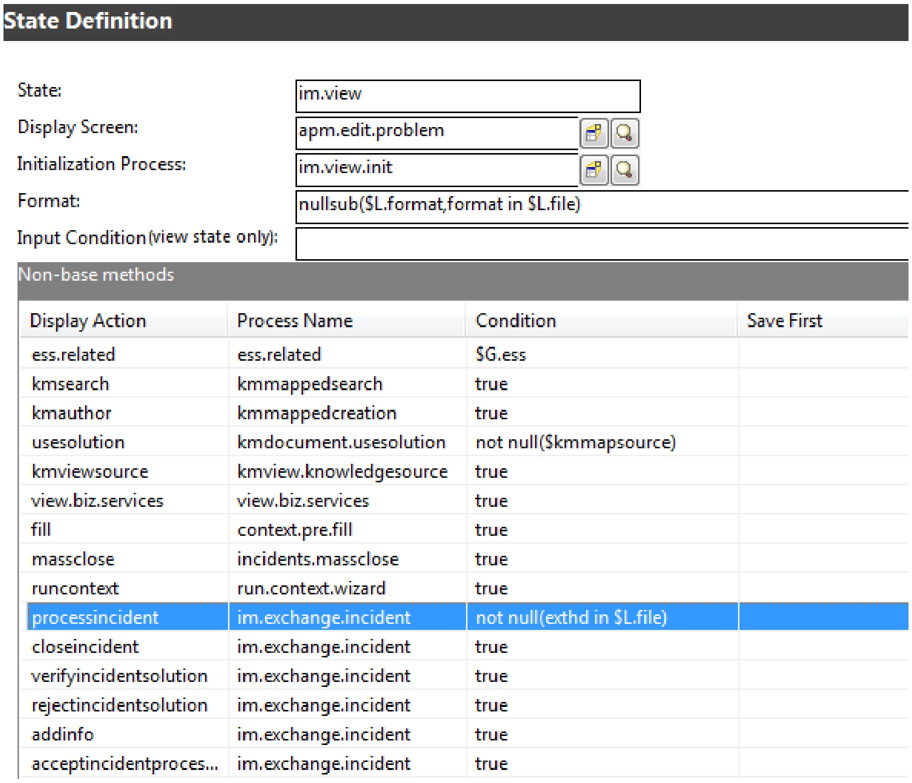
An action cannot be triggered multiple times, since the action request is sent asynchronously to the Incident Exchange. The exchange state of the Incident will only be updated (via Event In) during processing with the external helpdesk. Event In can only update the Incident that is not locked. This typically 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`.



State Definition

State: im.view

Display Screen: apm.edit.problem

Initialization Process: im.view.init

Format: nullsub($L.format,format in $L.file)

Input Condition (view state only):

Non-base methods

Display Action	Process Name	Condition	Save First
ess.related	ess.related	$G.ess	
kmsearch	kmmappedsearch	true	
kmauthor	kmmappedcreation	true	
usesolution	kmdocument.usesolution	not null($kmmapsource)	
kmviewsource	kmview.knowledgesource	true	
view.biz.services	view.biz.services	true	
fill	context.pre.fill	true	
massclose	incidents.massclose	true	
runcontext	run.context.wizard	true	
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	
acceptincidentproces...	im.exchange.incident	true	

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

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

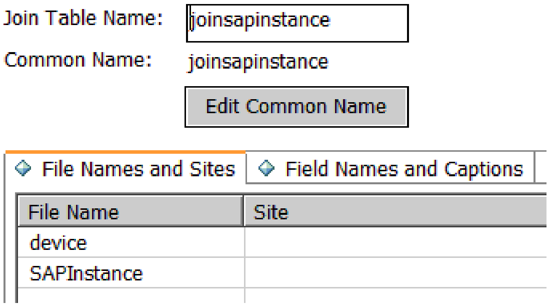
The following describes an example implementation of a new device type `SAPInstance` (created via **Configuration Management** → **Administration** → **Add New Device Type**). This device type needs fields for SAP SID, client and installation number. A new table SAPInstance should be created in advance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

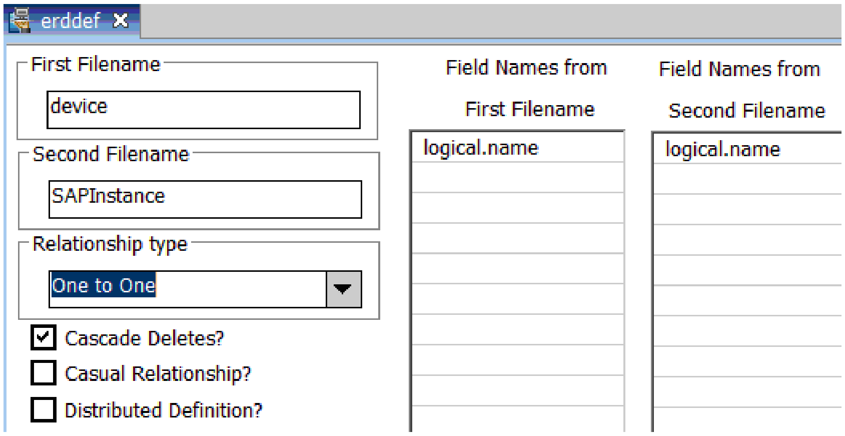


Join Table Name: joinsapinstance

Common Name: joinsapinstance

File Name	Site
device	
SAPInstance	

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



First Filename: device

Second Filename: SAPInstance

Relationship type: One to One

☑ Cascade Deletes?

☐ Casual Relationship?

☐ Distributed Definition?

Field Names from First Filename	Field Names from Second Filename
logical.name	logical.name

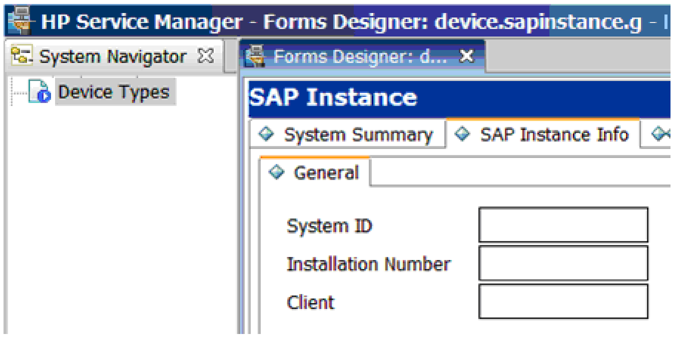
3 Create a form view for the device type you want to create in Service Manager.

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 7.0x*

Go to **Tailoring** → **Form Designer**, enter **`device.template.g`** 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`. Delete tabs Example Info, Financial, Scanner, and Subscribers.

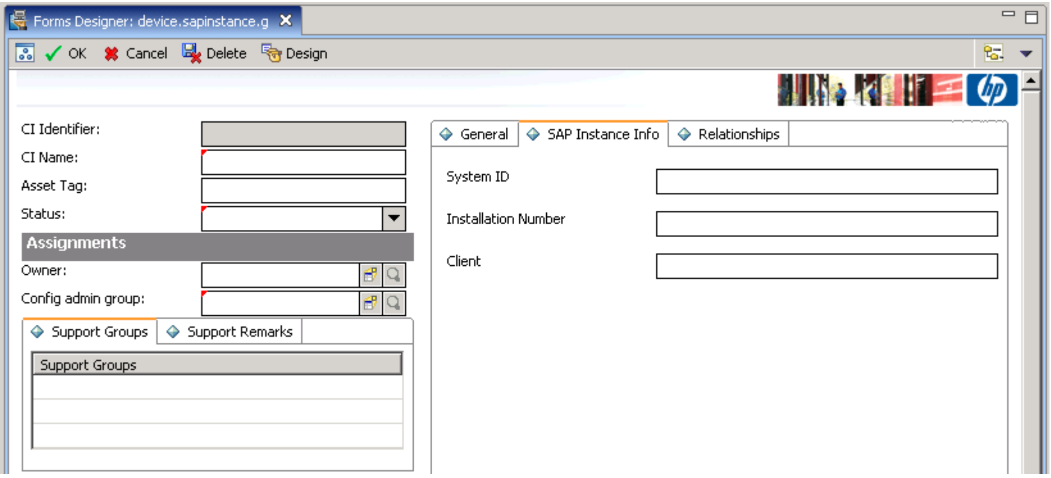
Then add a new tab **SAP Instance Info** in the form:



— *On Service Manager 7.10 and above*

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:

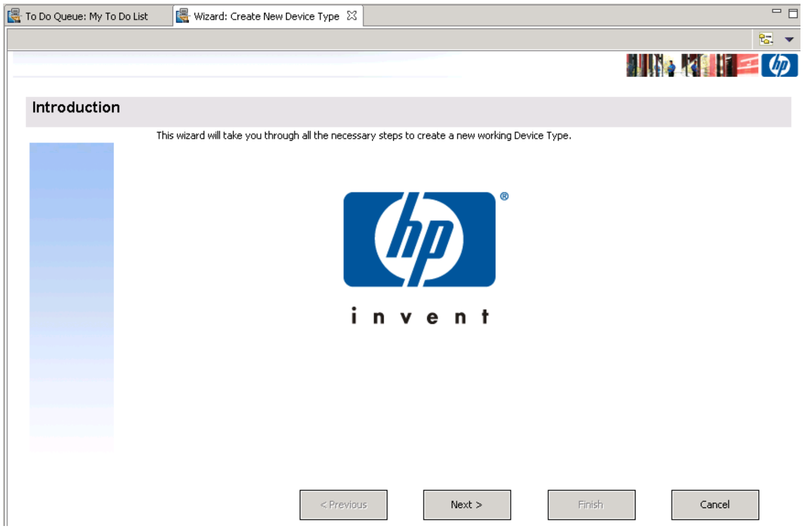


The SAP Instance Info tab form should include at least three fields: System ID, Installation Number, and Client.

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

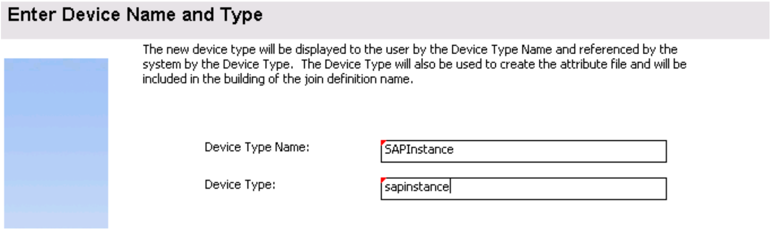
4 Add a new device type in Service Manager.

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



This wizard will take you through all the necessary steps to create a new working Device Type.

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



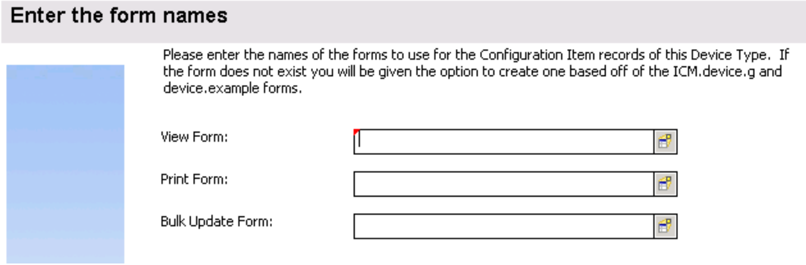
Enter Device Name and Type

The new device type will be displayed to the user by the Device Type Name and referenced by the system by the Device Type. The Device Type will also be used to create the attribute file and will be included in the building of the join definition name.

Device Type Name: SAPInstance

Device Type: sapinstance

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



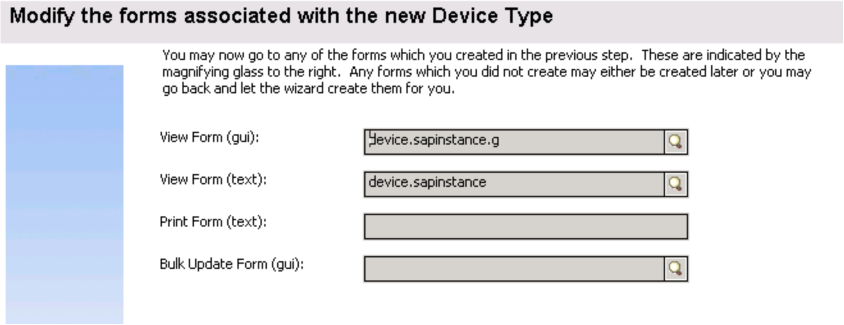
Enter the form names

Please enter the names of the forms to use for the Configuration Item records of this Device Type. If the form does not exist you will be given the option to create one based off of the ICM.device.g and device.example forms.

View Form:

Print Form:

Bulk Update Form:



Modify the forms associated with the new Device Type

You may now go to any of the forms which you created in the previous step. These are indicated by the magnifying glass to the right. Any forms which you did not create may either be created later or you may go back and let the wizard create them for you.

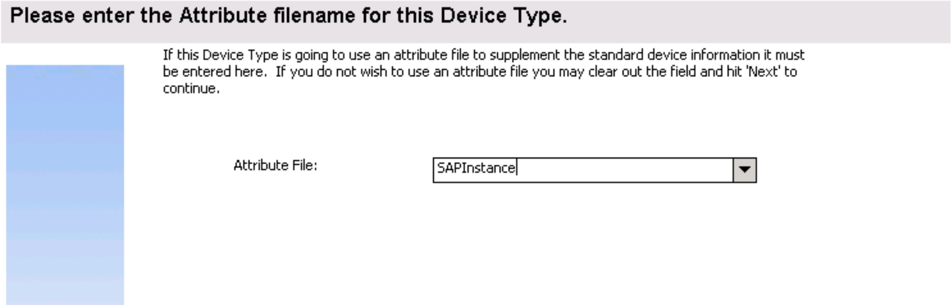
View Form (gui): device.sapinstance.g

View Form (text): device.sapinstance

Print Form (text):

Bulk Update Form (gui):

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

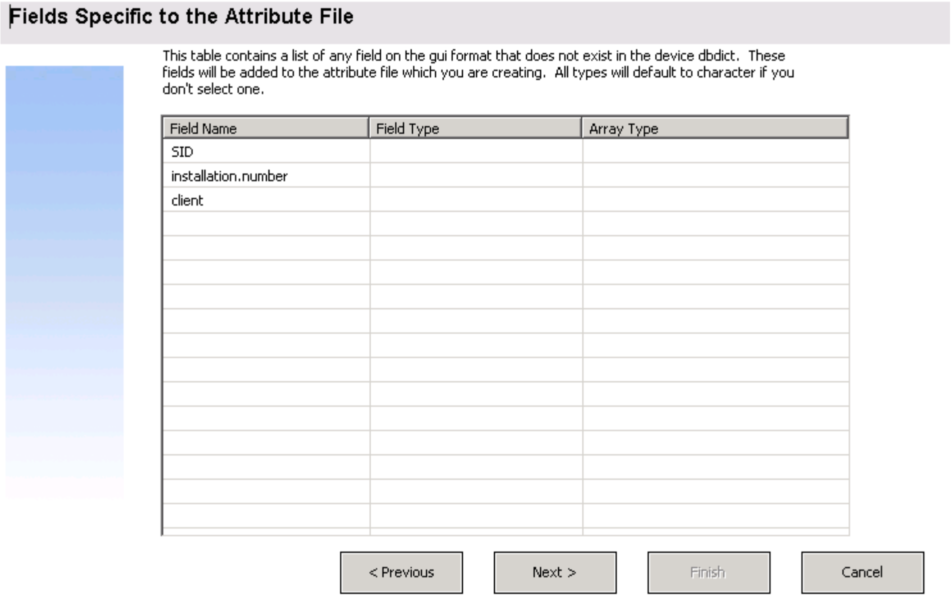


Please enter the Attribute filename for this Device Type.

If this Device Type is going to use an attribute file to supplement the standard device information it must be entered here. If you do not wish to use an attribute file you may clear out the field and hit 'Next' to continue.

Attribute File: SAPInstance

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

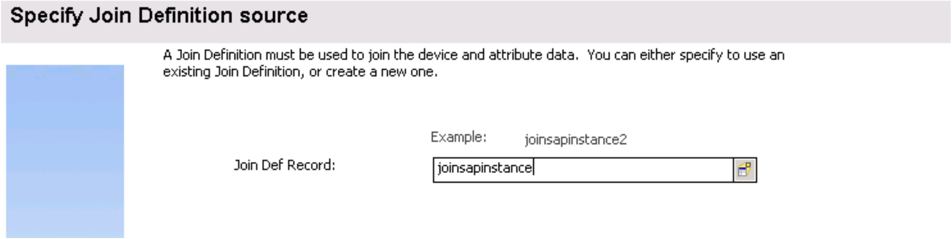


Fields Specific to the Attribute File

This table contains a list of any field on the gui format that does not exist in the device dbdict. These fields will be added to the attribute file which you are creating. All types will default to character if you don't select one.

Field Name	Field Type	Array Type
SID		
installation.number		
client		

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



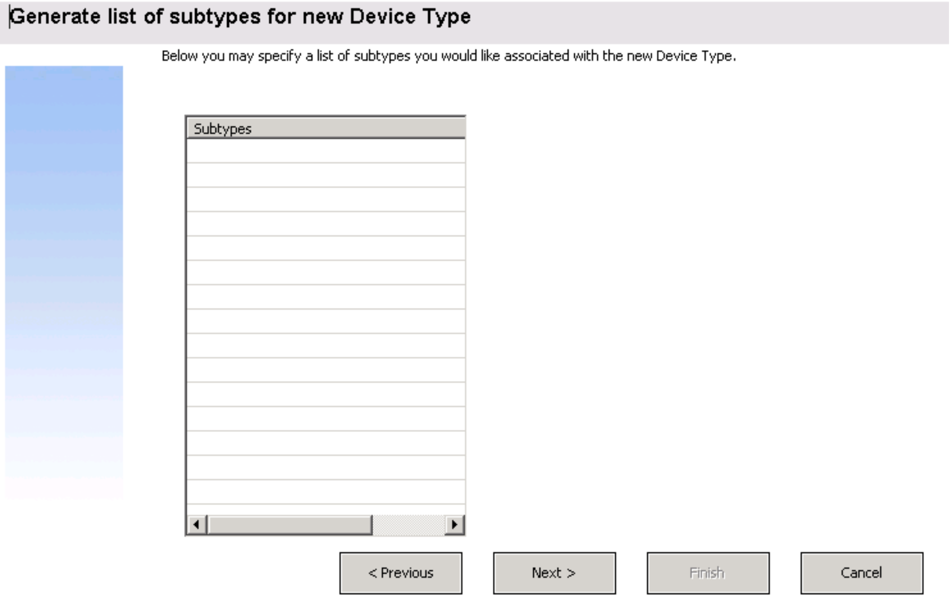
Specify Join Definition source

A Join Definition must be used to join the device and attribute data. You can either specify to use an existing Join Definition, or create a new one.

Example: joinsapinstance2

Join Def Record: joinsapinstance

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

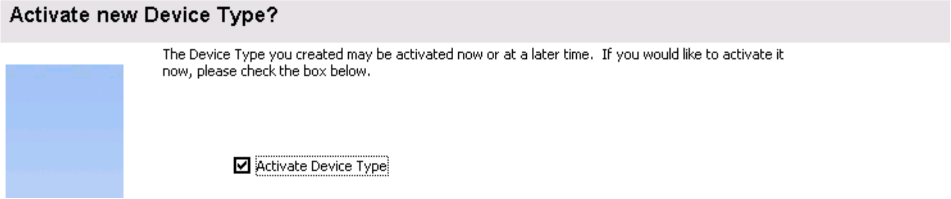


Generate list of subtypes for new Device Type

Below you may specify a list of subtypes you would like associated with the new Device Type.

Subtypes

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



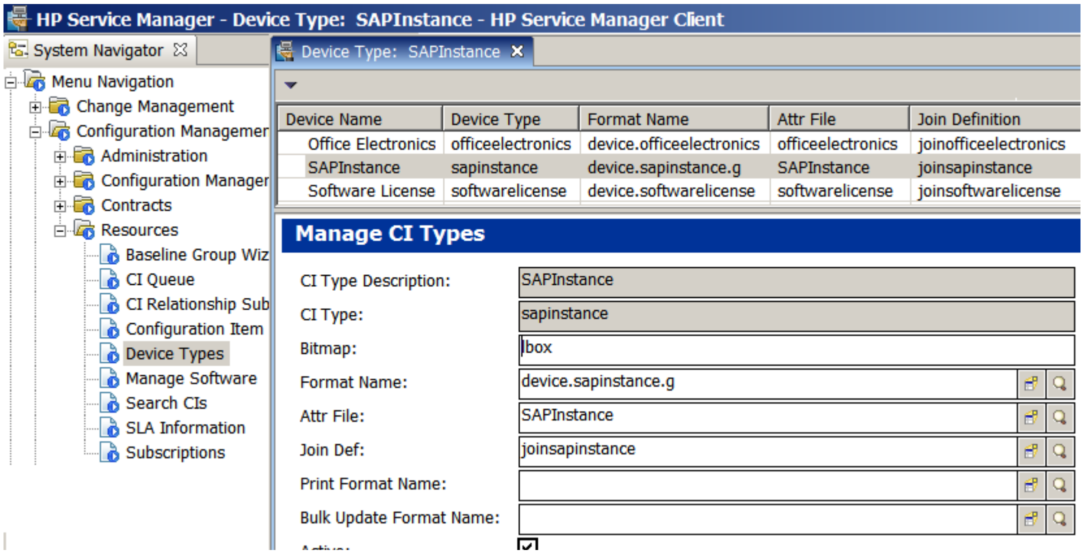
Activate new Device Type?

The Device Type you created may be activated now or at a later time. If you would like to activate it now, please check the box below.

☑ Activate Device Type

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

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



HP Service Manager - Device Type: SAPInstance - HP Service Manager Client

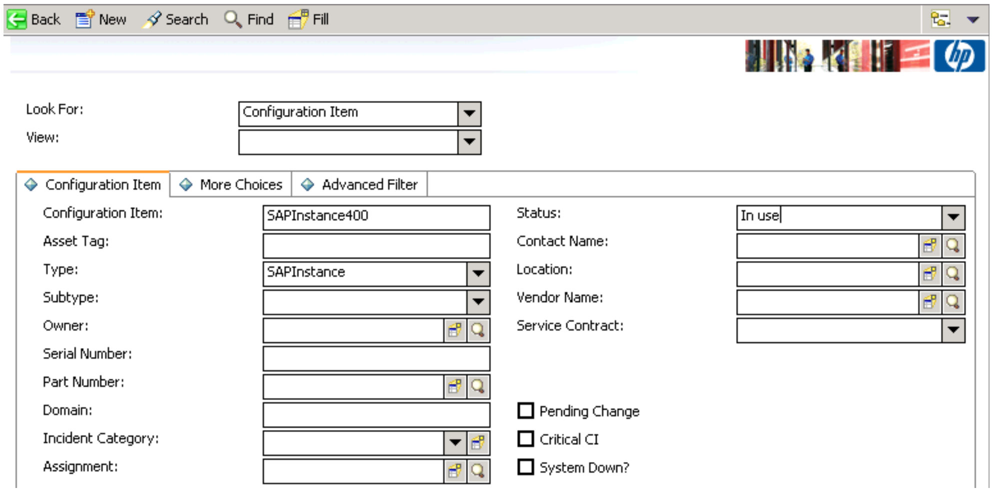
Device Name	Device Type	Format Name	Attr File	Join Definition
Office Electronics	officeelectronics	device.officeelectronics	officeelectronics	joinofficeelectronics
SAPInstance	sapinstance	device.sapinstance.g	SAPInstance	joinsapinstance
Software License	softwarelicense	device.softwarelicense	softwarelicense	joinsoftwarelicense

Manage CI Types

CI Type Description:	SAPInstance
CI Type:	sapinstance
Bitmap:	box
Format Name:	device.sapinstance.g
Attr File:	SAPInstance
Join Def:	joinsapinstance
Print Format Name:	
Bulk Update Format Name:	

6 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.

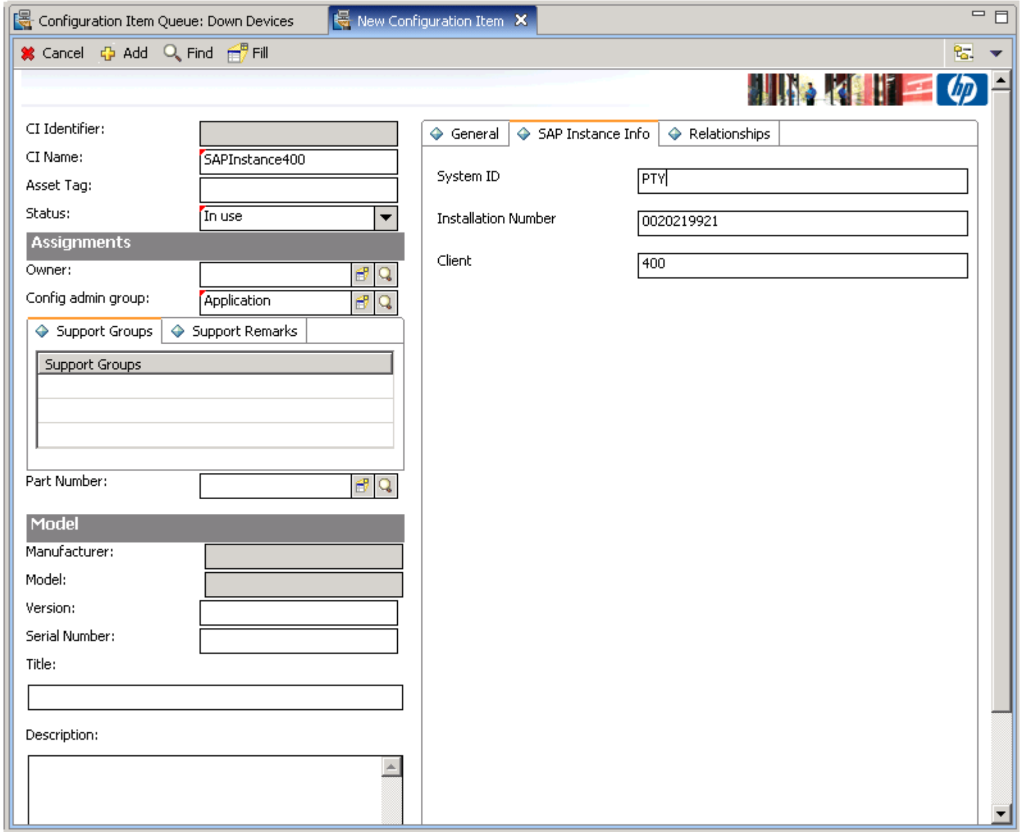


Look For: Configuration Item

View:

Configuration Item:	SAPInstance400	Status:	In use
Asset Tag:		Contact Name:	
Type:	SAPInstance	Location:	
Subtype:		Vendor Name:	
Owner:		Service Contract:	
Serial Number:			
Part Number:			
Domain:		☐ Pending Change	
Incident Category:		☐ Critical CI	
Assignment:		☐ System Down?	

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



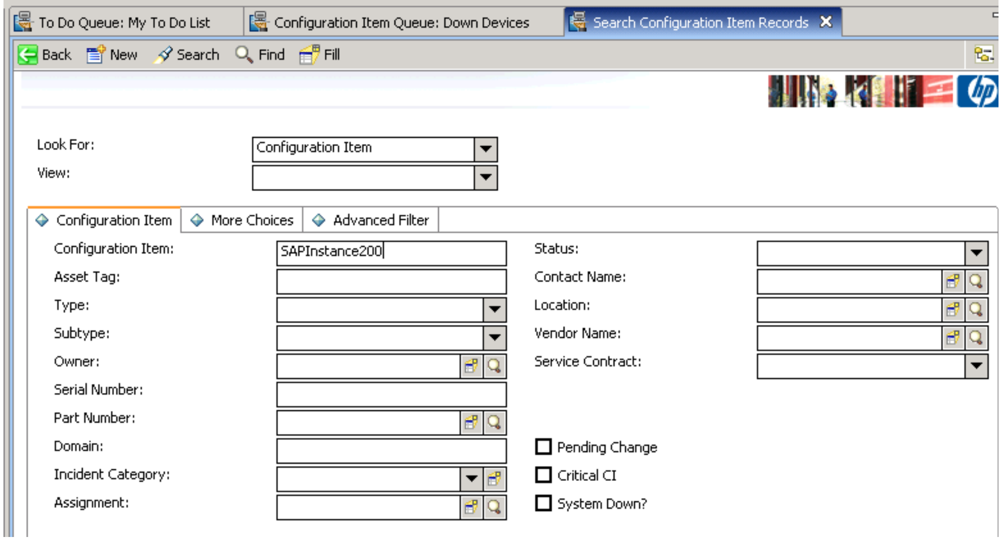
CI Identifier:
CI Name: SAPInstance400
Asset Tag:
Status: In use
Owner:
Config admin group: Application
Part Number:
Manufacturer:
Model:
Version:
Serial Number:
Title:
Description:

System ID: PTY
Installation Number: 0020219921
Client: 400

c Add relationship for the newly added SAP device item.

> This step is required for Service Manager 7.10 and above. If you are using a lower version, skip the following and jump to step 7 on page 57 directly.

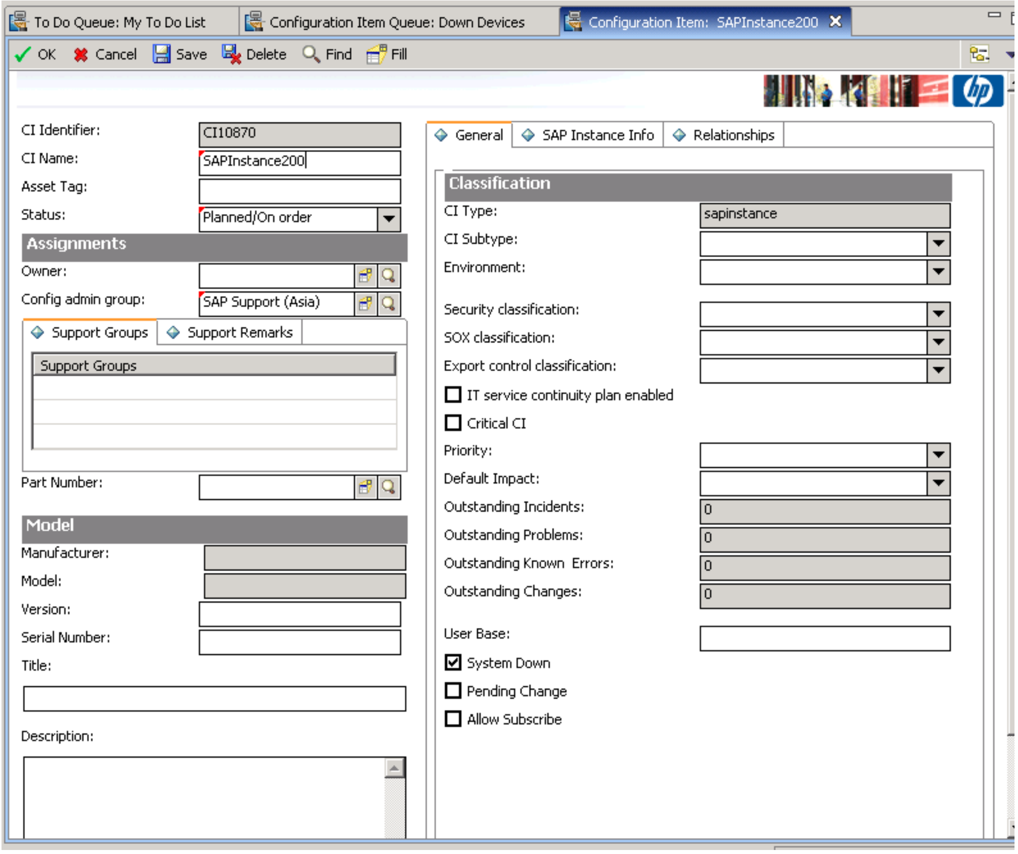
Go to **Configuration Management** → **Resources** → **CI Queue** → **New**. Enter the newly added SAP device item name in Configuration Item field, for example, **`SAPInstance200`**, and then click **Search**.


To Do Queue: My To Do List
Configuration Item Queue: Down Devices
Search Configuration Item Records
Back
New
Search
Find
Fill
Configuration Item
More Choices
Advanced Filter

Look For: Configuration Item
View:

Configuration Item: SAPInstance200
Asset Tag:
Type:
Subtype:
Owner:
Serial Number:
Part Number:
Domain:
Incident Category:
Assignment:
Status:
Contact Name:
Location:
Vendor Name:
Service Contract:
□ Pending Change
□ Critical CI
□ System Down?

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


To Do Queue: My To Do List
Configuration Item Queue: Down Devices
Configuration Item: SAPInstance200
OK
Cancel
Save
Delete
Find
Fill
General
SAP Instance Info
Relationships
Support Groups
Support Remarks

CI Identifier: CI10870
CI Name: SAPInstance200
Asset Tag:
Status: Planned/On order

Assignments

Owner:
Config admin group: SAP Support (Asia)

Support Groups

Part Number:

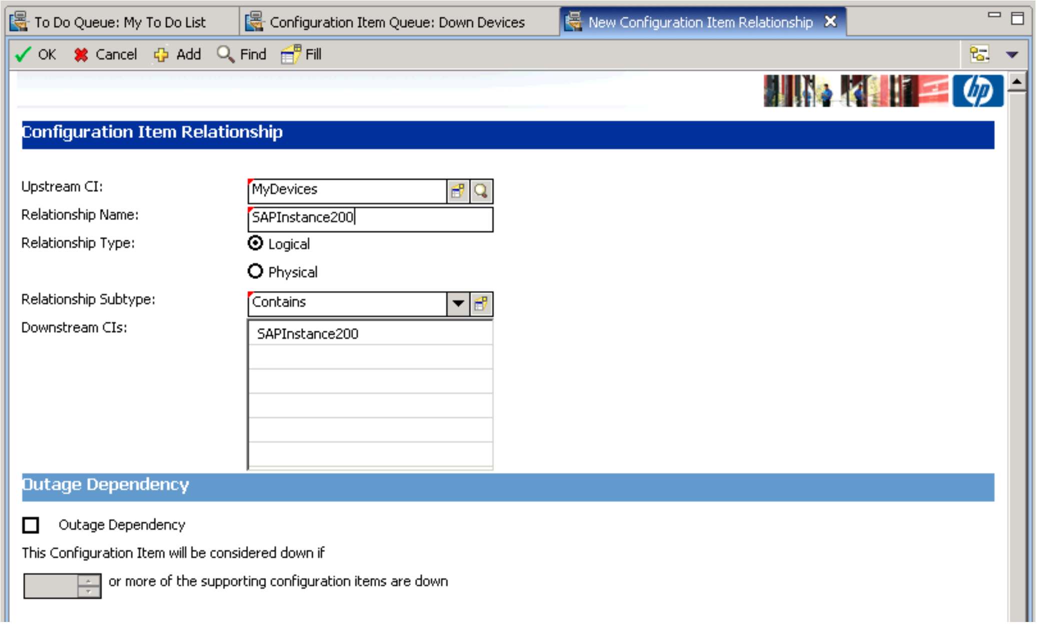
Model

Manufacturer:
Model:
Version:
Serial Number:
Title:
Description:

Classification

CI Type: sapinstance
CI Subtype:
Environment:
Security classification:
SOX classification:
Export control classification:
□ IT service continuity plan enabled
□ Critical CI
Priority:
Default Impact:
Outstanding Incidents: 0
Outstanding Problems: 0
Outstanding Known Errors: 0
Outstanding Changes: 0
User Base:
☑ System Down
□ Pending Change
□ Allow Subscribe

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


To Do Queue: My To Do List
Configuration Item Queue: Down Devices
New Configuration Item Relationship
OK
Cancel
Add
Find
Fill

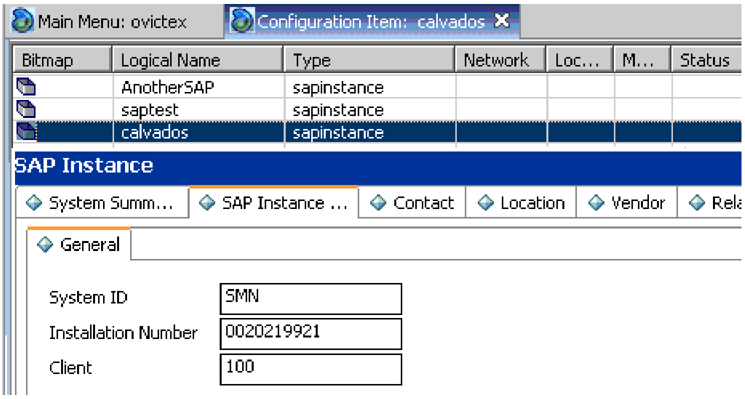
Configuration Item Relationship

Upstream CI: MyDevices
Relationship Name: SAPInstance200
Relationship Type: ● Logical ○ Physical
Relationship Subtype: Contains
Downstream CIs: SAPInstance200

Outage Dependency

□ Outage Dependency
This Configuration Item will be considered down if
or more of the supporting configuration items are down

7 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.



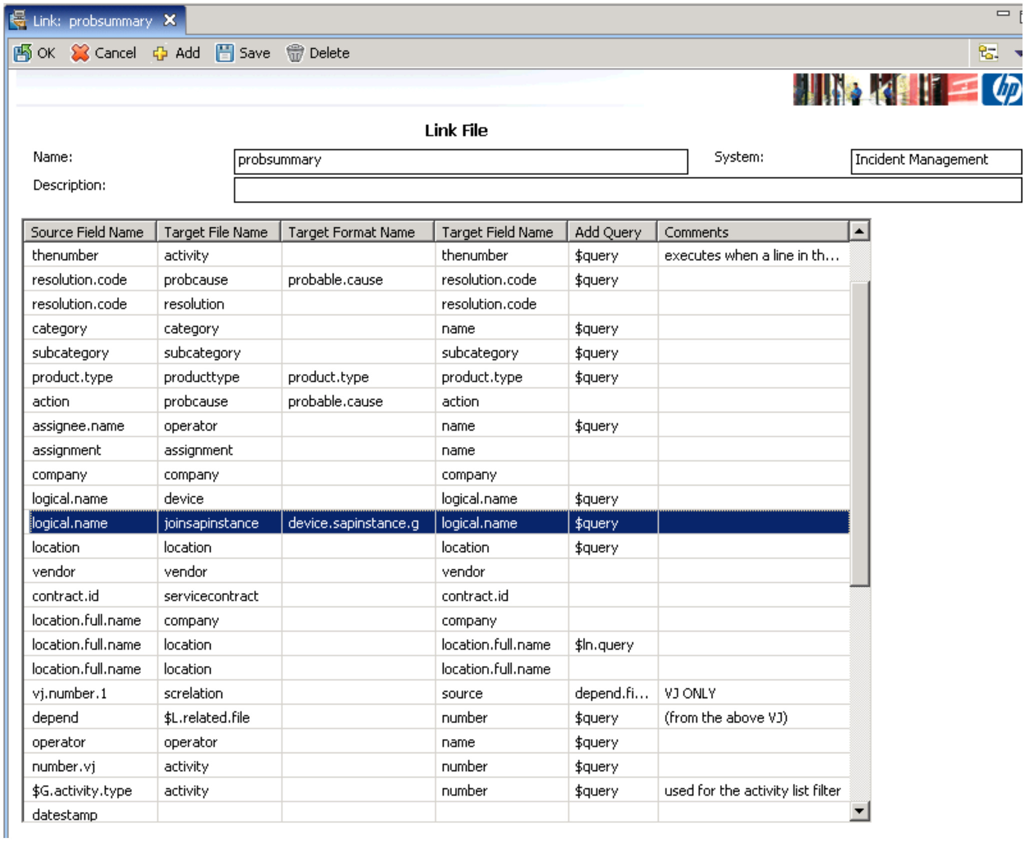
Bitmap	Logical Name	Type	Network	Loc...	M...	Status
	AnotherSAP	sapinstance				
	saptest	sapinstance				
	calvados	sapinstance				

SAP Instance

System ID	SMN
Installation Number	0020219921
Client	100

8 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`**.

9 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`.



Link File

Name:	probsummary	System:	Incident Management
Description:			

Source Field Name	Target File Name	Target Format Name	Target Field Name	Add Query	Comments
thenumber	activity		thenumber	$query	executes when a line in th...
resolution.code	probcause	probable.cause	resolution.code	$query	
resolution.code	resolution		resolution.code		
category	category		name	$query	
subcategory	subcategory		subcategory	$query	
product.type	producttype	product.type	product.type	$query	
action	probcause	probable.cause	action		
assignee.name	operator		name	$query	
assignment	assignment		name		
company	company		company		
logical.name	device		logical.name	$query	
logical.name	joinsapinstance	device.sapinstance.g	logical.name	$query	
location	location		location	$query	
vendor	vendor		vendor		
contract.id	servicecontract		contract.id		
location.full.name	company		company		
location.full.name	location		location.full.name	$ln.query	
location.full.name	location		location.full.name		
vj.number.1	screlation		source	depend.fi...	VJ ONLY
depend	$L.related.file		number	$query	(from the above VJ)
operator	operator		name	$query	
number.vj	activity		number	$query	
$G.activity.type	activity		number	$query	used for the activity list filter
datestamp					

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

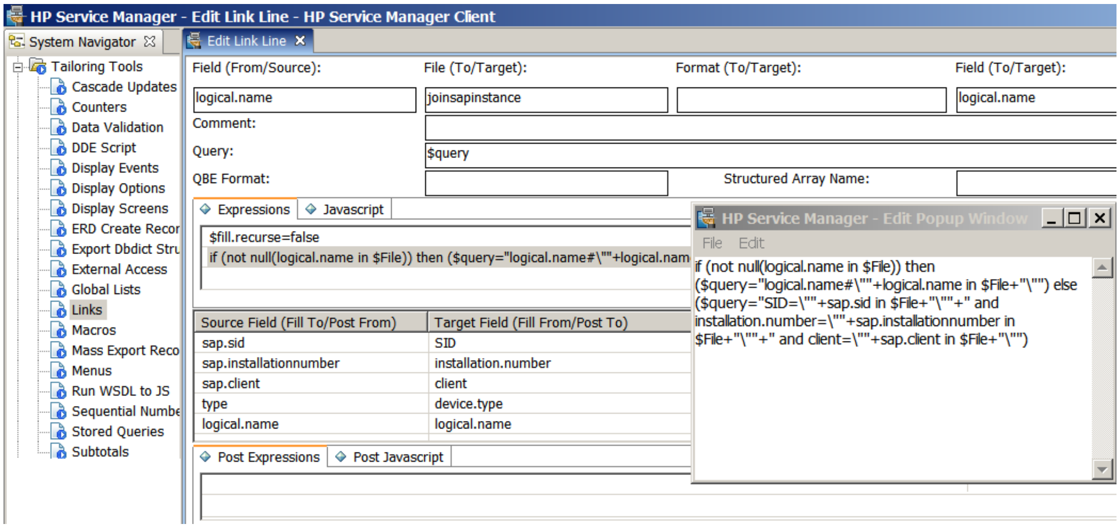
In Expressions tab, add the following two lines (**[LOGICAL.NAME_JOINSAPINSTANCE_EXPR]** in `code_sm7.txt`):

```
$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
type	device.type
logical.name	logical.name

On Service Manager 7.0x:



Field (From/Source): logical.name

File (To/Target): joinsapinstance

Format (To/Target):

Field (To/Target): logical.name

Comment:

Query: $query

QBE Format:

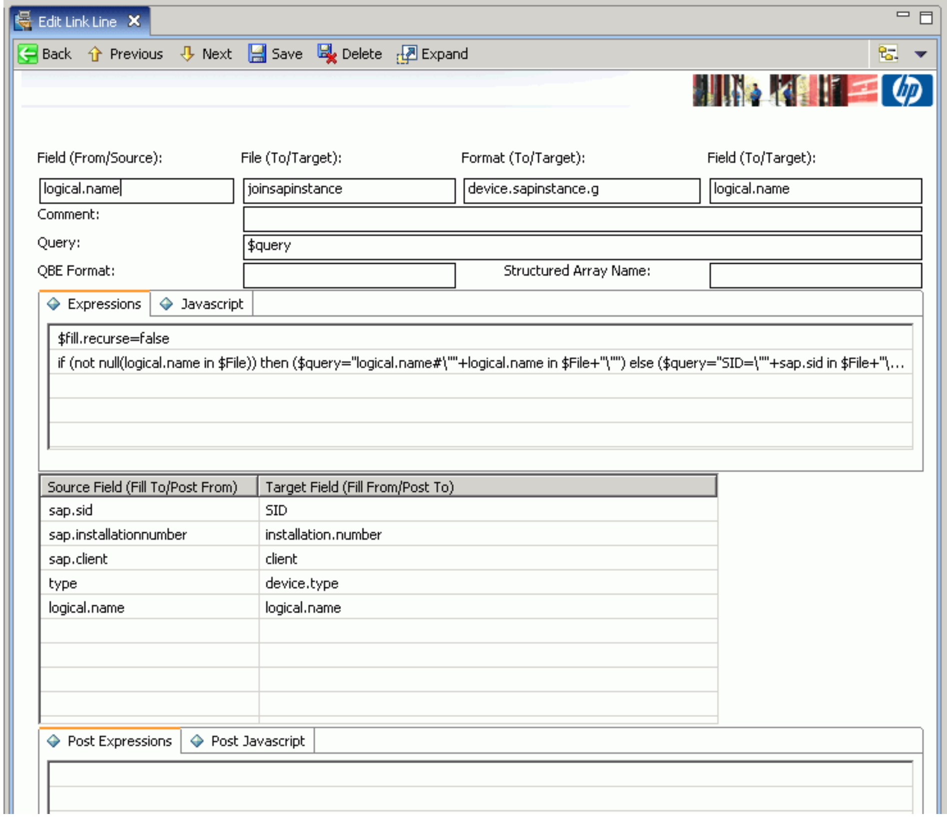
Structured Array Name:

```
$fill.recurse=false
if (not null(logical.name in $File)) then ($query="logical.name#\""+logical.nam
```

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

```
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+"\"")
```

On Service Manager 7.10 and above:



Field (From/Source):	File (To/Target):	Format (To/Target):	Field (To/Target):
logical.name	joinsapinstance	device.sapinstance.g	logical.name

Comment:

Query: $query

QBE Format:

Structured Array Name:

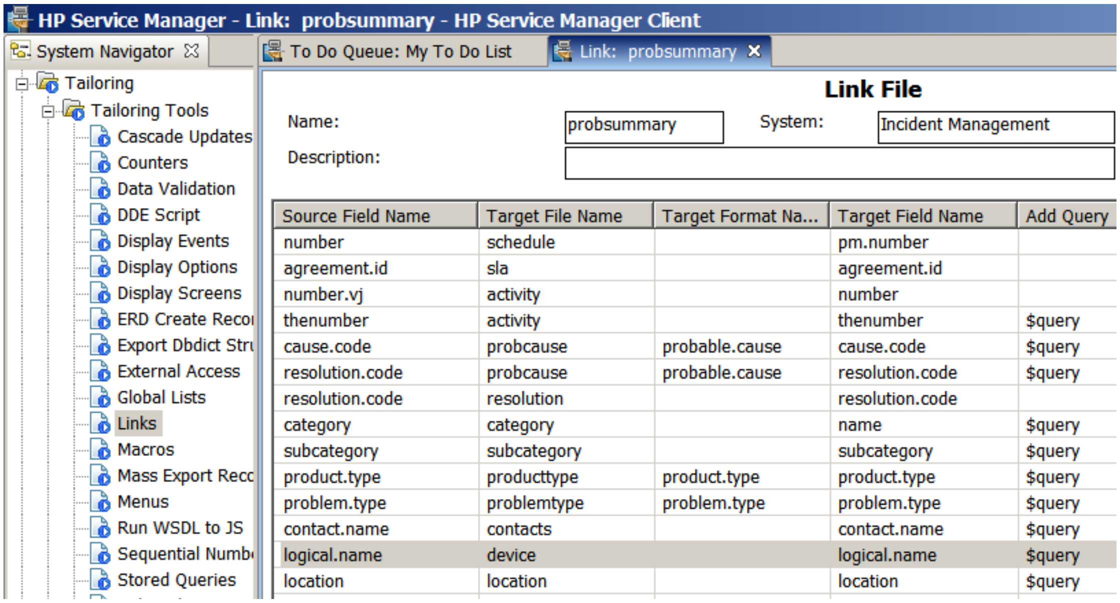
Expressions | Javascript

```
$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+"\...
```

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

Post Expressions | Post Javascript

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



Link File

Name: probsummary

System: Incident Management

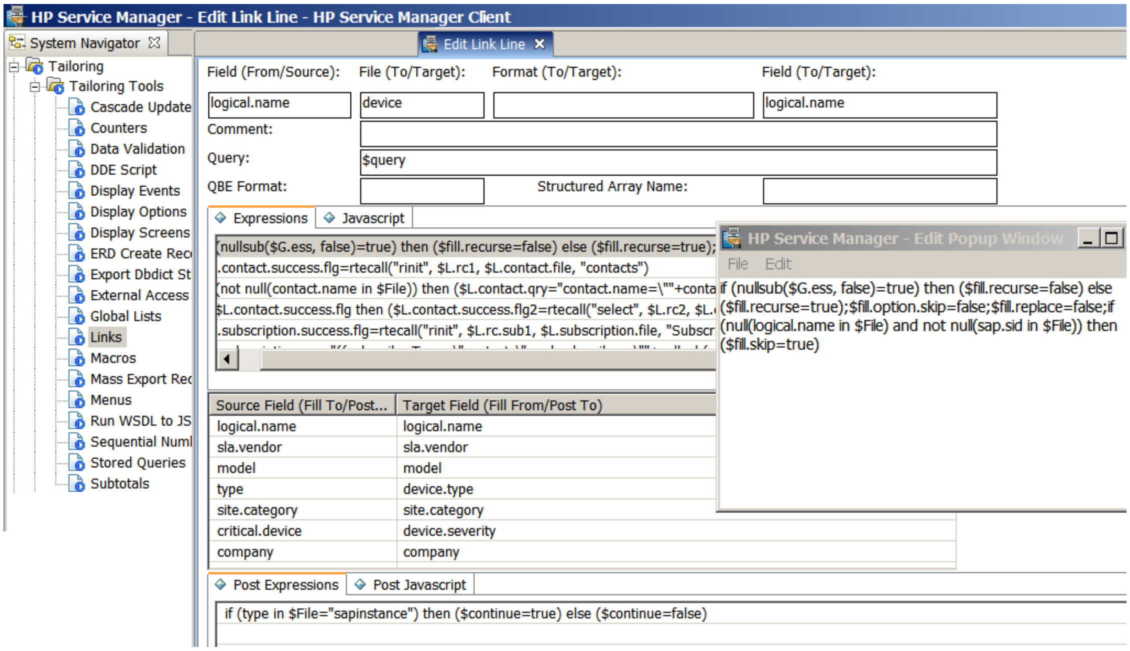
Description:

Source Field Name	Target File Name	Target Format Na...	Target Field Name	Add Query
number	schedule		pm.number	
agreement.id	sla		agreement.id	
number.vj	activity		number	
thenumber	activity		thenumber	$query
cause.code	probcause	probable.cause	cause.code	$query
resolution.code	probcause	probable.cause	resolution.code	$query
resolution.code	resolution		resolution.code	
category	category		name	$query
subcategory	subcategory		subcategory	$query
product.type	producttype	product.type	product.type	$query
problem.type	problemtype	problem.type	problem.type	$query
contact.name	contacts		contact.name	$query
logical.name	device		logical.name	$query
location	location		location	$query

12 Replace the first line with the following expression for Service Manager 7.0x; or, insert the following expression on Service Manager 7.10 and above (**[LOGICAL.NAME_DEVICE_EXPR]** in `code_sm7.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 7.0x:



Field (From/Source): logical.name
File (To/Target): device
Format (To/Target):
Field (To/Target): logical.name
Comment:
Query: $query
QBE Format:
Structured Array Name:

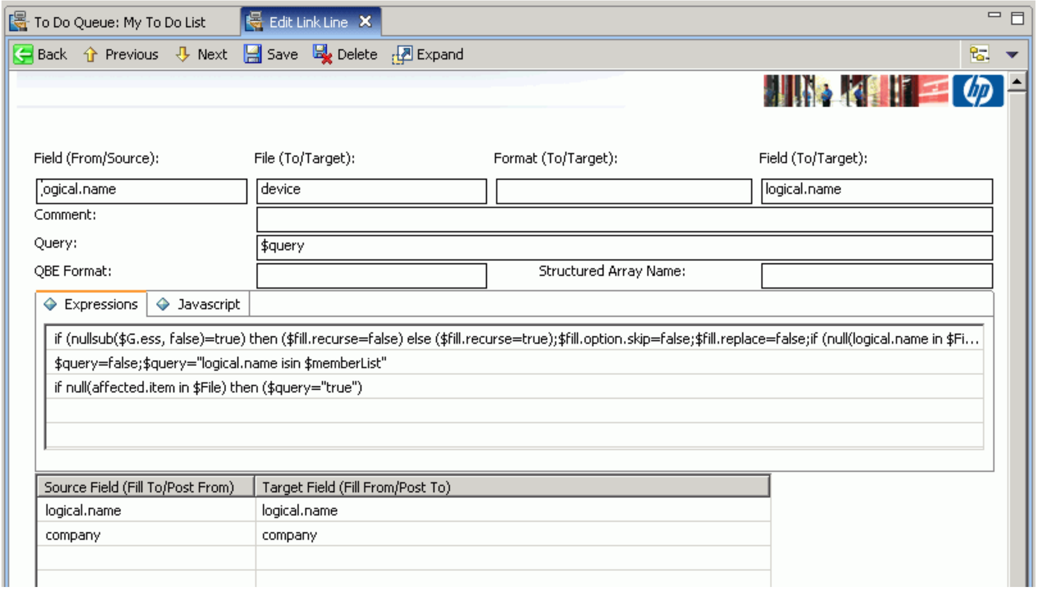
```
(nullsub($G.ess, false)=true) then ($fill.recurse=false) else ($fill.recurse=true);
.contact.success.flg=rtecall("rinit", $L.rc1, $L.contact.file, "contacts")
(not null(contact.name in $File)) then ($L.contact.qry="contact.name=\""+conta
$L.contact.success.flg then ($L.contact.success.flg2=rtecall("select", $L.rc2, $L.
.subscription.success.flg=rtecall("rinit", $L.rc.sub1, $L.subscription.file, "Subscr
```

```
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)
```

Source Field (Fill To/Post...	Target Field (Fill From/Post To)
logical.name	logical.name
sla.vendor	sla.vendor
model	model
type	device.type
site.category	site.category
critical.device	device.severity
company	company

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

On Service Manager 7.10 and above:



Field (From/Source): logical.name
File (To/Target): device
Format (To/Target):
Field (To/Target): logical.name
Comment:
Query: $query
QBE Format:
Structured Array Name:

```
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 $Fi...
$query=false;$query="logical.name isin $memberList"
if null(affected.item in $File) then ($query="true")
```

Source Field (Fill To/Post From)	Target Field (Fill From/Post To)
logical.name	logical.name
company	company

13 In the Post Expressions tab add the following expression ():

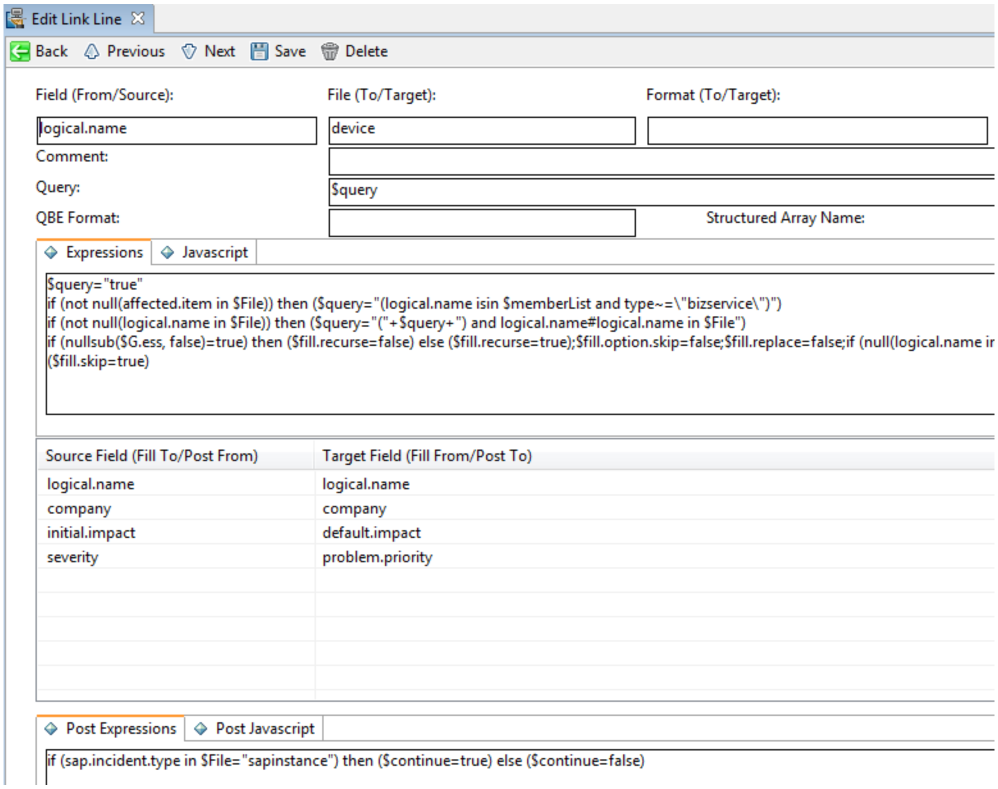
On Service Manager 7.0x ([LOGICAL.NAME_DEVICE_POST] in `code_sm7.txt`*):*

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

On Service Manager 7.10 and above ([LOGICAL.NAME_DEVICE_POST_7.10] in `code_sm7.txt`*):*

```
if (affected.item in $File="MyDevices") then ($continue=true) else
($continue=false)
```

On Service Manager 7.0x:



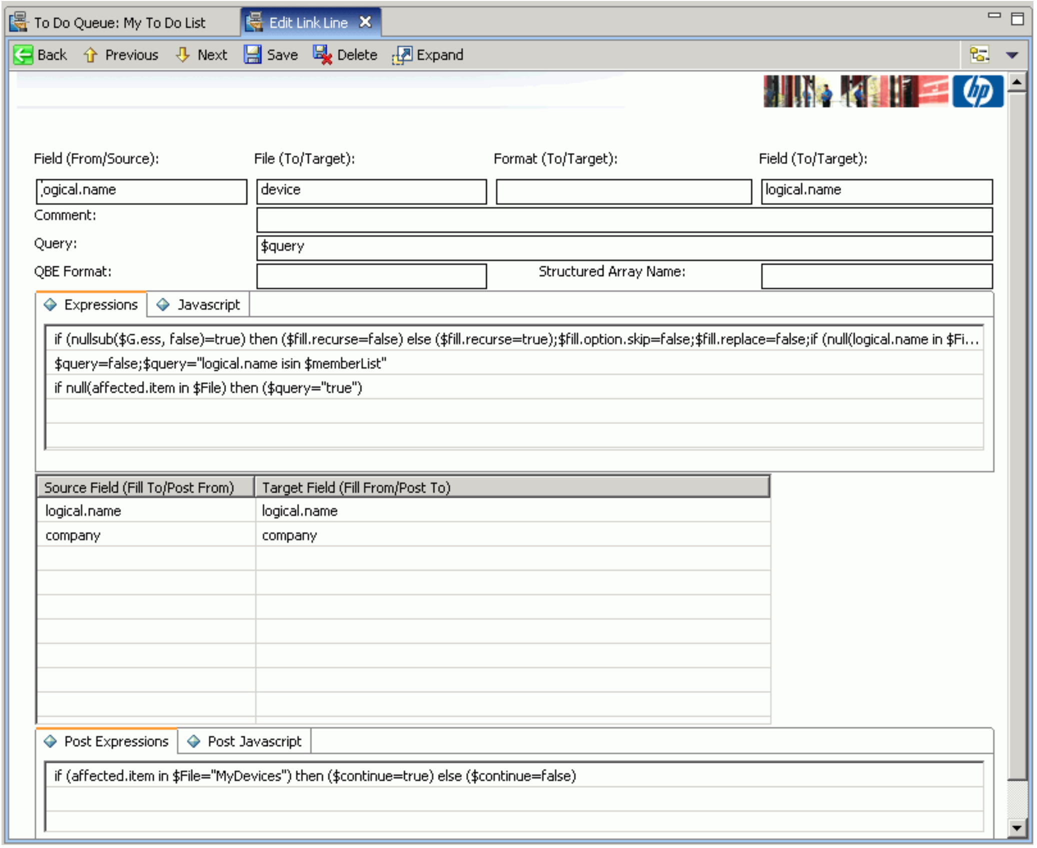
Field (From/Source):	File (To/Target):	Format (To/Target):
logical.name	device	
Comment:		
Query:	$query	
QBE Format:		Structured Array Name:

```
$query="true"
if (not null(affected.item in $File)) then ($query="(logical.name isin $memberList and type~=\"bizservice\")")
if (not null(logical.name in $File)) then ($query="("+$query+") and logical.name#logical.name in $File")
if (nullsub($G.ess, false)=true) then ($fill.recurse=false) else ($fill.recurse=true);$fill.option.skip=false;$fill.replace=false;if (null(logical.name ir
($fill.skip=true)
```

Source Field (Fill To/Post From)	Target Field (Fill From/Post To)
logical.name	logical.name
company	company
initial.impact	default.impact
severity	problem.priority

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

On Service Manager 7.10 and above:



Field (From/Source): logical.name

File (To/Target): device

Format (To/Target):

Field (To/Target): logical.name

Comment:

Query: $query

QBE Format:

Structured Array Name:

Expressions | Javascript

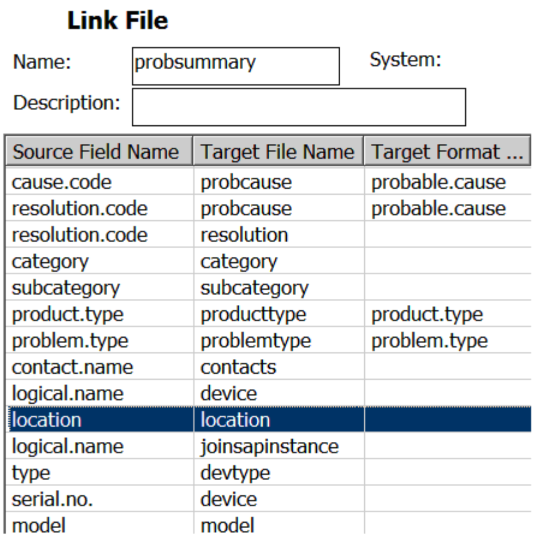
```
if (nullsub($G.ess, false)=true) then ($fill.recurse=false) else ($fill.recurse=true);$fill.option.skip=false;$fill.replace=false;if (null(logical.name in $Fi...
$query=false;$query="logical.name isin $memberList"
if null(affected.item in $File) then ($query="true")
```

Source Field (Fill To/Post From)	Target Field (Fill From/Post To)
logical.name	logical.name
company	company

Post Expressions | Post Javascript

```
if (affected.item in $File="MyDevices") then ($continue=true) else ($continue=false)
```

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



Link File

Name: probsummary

System:

Description:

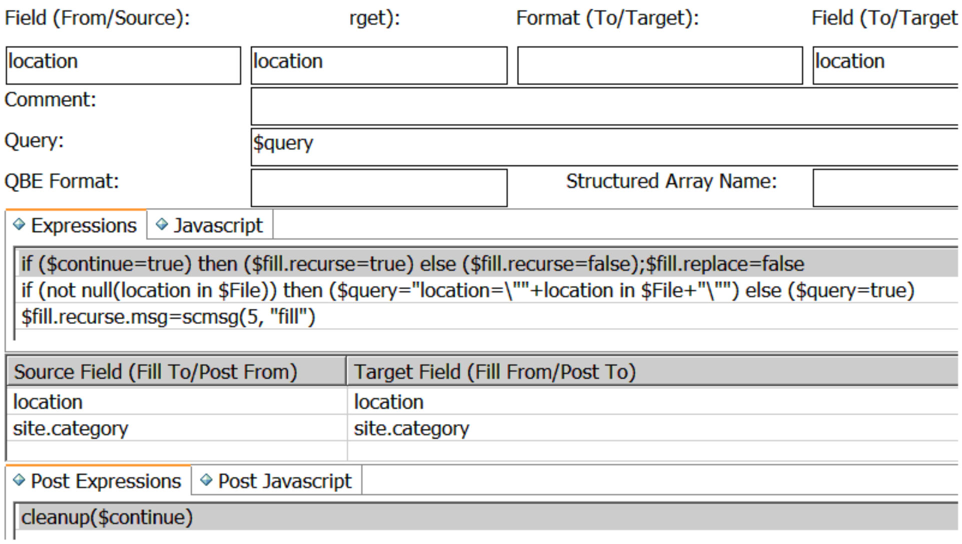
Source Field Name	Target File Name	Target Format ...
cause.code	probcause	probable.cause
resolution.code	probcause	probable.cause
resolution.code	resolution	
category	category	
subcategory	subcategory	
product.type	producttype	product.type
problem.type	problemtype	problem.type
contact.name	contacts	
logical.name	device	
location	location	
logical.name	joinsapinstance	
type	devtype	
serial.no.	device	
model	model	

15 In the Expressions tab, replace the first line with the following expression (**[LOCATION_LOCATION_EXPR]** in `code_sm7.txt`):

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

16 In the Post Expressions tab add the following expression:

`cleanup($continue)`



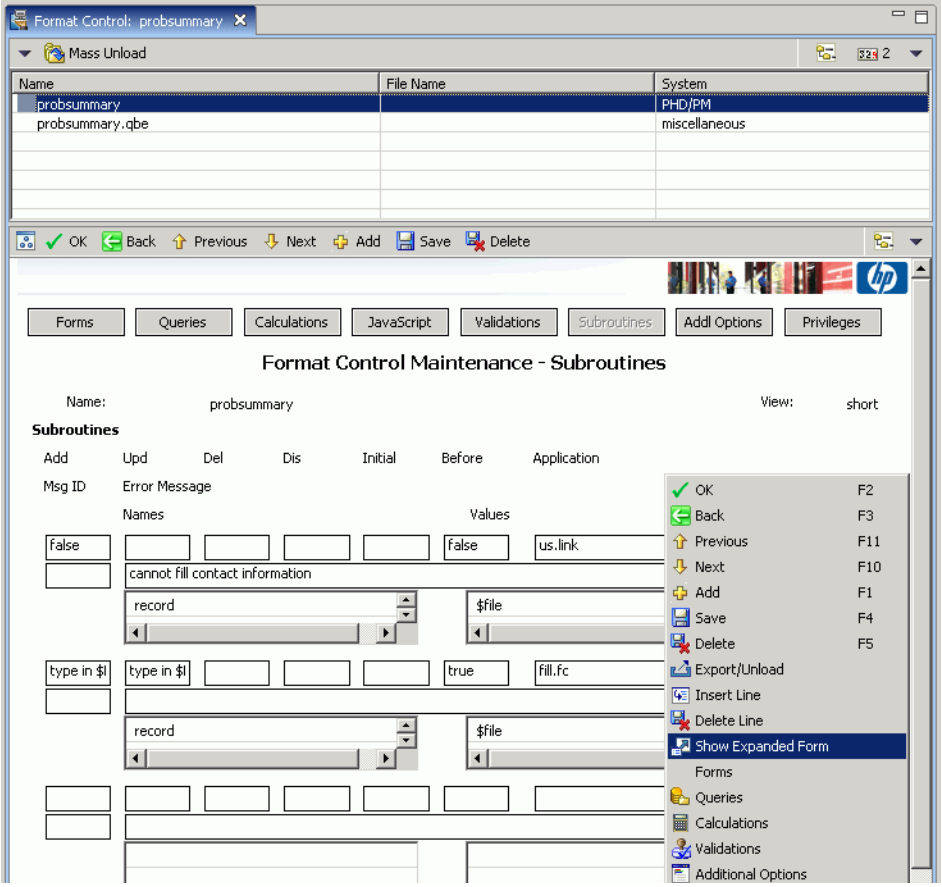
Field (From/Source):	rget):	Format (To/Target):	Field (To/Target
location	location		location
Comment:			
Query:	$query		
QBE Format:		Structured Array Name:	

```
if ($continue=true) then ($fill.recurse=true) else ($fill.recurse=false);$fill.replace=false
if (not null(location in $File)) then ($query="location=\""+location in $File+"\"") else ($query=true)
$fill.recurse.msg=scmsg(5, "fill")
```

Source Field (Fill To/Post From)	Target Field (Fill From/Post To)
location	location
site.category	site.category

```
cleanup($continue)
```

17 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.



Name	File Name	System
probsummary		PHD/PM
probsummary.qbe		miscellaneous

Format Control Maintenance - Subroutines

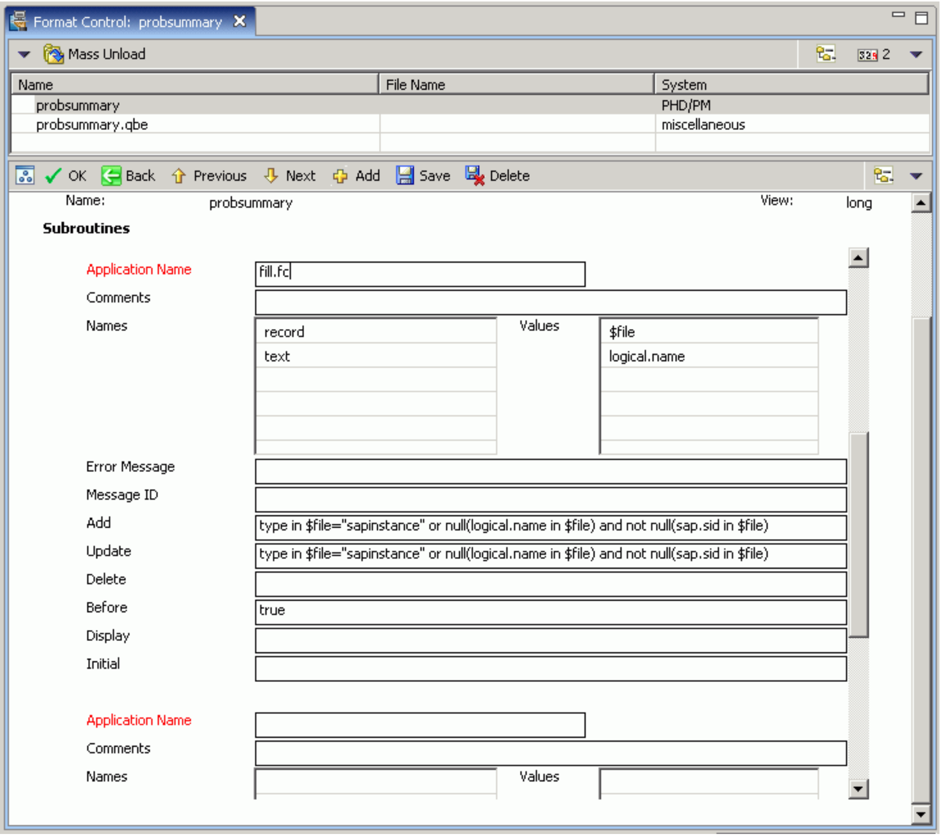
Name: probsummary View: short

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 $I	type in $I				true	fill.fc
	record				$file	

18 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_sm7.txt`)	type in $file="sapinstance" or null(logical.name in $file) and not null(sap.sid in $file)
Update (**[FILL.FC]** in `code_sm7.txt`)	type in $file="sapinstance" or null(logical.name in $file) and not null(sap.sid in $file)
Before	true



Name	File Name	System
probsummary		PHD/PM
probsummary.qbe		miscellaneous

Name: probsummary View: long

Subroutines

Application Name: fill.fc
Comments:
Names: record, text — Values: $file, logical.name
Error Message:
Message ID:
Add: type in $file="sapinstance" or null(logical.name in $file) and not null(sap.sid in $file)
Update: type in $file="sapinstance" or null(logical.name in $file) and not null(sap.sid in $file)
Delete:
Before: true
Display:
Initial:

Application Name:
Comments:
Names: — Values:

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.

Creating HPSAPSolutionManager Table

Do the following to create an SAP Solution Manager table. Service Manager Incident selects SAP Solution Manager from this table to exchange information.

1. Go to **System Definition** → **Tables**, right-click **Tables** and select **New Table**.
2. Enter **`HPSAPSolutionManager`** as the table name, and then click **OK**.
3. In **System Definition** → **Tables** → **HPSAPSolutionManager** → **Fields** click **New field**, enter **`name`** as Field name, and then click **OK**.
4. Select **Character** as Type.
5. Enter **`name`** as Caption.
6. Click **Save**.
7. In **Tailoring** → **Forms Designer** click **New**. Open the Form Wizard.
8. Enter **`HPSAPSolutionManager`** as the form name.
9. Select **`HPSAPSolutionManager`** as the table name.
10. Select **Detail of a Single Record**.
11. Set the Show column to **`true`** for fields `id` and `name`.
12. Set the Show column to **`false`** for all other fields.
13. Click **Proceed**, then click **OK**.
14. Select **Database Manager** from the drop-down list of the top right triangle, then click **Yes**.
15. Enter **`exthd`** as id (should be consistent with the value of `exthd.instances.id.<`*`number`*`>`).
16. Provide a value for the Name field, for example, **`SAP SolMan G11`**.
17. Click **Add** button to add the record.

Configuring WSDL Mapping

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

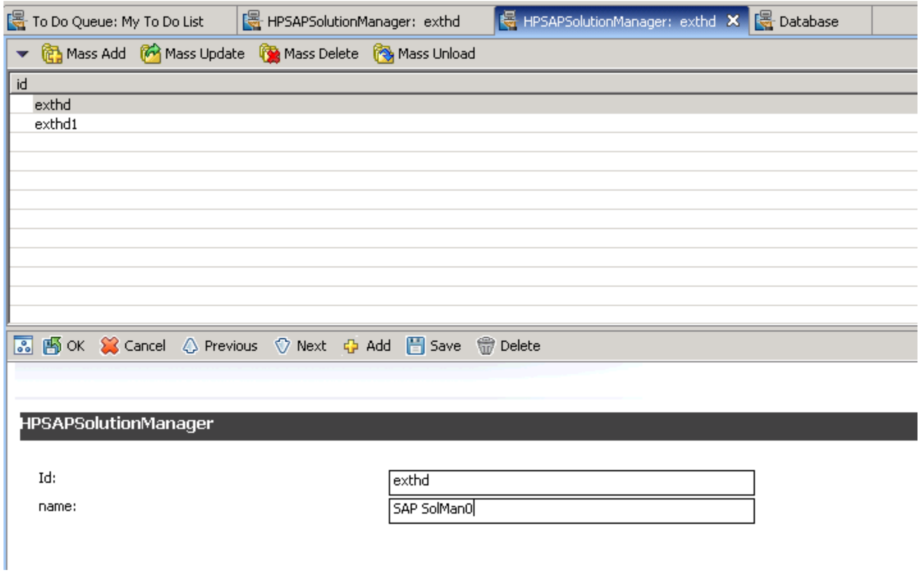
Field name	**Caption**
action	IncidentDescription
assignee.name	AssigneeName
brief.description	BriefDescription
initial.impact	InitialImpact
assignment	PrimaryAssignmentGroup
product.type	ProductType
resolution	Resolution
subcategory	SubCategory
severity	Urgency

Adding New Client for SAP Solution Manager

Do the following to add a new client for SAP Solution Manager:

1. Log in to Service Manager with a System Administrator account.
2. Click **Tailoring** → **Database Manager**. Database Manager opens.
3. Type **`HPSAPSolutionManager`** in the **Table** field and press the Enter key. The `HPSAPSolutionManager` table opens.

4 Click **Search**. The record list of the `HPSAPSolutionManager` table is displayed.



id
exthd
exthd1

Id: exthd

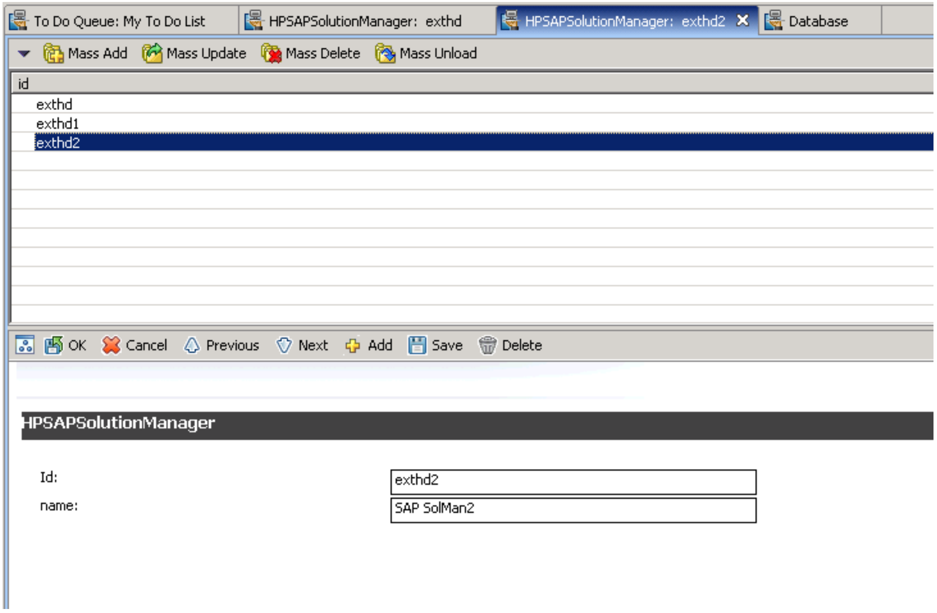
name: SAP SolMan0

5 Fill in the **Id** field and the **name** field to add a new client for SAP Solution Manager.

— **Id** refers to the properties file name, which matches the filename pattern `<Id>.properties`.

— **name** refers to the Solution Manager client that exchanges incidents with Service Manager.

6 Click **Add** to add the new record to the `HPSAPSolutionManager` table. As shown in the following screenshot, new client `exthd2` is added successfully:



id
exthd
exthd1
exthd2

Id: exthd2

name: SAP SolMan2

6 Configuring SAP Solution Manager

This chapter describes how to configure the SAP Solution Manager.

Prerequisites

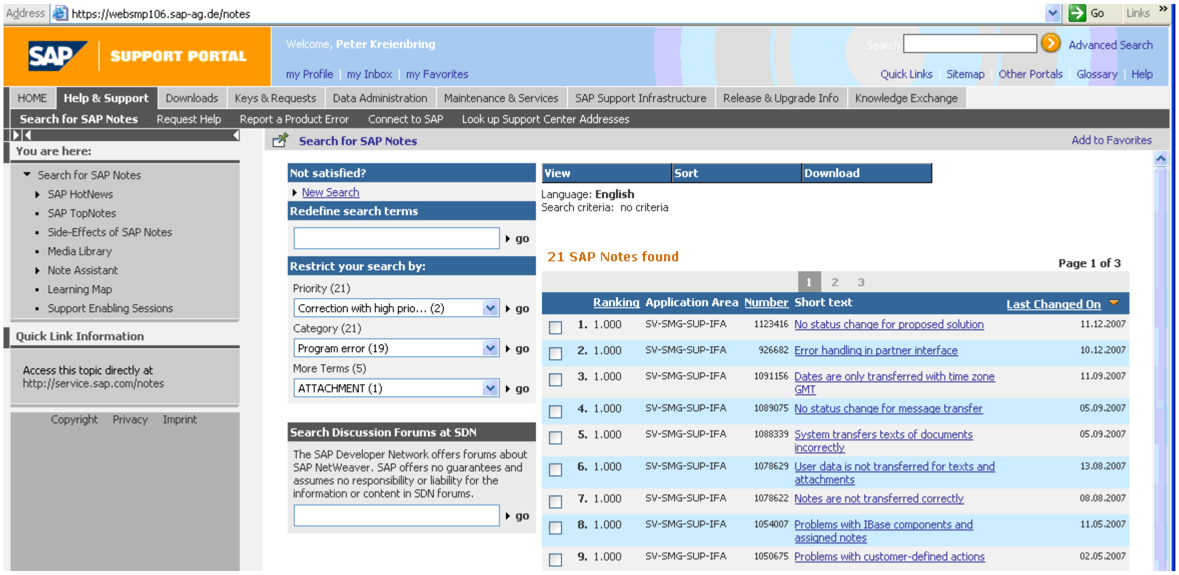
The prerequisites are:

- 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.



Access this topic directly at
http://service.sap.com/notes

Priority (21)
Correction with high prio... (2)
Category (21)
Program error (19)
More Terms (5)
ATTACHMENT (1)

The SAP Developer Network offers forums about SAP NetWeaver. SAP offers no guarantees and assumes no responsibility or liability for the information or content in SDN forums.

Language: **English**
Search criteria: no criteria

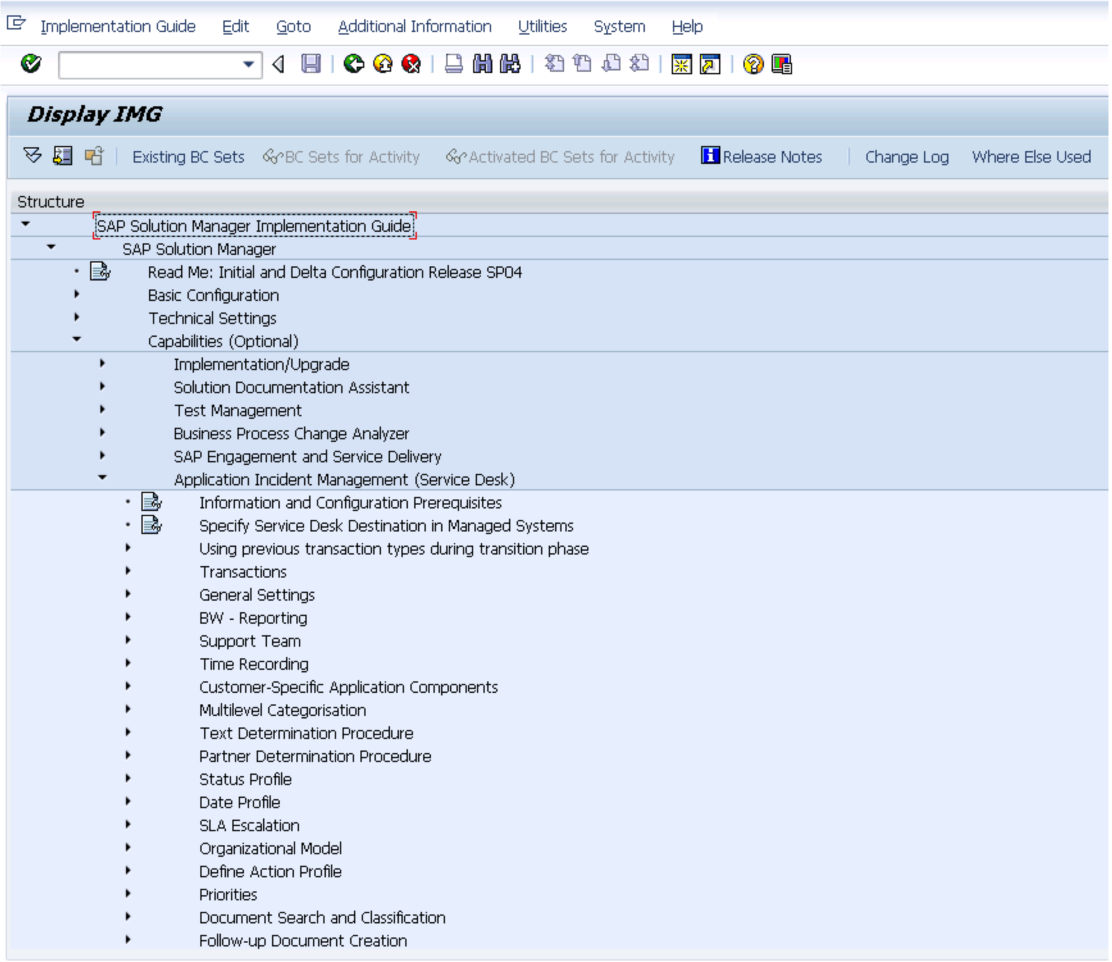
21 SAP Notes found

Page 1 of 3

		Ranking	Application Area	Number	Short text	Last Changed On
□	1.	1.000	SV-SMG-SUP-IFA	1123416	No status change for proposed solution	11.12.2007
□	2.	1.000	SV-SMG-SUP-IFA	926682	Error handling in partner interface	10.12.2007
□	3.	1.000	SV-SMG-SUP-IFA	1091156	Dates are only transferred with time zone GMT	11.09.2007
□	4.	1.000	SV-SMG-SUP-IFA	1089075	No status change for message transfer	05.09.2007
□	5.	1.000	SV-SMG-SUP-IFA	1088339	System transfers texts of documents incorrectly	05.09.2007
□	6.	1.000	SV-SMG-SUP-IFA	1078629	User data is not transferred for texts and attachments	13.08.2007
□	7.	1.000	SV-SMG-SUP-IFA	1078622	Notes are not transferred correctly	08.08.2007
□	8.	1.000	SV-SMG-SUP-IFA	1054007	Problems with IBase components and assigned notes	11.05.2007
□	9.	1.000	SV-SMG-SUP-IFA	1050675	Problems with customer-defined actions	02.05.2007

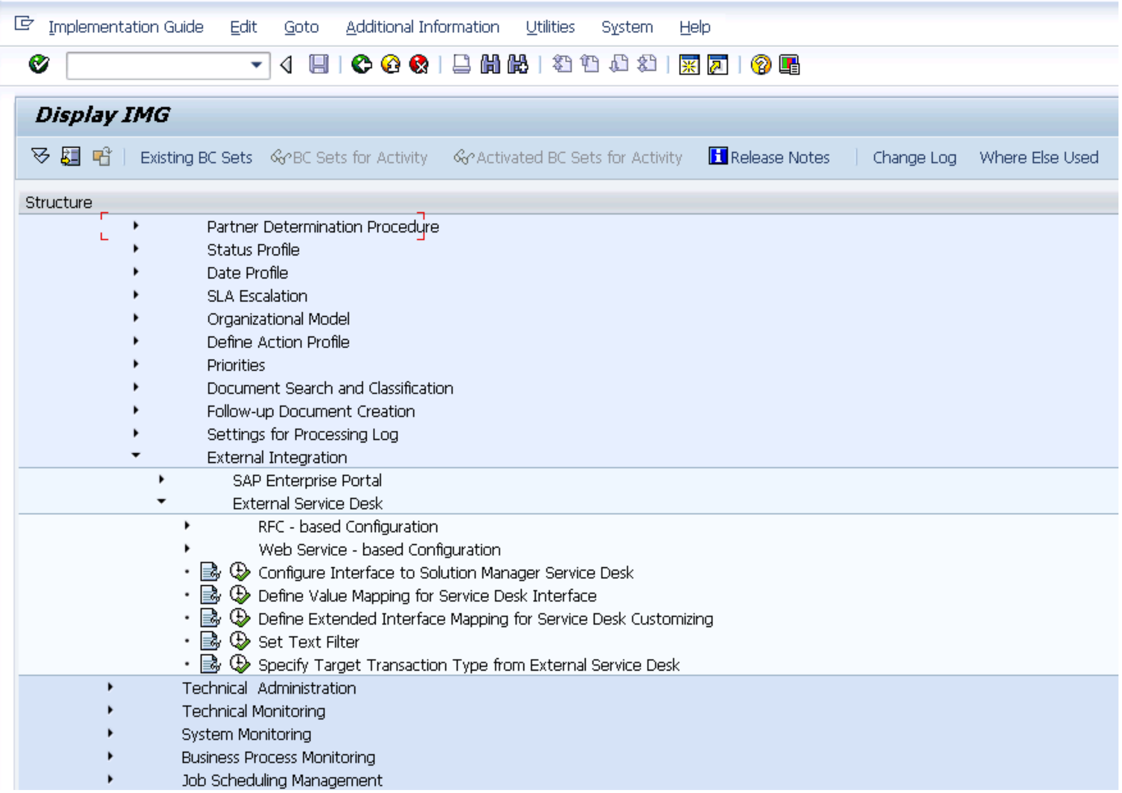
Configuring SAP Solution Manager External Service Desk Interface

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.



Structure

- SAP Solution Manager Implementation Guide
  - SAP Solution Manager
    - Read Me: Initial and Delta Configuration Release SP04
    - Basic Configuration
    - Technical Settings
    - Capabilities (Optional)
      - Implementation/Upgrade
      - Solution Documentation Assistant
      - Test Management
      - Business Process Change Analyzer
      - SAP Engagement and Service Delivery
      - Application Incident Management (Service Desk)
        - Information and Configuration Prerequisites
        - Specify Service Desk Destination in Managed Systems
        - Using previous transaction types during transition phase
        - Transactions
        - General Settings
        - BW - Reporting
        - Support Team
        - Time Recording
        - Customer-Specific Application Components
        - Multilevel Categorisation
        - Text Determination Procedure
        - Partner Determination Procedure
        - Status Profile
        - Date Profile
        - SLA Escalation
        - Organizational Model
        - Define Action Profile
        - Priorities
        - Document Search and Classification
        - Follow-up Document Creation



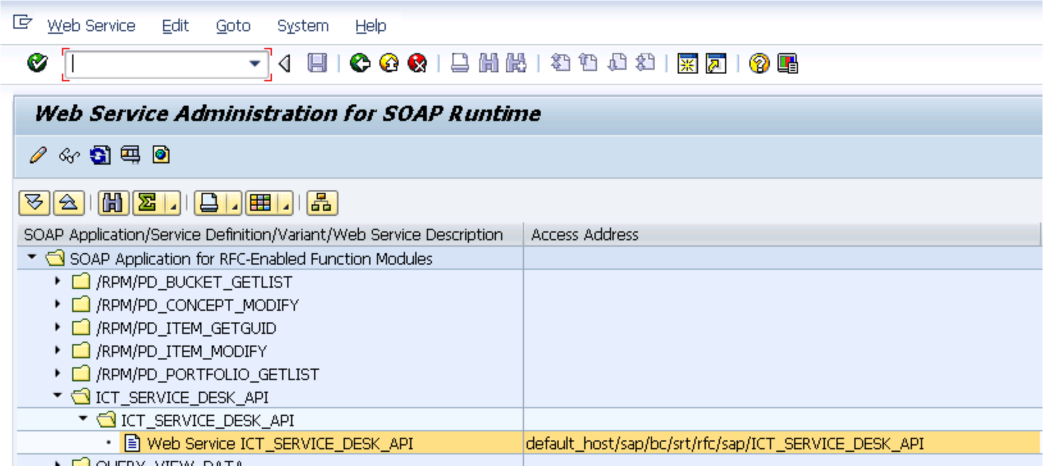
- Partner Determination Procedure
- Status Profile
- Date Profile
- SLA Escalation
- Organizational Model
- Define Action Profile
- Priorities
- Document Search and Classification
- Follow-up Document Creation
- Settings for Processing Log
- External Integration
  - SAP Enterprise Portal
  - External Service Desk
    - RFC - based Configuration
    - Web Service - based Configuration
    - Configure Interface to Solution Manager Service Desk
    - Define Value Mapping for Service Desk Interface
    - Define Extended Interface Mapping for Service Desk Customizing
    - Set Text Filter
    - Specify Target Transaction Type from External Service Desk
- Technical Administration
- Technical Monitoring
- System Monitoring
- Business Process Monitoring
- Job Scheduling Management

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`**.



SOAP Application/Service Definition/Variant/Web Service Description	Access Address
SOAP Application for RFC-Enabled Function Modules	
/RPM/PD_BUCKET_GETLIST	
/RPM/PD_CONCEPT_MODIFY	
/RPM/PD_ITEM_GETGUID	
/RPM/PD_ITEM_MODIFY	
/RPM/PD_PORTFOLIO_GETLIST	
ICT_SERVICE_DESK_API	
ICT_SERVICE_DESK_API	
Web Service ICT_SERVICE_DESK_API	default_host/sap/bc/srt/rfc/sap/ICT_SERVICE_DESK_API
QUERY_VIEW_DATA	

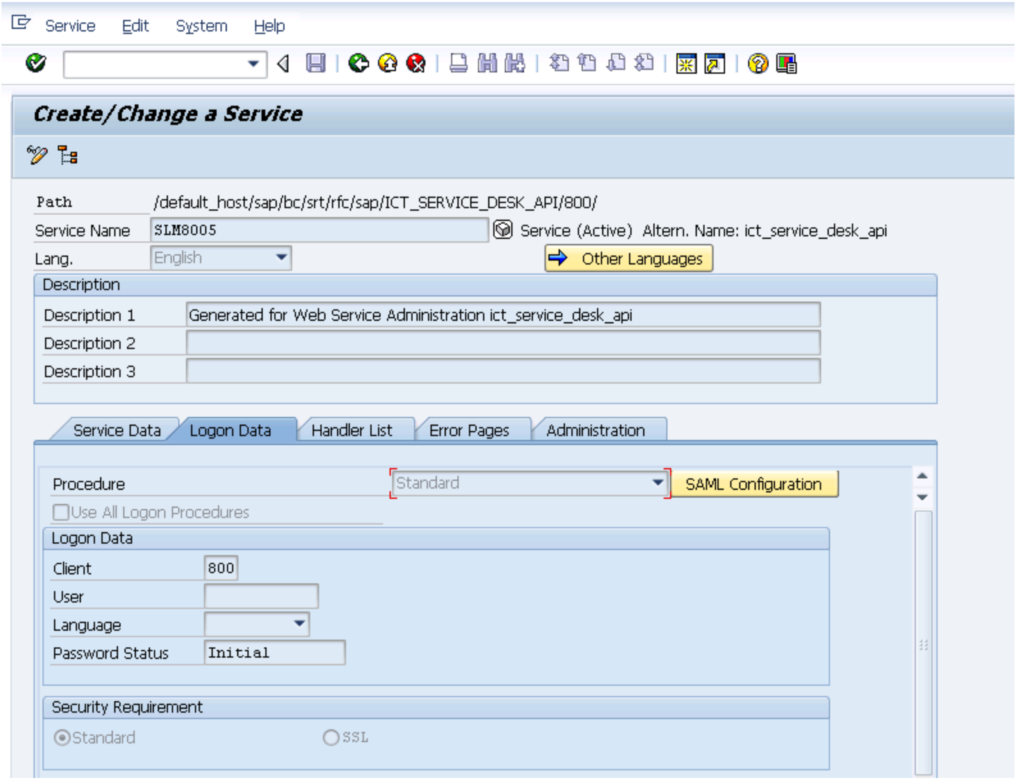
2 To determine the logon procedure of the Web Service for incoming requests, go to SAP transaction **/nsicf**.

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**.



Create/Change a Service

Path /default_host/sap/bc/srt/rfc/sap/ICT_SERVICE_DESK_API/800/

Service Name SLM8005 Service (Active) Altern. Name: ict_service_desk_api

Lang. English

Description

Description 1 Generated for Web Service Administration ict_service_desk_api

Description 2

Description 3

Procedure Standard

☐ Use All Logon Procedures

Logon Data

Client 800

User

Language

Password Status Initial

Security Requirement

● Standard ○ SSL

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.

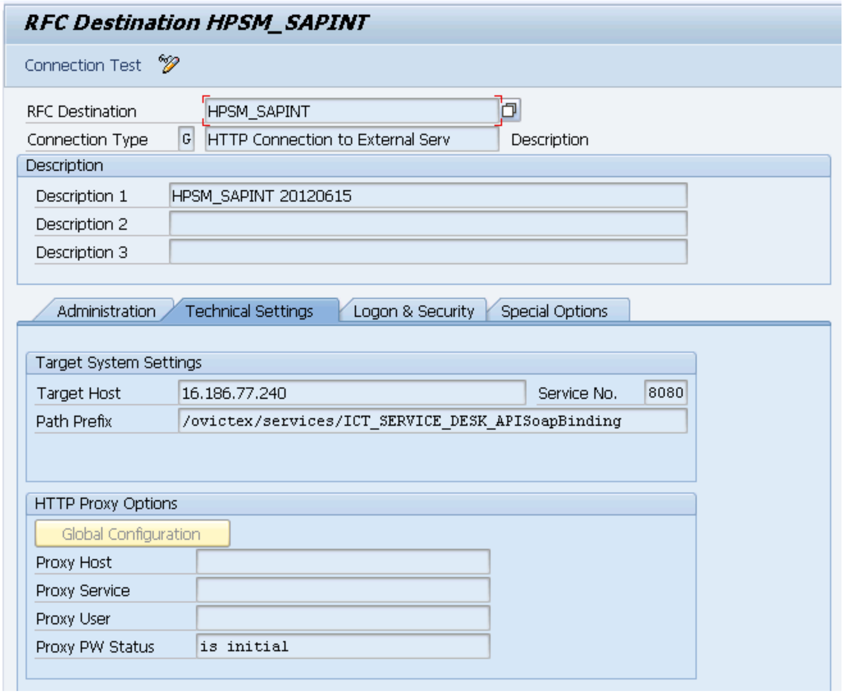
1 Select transaction **`/nsm59`**.

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`**.



RFC Destination HPSM_SAPINT

RFC Destination: HPSM_SAPINT
Connection Type: G HTTP Connection to External Serv Description

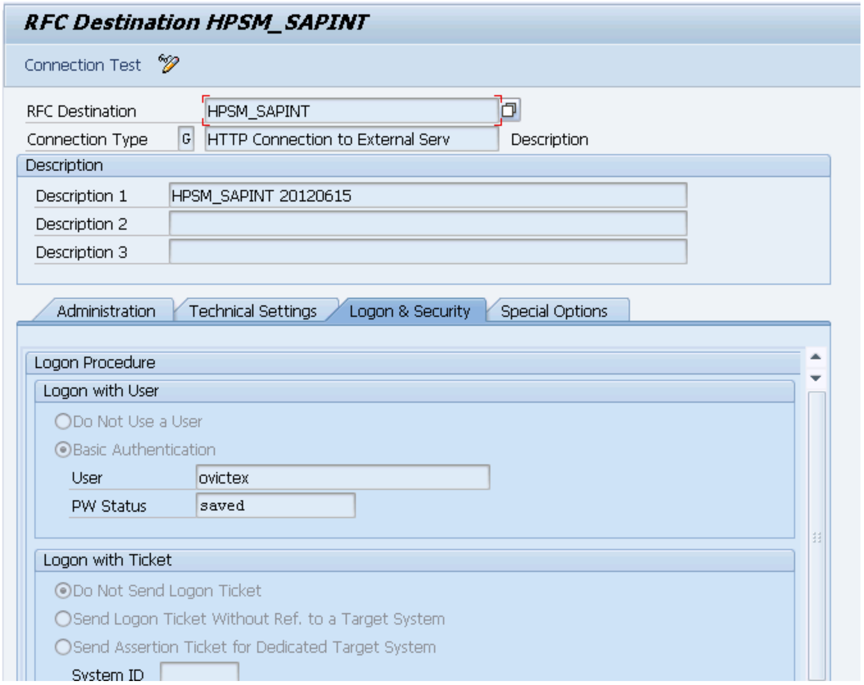
Description
Description 1: HPSM_SAPINT 20120615
Description 2:
Description 3:

Target System Settings
Target Host: 16.186.77.240 Service No. 8080
Path Prefix: /ovictex/services/ICT_SERVICE_DESK_APISoapBinding

HTTP Proxy Options

Proxy Host:
Proxy Service:
Proxy User:
Proxy PW Status: is initial

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”.



RFC Destination HPSM_SAPINT

RFC Destination: HPSM_SAPINT
Connection Type: G HTTP Connection to External Serv
Description

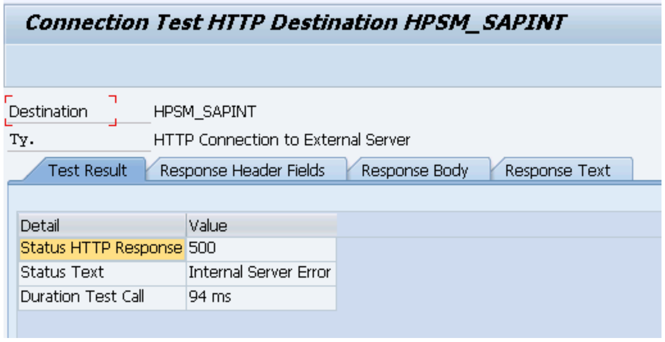
Description
Description 1: HPSM_SAPINT 20120615
Description 2:
Description 3:

Logon Procedure

Logon with User
○ Do Not Use a User
● Basic Authentication
User: ovictex
PW Status: saved

Logon with Ticket
● Do Not Send Logon Ticket
○ Send Logon Ticket Without Ref. to a Target System
○ Send Assertion Ticket for Dedicated Target System
System ID:

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



Connection Test HTTP Destination HPSM_SAPINT

Destination: HPSM_SAPINT
Ty.: HTTP Connection to External Server

Detail	Value
Status HTTP Response	500
Status Text	Internal Server Error
Duration Test Call	94 ms

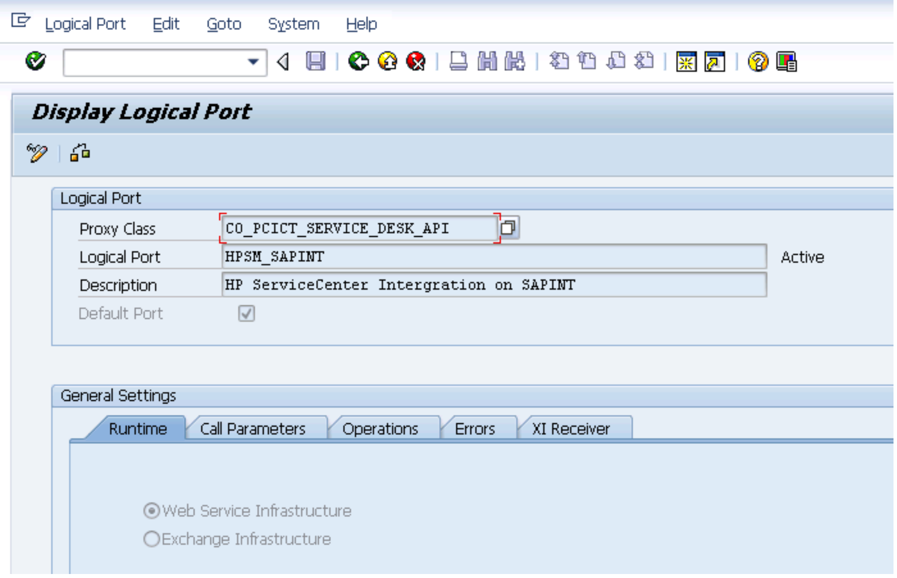
Create a Logical Port

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.



Display Logical Port

Logical Port

Proxy Class CO_PCICT_SERVICE_DESK_API

Logical Port HPSM_SAPINT Active

Description HP ServiceCenter Intergration on SAPINT

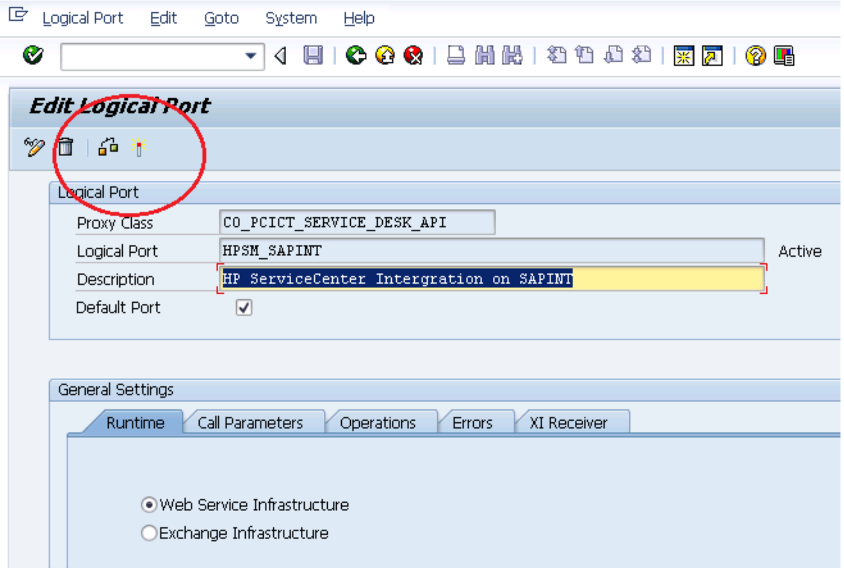
Default Port ☑

General Settings

● Web Service Infrastructure

○ Exchange Infrastructure

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



Edit Logical Port

Logical Port

Proxy Class CO_PCICT_SERVICE_DESK_API

Logical Port HPSM_SAPINT Active

Description HP ServiceCenter Intergration on SAPINT

Default Port ☑

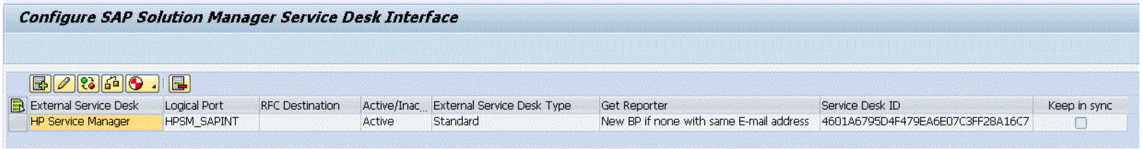
General Settings

● Web Service Infrastructure

○ Exchange Infrastructure

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.



Configure SAP Solution Manager Service Desk Interface

External Service Desk	Logical Port	RFC Destination	Active/Inac...	External Service Desk Type	Get Reporter	Service Desk ID	Keep in sync
HP Service Manager	HPSM_SAPINT		Active	Standard	New BP if none with same E-mail address	4601A6795D4F479EA6E07C3FF28A16C7	□

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

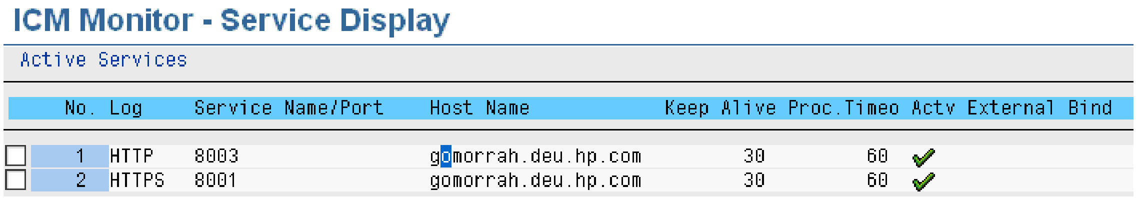
This IMG activity configures the value mapping between SAP Solution Manager Service Desk and SMSSMEX for ingoing and outgoing requests. Changing the default value mapping of the SAP Solution Manager is not required. If changes are necessary, use the field mapping file of the SMSSMEX configuration file. 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**.



ICM Monitor - Service Display

```
Active Services

    No. Log    Service Name/Port    Host Name            Keep Alive Proc.Timeo Actv External Bind

□     1 HTTP   8003                 gomorrah.deu.hp.com          30         60  ✓
□     2 HTTPS  8001                 gomorrah.deu.hp.com          30         60  ✓
```

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.

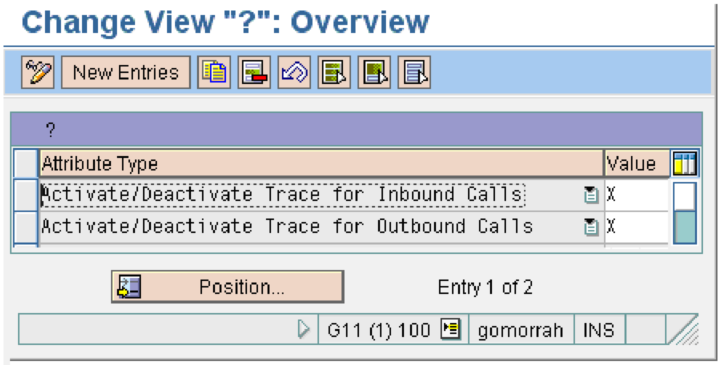
Enable tracing

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**



Change View "?": Overview

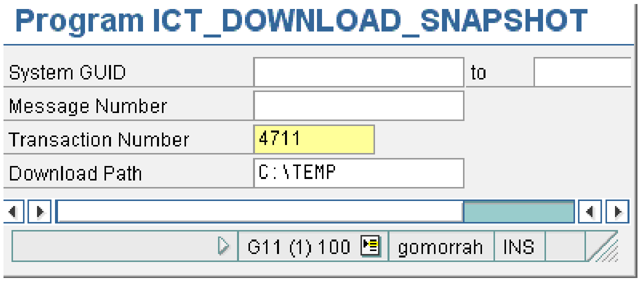
?

Attribute Type	Value
Activate/Deactivate Trace for Inbound Calls	X
Activate/Deactivate Trace for Outbound Calls	X

Entry 1 of 2

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).



Program ICT_DOWNLOAD_SNAPSHOT

System GUID		to	
Message Number			
Transaction Number	4711		
Download Path	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.

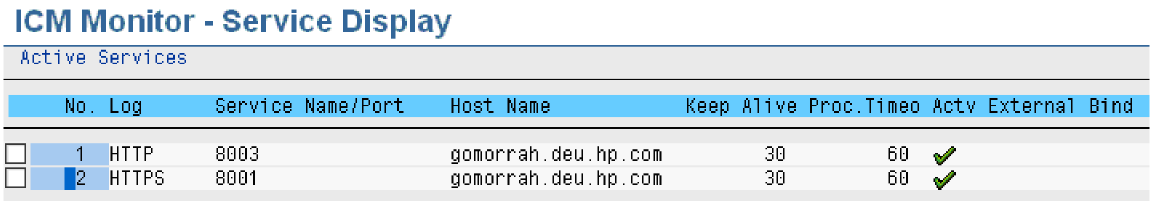
Checking SAP SSL Configuration

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.



ICM Monitor - Service Display

Active Services

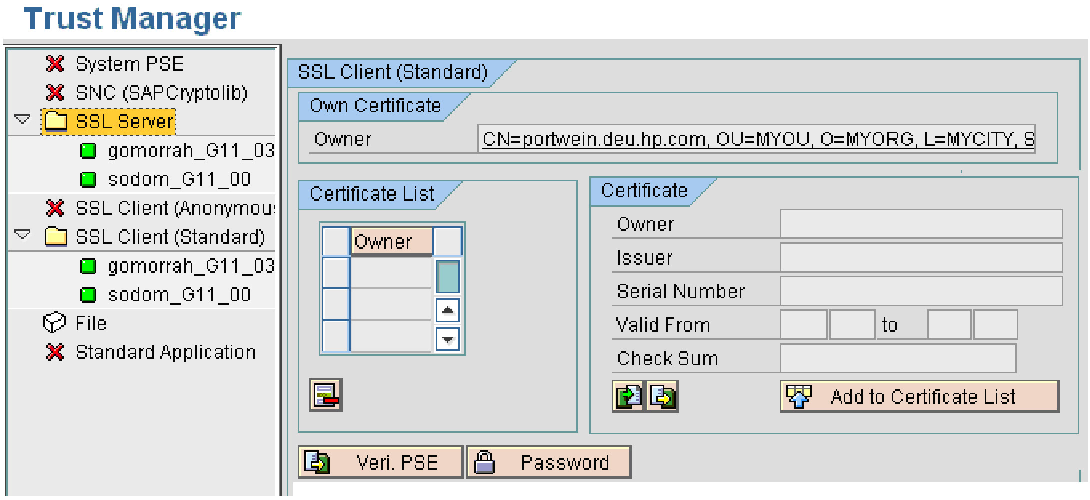
	No.	Log	Service Name/Port	Host Name	Keep Alive	Proc.Timeo	Actv	External Bind
□	1	HTTP	8003	gomorrah.deu.hp.com	30	60	✓	
□	2	HTTPS	8001	gomorrah.deu.hp.com	30	60	✓	

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.

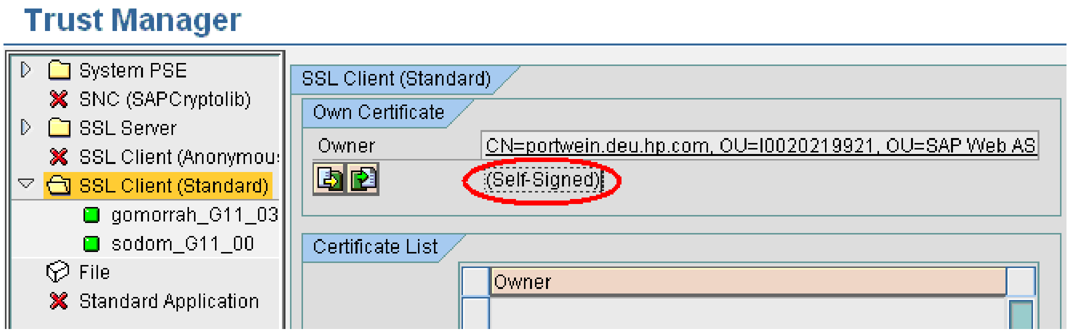
➤ Installing and configuring sapcryptolib requires a restart of the SAP WEB AS instance. The installation instructions are in the SAP online help. For more information, see Appendix B, Installing and Configuring SAPCRYPLIB.

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.



Owner CN=portwein.deu.hp.com, OU=MYOU, O=MYORG, L=MYCITY, S

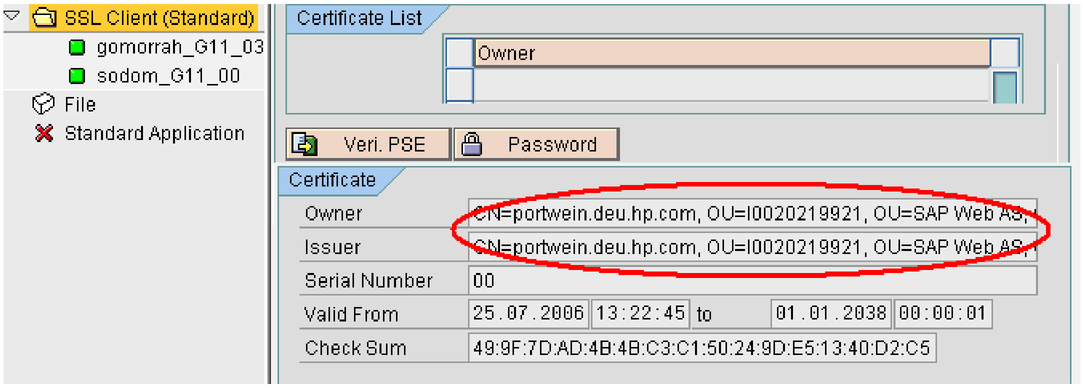
In the next diagram the certificate of the PSE “SSL Client (Standard)” is “Self Signed”. Self-signed certificates are not supported for communication with Apache Tomcat (the certificate must be signed against a CA). If the certificate is signed the 'Self signed' certificate text will disappear.



Owner CN=portwein.deu.hp.com, OU=I0020219921, OU=SAP Web AS

(Self-Signed)

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.



Owner CN=portwein.deu.hp.com, OU=I0020219921, OU=SAP Web AS,

Issuer CN=portwein.deu.hp.com, OU=I0020219921, OU=SAP Web AS,

Serial Number 00

Valid From 25.07.2006 13:22:45 to 01.01.2038 00:00:01

Check Sum 49:9F:7D:AD:4B:4B:C3:C1:50:24:9D:E5:13:40:D2:C5

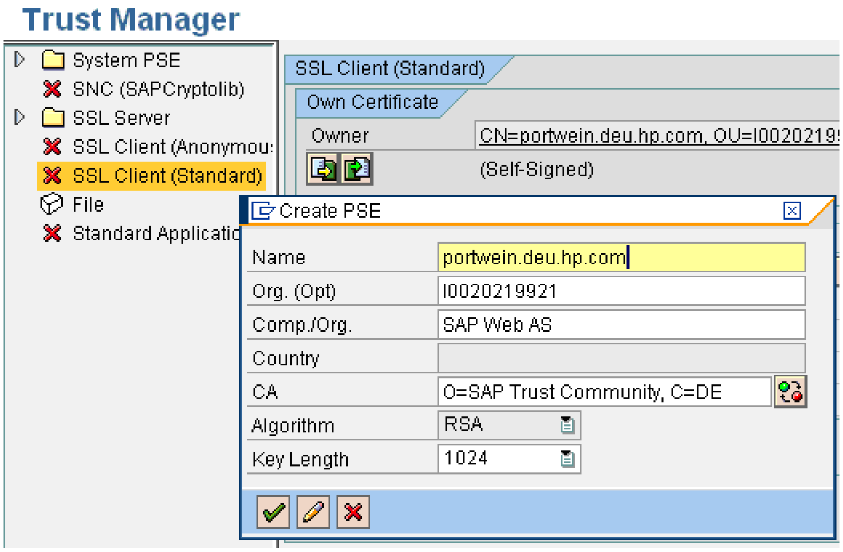
Creating a Client PSE in Trust Manager

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**.



Trust Manager

Owner CN=portwein.deu.hp.com, OU=I0020219!

(Self-Signed)

Create PSE

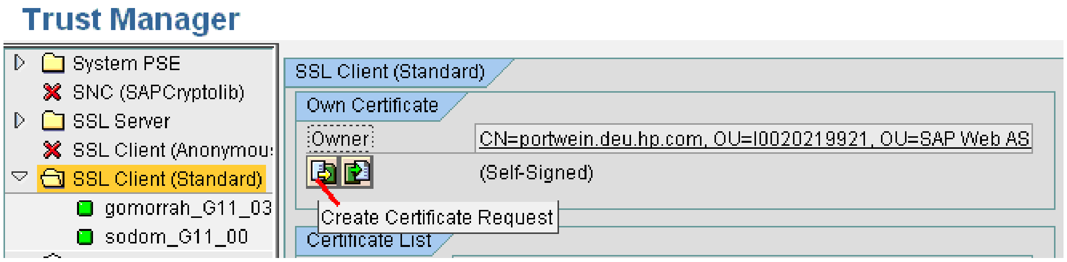
Name	portwein.deu.hp.com
Org. (Opt)	I0020219921
Comp./Org.	SAP Web AS
Country	
CA	O=SAP Trust Community, C=DE
Algorithm	RSA
Key Length	1024

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**.



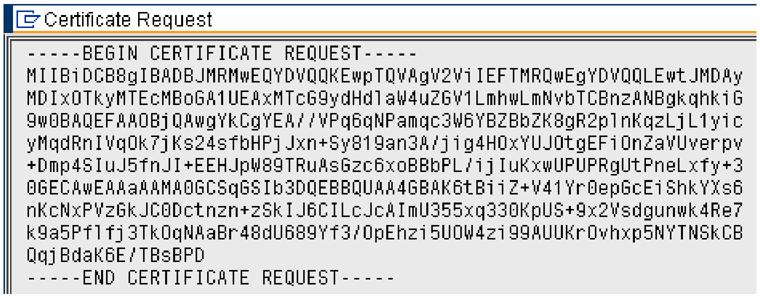
Trust Manager

Owner CN=portwein.deu.hp.com, OU=I0020219921, OU=SAP Web AS

(Self-Signed)

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

9 Certify the request with a CA.

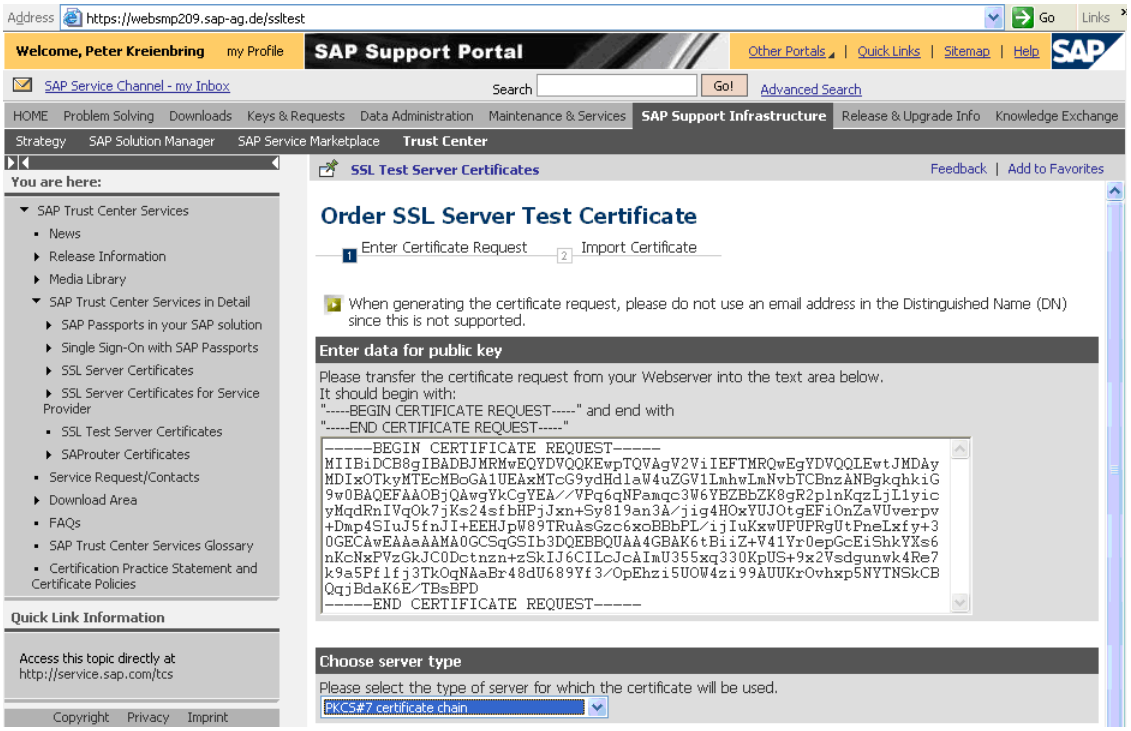


Certificate Request

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBiDCB8gIBADBJMRMwEQYDVQQKEwpTQVAgV2ViIEFTMRQwEgYDVQQLEwtJMDAy
MDIxOTkyMTEcMBoGA1UEAxMTcG9ydHdlaW4uZGV1LmhwLmNvbTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEA//VPq6qNPamqc3W6YBZBbZK8gR2plnKqzLjL1yic
yMqdRnIVqOk7jKs24sfbHPjJxn+Sy819an3A/jig4HOxYUJOtgEFiOnZaVUverpv
+Dmp4SIuJ5fnJI+EEHJpW89TRuAsGzc6xoBBbPL/ijIuKxwUPUPRgUtPneLxfy+3
0GECAwEAAaAAMA0GCSqGSIb3DQEBBQUAA4GBAK6tBiiZ+V41Yr0epGcEiShkYXs6
nKcNxPVzGkJC0Dctnzn+zSkIJ6CILcJcAImU355xq330KpUS+9x2Vsdgunwk4Re7
k9a5Pflfj3TkOqNAaBr48dU689Yf3/OpEhzi5UOW4zi99AUUKrOvhxp5NYTNSkCB
QqjBdaK6E/TBsBPD
-----END CERTIFICATE REQUEST-----
```

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).



Address https://websmp209.sap-ag.de/ssltest
Go
Links
Welcome, Peter Kreienbring
my Profile
SAP Support Portal
Other Portals
Quick Links
Sitemap
Help
SAP Service Channel - my Inbox
Search
Go!
Advanced Search
HOME
Problem Solving
Downloads
Keys & Requests
Data Administration
Maintenance & Services
SAP Support Infrastructure
Release & Upgrade Info
Knowledge Exchange
Strategy
SAP Solution Manager
SAP Service Marketplace
Trust Center
You are here:
SAP Trust Center Services
News
Release Information
Media Library
SAP Trust Center Services in Detail
SAP Passports in your SAP solution
Single Sign-On with SAP Passports
SSL Server Certificates
SSL Server Certificates for Service Provider
SSL Test Server Certificates
SAProuter Certificates
Service Request/Contacts
Download Area
FAQs
SAP Trust Center Services Glossary
Certification Practice Statement and Certificate Policies
Quick Link Information
Access this topic directly at http://service.sap.com/tcs
Copyright
Privacy
Imprint
SSL Test Server Certificates
Feedback
Add to Favorites

Order SSL Server Test Certificate

1 Enter Certificate Request 2 Import Certificate

When generating the certificate request, please do not use an email address in the Distinguished Name (DN) since this is not supported.

Enter data for public key

Please transfer the certificate request from your Webserver into the text area below.
It should begin with:
"-----BEGIN CERTIFICATE REQUEST-----" and end with
"-----END CERTIFICATE REQUEST-----"

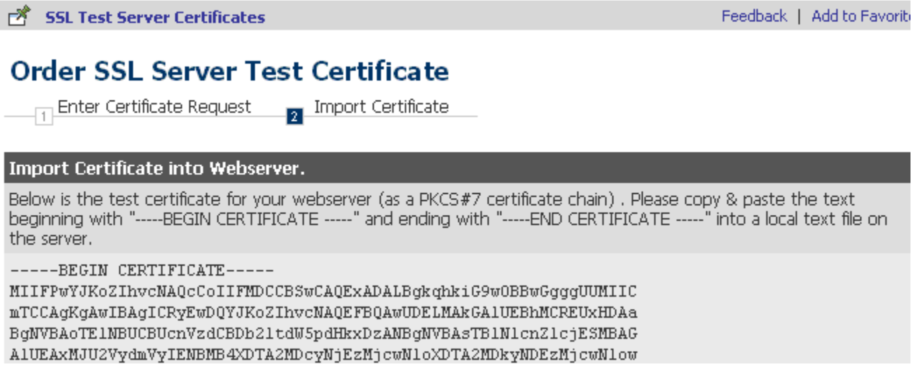
```
------BEGIN CERTIFICATE REQUEST------
MIIBiDCB8gIBADBJMRMwEQYDVQQKEwpTQVAgV2ViIEFTMRQwEgYDVQQLEwtJMDAy
MDIxOTkyMTEcMBoGA1UEAxMTcG9ydHdlaW4uZGV1LmhwLmNvbTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEA//VPq6qNPamqc3W6YBZBbZK8gR2plnKqzLjL1yic
yMqdRnIVqOk7jKs24sfbHPjJxn+Sy819an3A/jig4HOxYUJOtgEFiOnZaVUverpv
+Dmp4SIuJ5fnJI+EEHJpW89TRuAsGzc6xoBBbPL/ijIuKxwUPUPRgUtPneLxfy+3
0GECAwEAAaAAMA0GCSqGSIb3DQEBBQUAA4GBAK6tBiiZ+V41Yr0epGcEiShkYXs6
nKcNxPVzGkJC0Dctnzn+zSkIJ6CILcJcAImU355xq330KpUS+9x2Vsdgunwk4Re7
k9a5Pflfj3TkOqNAaBr48dU689Yf3/OpEhzi5UOW4zi99AUUKrOvhxp5NYTNSkCB
QqjBdaK6E/TBsBPD
------END CERTIFICATE REQUEST------
```

Choose server type

Please select the type of server for which the certificate will be used.

PKCS#7 certificate chain

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



Order SSL Server Test Certificate

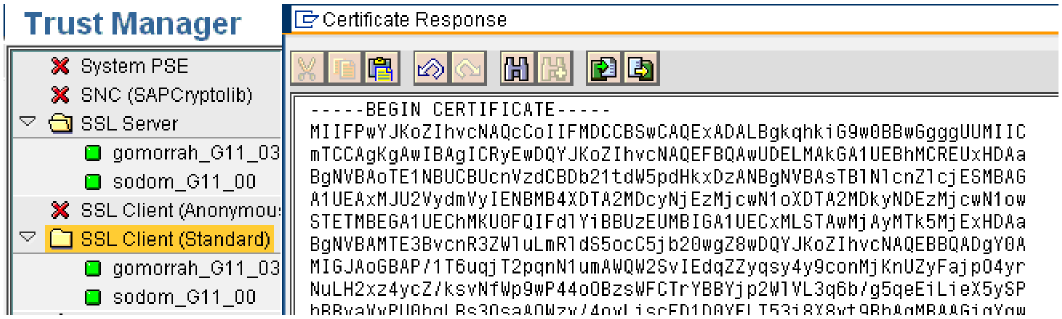
1 Enter Certificate Request 2 Import Certificate

Import Certificate into Webserver.

Below is the test certificate for your webserver (as a PKCS#7 certificate chain) . Please copy & paste the text beginning with "-----BEGIN CERTIFICATE -----" and ending with "-----END CERTIFICATE -----" into a local text file on the server.

```
-----BEGIN CERTIFICATE-----
MIIFPwYJKoZIhvcNAQcCoIIFMDCCBSwCAQExADALBgkqhkiG9w0BBwGgggUUMIIC
mTCCAgKgAwIBAgICRyEwDQYJKoZIhvcNAQEFBQAwUDELMAkGA1UEBhMCREUxHDAa
BgNVBAoTElNBUCBUcnVzdCBDb21tdW5pdHkxDzANBgNVBAsTBlNlcnZlcjESMBAG
A1UEAxMJU2VydmVyIENBMB4XDTA2MDcyNjEzMjcwNloXDTA2MDkyNDEzMjcwNlow
```

12 Copy the Certificate response to your client PSE.



```
-----BEGIN CERTIFICATE-----
MIIFPwYJKoZIhvcNAQcCoIIFMDCCBSwCAQExADALBgkqhkiG9w0BBwGgggUUMIIC
mTCCAgKgAwIBAgICRyEwDQYJKoZIhvcNAQEFBQAwUDELMAkGA1UEBhMCREUxHDAa
BgNVBAoTE1NBUCBUcnVzdCBDb21tdW5pdHkxDzANBgNVBAsTBlNlcnZlcjESMBAG
A1UEAxMJU2VydmVyIENBMB4XDTA2MDcyNjEzMjcwN1oXDTA2MDkyNDEzMjcwN1ow
STETMBEGA1UEChMKU0FQIFdlYiBBUzEUMBIGA1UECxMLSTAwMjAyMTk5MjExHDAa
BgNVBAMTE3BvcnR3ZWluLmRldS5ocC5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAP/1T6uqjT2pqnN1umAWQW2SvIEdqZZyqsy4y9conMjKnUZyFajp04yr
NuLH2xz4ycZ/ksvNfWp9wP44oOBzsWFCTrYBBYjp2WlVL3q6b/g5qeEiLieX5ySP
hBBvaVvPUAhqLBs3OsaAQWzv/4oyLiscED1D0YELT53i8X8vt9RhAgMBAAGjgYow
```

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

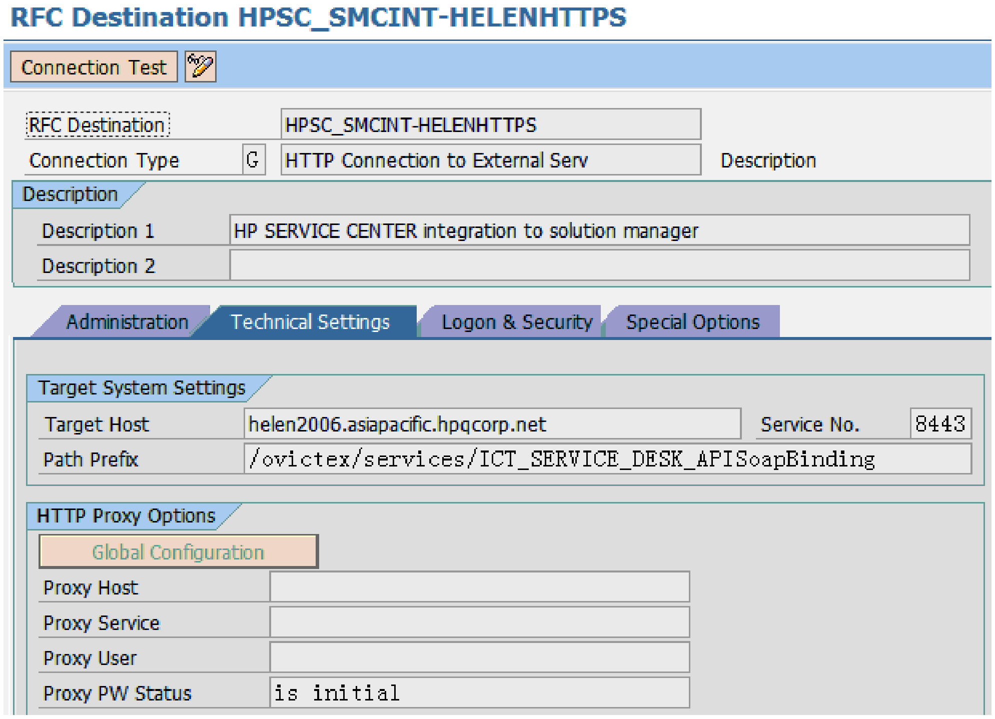
The outgoing connection from SAP Solution Manager to HP Apache Tomcat must be configured in SAP transaction **/nsm59**. Add a new or change an existing HTTP RFC destination with type G.

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).



RFC Destination HPSC_SMCINT-HELENHTTPS

RFC Destination: HPSC_SMCINT-HELENHTTPS

Connection Type: G HTTP Connection to External Serv Description

Description

Description 1: HP SERVICE CENTER integration to solution manager

Description 2:

Target System Settings

Target Host: helen2006.asiapacific.hpqcorp.net Service No.: 8443

Path Prefix: /ovictex/services/ICT_SERVICE_DESK_APISoapBinding

HTTP Proxy Options

Proxy Host:

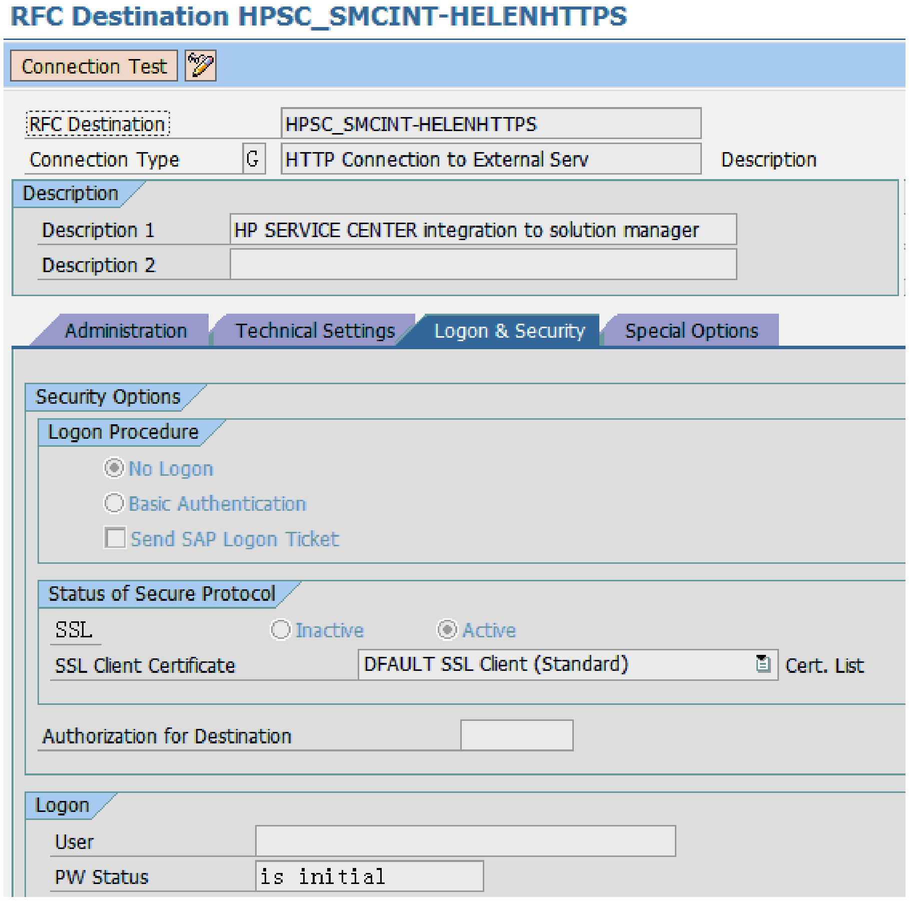
Proxy Service:

Proxy User:

Proxy PW Status: is initial

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)”.



RFC Destination HPSC_SMCINT-HELENHTTPS

RFC Destination: HPSC_SMCINT-HELENHTTPS

Connection Type: G HTTP Connection to External Serv — Description

Description

Description 1: HP SERVICE CENTER integration to solution manager

Description 2:

Security Options

Logon Procedure

● No Logon

○ Basic Authentication

□ Send SAP Logon Ticket

Status of Secure Protocol

SSL: ○ Inactive ● Active

SSL Client Certificate: DFAULT SSL Client (Standard)

Authorization for Destination:

Logon

User:

PW Status: is initial

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 74).

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 76).

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.

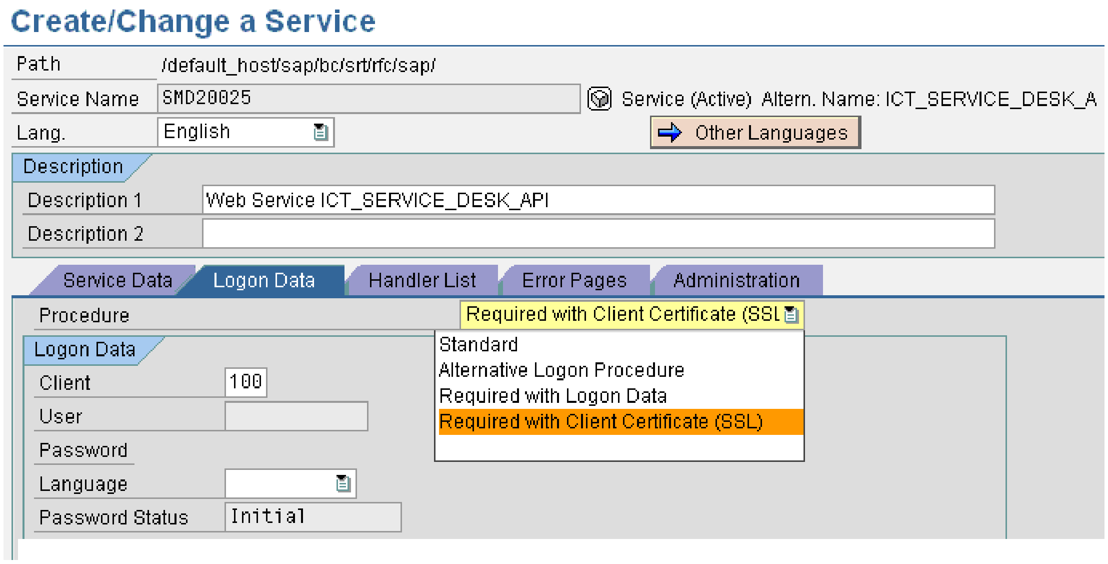
2 Execute the search of the service.

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.



Create/Change a Service

Path /default_host/sap/bc/srt/rfc/sap/

Service Name SMD20025 Service (Active) Altern. Name: ICT_SERVICE_DESK_A

Lang. English

Description

Description 1 Web Service ICT_SERVICE_DESK_API

Description 2

Logon Data

Procedure Required with Client Certificate (SSL

Standard
Alternative Logon Procedure
Required with Logon Data
Required with Client Certificate (SSL)

Logon Data

Client 100

User

Password

Language

Password Status Initial

➤ 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).

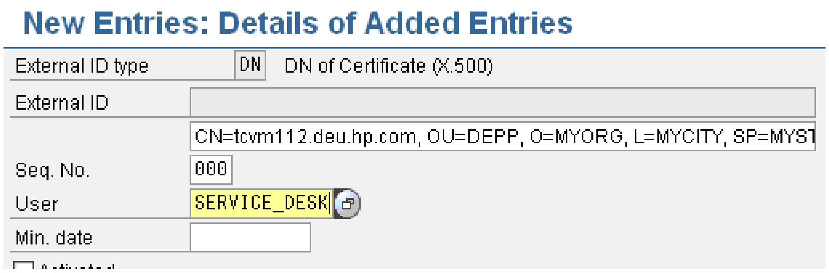
8 Select the Work Area **DN of Certificate X.500**.

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 86). 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.



New Entries: Details of Added Entries

External ID type DN DN of Certificate (X.500)

External ID

CN=tcvm112.deu.hp.com, OU=DEPP, O=MYORG, L=MYCITY, SP=MYST

Seq. No. 000

User SERVICE_DESK

Min. date

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).

If using ServiceCenter refer to Configuring HP ServiceCenter for SSL on page 163.

➤
- 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);
```

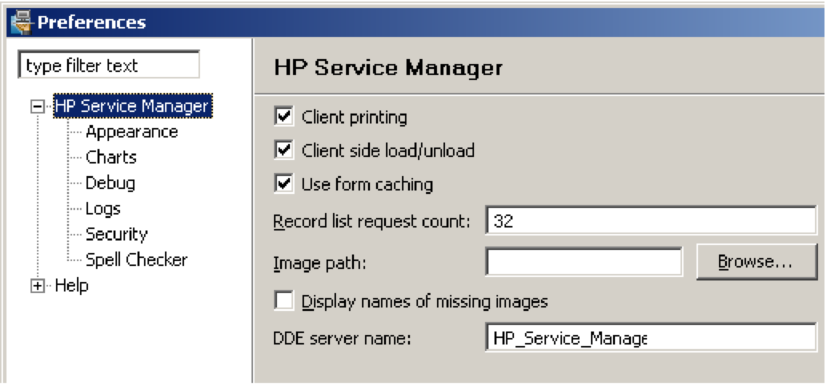
to

```
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.



☑ Client printing

☑ Client side load/unload

☑ Use form caching

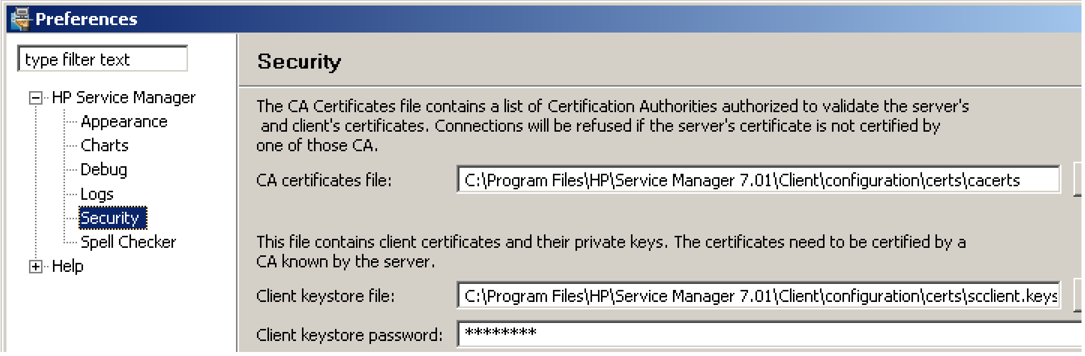
Record list request count: 32

Image path:

□ Display names of missing images

DDE server name: HP_Service_Manage

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



Security

The CA Certificates file contains a list of Certification Authorities authorized to validate the server's and client's certificates. Connections will be refused if the server's certificate is not certified by one of those CA.

CA certificates file: C:\Program Files\HP\Service Manager 7.01\Client\configuration\certs\cacerts

This file contains client certificates and their private keys. The certificates need to be certified by a CA known by the server.

Client keystore file: C:\Program Files\HP\Service Manager 7.01\Client\configuration\certs\scclient.keys

Client keystore password: *********

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>sm7client</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 118.

8 Upgrading SMSSMEX

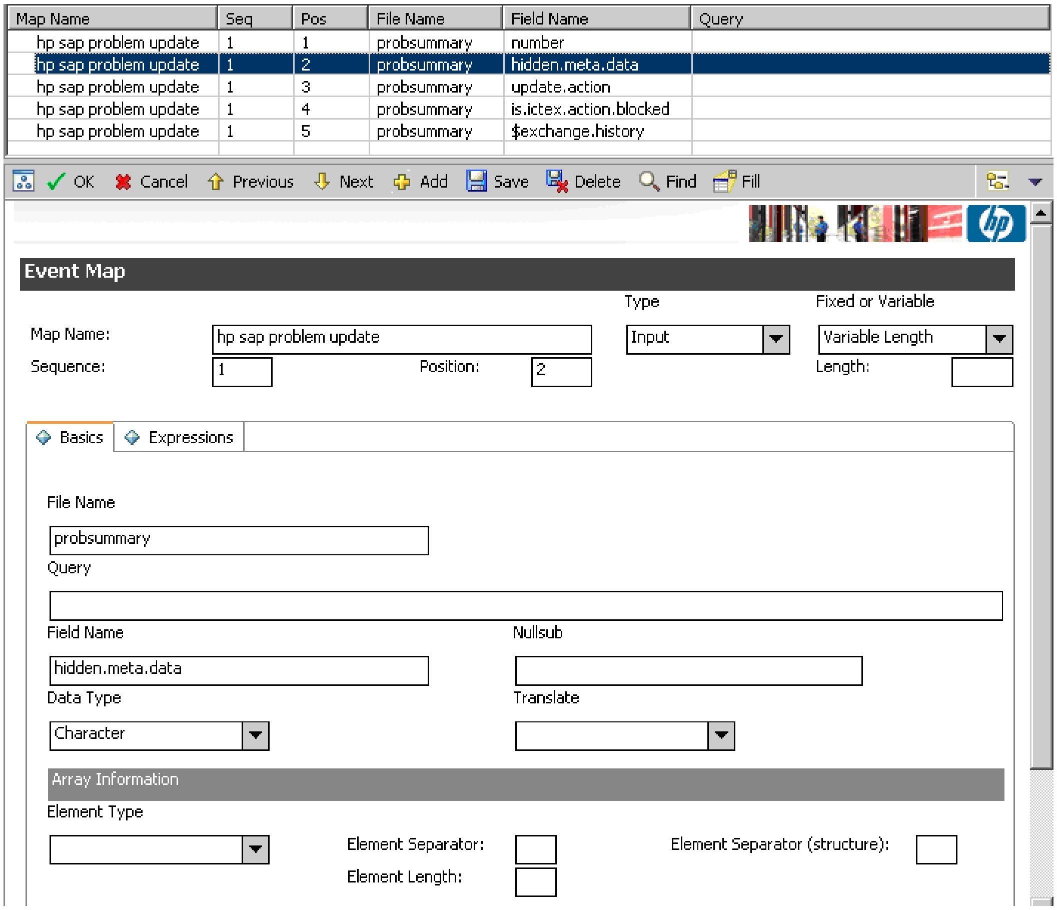
Upgrading SMSSMEX from V1.00 to V1.01

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

1 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**.



Map Name	Seq	Pos	File Name	Field Name	Query
hp sap problem update	1	1	probsummary	number	
hp sap problem update	1	2	probsummary	hidden.meta.data	
hp sap problem update	1	3	probsummary	update.action	
hp sap problem update	1	4	probsummary	is.ictex.action.blocked	
hp sap problem update	1	5	probsummary	$exchange.history	

Event Map

Map Name: hp sap problem update
Type: Input
Fixed or Variable: Variable Length
Sequence: 1
Position: 2
Length:

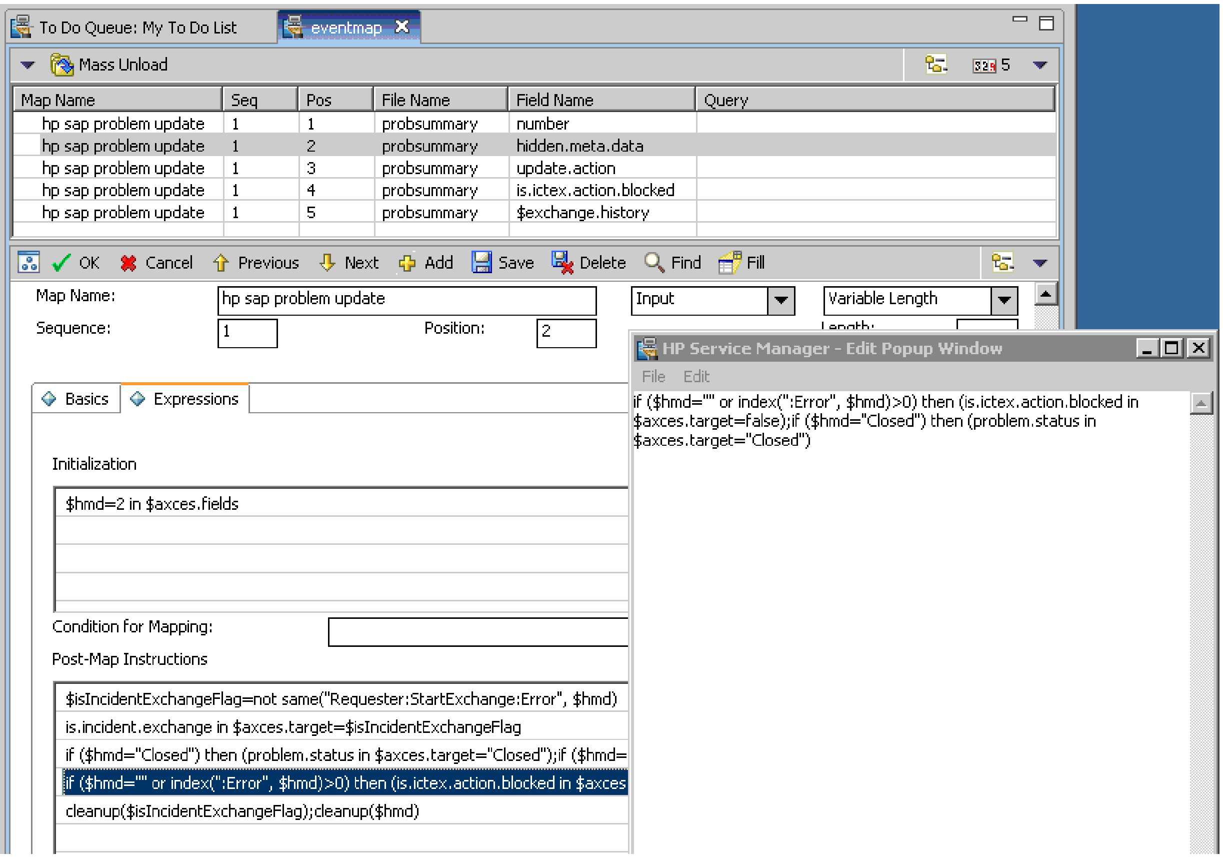
File Name: probsummary
Query:
Field Name: hidden.meta.data
Nullsub:
Data Type: Character
Translate:

Array Information

Element Type:
Element Separator:
Element Separator (structure):
Element Length:

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_sm7.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")
```



Map Name	Seq	Pos	File Name	Field Name	Query
hp sap problem update	1	1	probsummary	number	
hp sap problem update	1	2	probsummary	hidden.meta.data	
hp sap problem update	1	3	probsummary	update.action	
hp sap problem update	1	4	probsummary	is.ictex.action.blocked	
hp sap problem update	1	5	probsummary	$exchange.history	

Map Name: hp sap problem update   Input   Variable Length

Sequence: 1   Position: 2

Initialization

```
$hmd=2 in $axces.fields
```

Condition for Mapping:

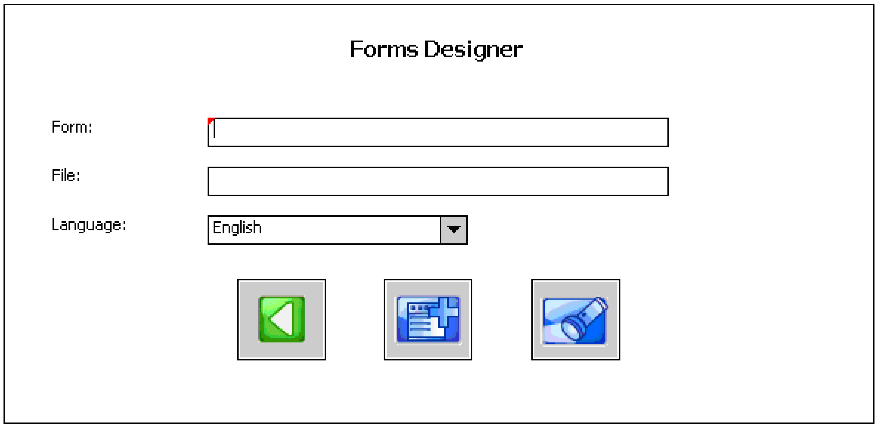
Post-Map Instructions

```
$isIncidentExchangeFlag=not same("Requester:StartExchange:Error", $hmd)
is.incident.exchange in $axces.target=$isIncidentExchangeFlag
if ($hmd="Closed") then (problem.status in $axces.target="Closed");if ($hmd=
if ($hmd="" or index(":Error", $hmd)>0) then (is.ictex.action.blocked in $axces
cleanup($isIncidentExchangeFlag);cleanup($hmd)
```

```
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")
```

2 Modify the `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.



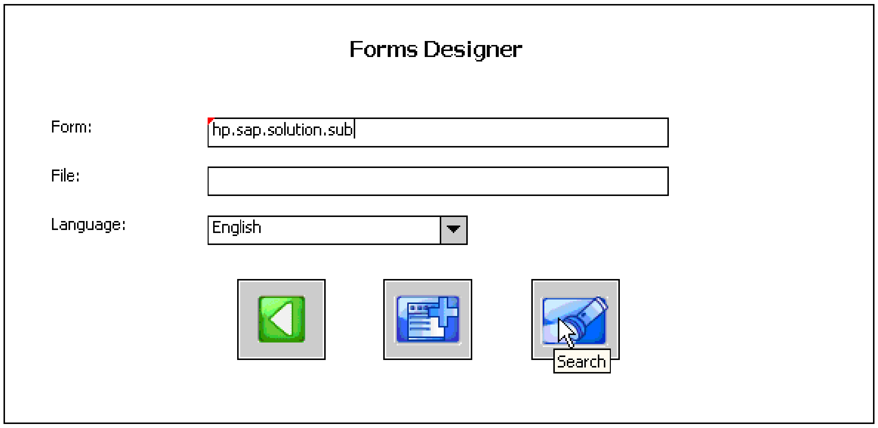
Forms Designer

Form:

File:

Language: English

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



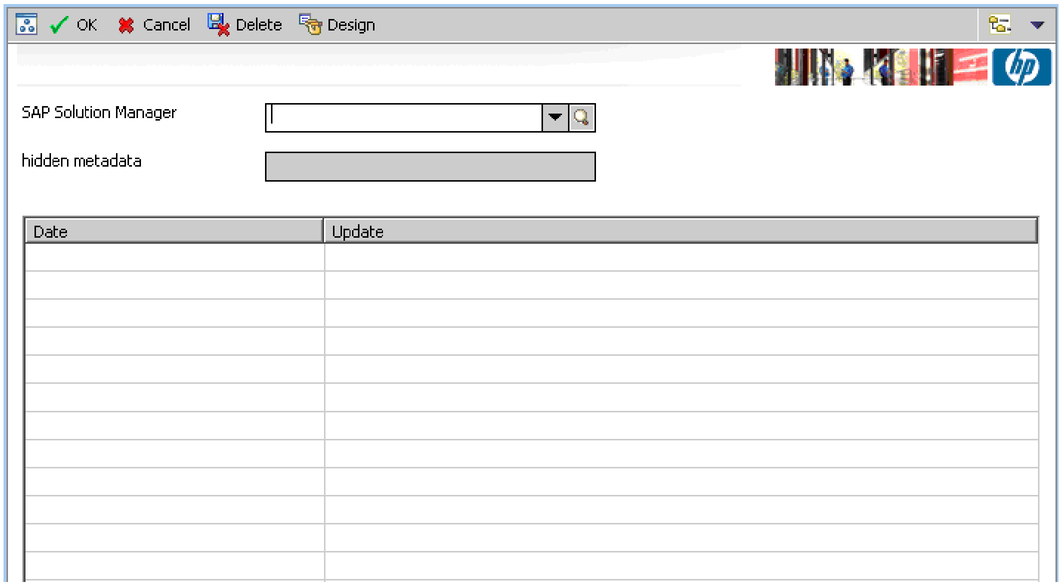
Forms Designer

Form: hp.sap.solution.sub

File:

Language: English

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

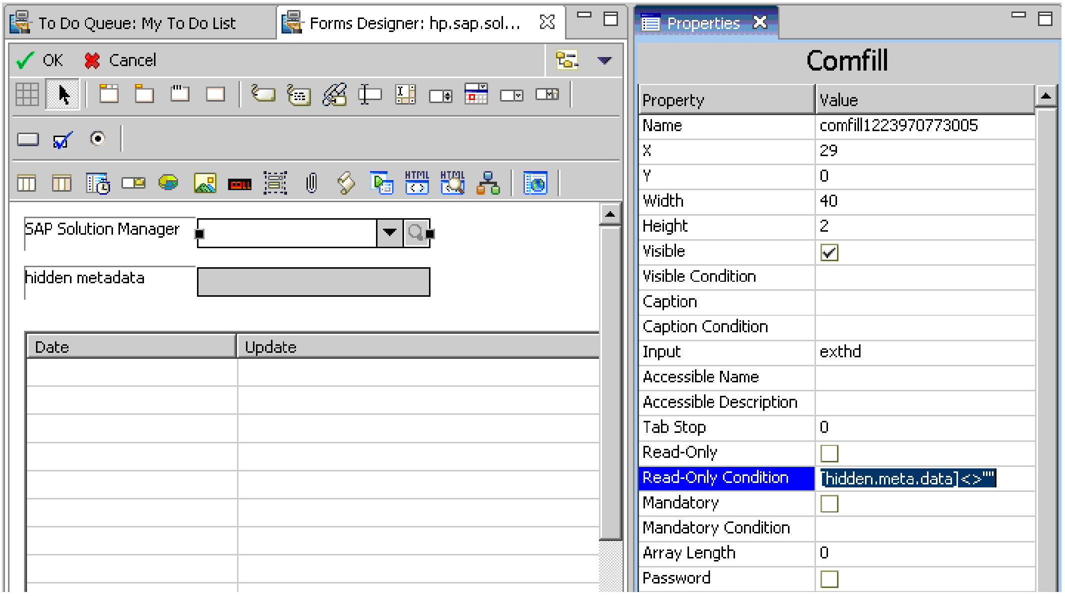


SAP Solution Manager

hidden metadata

Date	Update

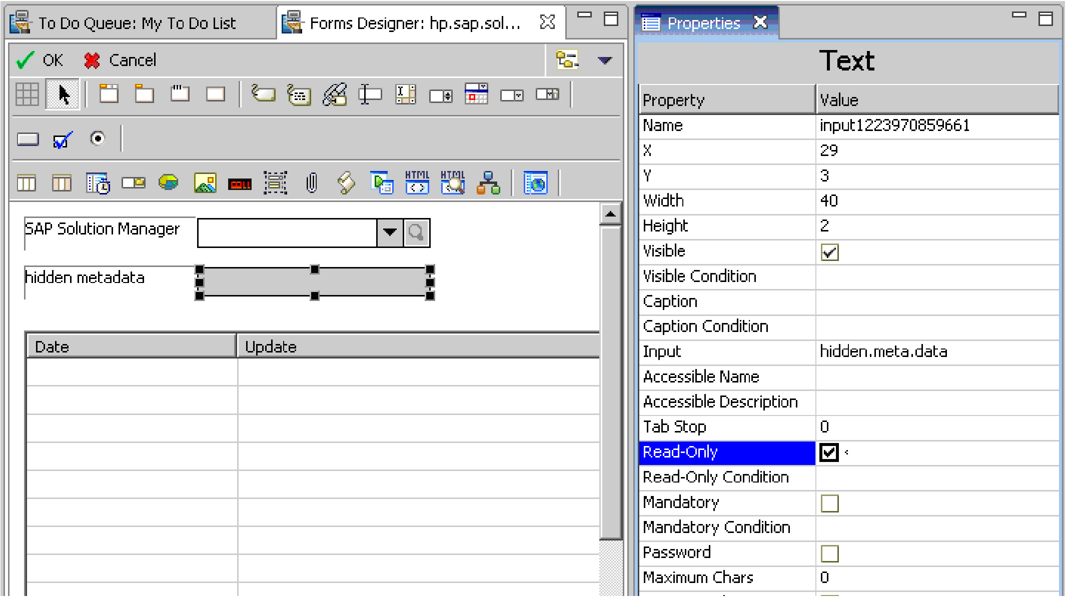
d Select **comfill** control from the dropdown list for **SAP Solution Manager** field, then in the Properties pane displayed, enter **`[hidden.meta.data]<>""`** as the value for Read-Only Condition property, and uncheck **Fill Button Visible** to hide the fill button.



Comfill

Property	Value
Name	comfill1223970773005
X	29
Y	0
Width	40
Height	2
Visible	☑
Visible Condition	
Caption	
Caption Condition	
Input	exthd
Accessible Name	
Accessible Description	
Tab Stop	0
Read-Only	□
Read-Only Condition	[hidden.meta.data]<>""
Mandatory	□
Mandatory Condition	
Array Length	0
Password	□

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



Text

Property	Value
Name	input1223970859661
X	29
Y	3
Width	40
Height	2
Visible	☑
Visible Condition	
Caption	
Caption Condition	
Input	hidden.meta.data
Accessible Name	
Accessible Description	
Tab Stop	0
Read-Only	☑
Read-Only Condition	
Mandatory	□
Mandatory Condition	
Password	□
Maximum Chars	0

3 Add the `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 44, Exchange History section in Chapter 5 for detailed instructions.

— On Service Center, see step 5 to step 6 on page 145, Exchange History section in Appendix C for detailed instructions.

4 Add a link record for field `exthd` of `probsummary` table:

— On Service Manager, see step 5 on page 45, Exchange History section in Chapter 5 for detailed instructions.

— On Service Center, see step 7 on page 145, Exchange History section in Appendix C for detailed instructions.

5 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 G, 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: 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 2: Configuring WSDL Mapping

See *Creating HPSAPSolutionManager Table* on page 65.

Task 3: 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 4: 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 5: 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_sm7.txt`.

4 Click **Save**.

Task 6: 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
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 7: 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 8: Renaming icons

Do the following:

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

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

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

Task 9: Checking if SAP Solution Manager is selected when clicking Send Incident

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 10: Upgrading SMSSMEX midware on Tomcat

Do the following:

1. Stop SMSSMEX V1.02.

   See *Starting/Stopping SMSSMEX* on page 27 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 19 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 24 to configure `ovictex.properties`.
   - c. See *Configuring File ovictexInternal.properties* on page 25 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 27 to start SMSSMEX V1.10.

Task 11: 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 29 to install SMSSMEX V1.10.

5 Configure SMSSMEX V1.10.

   a See *Configuring ovictex.properties* on page 29 to configure `ovictex.properties`.

   b See *Configuring File ovictexInternal.properties* on page 29 to configure `ovictexInternal.properties`.

   c See *External Helpdesks* on page 29 to configure external Helpdesks.

   d See *Configuring FieldMapping.xml* on page 29 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 30 to deploy the `ovictex.war` file on WebLogic.

7 Start SMSSMEX V1.10 and the WebLogic server.

Task 12: Triggerring URL

See *Trigger URL* on page 35 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"
   ```

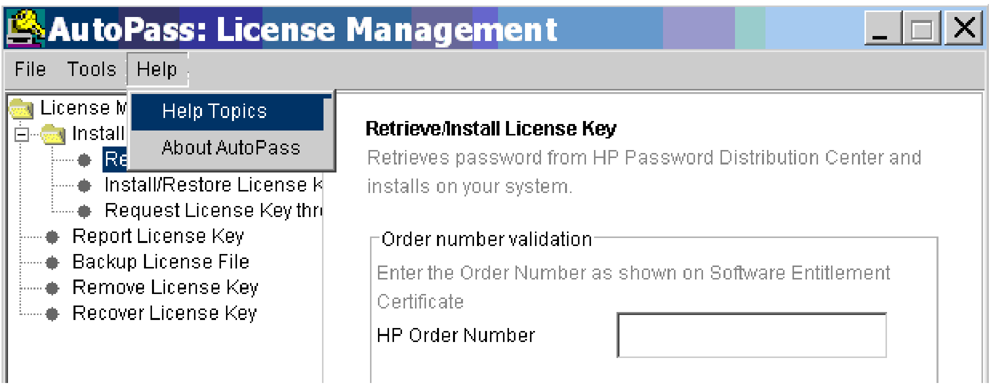
   b For HP-UX:

   ```
   /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).



Retrieve/Install License Key

Retrieves password from HP Password Distribution Center and installs on your system.

Order number validation

Enter the Order Number as shown on Software Entitlement Certificate

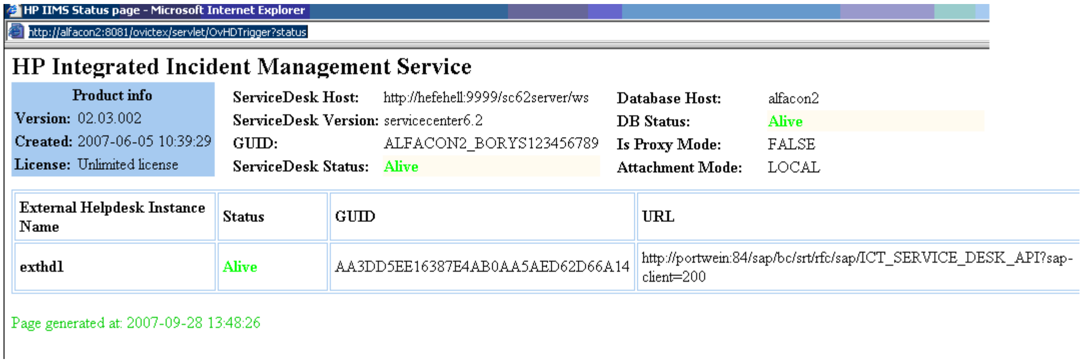
HP Order Number

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.



http://alfacon2:8081/ovictex/servlet/OvHDTrigger?status

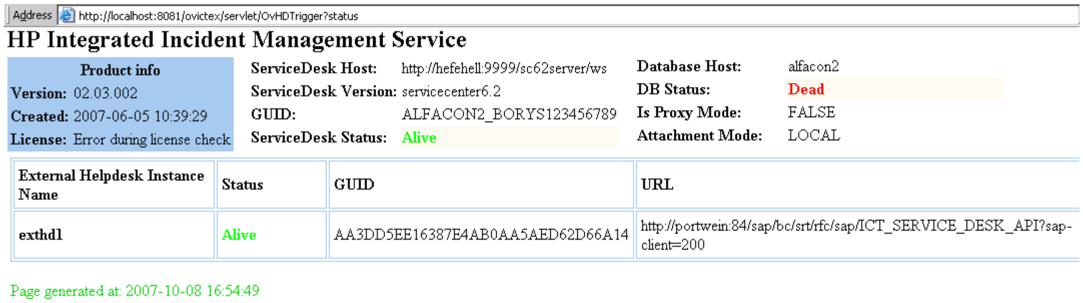
HP Integrated Incident Management Service

Product info				
Version: 02.03.002	ServiceDesk Host:	http://hefehell:9999/sc62server/ws	Database Host:	alfacon2
Created: 2007-06-05 10:39:29	ServiceDesk Version:	servicecenter6.2	DB Status:	Alive
License: Unlimited license	GUID:	ALFACON2_BORYS123456789	Is Proxy Mode:	FALSE
	ServiceDesk Status:	Alive	Attachment Mode:	LOCAL

External Helpdesk Instance Name	Status	GUID	URL
exthd1	Alive	AA3DD5EE16387E4AB0AA5AED62D66A14	http://portwein:84/sap/bc/srt/rfc/sap/ICT_SERVICE_DESK_API?sap-client=200

Page generated at: 2007-09-28 13:48:26

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



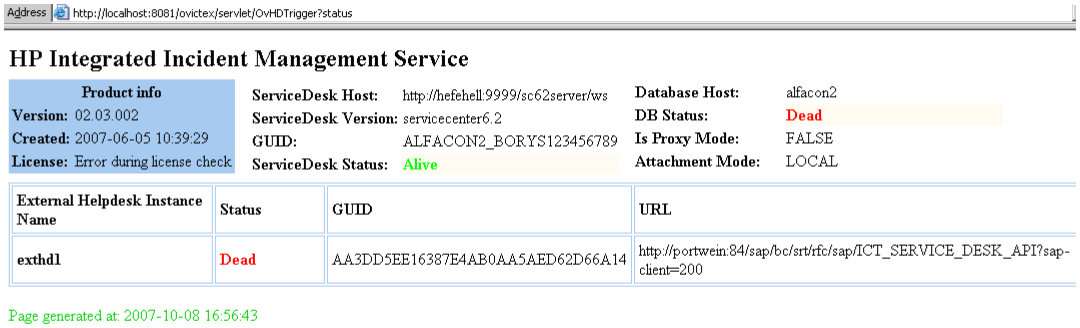
http://localhost:8081/ovictex/servlet/OvHDTrigger?status

HP Integrated Incident Management Service

Product info				
Version: 02.03.002	ServiceDesk Host:	http://hefehell:9999/sc62server/ws	Database Host:	alfacon2
Created: 2007-06-05 10:39:29	ServiceDesk Version:	servicecenter6.2	DB Status:	Dead
License: Error during license check	GUID:	ALFACON2_BORYS123456789	Is Proxy Mode:	FALSE
	ServiceDesk Status:	Alive	Attachment Mode:	LOCAL

External Helpdesk Instance Name	Status	GUID	URL
exthd1	Alive	AA3DD5EE16387E4AB0AA5AED62D66A14	http://portwein:84/sap/bc/srt/rfc/sap/ICT_SERVICE_DESK_API?sap-client=200

Page generated at: 2007-10-08 16:54:49



http://localhost:8081/ovictex/servlet/OvHDTrigger?status

HP Integrated Incident Management Service

Product info				
Version: 02.03.002	ServiceDesk Host:	http://hefehell:9999/sc62server/ws	Database Host:	alfacon2
Created: 2007-06-05 10:39:29	ServiceDesk Version:	servicecenter6.2	DB Status:	Dead
License: Error during license check	GUID:	ALFACON2_BORYS123456789	Is Proxy Mode:	FALSE
	ServiceDesk Status:	Alive	Attachment Mode:	LOCAL

External Helpdesk Instance Name	Status	GUID	URL
exthd1	Dead	AA3DD5EE16387E4AB0AA5AED62D66A14	http://portwein:84/sap/bc/srt/rfc/sap/ICT_SERVICE_DESK_API?sap-client=200

Page generated at: 2007-10-08 16:56:43

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

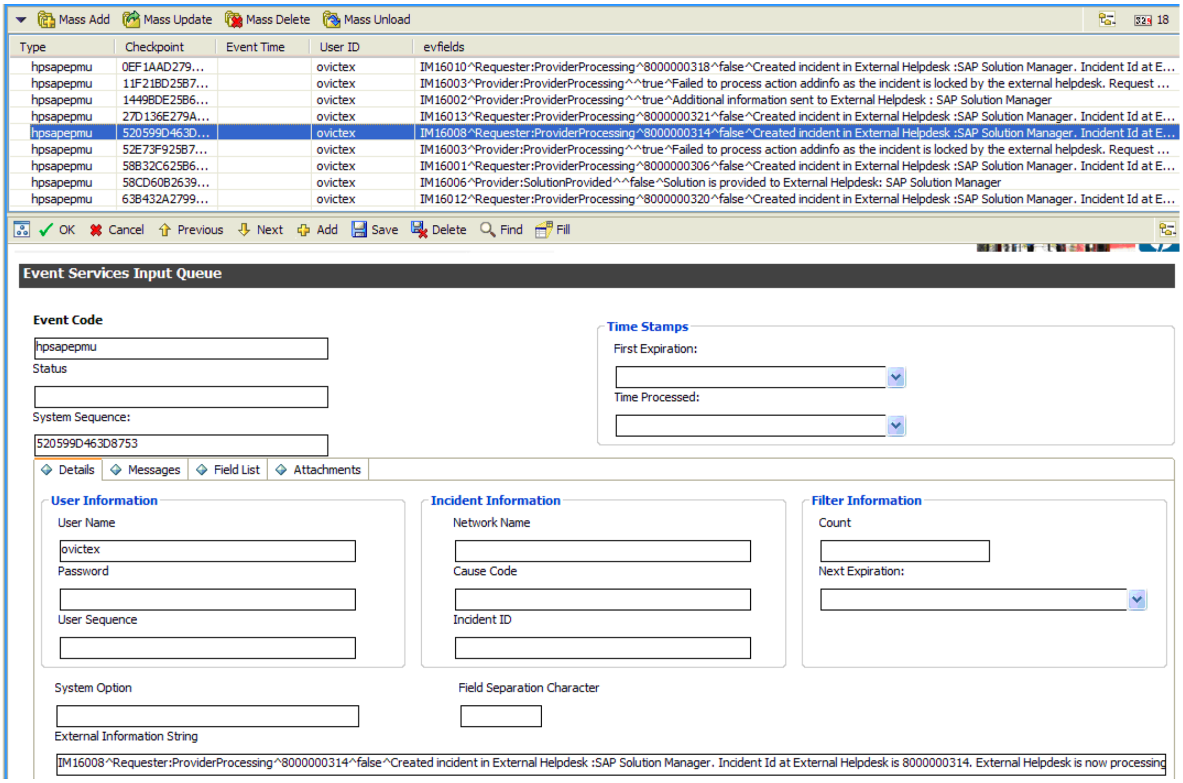
Problem

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).



Type	Checkpoint	Event Time	User ID	evfields
hpsapepmu	0EF1AAD279...		ovictex	IM16010^Requester:ProviderProcessing^8000000318^false^Created incident in External Helpdesk :SAP Solution Manager. Incident Id at E...
hpsapepmu	11F21BD25B7...		ovictex	IM16003^Provider:ProviderProcessing^^true^Failed to process action addinfo as the incident is locked by the external helpdesk. Request ...
hpsapepmu	1449BDE25B6...		ovictex	IM16002^Provider:ProviderProcessing^^true^Additional information sent to External Helpdesk : SAP Solution Manager
hpsapepmu	27D136E279A...		ovictex	IM16013^Requester:ProviderProcessing^8000000321^false^Created incident in External Helpdesk :SAP Solution Manager. Incident Id at E...
hpsapepmu	520599D463D...		ovictex	IM16008^Requester:ProviderProcessing^8000000314^false^Created incident in External Helpdesk :SAP Solution Manager. Incident Id at E...
hpsapepmu	52E73F925B7...		ovictex	IM16003^Provider:ProviderProcessing^^true^Failed to process action addinfo as the incident is locked by the external helpdesk. Request ...
hpsapepmu	58B32C625B6...		ovictex	IM16001^Requester:ProviderProcessing^8000000306^false^Created incident in External Helpdesk :SAP Solution Manager. Incident Id at E...
hpsapepmu	58CD60B2639...		ovictex	IM16006^Provider:SolutionProvided^^false^Solution is provided to External Helpdesk: SAP Solution Manager
hpsapepmu	63B432A2799...		ovictex	IM16012^Requester:ProviderProcessing^8000000320^false^Created incident in External Helpdesk :SAP Solution Manager. Incident Id at E...

Event Services Input Queue

Event Code: hpsapepmu

Status:

System Sequence: 520599D463D8753

Time Stamps

First Expiration:

Time Processed:

User Information

User Name: ovictex

Password:

User Sequence:

Incident Information

Network Name:

Cause Code:

Incident ID:

Filter Information

Count:

Next Expiration:

System Option:

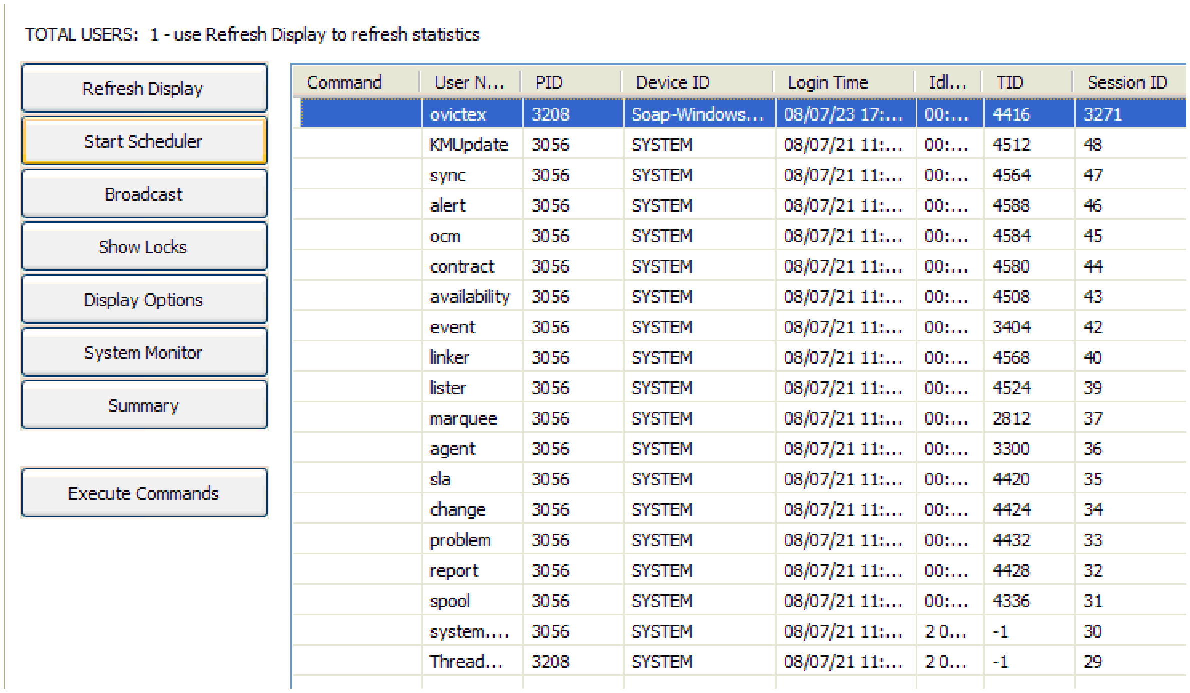
Field Separation Character:

External Information String: IM16008^Requester:ProviderProcessing^8000000314^false^Created incident in External Helpdesk :SAP Solution Manager. Incident Id at External Helpdesk is 8000000314. External Helpdesk is now processing

Solution

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

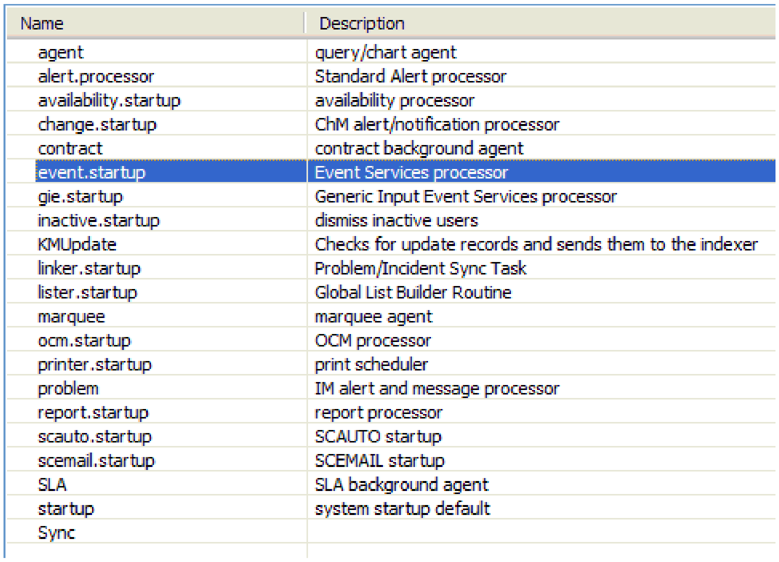
1 Go to **System Status**



TOTAL USERS: 1 - use Refresh Display to refresh statistics

Command	User N...	PID	Device ID	Login Time	Idl...	TID	Session ID
	ovictex	3208	Soap-Windows...	08/07/23 17:...	00:...	4416	3271
	KMUpdate	3056	SYSTEM	08/07/21 11:...	00:...	4512	48
	sync	3056	SYSTEM	08/07/21 11:...	00:...	4564	47
	alert	3056	SYSTEM	08/07/21 11:...	00:...	4588	46
	ocm	3056	SYSTEM	08/07/21 11:...	00:...	4584	45
	contract	3056	SYSTEM	08/07/21 11:...	00:...	4580	44
	availability	3056	SYSTEM	08/07/21 11:...	00:...	4508	43
	event	3056	SYSTEM	08/07/21 11:...	00:...	3404	42
	linker	3056	SYSTEM	08/07/21 11:...	00:...	4568	40
	lister	3056	SYSTEM	08/07/21 11:...	00:...	4524	39
	marquee	3056	SYSTEM	08/07/21 11:...	00:...	2812	37
	agent	3056	SYSTEM	08/07/21 11:...	00:...	3300	36
	sla	3056	SYSTEM	08/07/21 11:...	00:...	4420	35
	change	3056	SYSTEM	08/07/21 11:...	00:...	4424	34
	problem	3056	SYSTEM	08/07/21 11:...	00:...	4432	33
	report	3056	SYSTEM	08/07/21 11:...	00:...	4428	32
	spool	3056	SYSTEM	08/07/21 11:...	00:...	4336	31
	system....	3056	SYSTEM	08/07/21 11:...	2 0...	-1	30
	Thread...	3208	SYSTEM	08/07/21 11:...	2 0...	-1	29

2 Click **Start Scheduler**.



Name	Description
agent	query/chart agent
alert.processor	Standard Alert processor
availability.startup	availability processor
change.startup	ChM alert/notification processor
contract	contract background agent
event.startup	Event Services processor
gie.startup	Generic Input Event Services processor
inactive.startup	dismiss inactive users
KMUpdate	Checks for update records and sends them to the indexer
linker.startup	Problem/Incident Sync Task
lister.startup	Global List Builder Routine
marquee	marquee agent
ocm.startup	OCM processor
printer.startup	print scheduler
problem	IM alert and message processor
report.startup	report processor
scauto.startup	SCAUTO startup
scemail.startup	SCEMAIL startup
SLA	SLA background agent
startup	system startup default
Sync	

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

Incident Update or Process Action Fails

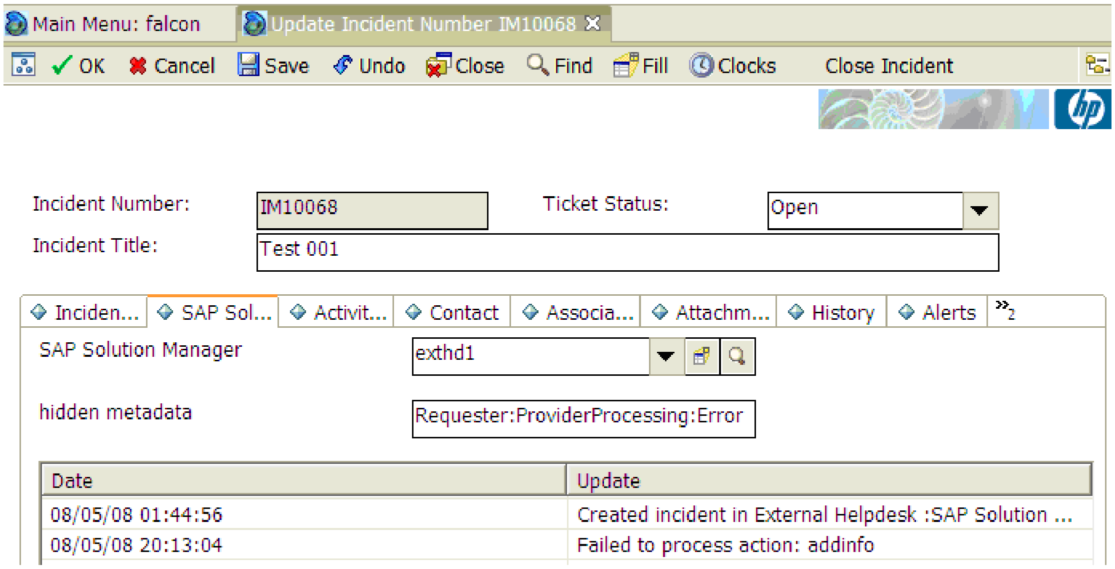
Problem

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:



Incident Number: IM10068

Ticket Status: Open

Incident Title: Test 001

SAP Solution Manager: exthd1

hidden metadata: Requester:ProviderProcessing:Error

Date	Update
08/05/08 01:44:56	Created incident in External Helpdesk :SAP Solution ...
08/05/08 20:13:04	Failed to process action: addinfo

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

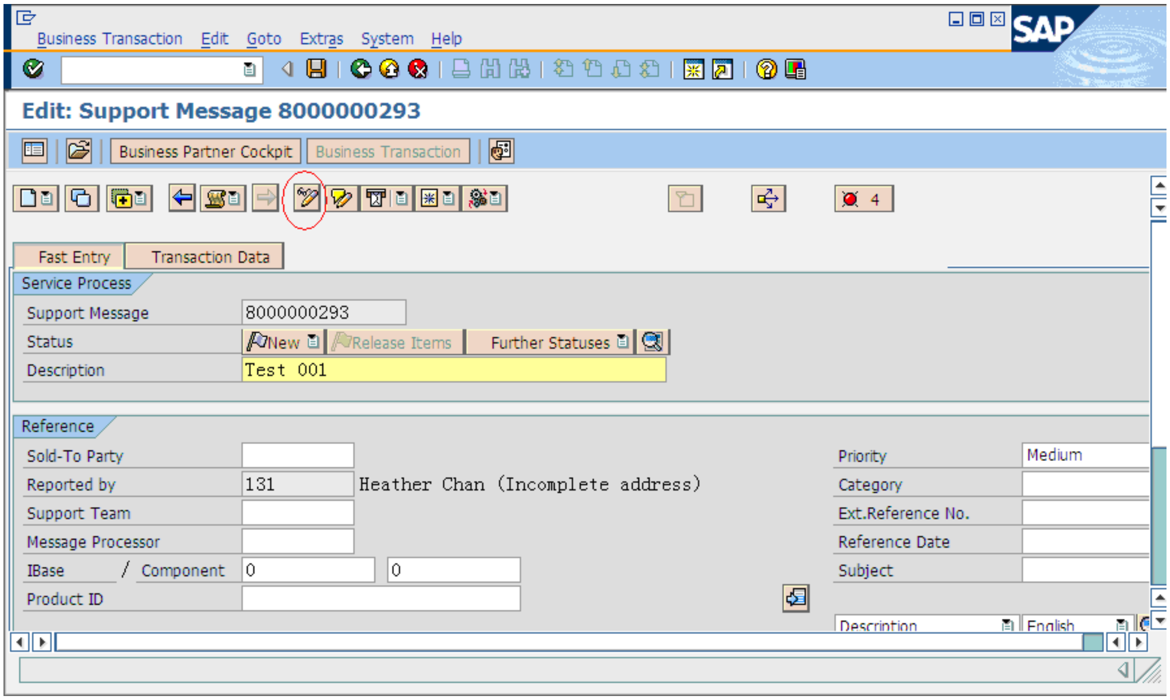
The incident in HP Service Manager or SAP Solution Manager is locked:

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.



Edit: Support Message 8000000293

Support Message 8000000293

Status New

Description Test 001

Sold-To Party

Reported by 131 Heather Chan (Incomplete address)

Support Team

Message Processor

IBase / Component 0 0

Product ID

Priority Medium

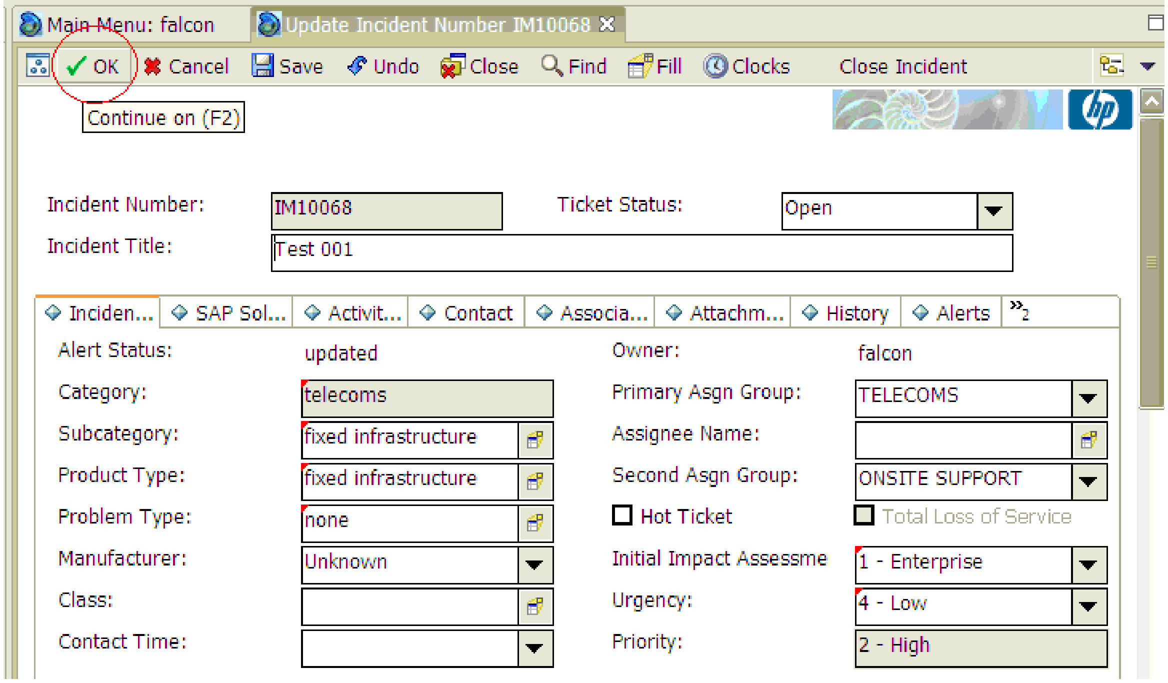
Category

Ext.Reference No.

Reference Date

Subject

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



Incident Number: IM10068

Ticket Status: Open

Incident Title: Test 001

Alert Status: updated

Owner: falcon

Category: telecoms

Primary Asgn Group: TELECOMS

Subcategory: fixed infrastructure

Assignee Name:

Product Type: fixed infrastructure

Second Asgn Group: ONSITE SUPPORT

Problem Type: none

□ Hot Ticket

□ Total Loss of Service

Manufacturer: Unknown

Initial Impact Assessme: 1 - Enterprise

Class:

Urgency: 4 - Low

Contact Time:

Priority: 2 - High

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:

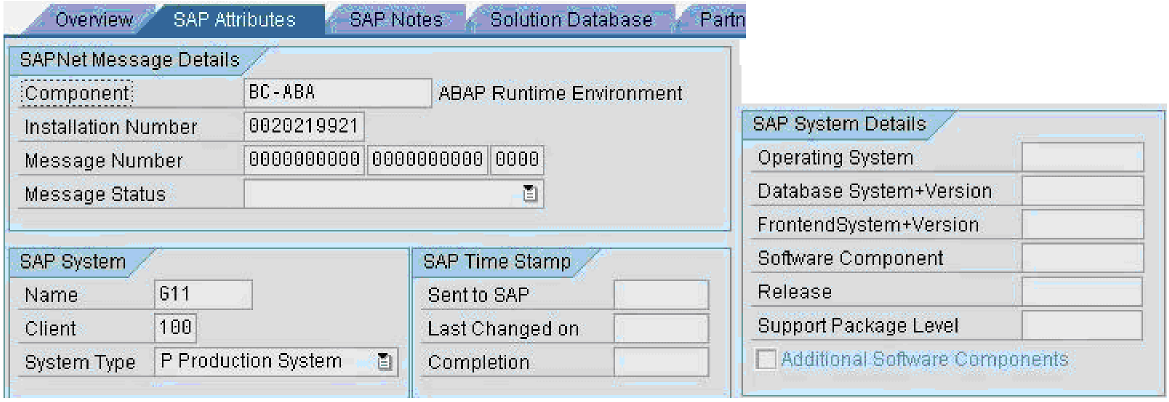
Table 9 Attribute Types of the SAP Solution Manager

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.



SAPNet Message Details

Component BC-ABA ABAP Runtime Environment

Installation Number 0020219921

Message Number 0000000000 0000000000 0000

Message Status

SAP System

Name G11

Client 100

System Type P Production System

SAP Time Stamp

Sent to SAP

Last Changed on

Completion

SAP System Details

Operating System

Database System+Version

FrontendSystem+Version

Software Component

Release

Support Package Level

□ Additional Software Components

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]
    ```

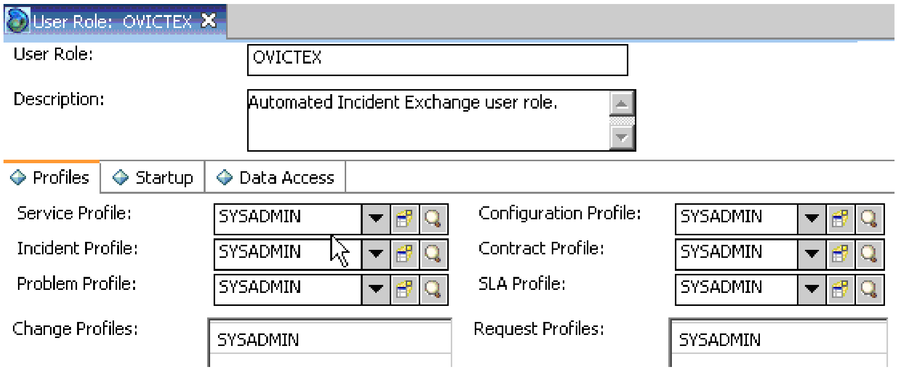
12. In transaction **`/nsmicm`** select from the **Administration** → **ICM** → **Restart** → **Yes** to restart ICM.

C Customizing HP ServiceCenter

This chapter describes the customization required for HP ServiceCenter for the integration. For Service Manager customization, see Chapter 5, Customizing HP Service Manager.

Creating a ServiceCenter User

Incident Exchange uses one ServiceCenter user to connect to ServiceCenter web services. The user and the user role should be dedicated to Incident Exchange.



User Role: OVICTEX

Description: Automated Incident Exchange user role.

Service Profile:	SYSADMIN	Configuration Profile:	SYSADMIN
Incident Profile:	SYSADMIN	Contract Profile:	SYSADMIN
Problem Profile:	SYSADMIN	SLA Profile:	SYSADMIN
Change Profiles:	SYSADMIN	Request Profiles:	SYSADMIN

Do the following:

1. Select **Utilities** → **Administration** → **Security** → **User Administration** → **Search for User Roles**.
2. Search for **SYSTEM ADMINISTRATOR**.
3. Enter **`OVICTEX`** as the User Role.
4. Change Description to **`Automated Incident Exchange user role`**.
5. Click **Add**.
6. Select **Utilities** → **Administration** → **Security** → **User Administration** → **User Quick Add Utility**.
7. Enter **`ovictex, INCIDENT EXCHANGE, Incident, Exchange, ovictex@hp.com`**.
8. Click **Next**.
9. Select to clone user **falcon**.
10. Click **Finish**.
11. Change User Role to **`OVICTEX`**.
12. In the Security tab
    a. Enter the operator password for `Password` field.
    b. Uncheck **Expire Password**.

c Check **Never Expire Password**.

13 Click **Save**.

Importing Customizations via Unload

Unloads are used to transfer customizations from one ServiceCenter to another. The Incident Exchange provides a core unload at <*SMSSMEX1.10 Release Package*>`\unloads\SC62x\core_sc6.2.unl`. This unload contains new ServiceCenter records that are unique to Incident Exchange and do not override any existing ServiceCenter records. The unload specifies settings for the following:

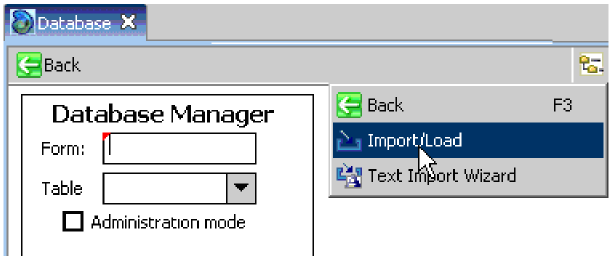
- Trigger URL
- Incident custom fields
- Web service exposure
- Event In
- Trigger process
- Template
- Incident form
- SAP Configuration Item handling

Core Unload

To import the unload do the following:

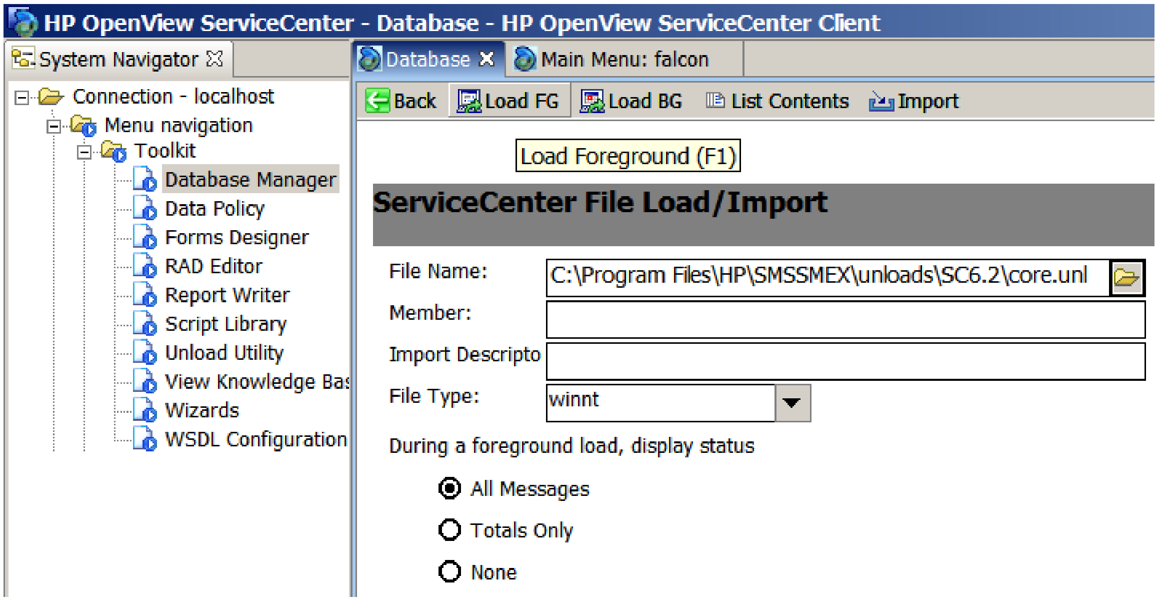
1 Select **Toolkit** → **Database Manager**.

2 Click **Import/Load**.



3 Select <*SMSSMEX1.10 Release Package*>`\unloads\SC62x\core_sc6.2.unl`.

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



ServiceCenter File Load/Import

File Name: C:\Program Files\HP\SMSSMEX\unloads\SC6.2\core.unl

Member:

Import Descripto

File Type: winnt

During a foreground load, display status

● All Messages

○ Totals Only

○ None

Demo Unload

For demo purposes, there is an additional demo unload with all customizations. It is based on an uncustomized, default installation of ServiceCenter. The demo unload allows you to quickly 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 will not work in a customized helpdesk. The demo unload overrides many standard ServiceCenter records, causing the helpdesk to work only for the demo. The demo unload can not be removed or undone.

To import the unload do the following:

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

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

3 Browse to the unload at <*SMSSMEX1.10 Release Package*>`\unloads\SC62x\demo_sc6.2.unl`.

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

Customizing Demo Unload Manually

This section describes how to manually customize the demo unload.

Incident Custom Fields and Web Service Exposure

Incident Exchange accesses ServiceCenter 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 **Navigation** → **System Definition** → **Tables** → **probsummary** → **Tab Fields and Keys**.
2. Create the following additional fields in table `probsummary` (check **v Include in API** for all fields).

Table 11 Incident custom fields and web service

Field name	Type	Caption	Field name in API	Field data type in API
custom.text.01	Text	CustomText01	CustomText01	StringType
custom.text.02	Text	CustomText02	CustomText02	StringType
...	...	...	...	...
custom.text.10	Text	CustomText10	CustomText10	StringType
sap.sid	Text	SAP SID	SapSid	StringType
sap.client	Text	SAP Client	SapClient	StringType
sap.installationnumber	Text	SAP installation number	SapInstallationNumber	StringType
hidden.meta.data	Text	Hidden meta data for Incident Exchange	HiddenMetaData	StringType
is.incident.exchange	Boolean	Flag for affiliation with Incident Exchange	IsIncidentExchange	BooleanType
exthd	Text	External helpdesk for Incident Exchange	Exthd	StringType

3. The history is written into an additional field in `probsummary`:

— 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, so a handshaking mechanism is required so that an action cannot be triggered 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.

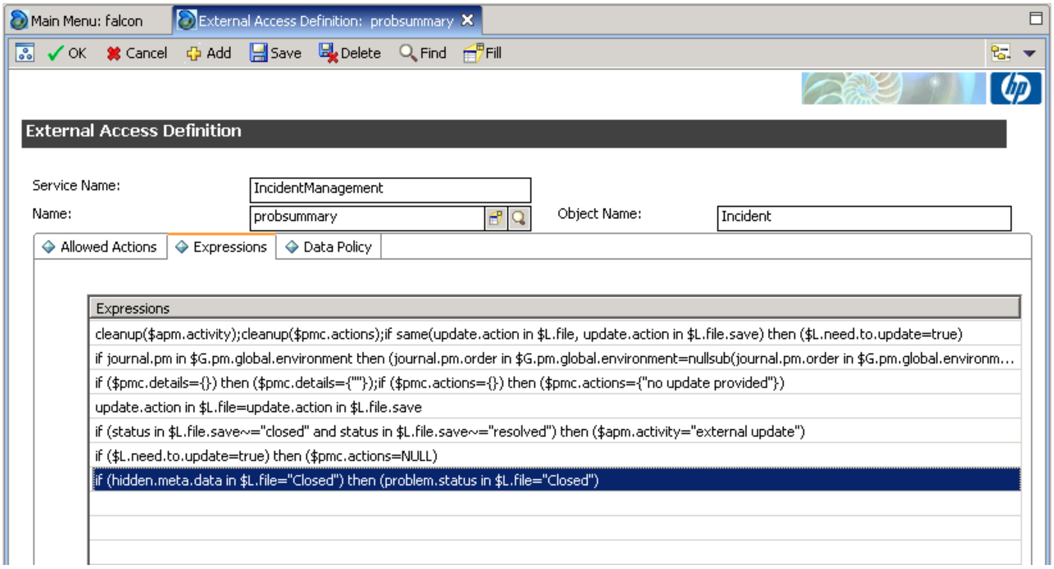
Field name	Type	Caption
exchange.history exchange.history date.stamp history.update	Array Structure Date/time Text	Log of Incident Exchange actions and events.
is.ictex.action.blocked	Boolean	Indicates if Incident Exchange is performing an exchange action.

5 Include the following `probsummary` table fields in the web service API.

Field name	Type	Caption	Field name in API	API Field type
priority.code	Text	Priority Code	PriorityCode	StringType
planned.end	Date/time	Planned End	PlannedEnd	DateTimeType

6 Go to **Utilities** → **Tools** → **Web Services** → **External Access**, enter **`IncidentManagement`** in Service Name field and click **Search**, and then add the following expression (**[INCIDENTMANAGEMENT]** in `code_sc6.txt`) in Expressions.

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



External Access Definition

Service Name: IncidentManagement

Name: probsummary

Object Name: Incident

Expressions
cleanup($apm.activity);cleanup($pmc.actions);if same(update.action in $L.file, update.action in $L.file.save) then ($L.need.to.update=true)
if journal.pm in $G.pm.global.environment then (journal.pm.order in $G.pm.global.environment=nullsub(journal.pm.order in $G.pm.global.environm...
if ($pmc.details={}) then ($pmc.details={""});if ($pmc.actions={}) then ($pmc.actions={"no update provided"})
update.action in $L.file=update.action in $L.file.save
if (status in $L.file.save~="closed" and status in $L.file.save~="resolved") then ($apm.activity="external update")
if ($L.need.to.update=true) then ($pmc.actions=NULL)
if (hidden.meta.data in $L.file="Closed") then (problem.status in $L.file="Closed")

Contacts Web Service Exposure

To expose contacts web service, do the following:

1 Select **Navigation** → **Utilities** → **Tools** → **Web Services** → **External Access**.

2 The `contacts` table is exposed in `ConfigurationManagement.wsdl`. The following fields already exist in table `contacts`, but need to be included in the web service API:

Field name	Type	Caption	Field name in API	Field data type in API
fax.phone	Text	fax.phone	Fax	StringType
operator.id	Text	operator.id	OperatorID	StringType

Event Services Web Service Exposure

To expose event services web service, do the following:

1 Select **Navigation** → **Utilities** → **Tools** → **Web Services** → **External Access**.

2 Enter

— Service Name: **`EventIn`**

— Name: **`eventin`**

— Object Name: **`Eventin`**

3 Insert

— Allowed Action: **`add`**

— Action Name: **`Create`**

4 Click **Add**.

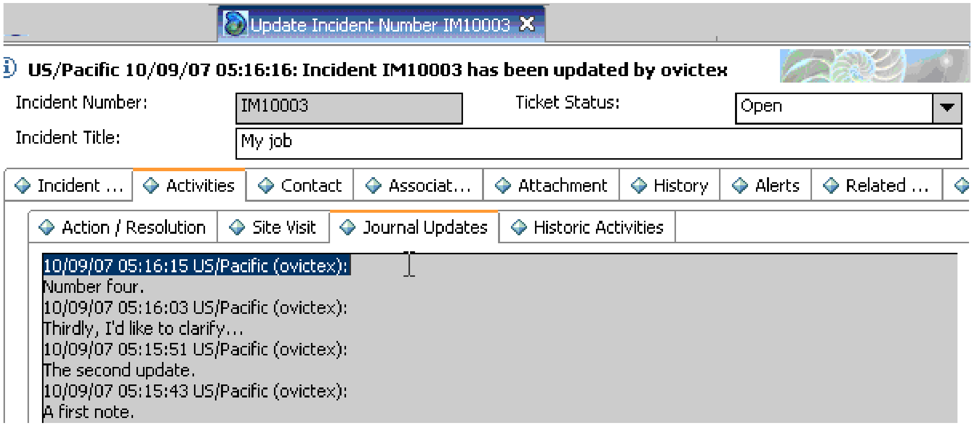
5 In the Data Policy tab, expose the following fields and save the definition.

Field name	Type	Caption	Field name in API	Field data type in API
evfields	Text	evfields		StringType
evtype	Text	evtype		StringType
evuser	Text	evuser		StringType
evsysseq	Text	evsysseq		StringType

6 Restart the ServiceCenter server.

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.



Update Incident Number IM10003

US/Pacific 10/09/07 05:16:16: Incident IM10003 has been updated by ovictex

Incident Number: IM10003

Ticket Status: Open

Incident Title: My job

```
10/09/07 05:16:15 US/Pacific (ovictex):
Number four.
10/09/07 05:16:03 US/Pacific (ovictex):
Thirdly, I'd like to clarify...
10/09/07 05:15:51 US/Pacific (ovictex):
The second update.
10/09/07 05:15:43 US/Pacific (ovictex):
A first note.
```

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 ServiceCenter.

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 ServiceCenter 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 **Navigation** → **Utilities** → **Tools** → **Document Engine** → **Processes**.

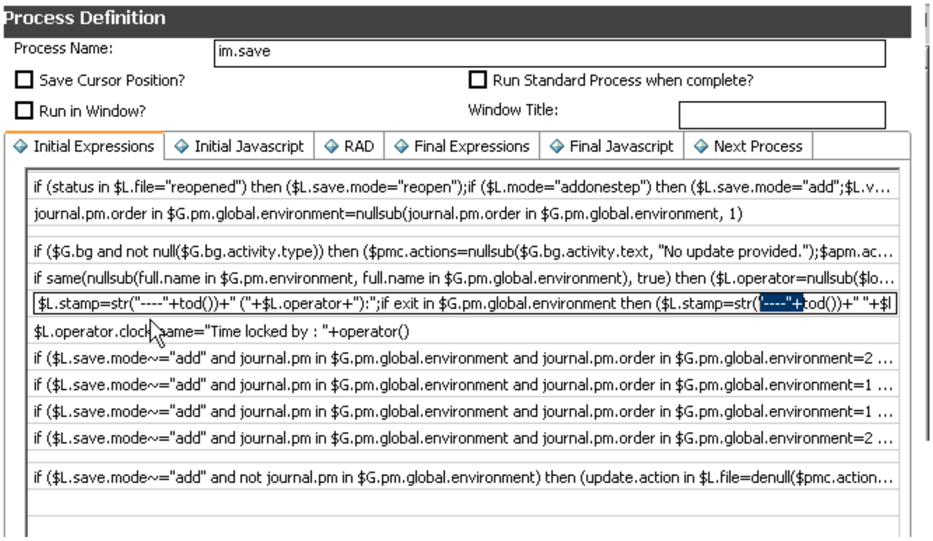
2 Search for all Processes starting with `im.`.

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+”):”.
```

The entire line (**[IM._JOURNAL]** in `code_sc6.txt`) is

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



Process Definition

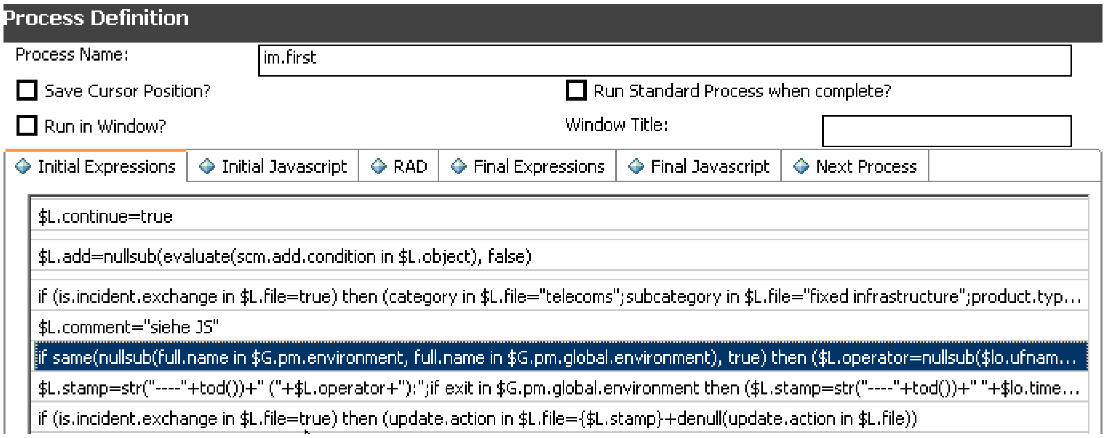
Process Name: im.save

□ Save Cursor Position? □ Run Standard Process when complete?

□ Run in Window? Window Title:

```
if (status in $L.file="reopened") then ($L.save.mode="reopen");if ($L.mode="addonestep") then ($L.save.mode="add";$L.v...
journal.pm.order in $G.pm.global.environment=nullsub(journal.pm.order in $G.pm.global.environment, 1)

if ($G.bg and not null($G.bg.activity.type)) then ($pmc.actions=nullsub($G.bg.activity.text, "No update provided.");$apm.ac...
if same(nullsub(full.name in $G.pm.environment, full.name in $G.pm.global.environment), true) then ($L.operator=nullsub($lo...
$L.stamp=str("----"+tod())+" ("+$L.operator+"):";if exit in $G.pm.global.environment then ($L.stamp=str("----"+tod())+" "+$l
$L.operator.clock.name="Time locked by : "+operator()
if ($L.save.mode~="add" and journal.pm in $G.pm.global.environment and journal.pm.order in $G.pm.global.environment=2 ...
if ($L.save.mode~="add" and journal.pm in $G.pm.global.environment and journal.pm.order in $G.pm.global.environment=1 ...
if ($L.save.mode~="add" and journal.pm in $G.pm.global.environment and journal.pm.order in $G.pm.global.environment=1 ...
if ($L.save.mode~="add" and journal.pm in $G.pm.global.environment and journal.pm.order in $G.pm.global.environment=2 ...

if ($L.save.mode~="add" and not journal.pm in $G.pm.global.environment) then (update.action in $L.file=denull($pmc.action...
```

4 Normally the first Journal entry is only entered *after* the Incident has been created. But Incident Exchange adds Journal entries (the external helpdesk Incident ID) during creation of an Incident. To ensure that the initial Journal entries also contain a separator (required for block detection by Incident Exchange), add the following statements in `im.first` to insert a separator.



Process Definition

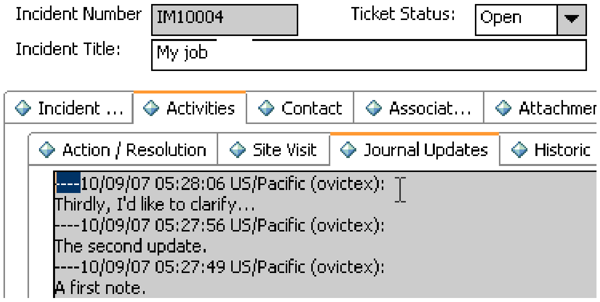
Process Name: im.first

□ Save Cursor Position? □ Run Standard Process when complete?

□ Run in Window? Window Title:

```
$L.continue=true
$L.add=nullsub(evaluate(scm.add.condition in $L.object), false)
if (is.incident.exchange in $L.file=true) then (category in $L.file="telecoms";subcategory in $L.file="fixed infrastructure";product.typ...
$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.ufnam...
$L.stamp=str("----"+tod())+" ("+$L.operator+"):";if exit in $G.pm.global.environment then ($L.stamp=str("----"+tod())+" "+$lo.time...
if (is.incident.exchange in $L.file=true) then (update.action in $L.file={$L.stamp}+denull(update.action in $L.file))
```

After the modification, all Journal updates should contain the separator as shown in the following diagram.



Incident Number: IM10004 Ticket Status: Open

Incident Title: My job

```
----10/09/07 05:28:06 US/Pacific (ovictex):
Thirdly, I'd like to clarify...
----10/09/07 05:27:56 US/Pacific (ovictex):
The second update.
----10/09/07 05:27:49 US/Pacific (ovictex):
A first note.
```

Template

When a new Incident is sent from SAP Solution Manager to ServiceCenter, Incident Exchange creates a new Incident with data for exchanged fields. The Incident management process inside ServiceCenter may require additional mandatory fields (e.g. `category`, `subcategory`, `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 ServiceCenter, the Process im.first is invoked when an Incident is submitted.

1 Select **Utilities** → **Tools** → **Document Engine** → **Processes**.

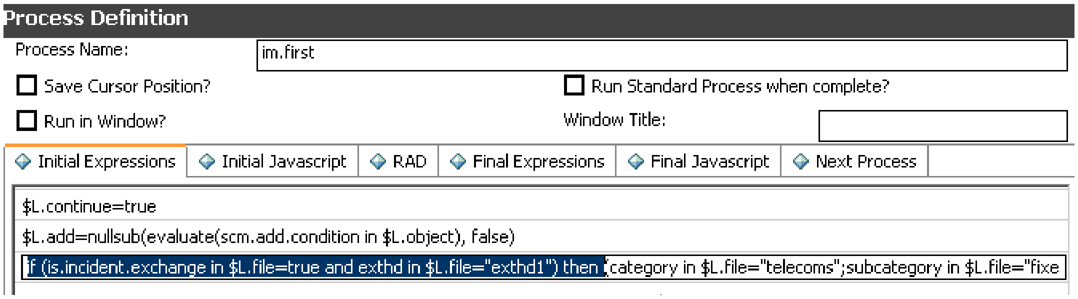
2 Search for **`im.first`**.

3 Add an Initial Expression that sets all required fields in `probsummary` that are not yet set by Incident Exchange.

> The Boolean field `is.incident.exchange` in `probsummary` is set by Incident Exchange and indicates if the Incident is opened by the Incident Exchange (or by some other means). If multiple external helpdesks are connected to ServiceCenter 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.

The following is an example expression:

```
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"})
```



Process Name: im.first

□ Save Cursor Position? □ Run Standard Process when complete?

□ Run in Window? Window Title:

```
$L.continue=true
$L.add=nullsub(evaluate(scm.add.condition in $L.object), false)
if (is.incident.exchange in $L.file=true and exthd in $L.file="exthd1") then (category in $L.file="telecoms";subcategory in $L.file="fixe
```

4 Add the following expressions (**[IM.FIRST_INIT]** in `code_sc6.txt`) to the Initial Expressions tab of `im.first`.

```
if (is.incident.exchange in $L.file=true) 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";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"})
$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))
```

Incident form

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

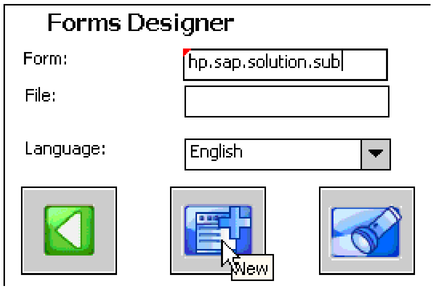
Status and Hidden Metadata

The hidden metadata field stores the current Incident Exchange state and ServiceCenter 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 ServiceCenter 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.



Forms Designer

Form: hp.sap.solution.sub

File:

Language: English

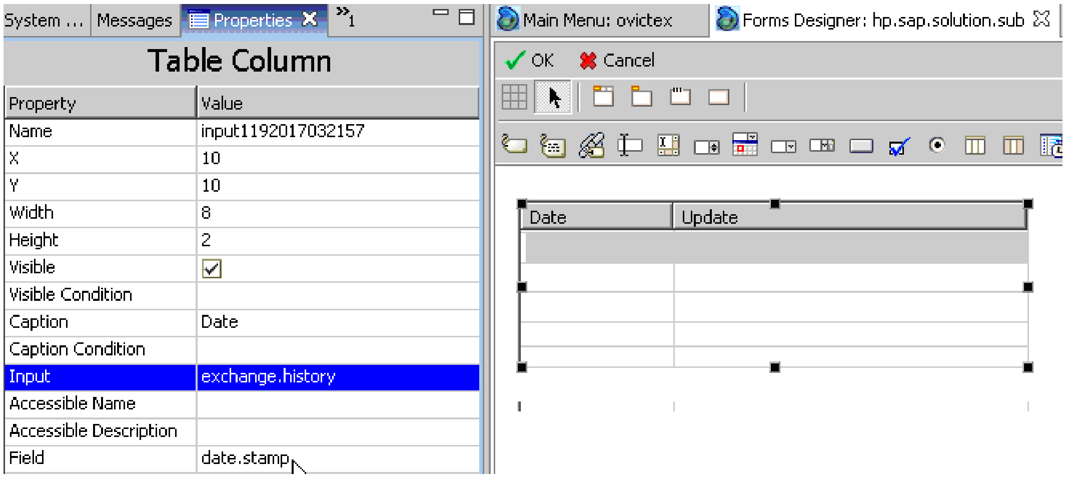


Table Column

Property	Value
Name	input1192017032157
X	10
Y	10
Width	8
Height	2
Visible	☑
Visible Condition	
Caption	Date
Caption Condition	
Input	exchange.history
Accessible Name	
Accessible Description	
Field	date.stamp

The `probsummary` table contains the Array field `exchange.history` that contains a Structure of `date.stamp` and `history.update`. The table can be show 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 Select **Toolkit** → **Forms Designer**.

2 Enter a name such **`hp.sap.solution.sub`** for the subform and press **New** (do not use the Forms wizard).

3 Insert a Table control.

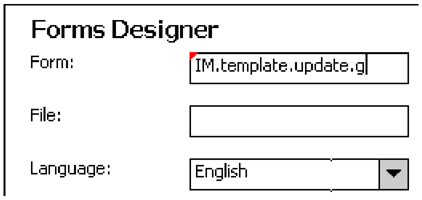
4 Connect the two Table control columns with the fields `date.stamp` and `history.update` using `exchange.history` as the input.

Table 12 Table Control

Control Component	Property	Value
“Date” column	Input	exchange.history
	Field	date.stamp
	Caption	Date
“Update” column	Input	exchange.history
	Field	history.update
	Caption	Update
Label	Caption	SAP Solution Manager
Label	Caption	hidden metadata

Table 12 Table Control

Control Component	Property	Value
Comfill control	Input	exthd
	Value List Condition	select("id","HPSAPSolution Manager","true","true")
	Display List Condition	select("name","HPSAPSolutionManager","true","true")
	Read-Only Condition	[hidden.meta.data]<>""
	Fill Button Visible	unchecked
Text control	Input	hidden.meta.data
	Read-Only	checked

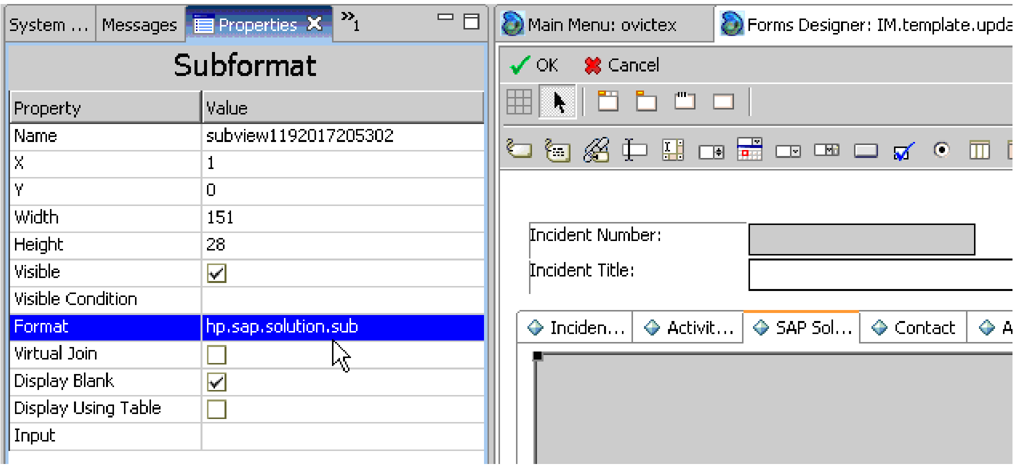


Forms Designer

Form: IM.template.update.g

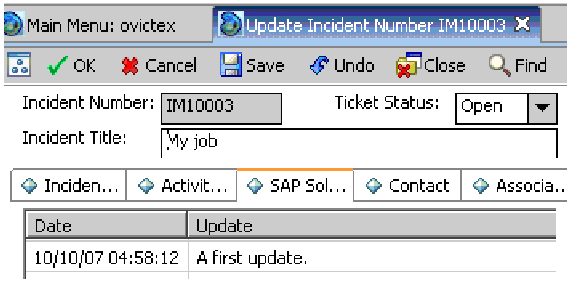
File:

Language: English



Subformat

Property	Value
Name	subview1192017205302
X	1
Y	0
Width	151
Height	28
Visible	☑
Visible Condition	
Format	hp.sap.solution.sub
Virtual Join	□
Display Blank	☑
Display Using Table	□
Input	



Incident Number: IM10003

Ticket Status: Open

Incident Title: My job

Date	Update
10/10/07 04:58:12	A first update.

5 Open all incident workflow Incident Forms:

```
IM.template.open.g
IM.template.update.g
apm.quick.g
IM.default.open.g
IM.default.update.g
```

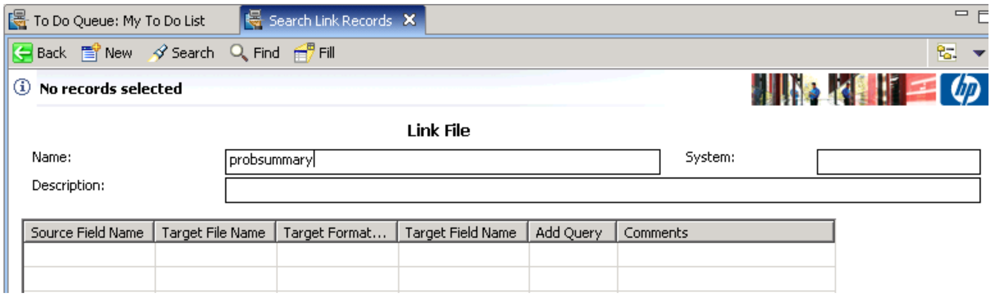
6 Add the created subform to the Incident form (in a new Notebook tab).

Table 13 Embedded Subform

Component	Property	Value
Notebook tab	Caption	Sap Solution Manager
Subform Control in “Sap Solution Manager” tab	Format	hp.sap.solution.sub

7 Add a link record for field `exthd` of `probsummary` table.

a Go to **Utilities** → **Tools** → **Links**. The Link File page appears. Enter **`probsummary`** in Name field, then click **Search**.



No records selected

Link File

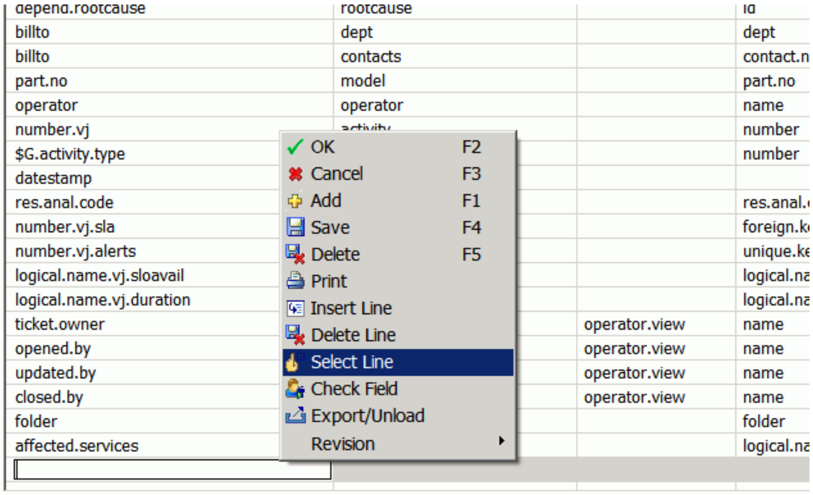
Name: probsummary

System:

Description:

Source Field Name	Target File Name	Target Format...	Target Field Name	Add Query	Comments

b Right click on an empty line in the table, select **Select Line** from the pop-up menu. The Edit Link page appears.



depend.rootcause	rootcause		id
billto	dept		dept
billto	contacts		contact.n
part.no	model		part.no
operator	operator		name
number.vj	activity		number
$G.activity.type			number
datestamp			
res.anal.code			res.anal.
number.vj.sla			foreign.k
number.vj.alerts			unique.ke
logical.name.vj.sloavail			logical.na
logical.name.vj.duration			logical.na
ticket.owner		operator.view	name
opened.by		operator.view	name
updated.by		operator.view	name
closed.by		operator.view	name
folder			folder
affected.services			logical.na

c Add the following values, and then click **Save**.

Field	Value
Field(From/Source)	exthd
File(To/Target)	HPSAPSolutionManager
Format(To/Target)	HPSAPSolutionManager
Field(To/Target)	id
Source Field (Fill To/Post From)	exthd
Target Field(Fill From/Post To)	id

Trigger Buttons

The Incident Exchange web service is triggered by an HTTP request from ServiceCenter. 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**, an so on. An implementation must follow the Incident Exchange state diagram. Display options that enable / 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 any updates with the external helpdesk (and can thus be invoked multiple times without updating the Incident in ServiceCenter.) 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 **Menu** → **Utilities** → **Tools** → **Display Options**.

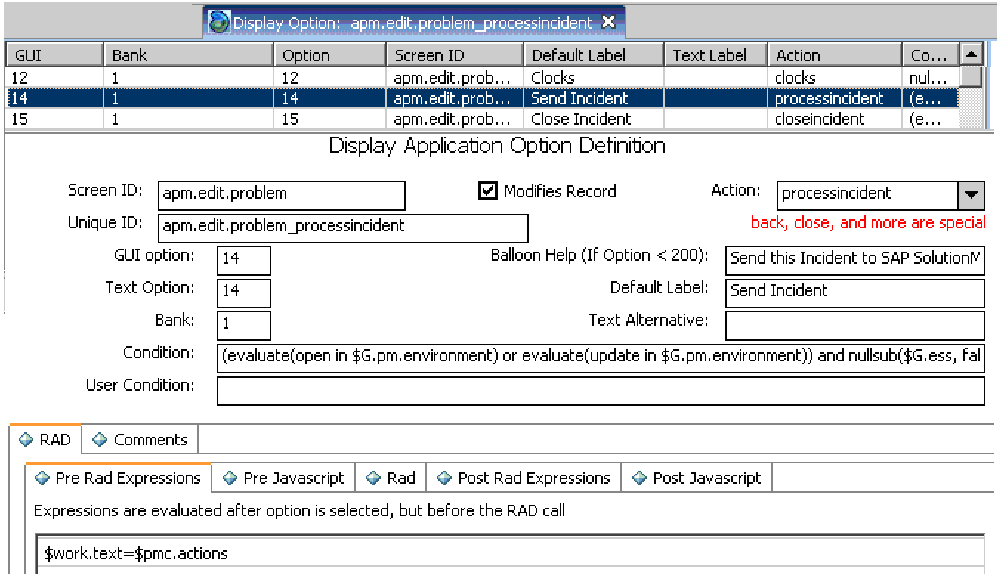
2 Search for `apm.edit.problem` in “Screen ID” field.

3 Add Display Options for all Incident Exchange actions:

- `processincident`
- `closeincident`
- `verifyincidentsolution`
- `rejectincidentsolution`
- `addinfo`
- `acceptincidentprocessing`

4 Provide a Condition that evaluates the hidden metadata in order to disable the button when the action is currently invalid. The demo unload provides a complete implementation of the Incident Exchange status model. This is an example condition for action `processincident`:

```
(evaluate(open in $G.pm.environment) or evaluate(update in
$G.pm.environment)) and nullsub($G.ess, false)=false and not
nullsub(is.ictex.action.blocked in $L.filed, false) and (hidden.meta.data
in $L.filed=NULL or hidden.meta.data in
$L.filed="Requester:StartExchange:Error")
```



GUI	Bank	Option	Screen ID	Default Label	Text Label	Action	Co...
12	1	12	apm.edit.prob...	Clocks		clocks	nul...
14	1	14	apm.edit.prob...	Send Incident		processincident	(e...
15	1	15	apm.edit.prob...	Close Incident		closeincident	(e...

Display Application Option Definition

Screen ID: apm.edit.problem

☑ Modifies Record

Action: processincident

back, close, and more are special

Unique ID: apm.edit.problem_processincident

GUI option: 14

Balloon Help (If Option < 200): Send this Incident to SAP SolutionM

Text Option: 14

Default Label: Send Incident

Bank: 1

Text Alternative:

Condition: (evaluate(open in $G.pm.environment) or evaluate(update in $G.pm.environment)) and nullsub($G.ess, fal

User Condition:

Expressions are evaluated after option is selected, but before the RAD call

```
$work.text=$pmc.actions
```

The following table list the display options to add.

Table 14 Trigger button display options

Unique ID	Property	Value
apm.edit.problem_processincident	Screen ID	apm.edit.problem
	GUI option	14
	Text Option	14
	Action	processincident
	Balloon Help(If Option < 200)	Send this Incident to SAP SolutionManager.
	Default Label	Send Incident
	Condition (**[SENDINCIDENT_COND]** in `code_sc6.txt`)	(evaluate(open in $G.pm.environment) or evaluate(update in $G.pm.environment)) and nullsub($G.ess, false)=false and not nullsub(is.ictex.action.blocked in $L.filed, false) and (hidden.meta.data in $L.filed=NULL or hidden.meta.data in $L.filed="Requester:StartExchange:Error")
	RAD > Pre Rad Expressions	$work.text=$pmc.actions
	Bank	1
	Modifies Record	Check
apm.edit.problem_closeincident	Screen ID	apm.edit.problem
	GUI option	15
	Text Option	15
	Action	closeincident
	Balloon Help(If Option < 200)	Close this Incident in SAP SolutionManager
	Default Label	Close in SAP
	Condition (**[CLOSEINCIDENT_COND]** in `code_sc6.txt`)	(evaluate(open in $G.pm.environment) or evaluate(update in $G.pm.environment)) and nullsub($G.ess, false)=false and not nullsub(is.ictex.action.blocked in $L.filed, false) and (hidden.meta.data in $L.filed)#"Requester" and hidden.meta.data in $L.filed~="Requester:StartExchange:Error"
	RAD > Pre Rad Expressions	$work.text=$pmc.actions
	Bank	1
	Modifies Record	Check

Table 14 **Trigger button display options (cont'd)**

Unique ID	Property	Value
apm.edit.problem_verifyincidentsolution	Screen ID	apm.edit.problem
	GUI option	16
	Text Option	16
	Action	verifyincidentsolution
	Balloon Help(If Option < 200)	Propose a solution for this Incident to SAP SolutionManager.
	Default Label	Send Solution
	Condition (**[SENDSOLUTION_COND]** in `code_sc6.txt`)	(evaluate(open in $G.pm.environment) or evaluate(update in $G.pm.environment)) and nullsub($G.ess, false)=false and not nullsub(is.ictex.action.blocked in $L.filed, false) and (hidden.meta.data in $L.filed="Provider:ProviderProcessing" or hidden.meta.data in $L.filed="Provider:ProviderProcessing:Error")
	RAD > Pre Rad Expressions	$work.text=$pmc.actions
	Bank	1
	Modifies Record	Check
apm.edit.problem_rejectincidentsolution	Screen ID	apm.edit.problem
	GUI option	17
	Text Option	17
	Action	rejectincidentsolution
	Balloon Help(If Option < 200)	Refuse the solution for this Incident as proposed by SAP SolutionManager.
	Default Label	Refuse Solution
	Condition (**[REJECTSOLUTION_COND]** in `code_sc6.txt`)	(evaluate(open in $G.pm.environment) or evaluate(update in $G.pm.environment)) and nullsub($G.ess, false)=false and not nullsub(is.ictex.action.blocked in $L.filed, false) and (hidden.meta.data in $L.filed="Requester:SolutionProvided" or hidden.meta.data in $L.filed="Requester:SolutionProvided:Error")
	RAD > Pre Rad Expressions	$work.text=$pmc.actions
	Bank	1
	Modifies Record	Check

Table 14 **Trigger button display options (cont’d)**

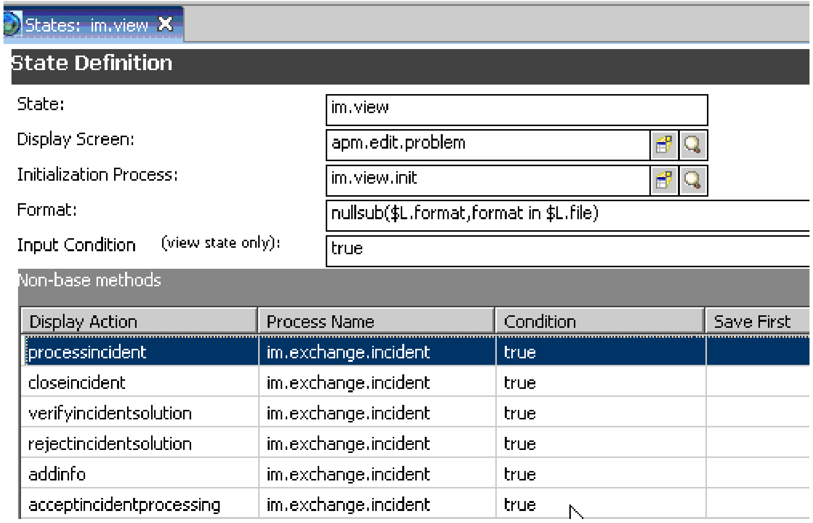
Unique ID	Property	Value
apm.edit.problem_addinfo	Screen ID	apm.edit.problem
	GUI option	18
	Text Option	18
	Action	addinfo
	Balloon Help(If Option < 200)	Add information to this Incident in SAP SolutionManager.
	Default Label	Add Info
	Condition ([ADDINFO_COND] in `code_sc6.txt`)	(evaluate(open in $G.pm.environment) or evaluate(update in $G.pm.environment)) and nullsub($G.ess, false)=false and not nullsub(is.ictex.action.blocked in $L.filed, false) and hidden.meta.data in $L.filed~=NULL and hidden.meta.data in $L.filed~="Requester:StartExchange:Error" and hidden.meta.data in $L.filed~="Closed"
	RAD > Pre Rad Expressions	$work.text=$pmc.actions
	Bank	1
	Modifies Record	Check

Table 14 Trigger button display options (cont'd)

Unique ID	Property	Value
apm.edit.problem_acceptincidentprocessing	Screen ID	apm.edit.problem
	GUI option	19
	Text Option	19
	Action	acceptincidentprocessing
	Balloon Help(If Option < 200)	Send this Incident back to SAP SolutionManager.
	Default Label	Send back
	Condition (**[SENDBACK_COND]** in `code_sc6.txt`)	(evaluate(open in $G.pm.environment) or evaluate(update in $G.pm.environment)) and nullsub($G.ess, false)=false and not nullsub(is.ictex.action.blocked in $L.filed, false) and (hidden.meta.data in $L.filed="Provider:ProviderProcessing" or hidden.meta.data in $L.filed="Provider:ProviderProcessing:Error" or hidden.meta.data in $L.filed="Requester:RequesterProcessing" or hidden.meta.data in $L.filed="Requester:RequesterProcessing:Error")
	RAD > Pre Rad Expressions	$work.text=$pmc.actions
	Bank	1
	Modifies Record	Check

5 Select **Utilities** → **Tools** → **Document Engine** → **States** and search for **`im.view`**.

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



State Definition

State: im.view

Display Screen: apm.edit.problem

Initialization Process: im.view.init

Format: nullsub($L.format,format in $L.file)

Input Condition (view state only): true

Non-base methods

Display Action	Process Name	Condition	Save First
processincident	im.exchange.incident	true	
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	

Display Action	Process Name	Condition
processincident	im.exchange.incident	true
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

Selection of External Helpdesk System

If ServiceCenter 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 ServiceCenter, then the value can be hardcoded. In any case, the exthd field must be set before the Incident is exchanged.

SAP Configuration Item Handling

This section describes how to handle SAP configuration items.

Overview

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

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

The 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 ServiceCenter, SAP CIs 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 ServiceCenter customization implements the bi-directional synchronization between the Incident fields and the ServiceCenter 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 **Services** → **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 15 New SAPInstance table fields

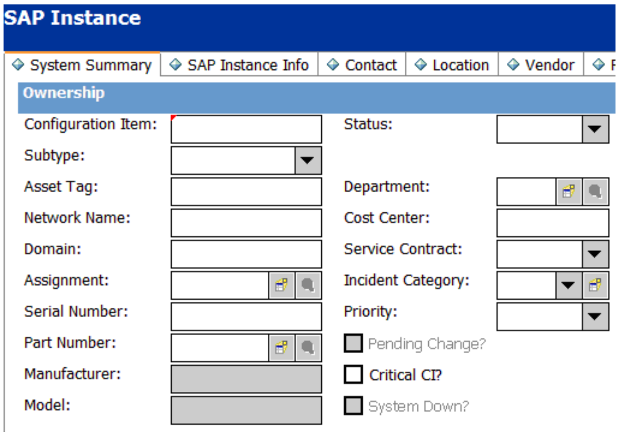
Field name	Type	Caption	Other Properties	
SID	Text	SID	Not null	Unique
client	Text	client	Not null	
installation.number	Text	installation.number	Not null	
logical.name	Text	logical.name		Unique

To implement the configuration described above, do the following:

1 Create a new form with the name `device.sapinstance.g` for SAP Solution Manager device type, you can copy an existing form of device. For example,

Go to **Toolkit** → **Form Designer**, enter **`device.template.g`** 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`. Delete tabs Example Info, Financial, and Scanner.

Then add a new tab **SAP Instance Info** in the form:.



SAP Instance

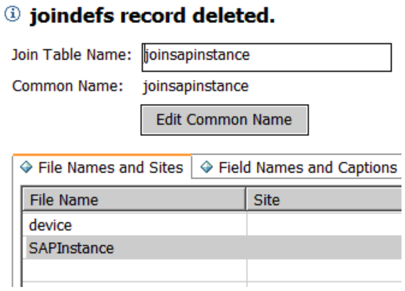
Ownership

Configuration Item:		Status:	
Subtype:			
Asset Tag:		Department:	
Network Name:		Cost Center:	
Domain:		Service Contract:	
Assignment:		Incident Category:	
Serial Number:		Priority:	
Part Number:		□ Pending Change?	
Manufacturer:		□ Critical CI?	
Model:		□ System Down?	

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

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



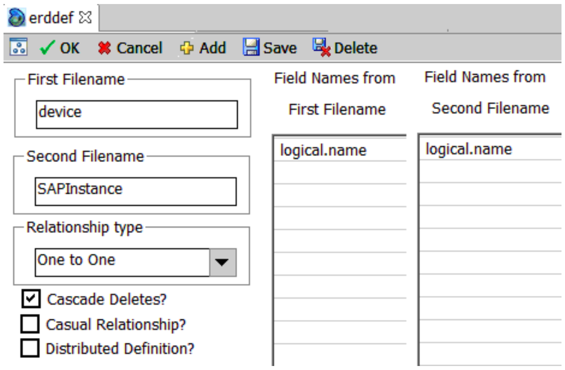
joindefs record deleted.

Join Table Name: joinsapinstance

Common Name: joinsapinstance

File Name	Site
device	
SAPInstance	

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



First Filename: device

Second Filename: SAPInstance

Relationship type: One to One

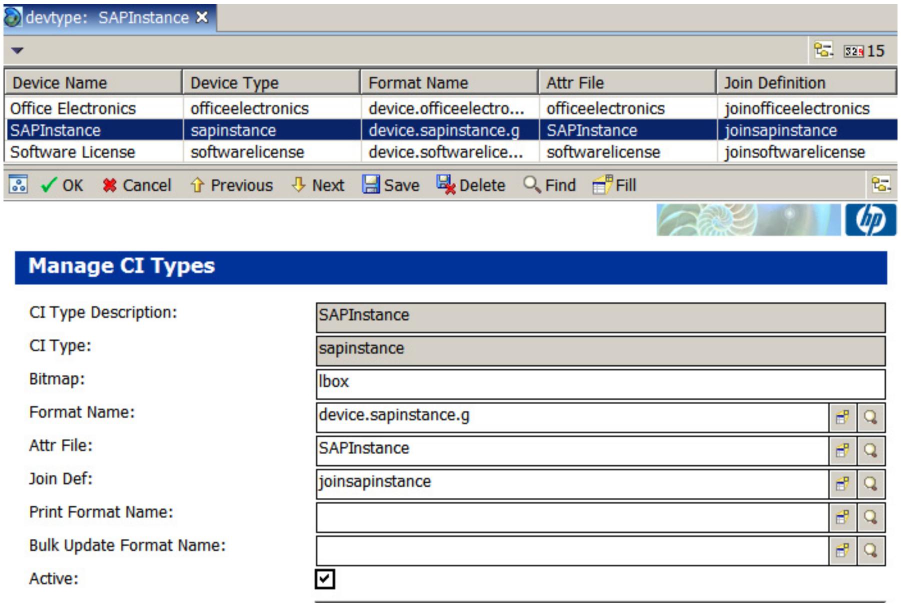
☑ Cascade Deletes?

☐ Casual Relationship?

☐ Distributed Definition?

Field Names from First Filename	Field Names from Second Filename
logical.name	logical.name

4 Generate a new device type in **Services** → **Configuration Management** → **Resources** → **Device Types**.

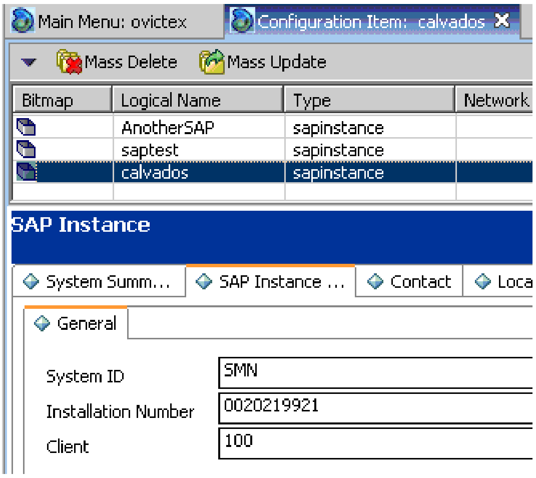


Device Name	Device Type	Format Name	Attr File	Join Definition
Office Electronics	officeelectronics	device.officeelectro...	officeelectronics	joinofficeelectronics
SAPInstance	sapinstance	device.sapinstance.g	SAPInstance	joinsapinstance
Software License	softwarelicense	device.softwarelice...	softwarelicense	joinsoftwarelicense

Manage CI Types

CI Type Description:	SAPInstance
CI Type:	sapinstance
Bitmap:	lbox
Format Name:	device.sapinstance.g
Attr File:	SAPInstance
Join Def:	joinsapinstance
Print Format Name:	
Bulk Update Format Name:	
Active:	☑

5 Create a custom form for entry of the SAP-specific CI attributes (the basic three plus any others) at **Services** → **Configuration Management** → **Resources** → **CI Queue** → **New**. Select SAPInstance in the Type dropdown list.

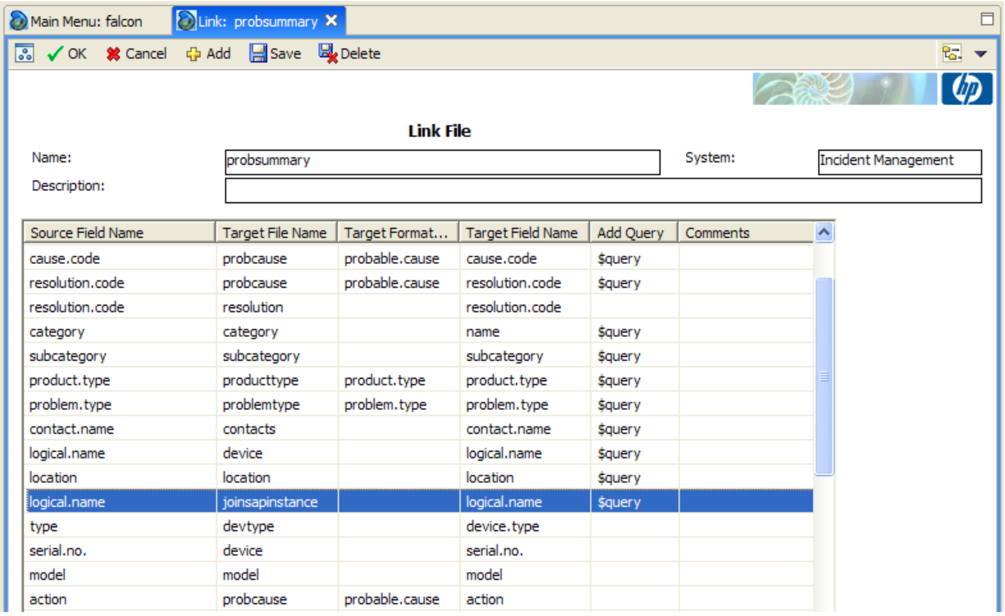


Bitmap	Logical Name	Type	Network
	AnotherSAP	sapinstance	
	saptest	sapinstance	
	calvados	sapinstance	

System ID	SMN
Installation Number	0020219921
Client	100

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

7 The existing CI lookup (links between `logical.name` and `device`) must be augmented. Insert an additional link line that links `logical.name` with `joinsapinstance`.



Link File

Name:	probsummary	System:	Incident Management
Description:			

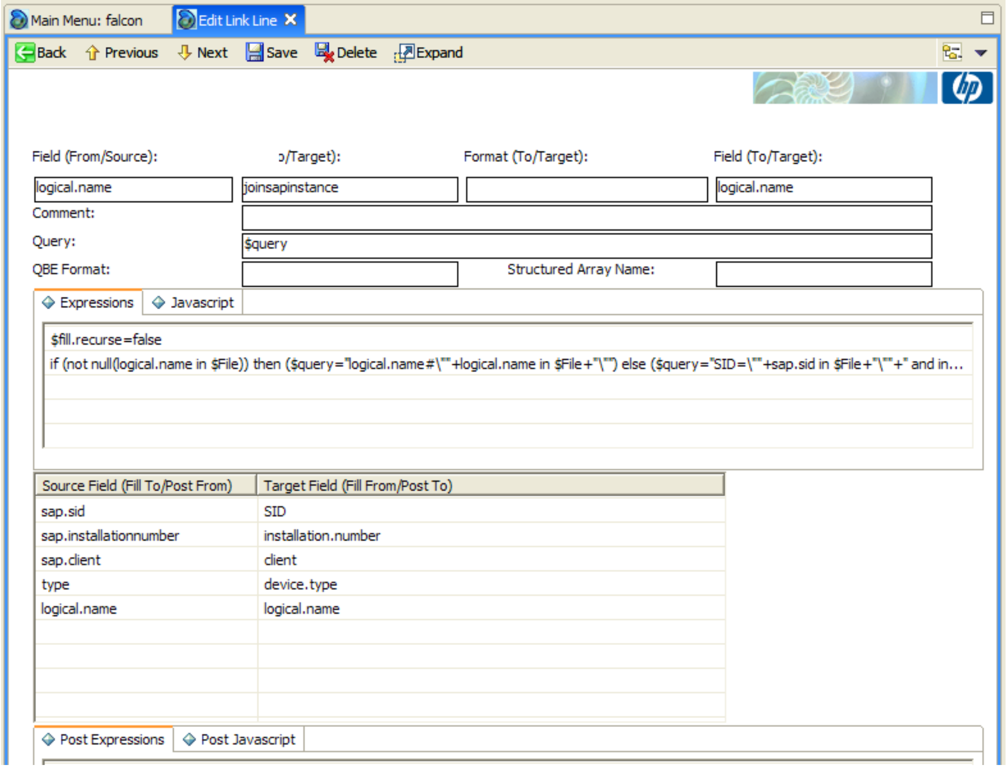
Source Field Name	Target File Name	Target Format...	Target Field Name	Add Query	Comments
cause.code	probcause	probable.cause	cause.code	$query	
resolution.code	probcause	probable.cause	resolution.code	$query	
resolution.code	resolution		resolution.code		
category	category		name	$query	
subcategory	subcategory		subcategory	$query	
product.type	producttype	product.type	product.type	$query	
problem.type	problemtype	problem.type	problem.type	$query	
contact.name	contacts		contact.name	$query	
logical.name	device		logical.name	$query	
location	location		location	$query	
logical.name	joinsapinstance		logical.name	$query	
type	devtype		device.type		
serial.no.	device		serial.no.		
model	model		model		
action	probcause	probable.cause	action		

8 The three attributes (that define the SAP CI) and the CI primary key must be linked. A query expression locates the correct CI based on the attributes in `probsummary`.

In Expressions tab, add the following two lines (**[LOGICAL.NAME_JOINSAPINSTANCE_EXPR]** in `code_sc6.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+"\"")
```

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



Field (From/Source): logical.name

ɔ/Target): joinsapinstance

Format (To/Target):

Field (To/Target): logical.name

Comment:

Query: $query

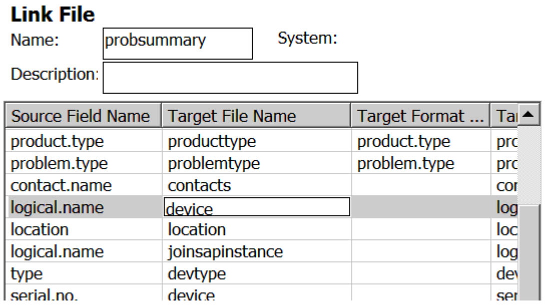
QBE Format:

Structured Array Name:

```
$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 in...
```

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

9 Select the line with `device` as Target File Name.



Link File

Name: probsummary System:

Description:

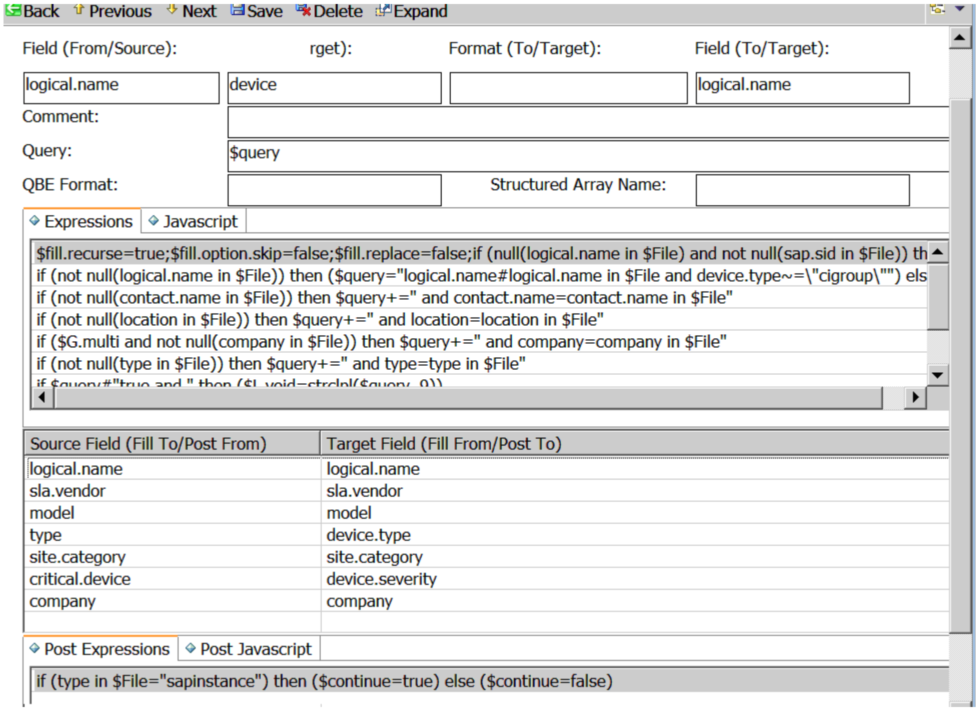
Source Field Name	Target File Name	Target Format ...	Ta
product.type	producttype	product.type	prc
problem.type	problemtype	problem.type	prc
contact.name	contacts		cor
logical.name	device		log
location	location		loc
logical.name	joinsapinstance		log
type	devtype		de\
serial.no.	device		ser

10 In the Expressions tab, replace the first line with following expression (**[LOGICAL.NAME_DEVICE_EXPR]** in `code_sc6.txt`):

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

11 In the Post Expressions tab, add the following expression (**[LOGICAL.NAME_DEVICE_POST]** in `code_sc6.txt`):

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



Field (From/Source):	rget):	Format (To/Target):	Field (To/Target):
logical.name	device		logical.name

Comment:

Query: $query

QBE Format:

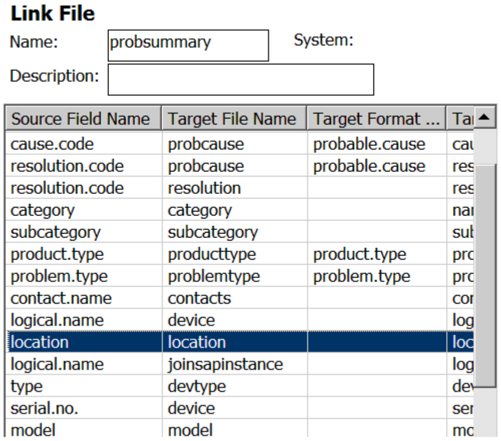
Structured Array Name:

```
$fill.recurse=true;$fill.option.skip=false;$fill.replace=false;if (null(logical.name in $File) and not null(sap.sid in $File)) th
if (not null(logical.name in $File)) then ($query="logical.name#logical.name in $File and device.type~=\"cigroup\"") els
if (not null(contact.name in $File)) then $query+=" and contact.name=contact.name in $File"
if (not null(location in $File)) then $query+=" and location=location in $File"
if ($G.multi and not null(company in $File)) then $query+=" and company=company in $File"
if (not null(type in $File)) then $query+=" and type=type in $File"
if $query#"true and " then ($L.void=strclpl($query, 0))
```

Source Field (Fill To/Post From)	Target Field (Fill From/Post To)
logical.name	logical.name
sla.vendor	sla.vendor
model	model
type	device.type
site.category	site.category
critical.device	device.severity
company	company

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

12 Select the line with `location` as Target File Name.



Link File

Name: probsummary

System:

Description:

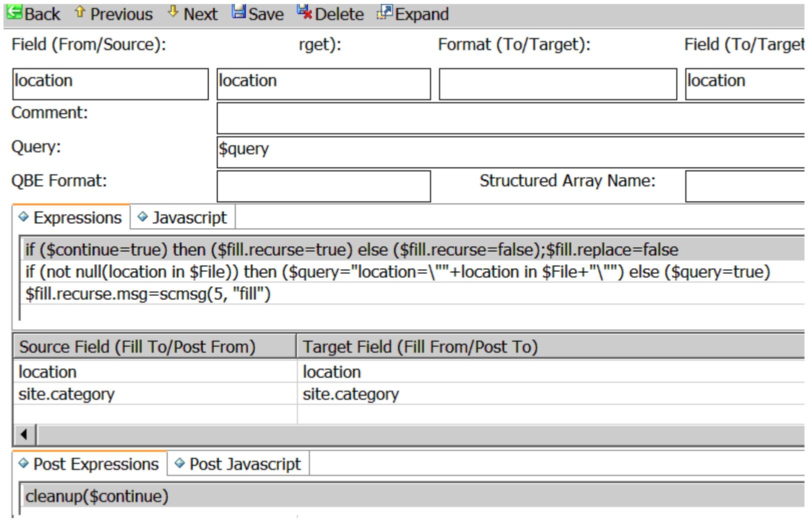
Source Field Name	Target File Name	Target Format ...	Ta
cause.code	probcause	probable.cause	cau
resolution.code	probcause	probable.cause	res
resolution.code	resolution		res
category	category		nai
subcategory	subcategory		sul
product.type	producttype	product.type	prc
problem.type	problemtype	problem.type	prc
contact.name	contacts		cor
logical.name	device		log
location	location		loc
logical.name	joinsapinstance		log
type	devtype		de\
serial.no.	device		ser
model	model		mc

13 In the Expressions tab, replace the first line with following expression (**[LOCATION_LOCATION_EXPR]** in `code_sc6.txt`):

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

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

```
cleanup($continue)
```



Field (From/Source): location
rget): location
Format (To/Target):
Field (To/Target location

Comment:

Query: $query

QBE Format:

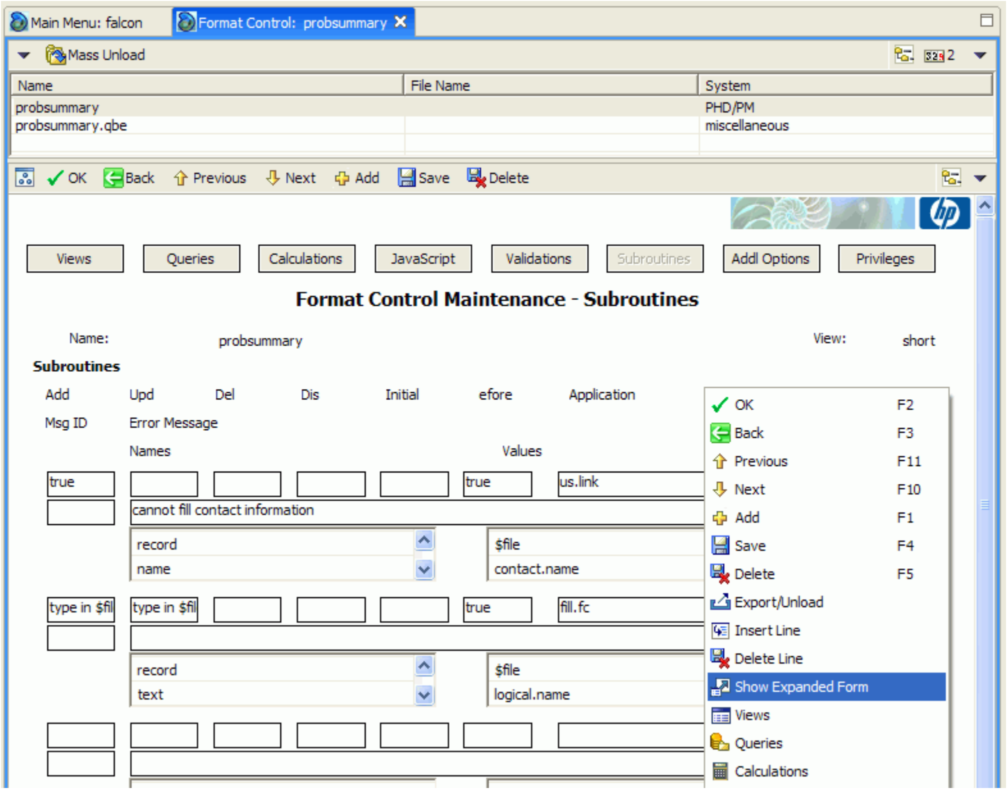
Structured Array Name:

```
if ($continue=true) then ($fill.recurse=true) else ($fill.recurse=false);$fill.replace=false
if (not null(location in $File)) then ($query="location=\""+location in $File+"\"") else ($query=true)
$fill.recurse.msg=scmsg(5, "fill")
```

Source Field (Fill To/Post From)	Target Field (Fill From/Post To)
location	location
site.category	site.category

```
cleanup($continue)
```

15 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 **Utilities** → **Tools** → **Format Control** and search for **probsummary** → **Subroutines** → **Show Expanded Form**.



Name	File Name	System
probsummary		PHD/PM
probsummary.qbe		miscellaneous

Format Control Maintenance - Subroutines

Name: probsummary

View: short

Subroutines

Add Upd Del Dis Initial efore Application

Msg ID Error Message

Names Values

true | | | | | true | us.link

cannot fill contact information

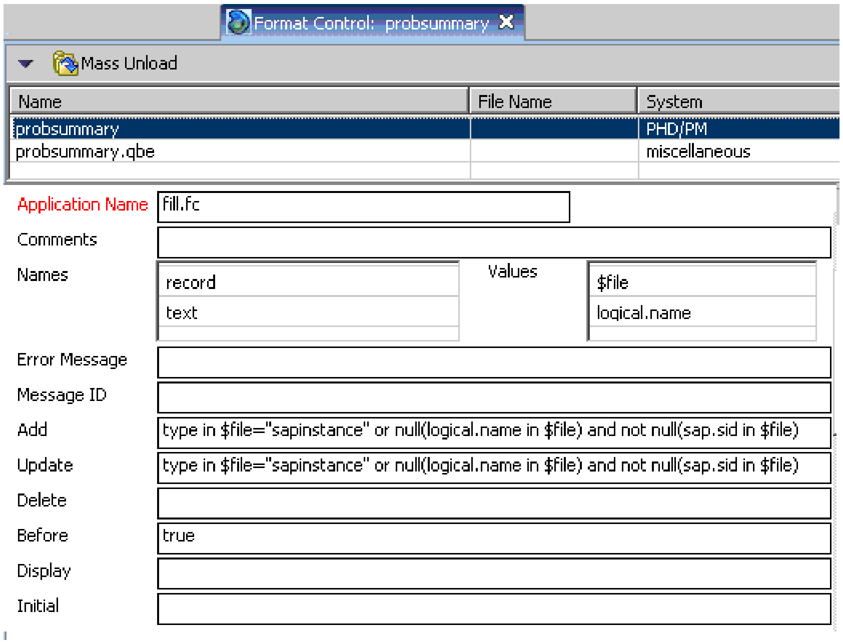
record | $file
name | contact.name

type in $fil | type in $fil | | | | true | fill.fc

record | $file
text | logical.name

16 Add a `fill.fc` application.

Field	Value
Application Name	fill.fc
Name	record
Value	$file
Name	text
Value	logical.name
Add	type in $file="sapinstance" or null(logical.name in $file) and not null(sap.sid in $file)
Update	type in $file="sapinstance" or null(logical.name in $file) and not null(sap.sid in $file)
Before	true



Name	File Name	System
probsummary		PHD/PM
probsummary.qbe		miscellaneous

Application Name	fill.fc
Comments	
Names	record text
Values	$file logical.name
Error Message	
Message ID	
Add	type in $file="sapinstance" or null(logical.name in $file) and not null(sap.sid in $file)
Update	type in $file="sapinstance" or null(logical.name in $file) and not null(sap.sid in $file)
Delete	
Before	true
Display	
Initial	

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 ServiceCenter 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

Above implementation assumes a simple CI setup. SAP CIs may be modeled in more complex ways.

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

The customization within ServiceCenter needs to be adapted to any SAP CI configuration. Incident Exchange directly interacts only with the Incident fields in the `probsummary` table. The synchronization with CIs inside ServiceCenter is the responsibility of the ServiceCenter customization.

HPSAPSolutionManager

An alternative to creating SAP Configuration Item handling is to create an SAP Solution Manager.

1. Go to **System Definition** → **Tables**, right-click **Tables** and select **New Table**.
2. Enter **`HPSAPSolutionManager`** as the table name.
3. Click **OK**.
4. In **System Definition** → **Tables** → **HPSAPSolutionManager** → **Fields**, click **New field**.
5. Enter **`name`** as Field name.
6. Click **OK**.
7. Select **Text** as Type.
8. Enter **`name`** as Caption.
9. Click **Save**.
10. In **Toolkit** → **Forms Designer** click **New**.
11. Use Form Wizard.
12. Enter **`HPSAPSolutionManager`** as the form name.
13. Select **`HPSAPSolutionManager`** as the table name.
14. Select **Detail of a Single Record**.
15. Set the Show column to **`true`** for fields `id` and `name`.
16. Set the Show column to **`false`** for all other fields.
17. Click **Proceed**.
18. Click **OK**.
19. Select **Database Manager** from the drop-down list of the top right triangle.
20. Click **Yes**.
21. Enter **`exthd`** as id (should be consistent with the value of `exthd.instances.id.<number>`).
22. Enter **`SAP SolMan G11`** (can be any value) as the Name.
23. Click **Add** button to add the record.

D 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>`

E Configuring HP ServiceCenter for SSL

This chapter describes how to configur HP ServiceCenter for SSL.

Setting Up ServiceCenter for SSL Communication with SMSSMEX

This section describes how to configure HP ServiceCenter for SSL.

- 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, another differentiating entry, such as the organization, must be made.
- When prompted, always use the fully qualified name (**`computer.domain.com`**) as the first/last name.

To configure HP ServiceCenter for SSL do the following:

1 Add `<OpenSSL installation path>/bin` in the PATH environment variable.

2 Verify that `openssl.exe` can be executed in a command line.

3 Set the current working path. Example path is

```
C:\my docs\smc\sc\ssl\sc6 certs
```

4 Put the `openssl.conf` file in the working path.

5 Generate a private/public key pair.

```
openssl genrsa -des3 -out <keystorefile> 2048
```

For example:

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

6 Export the public key as the self-signed root CA certificate (enter the fully qualified name of the machine, such as **`SCN4727.asiapacific.cpqcorp.net`**, case-sensitive).

```
openssl req -new -key <keystorefile> -x509 -days <days> -out
<certificatefile> -config <configfile>
```

For example:

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

7 Import the self-signed root CA's certificate into a keystore file.

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

For example:

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

8 Generate the ServiceCenter server keystore: Classic mode.

a Generate the private/public key pair and certificate request file.

```
openssl req -newkey rsa:2048 -keyout <keystorefile> -out <requestfile>
-config <configfile>
```

For example:

```
openssl req -newkey rsa:2048 -keyout scserverkey.pem -out
scservercert_request.pem -config ./openssl.conf
```

b 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 scservercert_request.pem -CA
mycacert.pem -CAkey cakey.pem -CAcreateserial -out
scservercert_classic.pem
```

9 Generate the ServiceCenter server keystore: Servlet mode.

a Generate the private/public key pair.

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

For example:

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

b Generate the request file.

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

For example:

```
keytool -certreq -alias scserver -keystore server.keystore -file
servercert_req.crs
```

c 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 servercert_req.crs -CA mycacert.pem
-CAkey cakey.pem -CAcreateserial -out scservercert_servlet.pem
```

d Import the root CA’s certificate into the server keystore.

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

For example:

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

e Import the signed certificate into the keystore.

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

For example:

```
keytool -import -trustcacerts -alias scserver -keystore server.keystore
-file scservercert_servlet.pem
```

10 Generate the client keystore.

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

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

For example:

```
keytool -genkey -alias scclient -keystore clientcerts
```

b Generate the request file.

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

For example:

```
keytool -certreq -alias scclient -keystore clientcerts -file
scclientcert_request.crs
```

c 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_request.crs -CA
mycacert.pem -CAkey cakey.pem -CAcreateserial -out scclient_cert.pem
```

d Import the root CA's certificate into the client keystore.

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

For example:

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

e 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 ./clientcerts
-file scclient_cert.pem
```

➤ Use the same steps to generate the client keystore for SMSSMEX. When asked for the first and last name, enter the fully qualified name of the incident exchange middleware host.

11 Generate the trust-list keystore for Service Center server (must be a `trusted.pem` file): Classic mode.

a If this `trusted.pem` file does not exist in the ServiceCenter RUN directory, then create the file by copying the client certificate file (`scclient_cert.pem`) to `trusted.pem`.

b If the file does already exist and you are adding new certificates, open the client certificate and copy and paste the contents to the bottom of the original trusted certificates `.pem` file (which must end with a carriage return and new line). For example:

```
-----BEGIN CERTIFICATE-----
MIIEDTCCAvUCCQCNvarTXFgHhjANBgkqhkiG9w0BAQQFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAkNBMRIwEAYDVQQHEwlTYW4gRGllZ28xGjAYBgNVBAoTEVBl
[...]
K06z24M9KPblX/9dN+5CtNAcODsPwpfKLbWOjLzGvSsPK2SFzQKGYREb5ULl5TKz
lQ5Rlfn17nhe9Ifwyn0EuPpe0GZbL5cgRqC7v6siH4Mo
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEDDCCAvQCCQDwJNn8mBramjANBgkqhkiG9w0BAQQFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAkNBMRIwEAYDVQQHEwlTYW4gRGllZ28xGjAYBgNVBAoTEVBl
[...]
LeSAtTshwhz5ZQ0OniRz/ZAW5kRBwc+OGwFasxCES2hogQA32NiGFVTISISzK/eq
HTXj+2FvCR2jCOXiorHeROX2+sgj1ScGVyeTRk5LZck=
-----END CERTIFICATE-----
```

12 Generate the trust-list keystore for Service Center server: Servlet mode.

a Export the certificate file.

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

For example:

```
keytool -export -alias scclient -keystore clientcerts -file
clientpubkey.crt
```

b Import the certificate into a jks file.

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

For example:

```
keytool -import -alias scclient -file clientpubkey.crt -keystore
trustedclients.jks
```

13 SSL Configuration in Service Center server (after modification of the `sc.ini` file, Service Center server should restart).

a Copy the generated files `mycacert.pem`, `scservercert_classic.pem`, `scserverkey.pem`, `trusted.pem`, `server.keystore`, `trustedclients.jks`, `cacerts` files into `<SC installation path}\Server\RUN>`.

b Add the following entries to `sc.ini`:

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

# -- classic mode --
cacertpem:mycacert.pem
```

```
certpem:scservercert_classic.pem
pkpem:scserverkey.pem
pkpempass:password
ssl_trustedClientspem:trusted.pem

# -- servlet mode --
keystoreFile:server.keystore
keystorePass:password
ssl_trustedClientsJKS:trustedclients.jks
ssl_trustedClientsPwd:password
truststoreFile:cacerts
truststorePass:password
```

c In the ServiceCenter Client application in **Toolkit** → **Script Library**, search for `HPSAPTrigger`.

d Change the following javascript code:

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

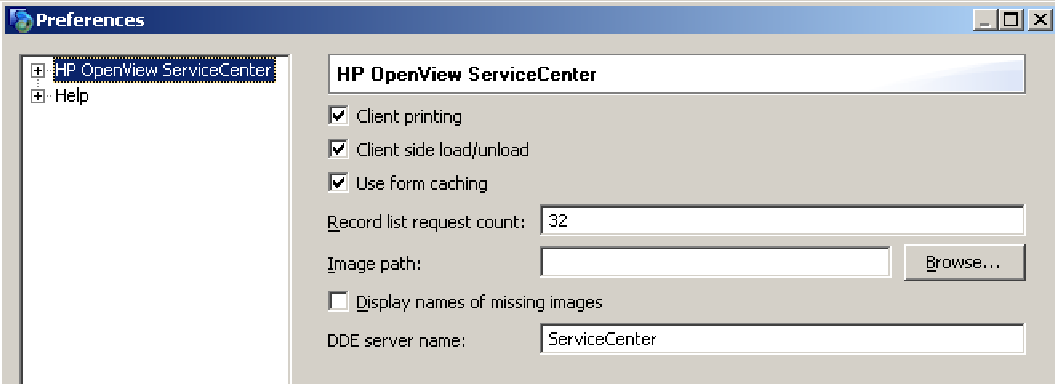
to

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

14 SSL Configuration in Service Center Client.

a Open the ServiceCenter Client.

b From the menu select **Window** → **Preference** to open the Preferences configuration window.



HP OpenView ServiceCenter

☑ Client printing

☑ Client side load/unload

☑ Use form caching

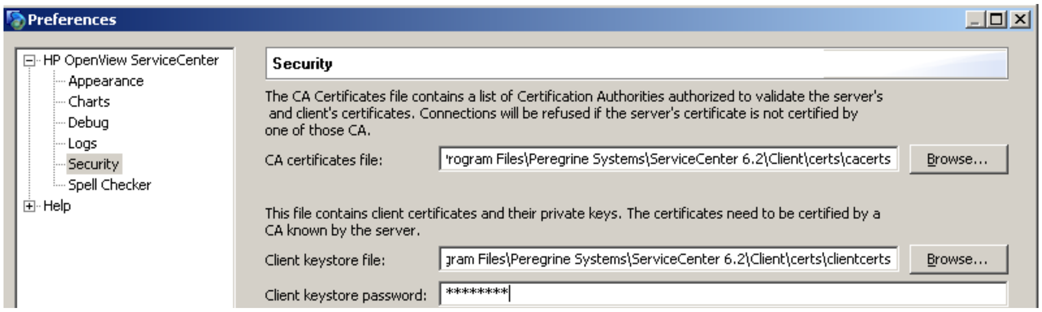
Record list request count: 32

Image path:

☐ Display names of missing images

DDE server name: ServiceCenter

c Expand `HP OpenView ServiceCenter` node in the left menu tree, and select Security to open the client Security configuration window.



Security

The CA Certificates file contains a list of Certification Authorities authorized to validate the server's and client's certificates. Connections will be refused if the server's certificate is not certified by one of those CA.

CA certificates file: 'rogram Files\Peregrine Systems\ServiceCenter 6.2\Client\certs\cacerts

This file contains client certificates and their private keys. The certificates need to be certified by a CA known by the server.

Client keystore file: ȝram Files\Peregrine Systems\ServiceCenter 6.2\Client\certs\clientcerts

Client keystore password: ********

d Click **Browse...** to set the CA certificates file and Client keystore file.

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

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

g Restart ServiceCenter Client to make the changes take effect.

h In the Connections dialog, the value of field **`Server host name`** must be the fully qualified name of the ServiceCenter server.

i In the Advanced tab, make sure **`Use SSL Encryption`** is checked.

15 ServiceCenter Web clients SSL configuration:

a Copy the servlet-mode trust keystore and client keystore files to the `WEB-INF` folder of the ServiceCenter Web Application Server.

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

c Modify the following configuration entry

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

For example:

```
<init-param>
  <param-name>sc.host</param-name>
  <param-value>SMCI02.chn.hp.com</param-value>
</init-param>
<init-param>
  <param-name>sc.port</param-name>
  <param-value>serverPort</param-value>
</init-param>
```

For example:

```
<init-param>
  <param-name>sc.port</param-name>
  <param-value>13080</param-value>
</init-param>
<init-param>
  <param-name>sc.ssl</param-name>
  <param-value>true</param-value>
</init-param>
<init-param>
  <param-name>sc.cacerts</param-name>
  <param-value>trustKeystore</param-value>
</init-param>
```

For example:

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

For example:

```
<init-param>
  <param-name>sc.keystore</param-name>
  <param-value>/WEB-INF/clientcerts</param-value>
</init-param>
<init-param>
  <param-name>sc.keystorepassword</param-name>
  <param-value>clientKeystorePassword</param-value>
</init-param>
```

For example:

```
<init-param>
  <param-name>sc.keystorepassword</param-name>
  <param-value>scclient</param-value>
</init-param>
```

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

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

to

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

Setting Up SMSSMEX for SSL Communication with ServiceCenter

To setup SMSSMEX for SSL communication with ServiceCenter, do the following.

1 Import the root CA into the SMSSMEX trust keystore.

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

For example:

```
keytool -import -trustcacerts -alias scca -keystore C:/Program Files/
HP/SMSSMEX/certs/ovictex.truststore -file mycacert.pem
```

2 Configure `ovictex.properties` (`keystore.password` and `truststore.password` should use `encryptPassword.bat` in <*SMSSMEX_installDir*>`/bin/`).

a Modify `sc.webservice.endpoint`.

```
sc.webservice.endpoint = https://schostname:port/sc62server/ws
```

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

```
schostname.keystore=<scclient certifiction file>
schostname.keystore.password=<keystore password>
schostname.truststore=<truststore file>
schostname.truststore.password=<truststore password>
```

For example:

```
sc.webservice.endpoint = https://SMCI02.chn.hp.com:13081/sc62server/ws
……
SMCI02.chn.hp.com.keystore= C:/Program Files/HP/SMSSMEX/certs/
clientcerts
```

```
SMCI02.chn.hp.com.keystore.password=~X1~H+7JAOrcX/R6kO5diPxV0w==
SMCI02.chn.hp.com.truststore= C:/Program Files/HP/SMSSMEX/certs/
ovictex.truststore
SMCI02.chn.hp.com.truststore.password=~X1~H+7JAOrcX/R6kO5diPxV0w==
```

F 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 7.0x / 7.10 / 7.11

  Copy button icons from `<SMSSMEX1.10 Release Package>\icons` folder to `<Client_Home>\plugins\com.hp.ov.sm.client.eclipse.user_7.xx\src\resources\icons\obj16`

  For example,

  ```
  C:\Program Files\HP\Service Manager
  7.01\Client\plugins\com.hp.ov.sm.client.eclipse.user_7.01\src\resourc
  es\icons\obj16
  ```

  For more information, see page 104 of *Service Manager 7.00 Installation Guide* and page 102 of *Service Manager 7.10 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 7.0x, `<WebApps_Root>\webtier-7.0x\images\obj16` directory.
- On Service Manager 7.10 / 7.11, `<WebApps_Root>\webtier-7.10\images\obj16` directory.

G 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.