# HP Service Oriented Architecture Policy Enforcer

## User Guide

**Version: 3.00**

**Windows®, HP-UX, Linux**


**June 2008**

# 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



## Trademark Notices

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

UNIX® is a registered trademark of The Open Group

To view open source code, see the Installation\license\thirdparty\source directory on the product installation media.

## Support

You can visit the HP Software support web site at:

**www.hp.com/go/hpsoftwaresupport**

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

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

**www.hp.com/managementsoftware/access_level**

To register for an HP Passport ID, go to:

**www.managementsoftware.hp.com/passport-registration.html**

# Contents

## Table of Contents

1

# HP SOA Policy Enforcer Overview

The HP SOA Policy Enforcer (SOA PE) is a part of the HP SOA Governance and SOA Management solution. SOA PE along with HP SOA Systinet products such as SOA Systinet Registry and Repository and HP Diagnostics help to implement end-to-end SOA governance. This includes design-time governance requirements such as service contract management, lifecycle management, and policy management. In addition, this solution also includes run-time governance requirements such as SOA run-time policy definitions, provisioning of Web-services, and distribution and enforcement of SOA run-time policies. SOA PE bundles the Diagnostics probe and helps capture SOA Management metrics during run time.

## SOA Governance

SOA Governance helps organizations gain visibility and control over the creation, deployment, and usage of SOA services.

Design-time governance tools such as HP SOA Systinet Registry and Repository provide a central system-of-record for all services and related information in an SOA environment. This is the common location where services are made available by providers and discovered by consumers.

Runtime Governance tools such as SOA PE offer the ability to easily configure and enforce runtime governance policies in an SOA environment. SOA PE also provides the required visibility into policy enforcement such as what policies are enforced for a service and so on. These policies may be related to various runtime aspects such as security, auditing, message format validations, transformations, and more and they may be enforced at various *enforcement points* within the SOA environment.

# SOA PE Concepts

It is essential to model the significant elements that participate in an SOA environment to be able to manage and govern these elements effectively. The models help abstract certain common properties of these elements that are relevant to SOA management and governance. This enables you to enforce policies across these elements and to manage them in a simple and effective manner. The following are the two key abstractions modeled in the SOA Policy Enforcer:

- *Policy Enforcement Resources*: Policy Enforcement Resources enforce policies to manage and govern the usage of services.
- *Services*: Services, as the name suggests provide some service to a consumer.

## Modeling the Policy Enforcement Resources

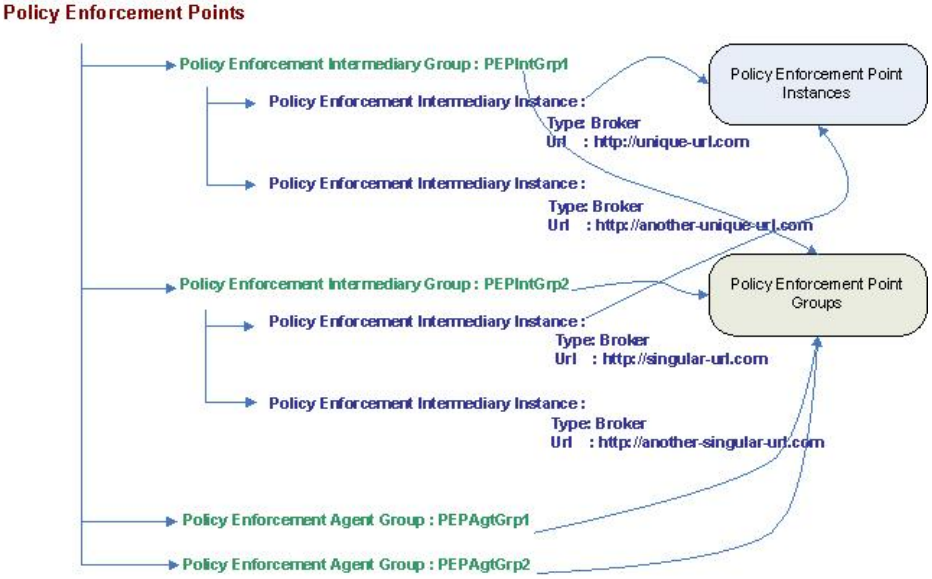
The following are the concepts used while modeling the policy enforcement resources:

- Policy: This refers to a set of configurable rules or constraints and corresponding actions that need to be applied on specified objects in an SOA environment. For example, security policy, schema validation policy, and so on.
- Intermediary: An *Intermediary* refers to a resource that mediates between the service consumer and the service provider and in the process, enforces policies.
- Agent: An *Agent* is deployed on a resource in the SOA environment to enforce policies. You can configure any third party agent with SOA PE. Contact HP Support (**www.hp.com/go/hpsoftwaresupport)** for any assistance regarding this.
- Policy Enforcement Point: A *Policy Enforcement Point* is a logical entity that represents one or more groups of known physical resources in the SOA environment. Each group contains resources that participate in the enforcement of one or more policies (such resources are henceforth referred to as *enforcement resources*). These groups are referred to as *Policy Enforcement Point Groups*
- Policy Enforcement Point Group: Based on the characteristics of the resource where the policy or policies are enforced, the enforcement resources are grouped into *Policy Enforcement Point Groups*. For example, *Policy Enforcement Intermediary Group* and *Policy Enforcement Agent Group*.
  - Policy Enforcement Intermediary Group: All enforcement resources that are *Intermediaries* are grouped under the *Policy Enforcement Intermediary Group*. There can be one or more instance of a *Policy Enforcement Intermediary Group,* each with one or more enforcement resources of the category *Intermediary*.
  - Policy Enforcement Agent Group: All enforcement resources that cannot by themselves enforce the policies created in SOA PE need *agents* to enforce the policies. These resources are grouped under the *Policy Enforcement Agent Group*. There can be one or more than one instance of a *Policy Enforcement Agent Group,* each with one or more than one enforcement resources of the category *Agent*.
- Policy Enforcement Instance: This refers to the physical instance of an enforcement resource. Depending upon the characteristics of the resource (Intermediary or Agent), there can be a *Policy Enforcement Intermediary Instance* or a *Policy Enforcement Agent Instance*. Every *Policy Enforcement Group Instance* is identified by a unique URL.

- Policy Enforcement Point Type: This refers to the vendor specific model of the enforcement resource, for example, *Broker*. The Broker is an installable component of the SOA PE product. It is an intermediary and therefore is a part of the *Policy Enforcement Intermediary Group*. *Policy Enforcement Point Type* can also take other values depending on the availability of support for that vendor in the SOA PE product. Currently, only the Broker is supported as the policy enforcement point.

The following illustration shows the relationship described above.

The words *Intermediary* and Broker are used to refer to the Broker in this document.




.
.
.

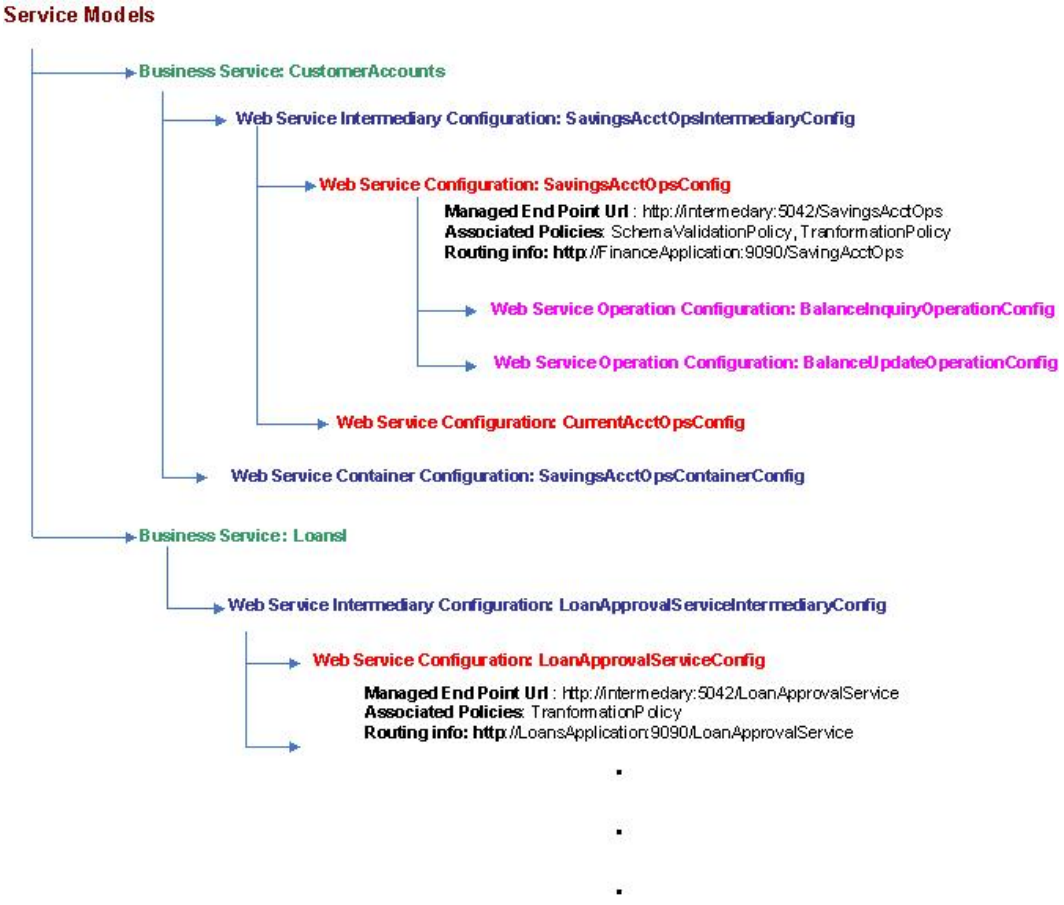
## Modeling Services

The following are the concepts used while modeling the services:

- *Web service*: This can refer to a proxy interface or the actual Web service interface. The address of the Web service interface is called the *endpoint*.
- *Intermediary Service*: An Intermediary service (service proxy) is a corresponding *functional endpoint* created by the Intermediary.
- *Business Service*: A business service is the virtualization of an application or a part of an application that delivers some customer identifiable business functionality to internal or external customers. In SOA PE, business service specifically refers to business functionality that is implemented by one or more Web services.

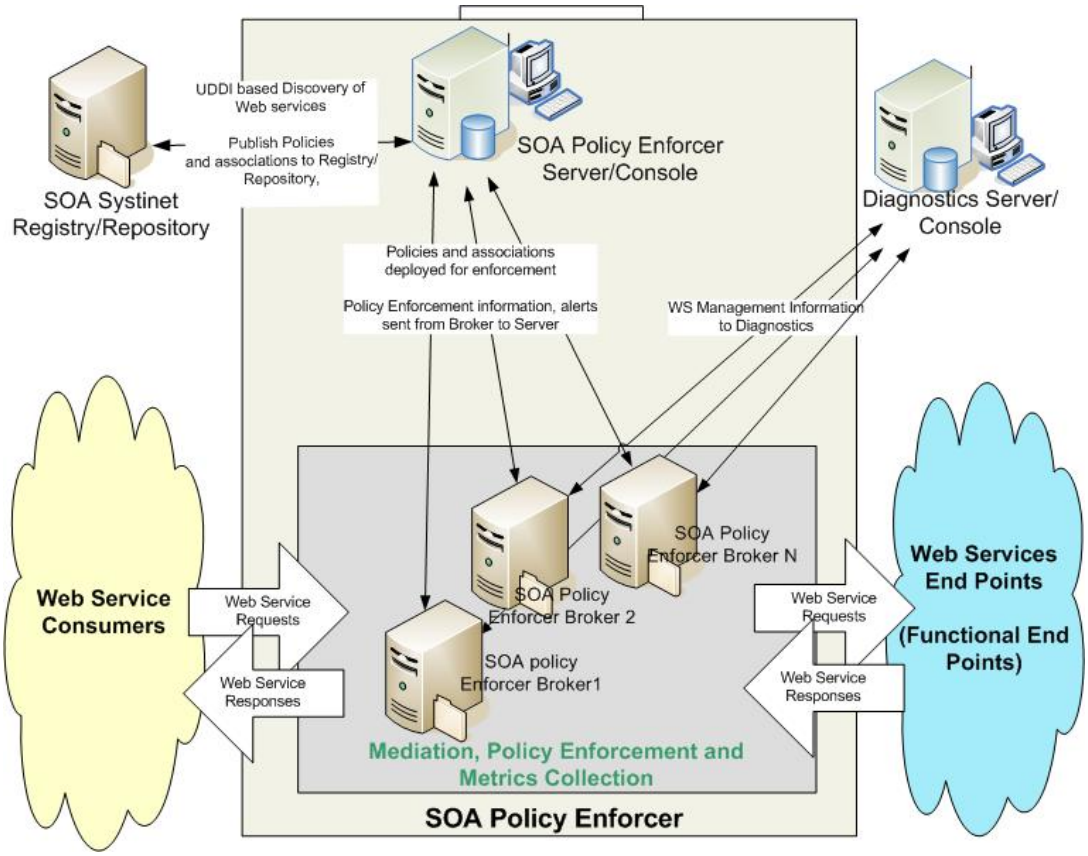
- *Web service configuration*: In SOA PE, a Web service is represented or modeled by a Web service configuration. A Web service configuration can be associated to one or more business services. Depending upon the category of the Policy Enforcement group instance where the policies for the Web service need to be enforced, there can be two types of Web Service configurations, namely, *Web service intermediary configuration* and *Web service container configuration*.
- Web service intermediary configuration: This is a Web service configuration of the type Intermediary. It is always bound to a Policy Enforcement Intermediary group.
- Web service container configuration: This is a Web Service configuration of the type Container. It is always bound to a Policy Enforcement Agent group.
- Web service operation configuration: In SOA PE, a Web service operation is represented or modeled by a Web Service operation configuration. One or more Web service operation configurations can be created for a Web service configuration.

The following illustration shows the relations described above.



# Architecture

Given below is a diagrammatic representation of an SOA PE deployment. Apart from the SOA PE components, the diagram also shows other HP software products that might also be deployed in the SOA environment and interact with SOA PE. Details regarding these products, their usage, and capabilities are out of scope of this document.




SOA PE consists of the following core components:

- **The SOA PE Server (also referred to as Network Services):** This component is the administrative console for the SOA PE product. It provides a web based GUI (SOA PE Server user interface), using which the user can execute various tasks that are required to configure enforcement of SOA runtime policies.

  The words user interface and web interface in this document refer to the SOA PE Server user interface.

  The following are the key functions that a user can execute using the SOA PE Server:

  — Define and maintain runtime governance policies (technical policies)

  — Save technical policies into the SOA Systinet Repository

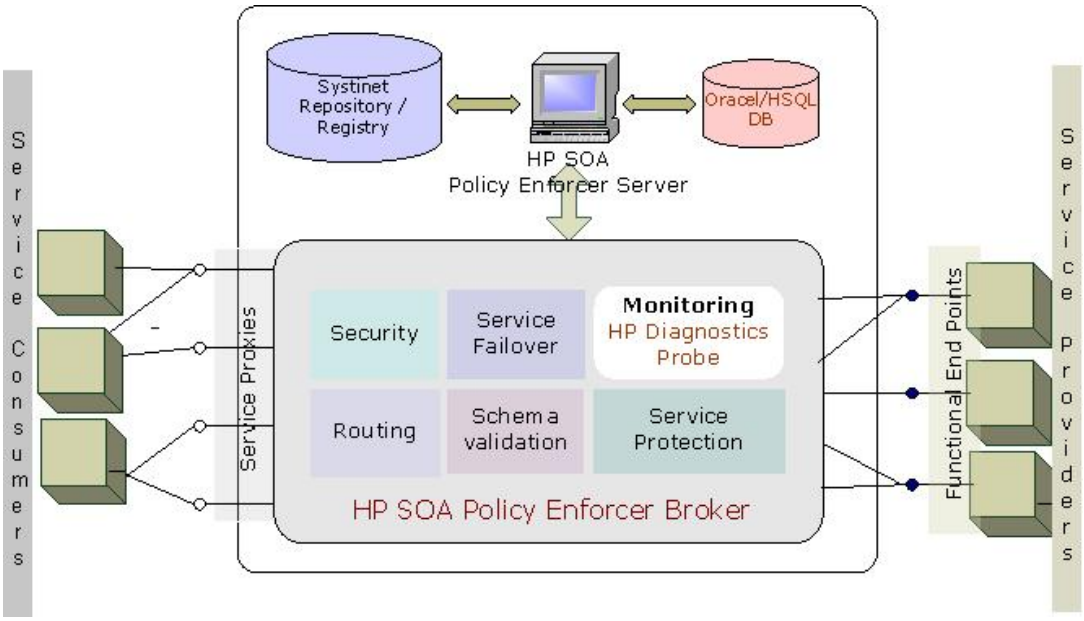
  — Import Web services and associated policies from the registry

  — Associate additional policies to the Web service as needed

  — Provision a Web Service along with the associated policies for enforcement across the various *Policy Enforcement Points* (PEPs).

- View information related to policy enforcement
- Configure and receive alerts for specific policy failures
- View management metrics by invoking the HP Diagnostic probe console

- **The Policy Enforcement Intermediary, WSM Intermediary, or the Broker:** This is the *Intermediary* component of the SOA PE and is used for policy enforcement. The Broker exposes *Intermediary Service* interfaces (also called *Service Proxies*) that act as a gateway for the *Web Service functional endpoints*. All Web service requests are routed through the Broker to the functional endpoints. The responses from the functional endpoints are again routed back to the Web service consumer through the Broker. The Broker enforces the policies associated with the Web Service as it routes the request to the functional endpoint and back. Policies can be enforced in both the request and the response paths.




A typical SOA PE installation includes a single SOA PE server that interacts with any number of policy enforcement intermediaries. The communication between the software components is SOAP/HTTP(S). This communication channel is often referred to as the *Management Channel*.

## Security

SOA PE aggregates Web service message trace information that is collected by policy enforcement intermediaries. Trace information provides historical data related to a Web service performance, access history, security, size, source and destination endpoints, successes, and failures, and can also include the SOAP request-response payloads and profile data.

Trace information is persisted to the SOA PE database (Oracle or HSQL DB) at regular intervals. The web interface is used to view the trace information and generate reports. Refer to the *Viewing Reports* section of this guide for more information about reports.

The SOA PE software can be secured in several ways. Management communication between the SOA PE server and the WSM Intermediaries can be secured using SSL and HTTPS. The communication channel between these core components is often referred to as the management channel. This channel contains sensitive management information and can also be used to affect the resources that are being managed. For these reasons, security on this channel is very important.

The SOA PE software provides security capabilities when using a WSM Intermediary. The capabilities can secure communication between a SOAP client, the WSM Intermediary, and the final Web service endpoint. This communication channel is often referred to as the application channel. Communication on this channel can be secured at both the transport layer (SSL and HTTPS) and at the message layer (WS-Security). This type of security is often implemented when a WS container does not offer native security support.

For instructions on securing the application channel when using an Intermediary, refer to the *SOA PE Security* chapter.

## Integration Points

The SOA PE software provides many integration points that allow custom integrations with existing software assets in an IT environment. Integrations with the SOA PE software provide greater reusability and the flexibility to create solutions that are specific to a particular IT environment. This guide does not provide detailed integration instructions. Detailed instructions for common integrations are provided in the *SOA PE Integration Guide*, located in the `/Documentation` directory of the distribution.

A single instance of the WSM Intermediary can manage multiple intermediary services. In addition, multiple Intermediaries can be used on a single host or can be distributed across hosts. In scenarios where a single service is replicated across multiple machines, management data and metrics are aggregated.

# Using SOA PE for SOA Governance

The following sections describe the components you can use to implement runtime governance using SOA PE. This section also provides an overview about the steps involved in defining a service model using SOA PE.

## Defining Technical Policies

A technical policy refers to a set of configurable rules or constraints and corresponding actions that can be applied on specified components in an SOA environment. Technical policies in SOA PE allow you to implement runtime governance on services. By default, SOA PE includes a set of different supported policy types such as route, log, audit, event, and, so on referred to as *bundled policies*. You can create any of the following policy types to implement runtime governance for your service:

- Audit policy: This policy allows you to track all the events associated with the service to which the policy is attached. Trace information provides historical data related to a Web service performance, access history, security, size, source and destination endpoints, successes, and failures, and can also include the SOAP request-response payloads and profile data. You can view the audit message traces report to view the traced audit information.
- Service protection policy: You can use a service protection policy to prevent an endpoint from getting overloaded with service requests. You can use this type of a policy to specify the number of service requests that an endpoint can accept.
- Scheduled availability policy: You can use a scheduled availability policy to allow or deny access to a service based on the scheduled availability time period specified for that service.
- Event policy: You can use an event policy to generate an alert based on the performance of an operation.
- Content detection policy: You can use the content detection policy to verify the presence of content in a service request message. Based on the presence or absence of the content, the Intermediary either rejects the message from the client and returns a fault code to the client or forwards the message to the endpoint.
- WS-Addressing policy: This policy allows you to enforce communication between service providers and service consumers using WS-Addressing. Refer to the *WS-Addressing* chapter in this guide for more information.
- Schema validate policy: You can use the schema validate policy to validate a schema included in the body of an incoming SOAP request message.
- Transform policy: You can use a transform policy to transform request messages or response messages based on the request or response XSL template that you specify.
- JMS mediation policy: You can use this policy to support SOAP/XML requests over JMS at inbound.
- Message security policy: You can use this policy to make inbound and outbound messages secure.
- Transport security policy: You can use this policy to implement security at the transport level for a message.

Refer to section *Technical Policies* in *Chapter 4 SOA PE Administration* of this guide for more information about technical policies.

## Provisioning a Service

Provisioning a Web service implies enforcing policies for runtime governance using service provisioning. Service provisioning involves multiple steps which you can perform using the service provisioning wizard provided by SOA PE. Refer to the *Service Provisioning* section in *Chapter 4 SOA PE Administration* for more information about service provisioning.

### Enforcement of Runtime Policies

SOA PE enforces runtime governance using technical policies and the PEP. After you provision services and deploy the services to a PEP, the PEP acts as a gateway between the service consumer and the endpoint. Based on the configuration specified in the policies attached to the service, the PEP verifies the requests and responses between the end point and the service consumer. The PEP therefore performs the role of enforcing policies attached to a particular service.

### Configuring Alerts and Notifications

You can configure alerts and notifications in SOA PE to notify about of the state of a service. For example, an alert can be configured to notify you when a policy or a service fails. A notification can be a message that notifies you when the deployment of a particular provisioned service was successful. As an operator, you can see these alerts on the SOA PE UI and take the necessary steps that might be required based on the alert or the notification.

## Monitoring Services

SOA PE integrates with HP Diagnostics to provide service performance monitoring capabilities. By default when you install SOA PE, the installer installs the Diagnostics probe. The Diagnostics probe collects service performance metrics and sends this information to the Diagnostics Server. You can view the following service performance metrics from the SOA PE user interface:

- Availability
- Throughput
- SLO violations
- Security Violations

## Defining Service Models

The SOA PE user interface provides a graphical way of creating, editing, and viewing service models. The service model functionality is spread across different screens; each screen is specific for the structural element of the model being defined. The following tasks outline the typical manner in which a service model is defined. This section does not provide detailed procedural steps. Detailed procedures for these and many other tasks are included in this guide.

In general there are three ways of creating the service model:

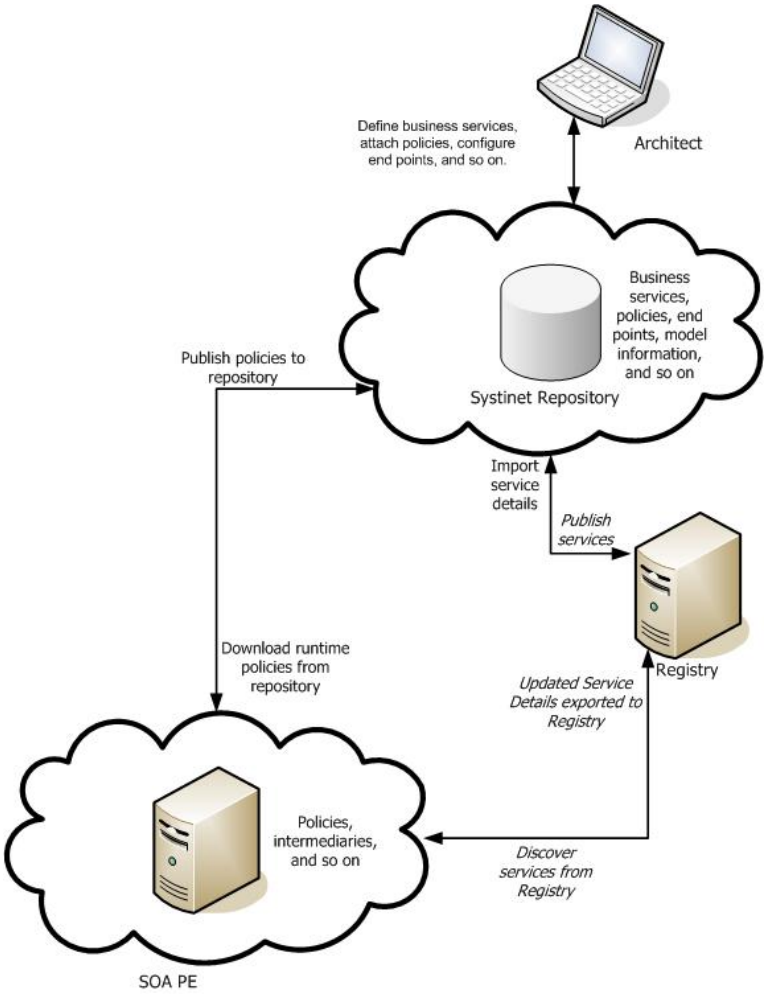
- Using Provisioning Wizard
  - **Create a Policy Enforcement Point** – This step involves the creation of the Policy Enforcement Points that are required to deliver a business application. Refer to chapter *Managing Resources Using PEP Groups* for more information about defining PEPs.

- **Create Policies** – This step involves creation of policies that must be associated with the service. This is optional. Refer to section *Technical Policies* in *Chapter 4 SOA PE Administration* for more information about creating technical policies.
- **Provision Service** – This step involves creation of web service, association of policies to the web service, definition of business service, association of web service to business service and deployment of web service to a PEP. Refer to the *Service Provisioning* section in *Chapter 4 SOA PE Administration* for more information about service provisioning.
- **Life Cycle Status** This step involves verification of the status of a provisioned Web service or deployment of a service saved for provisioning. Refer to section *Lifecycle Management* in *Chapter 6 SOA PE Administration* for more information.

- From Systinet
  - **Configure Systinet** - Configure the location of the Systinet registry/repository in SOA PE from the SOA PE user interface. SOA PE will then publish the Policies to Systinet or download policies from Systinet.

    **Note:** During installation of SOA PE if you have chosen not to install the default policies that are available out of the box with SOA PE, the SOA PE runtime policies available in Systinet are downloaded into SOA PE. If you have chosen to install the default policies, then the default policies are published to the Repository. For outbound security policies downloaded from Systinet, the credentials relevant in new environment need to be specified again in the policy.
  - **Create Service Model** - Model the services in Systinet by defining the business service, implementation service and policy information and export to Registry. Refer to HP SOA Systinet documentation for more information on creating service model in the Repository. Once the service model is exported, SOA PE discovers services from Registry.
  - **Provision Service** - Provision the services discovered by SOA PE to bring the services under runtime governance. SOA PE then publishes the proxy access point and the changed policy association to Systinet.

The flow of policies and service models between SOA PE and Systinet is shown in the following figure.




- Using manual steps
  - **Create a Policy Enforcement Point** – This step involves the creation of the Policy Enforcement Points that are required to deliver a business application. Refer to chapter *Managing Resources Using PEP Groups* for more information about defining PEPs.
  - **Create a Business Service** – These steps involve creation of a business service that you can associate with the Web service to be brought under runtime governance.
  - **Add a configuration** – These steps involve adding Policy Enforcement Points configuration that is required to register a PEP.
  - **Add Resources** – These steps involve configuring the resources that are required to be brought under runtime governance with the business service.

You can also configure alerts and notifications for the provisioned service. Refer to the chapter *Using Alert Notifications* for more information about configuring alerts and notifications.

## Using HP SOA Policy Enforcer with HP SOA Systinet

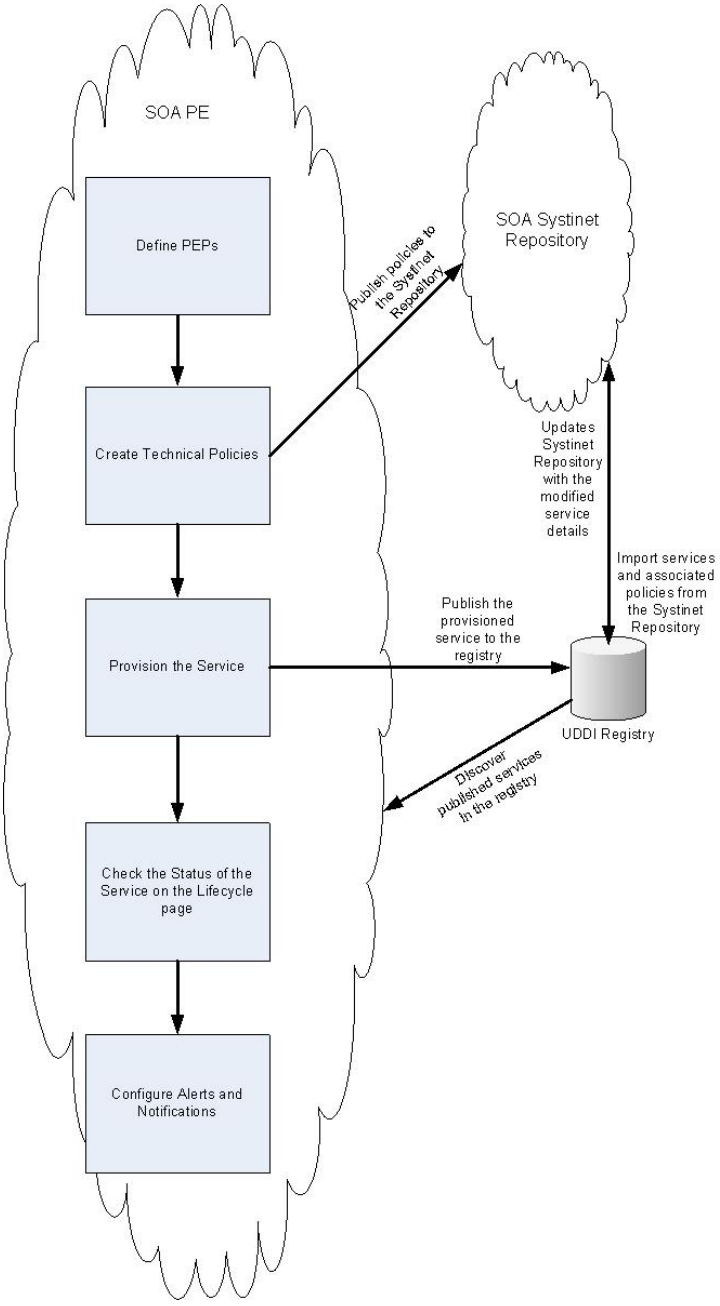
SOA PE 3.00 together with HP SOA Systinet 3.00, known as the HP SOA solution provides a complete solution across the lifecycle of SOA and allows to achieve end-to-end SOA governance. The HP SOA solution offers you a complete integrated SOA governance environment. HP provides end-to-end design-time and run-time governance by integrating SOA PE with HP SOA Systinet using GIF (Governance Interoperability Framework). The GIF is designed to provide a standards-based approach for publishing and discovering business services information in a SOA environment across multiple vendor products. GIF improves SOA visibility, governance, and lifecycle management. During a design phase, as an architect, you can define your business services, attach policies to the service, maintain information about the service model, end points, and so on (collectively known as artifacts) using HP SOA Systinet. SOA Systinet maintains all this information in a location known as the repository. Systinet publishes the artifacts to a registry.

You can use SOA PE to implement run time governance and management of the SOA environment. SOA PE allows you to author runtime policies and publish the policies to the Systinet Repository. When installing SOA PE, you can choose to install the default policies bundled with SOA PE. If you choose not to install the default policies during installation, you can download the policies from the Systinet Repository after installing SOA PE. You can also download the runtime policies from the Systinet Repository.

SOA PE allows you to receive information about published service-related artifacts from the registry. You can use SOA PE to associate more policies to the published Web service, associate Intermediaries to the Web service to bring the service under runtime governance, and then publish the updated service back to the registry.

**Note:** When the SO PE Server is configured to work with the repository, the Systinet Repository must be running before the HP SOA PE Server can be started up.

The following diagram lists the steps detailed above to create a service using the SOA PE user interface. The diagram also shows how service models are published to the registry and the policies published to the Systinet Repository.



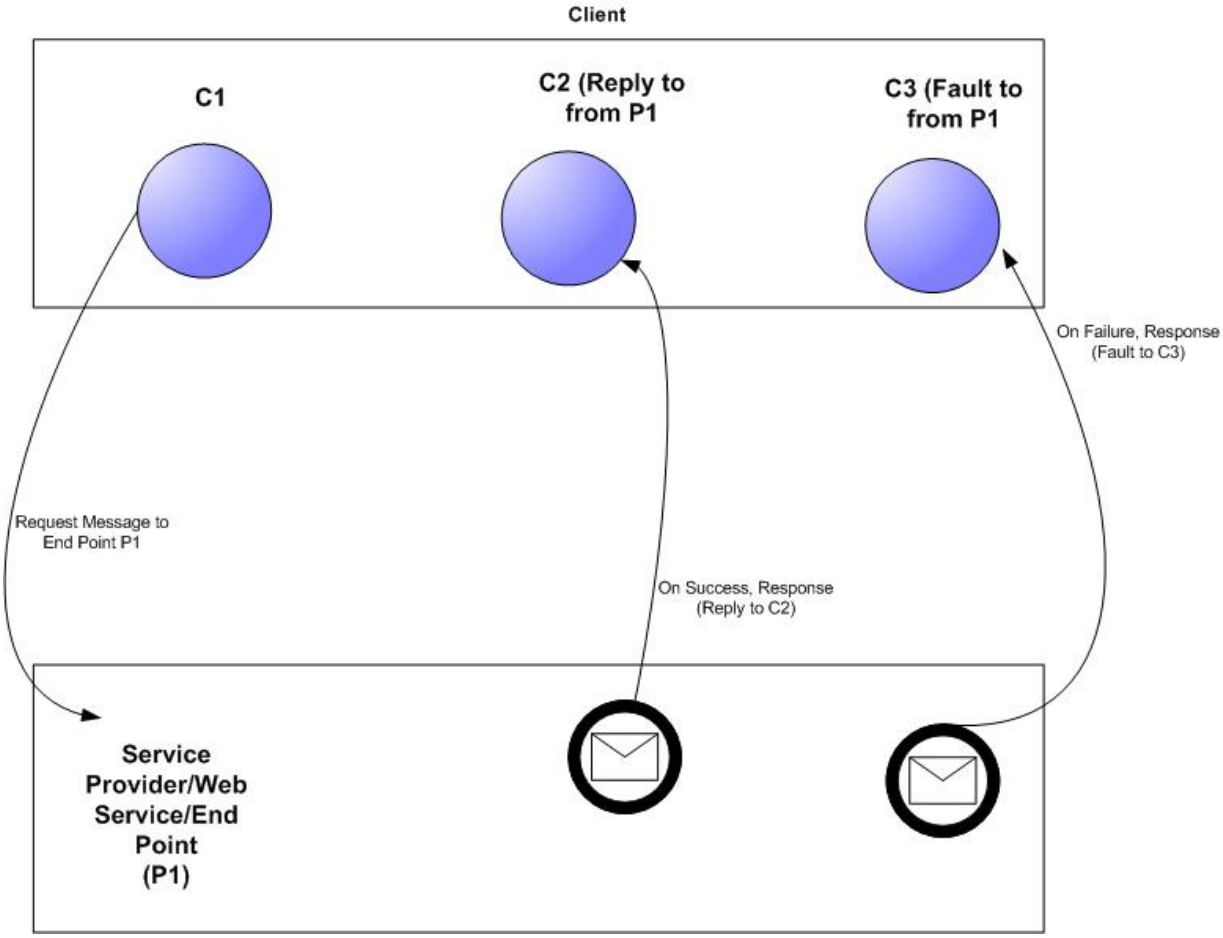

After you publish the service to the registry, SOA Systinet updates its repository with the updated service details. The registry acts as a common location between SOA PE and HP SOA Systinet to discover and publish services.

# WS-Addressing- An Overview

A SOAP request message does not allow you to specify addressing information in messages. The transport layer handles the addressing of messages which makes the SOAP message dependent on the transport layer.

The concept of WS-Addressing provides a transport-independent (protocol independent) mechanism to allow Web Services to communicate addressing information. This concept allows a standard way to route messages across multiple transport mechanisms. WS-Addressing allows you to achieve both synchronous (using anonymous addressing) and asynchronous (using non anonymous addressing) message addressing.

In an asynchronous addressing scenario, you can specify the Web Service to which a request must be sent (in the To endpoint). You can configure different recipients at the service consumer for receiving response messages (Reply to from end point). You can also configure recipients at the service consumer side to receive fault messages (fault to from end point) in the event of an error. Refer to the following diagram that illustrates this scenario.




In this scenario, you can see that C1 sends the request message to the Web Service. Based on the success or failure of the message at the Web Service, the Web Service routes the message to C2 or C3. In this scenario, the resources at C1 are free as soon as it sends a message to the Web Service. C2 at the service consumer side receives successful response messages from the Web Service. In the event of a failure of the request message the Web Service side, C3 receives the fault message from the Web Service. An Anonymous communication scenario helps you to configure different points to send or receive messages to ensure optimal usage of system resources.

You can refer to the World Wide Web Consortium (W3C) site (http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509) for more information about WS-Addressing specification.

2

# Getting Started

This chapter explains basic tasks that are associated with using SOA PE and covers the following topics:

- Finance Example Overview
- Starting and Stopping the SOA PE
- Starting the SOA PE User Interface
- Configuring HTTP Settings
- Configuring an Oracle Database
- Adding User Roles
- Using XPL Logging
- Using XPL Tracing

## Finance Example Overview

A Web services-based example application is provided in the distribution and can be used to test the SOA PE software. The Finance application is included as a convenience if you do not have a Web service-based application to test with while setting up the software. The Finance application is also used as part of the SOA PE HTML-based tutorials that are located in the `/Documentation` directory of the distribution. The tutorials also include setup instructions.

The Finance application is located in the `/Examples` directory of the distribution. The example includes a Web service for the Tomcat, BEA WebLogic Server (WLS), and the .NET platform. In addition, a client is included with the example. The client is only available for the Windows platform.

The Microsoft .NET Redistributable Package and Microsoft WSE 2.0 must be installed on the computer where the Finance application client is installed.

## Setting Up the Finance Application

To set up the Finance application:

1. Deploy the finance service (`axis.war`, `finance-service.ear`, or `FinanceServiceInstaller.msi`) to either the Tomcat, WLS, or .NET platform respectively.
2. Install the Finance client, using `/Examples/FinanceService/client/FinanceSetup.msi`.
3. From the directory where you installed the Finance client, click `FinanceClient.exe`. The HP Finance Client (.NET) application starts.
4. Click the **Configuration** tab.
5. In the Server URL field, enter the URL for the deployed finance server. For example,

   When using Tomcat enter:

   `http://<host:port>/axis/services/FinanceServiceSoap?wsdl`

   When using WLS enter:

   `http://<host:port>/FinanceService/FinanceService`

   When using .NET enter:

   `http://<iis_host>/FinanceService/FinanceService.asmx`
6. Click **Apply**.
7. Click the **Quotes and Information** tab.
8. In the Symbol field, enter `hpq` and click **Get Quote**. The quote information is returned in the Results section. You can also enter `MSFT`, `IBM`, and `BEAS`. Any other symbol will generate an exception.

# Starting SOA PE

A script for both Windows and UNIX is provided to start SOA PE. The script is located in <*install_dir*>`/bin/win32` and <*install_dir*>`/bin/unix` respectively.

Windows users can choose to create product icons during the SOA PE installation. If you accepted the default program group during installation, you can start the SOA PE server by clicking **Start | Program Files | SOA Policy Enforcer 3.00 | Network Services**.

Make sure an environment variable MIP_JAVA_HOME was created during the SOA PE installation. SOA PE will not start if the environment variable is not set. This variable must be set to the JDK you want SOA PE to use. See the *SOA PE Installation Guide* for Java version requirements.

To start SOA PE:

1. Open a command prompt.
2. Depending on your platform, change directories to <*install_dir*>`\bin\win32` or <*install_dir*>`\bin\unix`.

Run the `networkservices` startup script. The console outputs log messages as SOA PE starts.

During the SOA PE installation, you had the option to install SOA PE as a Windows Service. If you chose this option, SOA PE is already running. Attempting to start SOA PE again causes an error.

## Installing the SOA PE as a Windows Service

If you choose not to install SOA PE as a Windows service during the installation, a batch script is provided that installs SOA PE as a Windows service. This allows the server to automatically start whenever Windows is started. The script can also be used to remove SOA PE from being a Windows service.

To install SOA PE as a Win 32 Service:

1. Open a command window.
2. Change directories to <*install_dir*>`\bin\win32\services`.
3. Run `service-manager.bat` and specify the following arguments:

```
service-manager.bat -install networkservices <install_dir>
```

The service has been successfully installed when the following message appears in the console:

```
Service "SOA Policy Enforcer v3.00 networkservices" installed.
```

The script configures SOA PE to automatically start the next time Windows is started. You must use the Windows Computer Management Console to change this behavior.

To remove the service, run the service-manager script and specify `-remove`. For example,

```
service-manager.bat -remove networkservices <install_dir>
```

# Stopping SOA PE

SOA PE can be stopped using the stop process methods that are appropriate for the host operating system.

## Windows

Switch to the command window where the server process is running and type `Ctrl+c`. Then type `y` to terminate the process.

If SOA PE is running as a Windows service, the service must be stopped. To stop a Windows service, open the Control Panel and select **Administrative Tools**. From the Administrative Tools screen, select **Services**. From the Services screen, right-click the SOA PE service and select **Stop**.

### UNIX

When using Linux or HP-UX, open a terminal window and issue the following command:

```
ps -ef | grep java
```

The command lists all current Java processes, including the process number. Find the SOA PE process and issue the `kill` command to stop the process. For example:

`kill` <*process number*>

## Starting the SOA PE User Interface

SOA PE is administered through the SOA PE user interface. The user interface is a web application that runs on port 5002. A different port can be specified in the <*install_dir*>`\conf\networkservices\mipServer.xml` file.

To start the web interface:

1. Start SOA PE as described previously.
2. Open a browser.
3. Enter the following URL and substitute <*host*> with the host name where the SOA PE server is running:

   ```
   http://<host>:5002/bse
   ```

4. The default credentials are `admin` for the user name and `password` for the password.
5. Click **Login**. The Dashboard is displayed.

The SOA PE version (including installed patches) is located above the copyright statement at the bottom of each page.

### Assigning Access to the Web Interface

The <*install_dir*>`\conf\networkservices\mipServer.xml` file allows you to define user credentials for accessing the web interface. In particular, you can define user names and passwords for accessing the console. A single role, `admin`, has been implemented. All users must be associated with this role. This feature is typically only used while testing the SOA PE software.

To add access rights for a user:

1. Stop SOA PE if it is currently started.
2. Use a text editor to open <*install_dir*>`\conf\networkservices\mipServer.xml`.

3 Comment out the following entry:

```
<entry name="com.hp.mip.security.provider.console">default</entry>
```

4 Add a new user and password entry. For example:

```
<entry name="com.hp.mip.server.security.user">Joe User</entry>
<entry name="com.hp.mip.server.security.password">password</entry>
```

5 Save and close the file.

6 Restart SOA PE.

# Configuring HTTP Settings

SOA PE contains an HTTP server. The server is used to accept HTTP requests for the SOA PE web interface. This step is optional.

## Configuring the HTTP Server Port Number

SOA PE contains a Java HTTP Server that listens for HTTP messages and is used by the SOA PE web interface console. The HTTP Server is configured in the <*install_dir*>\conf\ networkservices\mipServer.xml file. The default port used by the HTTP Server is 5002. If port 5002 is currently being used, SOA PE will not start.

To configure the port number:

1 Stop SOA PE if it is currently started.

2 Use a text editor to open <*install_dir*>\conf\networkservices\mipServer.xml.

3 Change the port number for the com.hp.http.server.port entry. For example:

```
<entry name="com.hp.http.server.port">5003</entry>
```

4 Save and close the file.

5 Restart SOA PE.

## Configuring HTTP Server Thread Settings

You can change the manner in which the HTTP server manages threads. Thread management can help increase performance and improve latency for the HTTP Server. There are three thread settings:

- <entry name="com.hp.http.threads.max"> – The maximum number of threads allowed to be used by the HTTP server.
- <entry name="com.hp.http.threads.min"> – The minimum number of threads allowed to be used by the HTTP server.
- <entry name="com.hp.http.threads.maxIdle"> – The maximum amount of time in milliseconds that an HTTP server thread can remain idle.

To configure the HTTP server thread settings:

1 Stop SOA PE if it is currently started.

2 Use a text editor to open <*install_dir*>\conf\networkservices\mipServer.xml.

3 Configure the HTTP server thread settings. For example:

```
<entry name="com.hp.http.threads.max">50</entry>
<entry name="com.hp.http.threads.min">2</entry>
<entry name="com.hp.http.threads.maxIdle">60000</entry>
```

4 Save and close the file.

5 Restart SOA PE.

# Configuring the Refresh Setting

SOA PE's web interface contains a refresh feature that periodically auto-refreshes screens that have dynamic information. If this feature is disabled, you must manually refresh a screen to view the most current information. The feature can be configured to refresh at any interval (in seconds).

The refresh feature is disabled by default. When enabled, the refresh image in the top right corner of the web interface is animated. You can also enable or disable this by clicking on the refresh image.

To configure the refresh setting:

1 Log in to the SOA PE web interface as an administrator.

2 From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.

3 From the General Settings tab, enter the interval (in Seconds) to wait before an auto-refresh.

4 Click to select the **Refresh enabled** check box.

5 Click **Save**.

# Configuring an Oracle 10g Database

SOA PE stores service messages, service trace messages, and alerts to a database. The SOA PE software includes the HSQL database which is a light-weight database. This database can be used for testing. However, for production environments, a database schema for creating the data tables in Oracle 10g is provided. See the Oracle 10g documentation if you are not familiar with creating data tables using a schema file.

The schema for creating the tables in Oracle is located at <*install_dir*>\data\oracle\Create-Tables-Oracle.SQL. After you create the database and run the schema, configure SOA PE to use the database.

You must copy the 10g version of the oracle thin JDBC driver (oracle_ojdbc14.jar and oracle_nls_charset12.jar) into the <*install_dir*>/lib/ext directory. These .jar files are available from the Oracle website.

To configure SOA PE to use the Oracle 10g database:

1. Stop SOA PE if it is currently started.
2. Open <*install_dir*>\conf\networkservices\mipServer.xml.
3. Uncomment the Database Properties section and add your database information. For example:

```
<entry name="com.hp.db.demo">false</entry>

<!-- The demo entry must be set to false. -->

<entry name="com.hp.db.driver">
   oracle.jdbc.driver.OracleDriver</entry>
<entry name="com.hp.db.url">
   jdbc:oracle:thin:@host:1521:DB1</entry>
<entry name="com.hp.db.user">admin</entry>
<entry name="com.hp.db.password">admin</entry>
```

4. If the demo entry is set to true, you must set it to false. (for example, `entry name="com.hp.db.demo">false</entry>`)
5. Specify the host name (@host), port number (for example, 1521 in the sample code above), and the SID (for example, DB1 in the example above).
6. Save and close the file.
7. Restart SOA PE.

## Configuring Time Zones

Oracle database versions less than 9.2.0.5 use the small time zone file (`timezone.dat`) by default. This file does not contain several time zone region names including many European time zone names. If you are running SOA PE in a time zone that is not in the Oracle small time zone file, check to see if the time zone is in the large time zone file (`timezlrg.dat`).

If your Oracle installation is on UNIX, you can configure Oracle to use the large time zone file by setting an environment variable:

```
ORA_TZFILE=$ORACLE_HOME/oracore/zoneinfo/timezlrg.dat
export ORA_TZFILE
```

If your Oracle installation is on Windows, you must modify the Windows registry and add the `ORA_TZFILE` parameter to the `HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOMEID` subkey and set it to `$ORACLE_HOME/oracore/zoneinfo/timezlrg.dat`.

You must restart the database for the change to take effect.

## Performing Database Maintenance

As with all databases, you must monitor the database and periodically do maintenance to control the size of the database. Some example SQL scripts are provided to remove old alerts and trace messages from the SOA PE database. The SQL scripts are located in:

<*install_dir*>\data\oracle\CleanAlerts-Preview-Oracle.SQL

<*install_dir*>\data\oracle\CleanAlerts-Oracle.SQL

<*install_dir*>\data\oracle\CleanAudits-Preview-Oracle.SQL

<*install_dir*>\data\oracle\CleanAudits-Oracle.SQL

## Migrating an SOA PE Database

Refer to the *SOA PE Installation Guide* for instructions to migrate an SOA PE database.

# Adding Users

SOA PE let you to assign roles for users. You must have administrative privileges to create new users. SOA PE supports the following roles:

- Administrator: A user with an administrator role can access all the resources, create new users, and modify the roles of existing users.
- Stakeholder: A user with the stakeholder role can access limited resources.
- Guest: A user with the guest role can only view specific information.

You cannot assign multiple roles to a single user.

SOA PE performs authentication or authorization using the security provider configured. A default security provider is bundled along with SOA PE. When using the default provider the built in administrator user name and password are the following:

  - User name: admin
  - Password: password
- You can create more users and assign roles to them by using the **View->StakeHolder Roles** option in the SOA PE web interface.

You can also configure SOA PE to use a different provider. Contact HP Software support at www.hp.com/go/hpsoftwaresupport for instructions to configure a different security provider for SOA PE.

When using a provider other than the default user authentication is accomplished by using users configured in the security provider.

You must configure users configured in other third party providers in SOA PE for assigning services and resources to the user.

## Adding Users

To add users, follow these steps:

1. Log in to the SOA PE web interface as an administrator.
2. Click StakeHolder Roles in the View drop-down list present on the left pane of SOA PE web interface. This People dialog box opens.
3. Click Add. The Add dialog box opens.

4 Type the appropriate details in the boxes and select one of the following from the Role:* drop-down list:

- ROLE_ADMIN
- ROLE_STAKEHOLDER
- ROLE_GUEST

- The suffix of your selection determines the type of user role that you want to create. Check if these roles are still named the same way. The password field is shown only when using default provider.

- The userid must not be more than 20 characters

5 Click **Save** to add the new user role.

## Modifying Existing User

To modify the existing user, follow these steps:

1 Log in to the SOA PE web interface as an administrator.

2 Click StakeHolder Roles in the View drop-down list on the left pane. The People dialog box opens.

3 Click the user role that you want to modify. This displays the details of the selected user role.

4 Click Edit. The Edit dialog box opens and lets you modify the details of the user.

5 Click Save to save your changes.

You cannot change the role of a user. To change the role of a user, you must delete the user and add the user again.

## Removing Existing Users

To remove existing user roles, follow these steps:

1 Log in to the SOA PE web interface as an administrator.

2 Click StakeHolder Roles in the View drop-down list on the left pane. The People dialog box opens.

3 Click the user that you want to remove. This displays the details of the user role that you selected.

4 Click Remove. This displays the warning that all business and PEP relationships will be removed.

5 Click R**emove** to confirm the removal of the selected user role.

# Using XPL Logging

SOA PE uses HP Software Cross Platform (XPL) logging. The following sections describe Installation, configuration, and usage.

## Installing XPL Logging

During the SOA PE installation, you may have been prompted to select the HP Software installation and data directories. You are only prompted for this information the first time you install an HP Software product.

The default value for the installation directory is `C:\Program Files\HP Software` on Windows and `/opt/OV` on UNIX. The default value for the data directory is `C:\Program Files\HP Software\data` on Windows and `/var/opt/OV` on UNIX. The SOA PE log files are created in the log subdirectory of the data directory. If you do not run SOA PE as an administrator, you may need to change the permissions for the log subdirectory.

### XPL Tools

The HP Software Cross Platform Component contains logging and tracing tools. If you need to change the default log file configuration parameters, install the component. Run the appropriate installer in the /Support directory of the SOA PE CD.

## Configuring XPL

SOA PE automatically creates log files in the log subdirectory of the HP Software data directory. The SOA PE log file name has the following format:

*networkservices[unique].sequence.locale*

For example:

`networkservices0.0.en_US`

This file is the first SOA PE log file created for the US English locale.

SOA PE creates a log file for an English locale and a second file for your system's locale if it is different from English.

SOA PE creates up to 10 log files. Each file contains up to 1 megabyte of data. The log files have sequence numbers 0 through 9. When the maximum number of log files is exceeded, the sequence 0 log file is overwritten.

You can change the maximum number of log files and log file size using the HP Software Cross Platform tool, `ovconfchg`. After installing the HP Software Cross Platform Component, this program is in the `/bin` directory of the HP Software installation directory. An example of using this tool is the following.

```
ovconfchg -ns xpl.log.OvLogFileHandler -set filecount 12
-set filesize 2
```

This command sets the maximum number of log files to 12 and the maximum log file size to 2 megabytes.

Restart SOA PE for the new configuration to take effect.

You can see the current configuration using the following command:

```
ovconfget
```

For more information about `ovconfchg` and `ovconfget`, see the help documentation in the help subdirectory of the HP Software installation directory.

## Configuring Log Levels

You can change SOA PE log levels using the SOA PE web interface. You can also change the log levels by editing the `logging.properties` file in the JDK `/lib` directory or the `xpllogging.properties` in the <*install_dir*>`/conf/networkservices` directory. The log levels are `SEVERE`, `WARNING`, `INFO`, `FINE`, `FINER`, and `FINEST`. By default the log level is set to `INFO`.

### *Using the SOA PE User Interface*

The edit/query log level feature provides the ability to edit/query log levels for different log categories that are configured for SOA PE. Different log levels and log categories provide varying levels of log details that can help identify process events that are occurring in SOA PE.

To edit/query log levels:

1. From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens and the General Settings tab is selected by default.
2. Click the **Edit/Query SOA PE Server Log Levels**. The Edit/Query Log Levels screen opens in a new browser window. The default root logger and its current log level are displayed.
3. Using the Log Level drop-down list, select a new log level.
4. Click **Update Settings**. The log level for this logger is updated on SOA PE. The log file will display any log messages that are sent to this log level. If no code uses this logger or this particular log level, no new messages are displayed.
5. In the Logger field, change the Root logger to `MIP`. Any string can be entered in the Logger field. You can also set the log level for individual packages. SOA PE packages begin with `com.hp.ov.mip`.
6. Click **Query**. The Log Level field updates and displays the current log level for the logger. If you query a logger that is not currently implemented, the Log Level field displays the word Unknown. If you save a logger that is not currently implemented in SOA PE, the logger is created on SOA PE and the log level selected is set. However, because no code is using the logger, no new messages are displayed.
7. Repeat steps 3 and 4 to change the log level.
8. Click **Cancel** to close the Edit/Query Log Levels screen.

### Using JRE Properties File

You can change the log level for SOA PE by editing the `logging.properties` file in the `JRE /lib` directory. You must restart SOA PE for the changes take effect. For example, you can add the following line to the end of the file:

```
com.hp.ov.mip.level = FINE
```

This sets the log level for the SOA PE to `FINE`.

### Using the XPL Properties File

You can change the log level for SOA PE by editing the `xpllogging.properties` in the <*install_dir*>`/conf/networkservices` directory. You must restart SOA PE for the changes take effect. For example, you can add the following line to the end of the file:

```
com.hp.ov.mip.level = FINE
```

This sets the log level for SOA PE to `FINE`.

## Viewing Logs

You can use an editor or the SOA PE web interface to view the SOA PE log files. From the **Actions** drop-down menu, click **Change Settings** to go to the Settings page and then click **View SOA Policy Enforcer Server Log**. You can also use an editor to view the SOA PE log files in the HP Software data log directory.

# Using XPL Tracing

SOA PE uses the HP Software Tracing tools for tracing. See the *HP Software Tracing Concepts Guide* for detailed information on how to use the trace feature. The guide is located on the SOA PE CD in the `/Documentation` directory.

## Installation

Before starting, verify if the HP Software Tracing tools are already installed on your system. You can check to see if the trace server is installed. On Windows, the trace server is installed as `C:\Program Files\HP Software\bin\ovtrcsvc.exe`. On UNIX, the trace server is installed as `/opt/OV/lbin/xpl/trc/ovtrcd`.

The tracing tools are located on the SOA PE CD in the `/Support` directory.

### Windows

To install the tracing tools on a Windows system, double-click `/Support/HPOvXpl-`<*version*>`-release.msi`.

### HP-UX

To install the tracing tools on an HP-UX system, run the following command:

```
swinstall -s /Support/HPOvXpl-<version>-HPUX11.0-release.depot \*
```

### Linux

To install the tracing tools on a Linux system, run the following command:

```
rpm -Uhv /Support/HPOvXpl-<version>-Linux2.4-release.rpm
```

## Example Configuration Entries

The following SOA PE entries are example entries for the XPL configuration file:

```
TCF Version 3.2
APP: "networkservices"
SINK: Socket "system1.acme.com" "node=192.1.60.106;"
TRACE: "mip.config" "Operation" Info Error
TRACE: "mip.config" "Parameters" Info Error
TRACE: "mip.config" "Procedure" Info Error
TRACE: "mip.metrics" "Operation" Info Error
TRACE: "mip.metrics" "Parameters" Info Error
TRACE: "mip.metrics" "Procedure" Info Error
TRACE: "mip.slos" "Operation" Info Error
TRACE: "mip.slos" "Parameters" Info Error
TRACE: "mip.slos" "Procedure" Info Error
TRACE: "mip.deploy" "Operation" Info Error
TRACE: "mip.deploy" "Parameters" Info Error
TRACE: "mip.deploy" "Procedure" Info Error
```

3

# Managing Resources Using PEP Groups

This chapter explains how to create and maintain PEP groups and register PEP resource using the SOA PE web interface. PEP groups are part of the service model definition and are an integral part of managing SOA resources using the SOA PE software.

## Overview

PEP is an abstract concept that can mean different things. Within the scope of the SOA PE's service model, a PEP is the virtualization of management information and capabilities of a group of resources. PEP can represent a single resource or can be a collection of resources that are managed together in some meaningful way. Typically, this model is used to organize resources that are similar.

The PEP groups that are supported include the following:

- **Policy Enforcement Intermediary Groups**: This type of PEP captures the management of policy enforcement intermediary groups and their hosted intermediary services. The policy enforcement intermediary groups support the deployment and discovery of an intermediary group.

## Creating a PEP

The following instructions are specific to the type of PEP that is being created. Once a PEP is created, you can register any number of resources to the PEP. Instructions for registering resources to a PEP are provided in the "Registering Resources" section below.

### Create a Policy Enforcement Intermediary Group

To create a policy enforcement intermediary group, follow these steps:

1 From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Point Summary screen opens.

2 From the policy enforcement intermediary group section, click **Add**. The Add policy enforcement intermediary group screen opens.

3 Complete the following fields:

- **Name**: A descriptive name for the PEP.
- **Description**: A description for the PEP.
- **Owner**: Use the drop-down list to select an owner for the PEP. The owner of a PEP is generally responsible for lifecycle management and publishing of the service.
- **Support**: Use the drop-down list to select a support person for the PEP. The person or group responsible for supporting deployed instances of the service.
- **Availability**: This check box indicates that an alert is generated when the PEP is not operational (for example, when a managed policy enforcement intermediary group that is contained in the PEP is not available).
- **Alert Recipients**: The alert categories that are used for this PEP. Use the respective drop-down lists to select alert categories for both degraded and failed alerts.

4 Click **Save**. The PEP is created and its view screen opens.

5 Repeat this procedure to create additional policy enforcement intermediary group or refer to the “Registering Resources” section to add a resource to this PEP.

# Registering Resources

The following instructions should be completed after completing the instructions in the previous section. The instructions in this section are organized based on the type of resource that is registered.

## Registering Policy Enforcement Intermediary Resources

When a managed policy enforcement intermediary is registered, its hosted Web services or intermediary services are automatically discovered and registered as well. As services are added and removed from an intermediary, they are automatically added and removed from the PEP.

To register an intermediary resource, follow these steps:

1 Make sure the managed intermediaries that you want to register are started.

Instructions for setting up managed intermediary are located in separate sections of this guide.

Some SOA PE features may not work as expected when using intermediary versions that are different than the SOA PE version. It is recommended that the SOA PE version and the policy enforcement intermediary versions match.

2 From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Service Summary screen opens.

3 Select the PEP you want to contain the resource. The PEP View screen displays for the selected PEP.

4 From the Contained Policy Enforcement Intermediary Instances section, click **Add**. The Add Policy enforcement intermediary screen opens.

5 From the **Type** drop-down box, select the type of resource you want to register.

6 Using the fields provided, enter the host and port where the managed resource is installed.

The policy enforcement intermediaries publish their management interface (WSDL) to a URL. The SOA PE web interface uses the information entered in this step to construct the URL. Once you become familiar with the URL format, you can use the URL text box to enter the URL to the management WSDL.

7 Click the SSL check box if you want the management channel to this resource to be secured. To use this feature you must first set up the appropriate security components. See the "Using SSL for the Management Channel" chapter.

8 Click **Add**. The Add Policy Enforcement Intermediary screen reopens and the Contained Web Services section lists the Web services that are discovered in the managed Policy enforcement intermediary.

9 Click **Add**. The Policy Enforcement Intermediary screen opens and the Contained Web Services section lists the resources that are now registered in the PEP.

10 Repeat this procedure to register additional resources for this PEP.

# Managing Web Service Resources

Registered resources have a view screen that provides details about the resource as well as basic operations that allow you to interact and manage the resource.

From this screen, you can do the following:

- Select a Web or an intermediary service to view its details
- View log traces for the resource
- Check the availability of a resource
- View/Acknowledge alerts that are currently active for a resource
- Delete a resource

## Viewing Registered Resources

To view details about a registered resource, follow these steps:

1 From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.

2 From the list of PEPs, click the PEP you want to view. The PEP View screen opens for the selected PEP.

3 From the Contained Policy Enforcement Intermediary Instances section, click a resource to view it. The resource's view screen opens. Each service in the resource is listed in the Contained Web services section. You can click a service link to view the service's details including a performance graph and a list of Web service operations. In addition, you can click an operation to see its properties and performance graph.

## Viewing Log Traces

The log trace feature is a convenient way to view the log file for a registered policy enforcement intermediary from within the SOA PE web interface without having to log on to multiple remote computers. The log traces are used to troubleshoot problems or to verify that a policy enforcement intermediary is operating successfully.

To view log traces, follow these steps:

1 From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.

2 From the list of PEPs, click the PEP you want to view. The PEP View screen opens for the selected PEP.

3 From the Contained Policy Enforcement Intermediary Instances section, click a resource to view it.

4 From the Logging Level section, click **View Log**. A new browser window opens and lists the last 20 log messages.

5 Use the text box to change the amount of entries to be displayed.

6 Click **Go** to refresh the window.

7 When you are done viewing the log messages, click **Close** to close the browser window.

### Editing and Querying Log Levels

The edit/query log level feature provides the ability to query/edit log levels for different loggers that are configured in a policy enforcement intermediary. Different log levels and loggers provide varying levels of log details that can help identify process events.

A policy enforcement intermediary contains a predefined set of loggers. For .NET, two categories (Catalog and libraries) are used. For policy enforcement intermediaries and loggers are defined in each intermediary's XPL configuration file. In addition, any custom loggers that are implemented in a policy enforcement intermediary can also be configured.

To edit/query log levels, follow these steps:

1 From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.

2 From the list of PEPs, click the PEP you want to view. The PEP View screen opens for the selected PEP.

3 From the Contained Policy Enforcement Intermediary Instances section, click a resource to view it.

4 From the Logging Level section, click **Edit/Query Log Levels**. The Edit/Query Log Levels screen opens in a new browser window. The default root logger and its current log level open.

5 Using the Log Level drop-down list, select a new log level.

6 Click **Save**. The log level for this category is updated on the policy enforcement intermediary. The log file will now display any log messages that are sent to this category's log level. If no code uses this logger or this particular log level, no new messages are displayed.

7 In the Logging Category field, replace `MIP` with a logging category that is implemented on this policy enforcement intermediary. Any string can be entered in the Logging Category field.

8 Click **Query**. The Log Level field updates and displays the current log level for the category.

If you query a logging category that is not currently implemented in the policy enforcement intermediary, the Log Level field displays the word `Unknown`. If you save a logging category that is not currently implemented, the logging category is created and the log level selected is set. However, because no code is using the logging category, no new messages are displayed.

9 Repeat steps 5 and 6 to change the log level for the logger.

10 Click **Cancel** to close the Edit/Query Log Levels screen.

## Enabling Availability Notifications

The availability feature allows an alert notification to be sent to alert recipients whenever a registered policy enforcement intermediary is not available. Enabling this feature will quickly notify individuals when a policy enforcement intermediary is not operational and can help you determine why a Web service is failing.

To enable availability notifications, follow these steps:

1 From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.

2 From the list of PEPs, click the PEP you want to view.

3 From the Contained Policy Enforcement Intermediary Instances section, click a resource to view it.

4 From the Status section, click **Edit Alerts**. The Availability screen opens.

5 From the Availability section, click the **Alert when unavailable** check box. A check indicates that availability notifications are enabled.

6 From the Alert Recipients section, use the Alert Recipient drop-down list to select a Recipient Category to receive the alert.

7 Click **Save**. Alerts are displayed in the Resource Alerts section.

To see a generated availability alert, manually shutdown the policy enforcement intermediary for which you enabled availability alerts. Refresh the screen. An alert message opens in the Alerts section and indicates that the policy enforcement intermediary is unavailable. Restart the intermediary. When the intermediary becomes available, an alert message opens in the Alerts section and indicates that the intermediary is available.

## Deleting a Resource

You can delete a PEP at any time. This procedure is typically completed when a policy enforcement intermediary host is decommissioned or is no longer used to host Web services. When you delete an intermediary, it is removed from SOA PE. If the resource is part of a PEP, it is removed from the PEP as well.

Deleting a policy enforcement intermediary also removes its Web services (or intermediary services).

To deregister a policy enforcement intermediary, follow these steps:

1. From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.
2. From the list of PEPs, click the PEP you want to view.
3. From the Contained Policy Enforcement Intermediary Instances section, click a resource to view it.
4. Click **Remove**. A remove screen opens in a new browser window.
5. Click **Remove**. The Business Services screen opens.
6. From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.
7. From the list of PEPs, click the PEP you want to view. The resource is removed from the Contained Policy Enforcement Intermediary Instances section.

# Managing Policy Enforcement Intermediaries

The View PEP screen provides details about a PEP. The screen is a convenient way to view managed policy enforcement intermediaries from the context of their PEP. From this screen, you can do the following:

- View a PEP and its details
- Edit a PEP
- Add/Remove resources from a PEP
- View/Acknowledge alerts that are currently active for the PEP
- Delete a policy enforcement intermediary PEP

## Viewing a Policy Enforcement Intermediary

A PEP service view screen provides information about the PEP, such as alerts, as well as features for editing the PEP.

To view a PEP, follow these steps:

1. From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens. Alert and status information for each PEP is also listed.
2. Select the PEP you want to view. The view screen opens and contains a section for general information, a section for alerts, and a section that lists all policy enforcement intermediary resources that are contained in the PEP. The Contained Policy Enforcement Intermediary Instances section also lists the current version of the policy enforcement intermediary resource as well as the resource's management WSDL.

Some SOA PE features may not work as expected when using policy enforcement intermediary versions that are different than the SOA PE version. It is recommended that the SOA PE version and the policy enforcement intermediary versions match.

## Adding Resources

Any policy enforcement intermediary that is already registered with SOA PE can be added to a PEP. Typically, this procedure is used to add a policy enforcement intermediary in multiple PEPs or move a policy enforcement intermediary between PEPs. The latter is required when you delete a PEP.

To add a policy enforcement intermediary to a PEP, follow these steps:

1. From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.
2. From the list of PEPs, click the PEP you want to configure. The view screen opens for the selected PEP.
3. From the Contained Policy Enforcement Intermediary Instances section, click **Edit**. The Edit WS Intermediary Policy Enforcement Points screen opens.
4. From the list of intermediaries, click the **Contains** check box for each resource you want to add to this PEP. A check mark indicates that the policy enforcement intermediary is selected.
5. Click **Save**.

## Removing Resources

To remove a managed policy enforcement intermediary from a PEP, follow these steps:

1. From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.

2 From the list of PEPs, click the PEP you want to configure. The view screen opens for the selected PEP.

3 From the Contained Policy Enforcement Intermediary Instances section, click **Edit**. The Edit WS Intermediary Policy Enforcement Point screen opens.

4 From the list of policy enforcement intermediaries, click the **Contains** check box for each resource you want to remove from this PEP. An empty check box indicates that the resource is no longer selected.

5 Click **Save**.

## Enabling Availability Notifications

Availability notifications generate alerts for PEP whenever a managed policy enforcement intermediary that is contained in the PEP fails. This can be used to troubleshoot any problems that are encountered when managing Web services. Alerts are sent to an alert category that contains any number of alert recipients. For more information on setting up alert recipients and creating alert recipient categories, see chapter 5 “Using Alert Notifications”.

The PEP list indicates the alert status of all PEP groups (intermediary). The View PEP screen provides the details of the alert and also indicates which managed policy enforcement intermediary caused the alert.

It is good practice to enable availability notifications for a managed policy enforcement intermediary that is contained in a PEP. See the previous section “Enabling Availability Notifications” for more information.

To enable availability notifications for a PEP, follow these steps:

1 From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.

2 From the list of PEPs, click the PEP you want to view. The view screen opens for the selected PEP.

3 Click **Edit**. The Edit WS Intermediary Policy Enforcement Points screen opens.

4 From the Availability section, click the check box to enable availability notifications. A check indicates that availability notifications are enabled.

5 From the Alert Recipients section, use the drop-down list to select an alert category for both the Degraded and Unavailable availability status.

6 Click **Save**.

## Deleting a Policy Enforcement Intermediary PEP

You can delete a PEP at any time. When you delete a PEP, its alerts are removed. However, any managed policy enforcement intermediaries that are contained in the PEP are not removed from SOA PE and can be added to another PEP.

To delete a PEP, follow these steps:

1. From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.
2. From the list of PEPs, click the PEP you want to view. The view screen opens for the selected PEP.
3. Click **Remove**. A warning screen opens.
4. Click **Remove**. The Policy Enforcement Points Summary screen opens and the PEP is removed.

4

# SOA PE Administration

This chapter provides an overview of the features and functionalities provided by SOA PE. The topics covered are as follows:

- Technical Policies
- Setting Up the Audit Components
- Viewing Audit Information
- Service Provisioning
- Life Cycle Management
- Dashboard
- Configuring the Systinet Repository
- Viewing Reports

## Technical Policies

You can specify a set of requirements that Web services must adhere to when communicating with each other. This set of specific requirements is referred to as a technical policy.

By default, SOA PE includes a set of different supported policy types such as route, log, audit, event, and so on. A technical policy comprises a collection of assertions. A technical policy is associated with a service. Technical policy documents follow the WS- Policy standards.

### Bundled Policies

After installation, SOA PE provides a set of pre-configured policies (bundled policies) by default that belong to the policy types listed above. You can use these policies or modify them based on your requirements. To view the bundled policies, perform the following steps:

1 Log in to the SOA PE user interface as an administrator.

2 Click **Technical Policies** from the View drop-down menu. This displays the Technical Policies page with the list of bundled policies.

You can refer to the Description column adjacent to the policy name for a description for each listed policy. You can enforce policies in a SOA runtime environment by using Policy Enforcement Points (PEP).

## Policy Enforcement Point

You can enforce policies in a SOA runtime environment by using Policy Enforcement Points (PEP).

## Creating a New Technical Policy

You can create a new technical policy by specifying values for all the conditions (assertion parameters) that the policy must support. To create a technical policy follows these steps:

1 Log in to SOA PE user interface as an administrator.

2 From the **Actions** panel on the left pane, click **Add Policies**. The Technical Policies page opens.

3 Click **Add**. The Add New Technical Policy page opens.

4 Type the Name, Description, and Version in the respective boxes on the Add New Technical Policies page.

Make sure that you type meaningful names and description for the policies. This will help in easy identification of the policies.

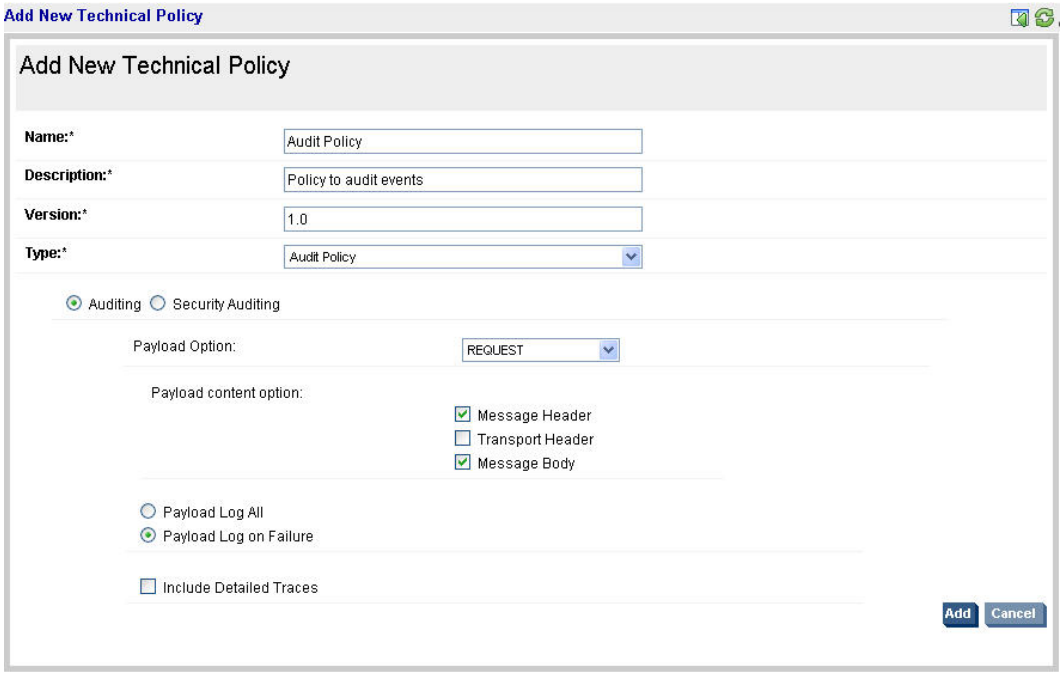
5 Based on the type of policy that you want to create, perform the steps listed in the following sections to create the specific policy. SOA PE saves the policy as Web service (WS) policy document.

- Create an audit policy
- Create a service protection policy
- Create a scheduled availability policy
- Create an event policy
- Create a content detection policy
- Create a WS-Addressing policy
- Create a schema validate policy
- Create a transform policy
- Create a JMS mediation policy
- Create a message security policy
- Create a transport security policy

## Creating an Audit Policy

You can use an audit policy to audit traces related to an event. To create an audit policy, perform step 1 through step 5 in the *Creating a New Technical Policy* section and then follow these steps:

1 Select Audit Policy from the **Type** drop-down list. This displays the parameters that you can specify for an audit policy as shown in the following figure.




2 Select **Auditing** if you want audit traces to be recorded in SOA PE or select **Security Auditing** if you want to log audit traces to security validator.

**Note:** While Security Auditing is turned on, security audit log is created under <install_dir>/conf/broker/securityaudit.log when default or ldap security providers is configured in SOA PE broker.

To modify the security audit configuration, modify the SECURITY_AUDIT_FILE section present in <install_dir>/conf/broker/logging.properties file.

3 Select **NONE**, **REQUEST**, **RESPONSE**, or **REQUEST-RESPONSE** from the **Payload Option:** drop-down list. This option signifies whether the payload for the message must be collected for the request message, response message, or both the request and response messages. A payload signifies the data in a message. If you select **NONE**, SOA PE does not display the payload content option discussed in step 4 of this procedure.

4 From the **Payload content option**, choose one or all of the following options to specify the information that must be traced:

— Message Header: This option specifies that the message header for the payload must be audited. This option is available for a SOAP service only.

- Transport Header: This option specifies that the transport header for the payload must be logged. In the case of HTTP this option corresponds to HTTP headers and when JMS is used as transport, this option corresponds to JMS propertied in the JMS message.
- Message body: This option specifies that the body of the message must be traced and logged. In the case of XML service, SOA PE audits the complete message. In the case of a SOAP message SOA PE audits only the SOAP body in the SOAP envelope.

5 Select the **Payload Log All** option to collect payload for all messages (successful messages and failed messages) or select the **Payload Log on Failure** option to collect payload only for the failed messages.

6 Select **Include Detailed Traces** to enable all tracing details.

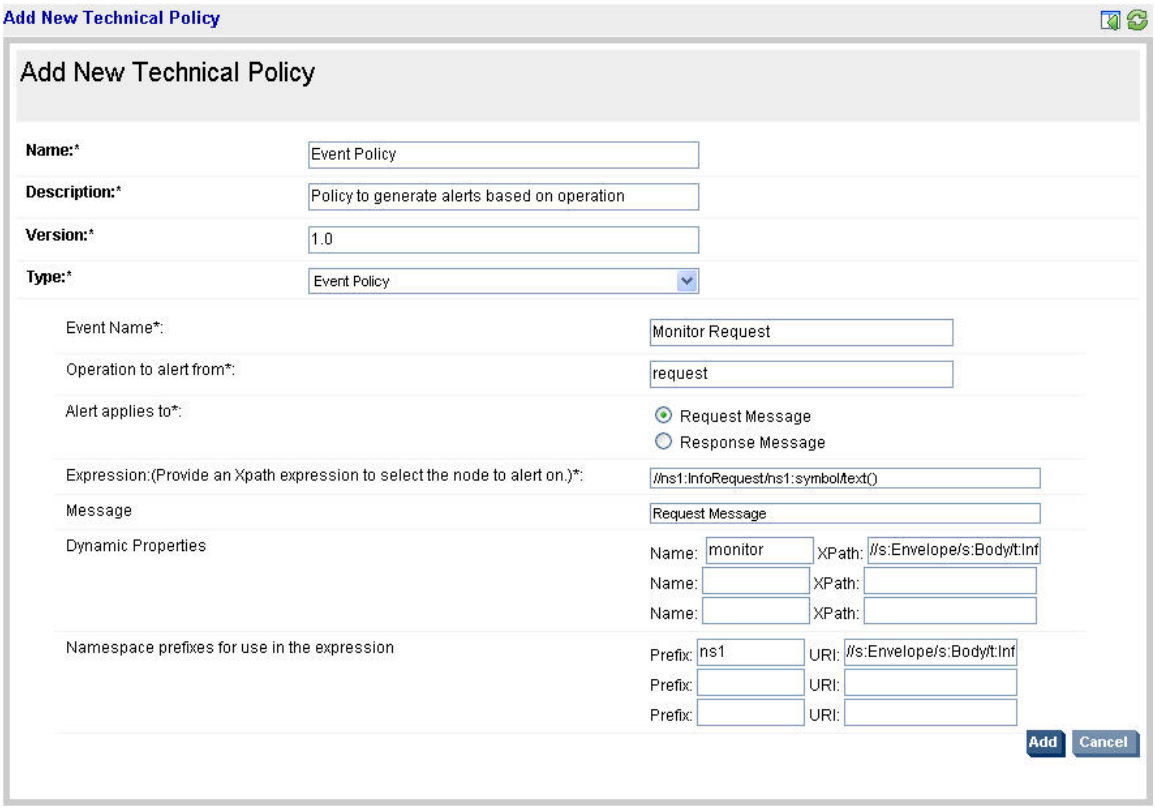
7 Click **Add** to complete the creation of the new audit policy.

You can view the audit message traces report to view the traced audit information according to your specifications in the audit policy. Refer to section Audit Message Traces Reports for more information about viewing the report.

## Creating an Event Policy

You can use an event policy to generate an alert based on the performance of an operation; for example, a business content alert. To create an event policy, perform step 1 through step 5 in the **Creating a New Technical Policy** section and then follow these steps:

8 Select **Event Policy** from the **Type** drop-down list. This displays the parameters that you can specify for an event policy.




9 Type the name of the event in the **Event Name** box.

10 Type the name of the operation that contains the business service you want to monitor in the **Operation to alert from** box.

11 Select **Request Message** or **Response Message** from the **Alert Applies to** section. This signifies that the generated alerts by the operation are applicable either to the request messages or the response messages.

12 Type the XPath expression in the **XPathExpression** box. By using this expression, you can specify the business content to be extracted from the operation. For example, if you provide the XPath expression, //ns1:InfoRequest/ns1:symbol/text(), this expression scans the SOAP message for the InfoRequest node and extracts the business content for the node by the name symbol, present under the InfoRequest node.

13 Type a message that you want the alert to display in the **Message** box.

14 Type variable names for the event name and XPath expression in the **Name** and **XPath** boxes, located in the **Dynamic Properties** section. For example, you can specify the name as follows: Name: monitor, where monitor is the variable defined for the event name. You can specify the Xpath expression variable as follows //s:Envelope/s:Body/t:InfoRequest/t:symbol/text(). You can use these variables in the **Message:(MessageHelp)** when you create messages that must either include the event name or the XPath expression. XPath expression specified here is evaluated on the business content selected by the XPath expression specified in Step 5.

15 Type a namespace prefix that is included with the XPath expression you typed, in the **Namespace prefixes for use in this expression** box. For example, in step 5 of this procedure, in the sample XPath expression, the namespace prefix is ns1.

16 Click **Add** to complete the creation of the new event policy.

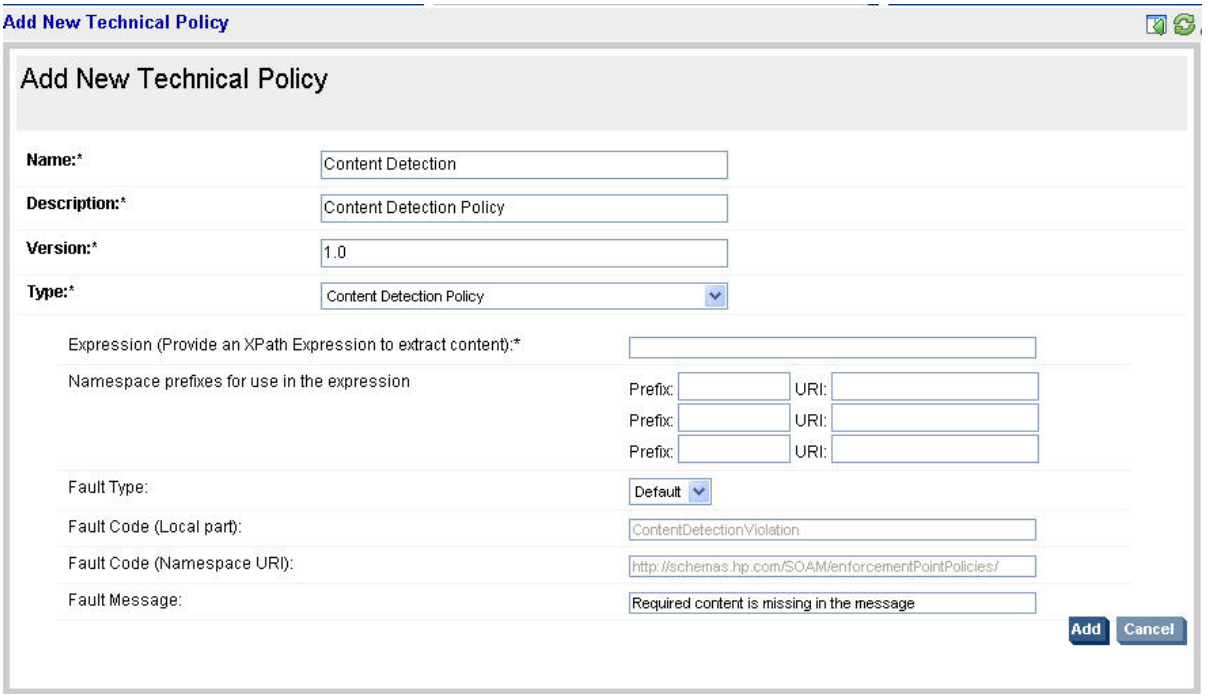
## Creating a Content Detection Policy

You can use the content detection policy to verify the presence of content in a message at the Intermediary. You can extract the header or the body of the message using this policy. This policy allows you to specify an XPath expression to extract the content from the message that you want to detect. Based on the presence or absence of the XPath expression in the extracted content, the Intermediary either rejects the message from the client and returns a fault code to the client or forwards the message to the endpoint.

The audit handler logs all messages and fault information to the audit logs. SOA PE displays the success or failure status for the presence or absence of the extracted content in the message on the SOA PE dashboard.

To create a content detection policy, perform step 1 through step 5 in the **Creating a New Technical Policy** section and then follow these steps:

1 Select **Content Detection Policy** from the **Type** drop-down list. This displays the parameters that you can specify for this policy as shown below.




2 Type an XPath expression in the **Expression box**, for example, //ns1:InfoRequest/ns1:symbol/text(). this expression scans the SOAP message for the InfoRequest node and extracts the business content for the node by the name symbol, present under the InfoRequest node.

3 Type a namespace prefix that is included with the XPath expression you typed, in the **Namespace prefixes for use in this expression** box.

4 Select **Default** or **Other** from the **Fault Type** drop-down list. If you select **Default**, SOA PE displays the values for the Fault Code (Local part) and the Fault Code (Namespace URI) fields. If you select **Other**, you can specify the values of your choice for the Fault Code fields listed below:

- Fault Code (Local Part): you can specify a fault code identifier of your choice in this field.
- Fault Code (Namespace URI): you can specify the URI to the schema that you want to use if a fault code is generated.

5 Type a message in the **Fault Message** box that you want the SOA PE Dashboard to display if the content detection fails.

6 Click **Add** to complete the creation of the new Content Detection policy.

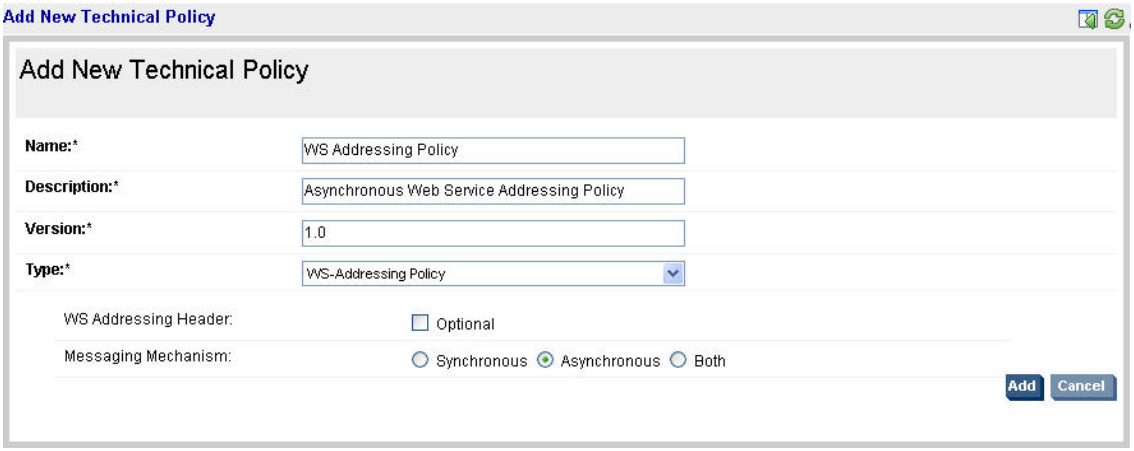
## Create a WS-Addressing Policy

The concept of WS-Addressing provides a transport-independent (protocol independent) mechanism to allow Web Services to communicate addressing information. This concept allows a standard way to route messages across multiple transport mechanisms. WS-Addressing allows you to achieve both synchronous (using anonymous addressing) and asynchronous (using non anonymous addressing) message addressing.

In an asynchronous addressing scenario, you can specify the Web Service to which a request must be sent. You can configure different recipients at the service consumer for receiving response messages. You can also configure recipients at the service consumer side to receive fault messages in the event of an error. Refer to the *WS Addressing – An Overview* chapter for more information about WS-Addressing.

To create a WS-Addressing policy, perform step 1 through step 5 in the **Creating a New Technical Policy** section and then follow these steps:

1 Select **WS-Addressing Policy** from the **Type** drop-down list. This displays the parameters that you can specify for this policy as shown below.




2 Select **Optional** from the **WS Addressing Header** option to specify if the presence of the WS Addressing header in the message is optional or mandatory.

3 Select one of the following options from the **Message Mechanism** option:

- **Synchronous**: Select this option to specify that the policy must support anonymous message addressing.
- **Asynchronous**: Select this option to specify that the policy must support non anonymous message addressing.

— **Both**: Select this option to specify that the policy must support both anonymous and non anonymous message addressing.

4 Click **Add** to complete the creation of the WS-Addressing policy. You can see the policy on the Technical Policies page.

## Creating a Schema Validate Policy

You can use the schema validate policy to validate a schema included in the body of an incoming SOAP request message. To create a schema validate policy, perform step1 through step 5 in the **Creating a New Technical Policy** section and then follow these steps:

1 Select **Schema Validate Policy** from the **Type** drop-down list.

2 Click **Add** to complete the creation of the Schema Validate policy.

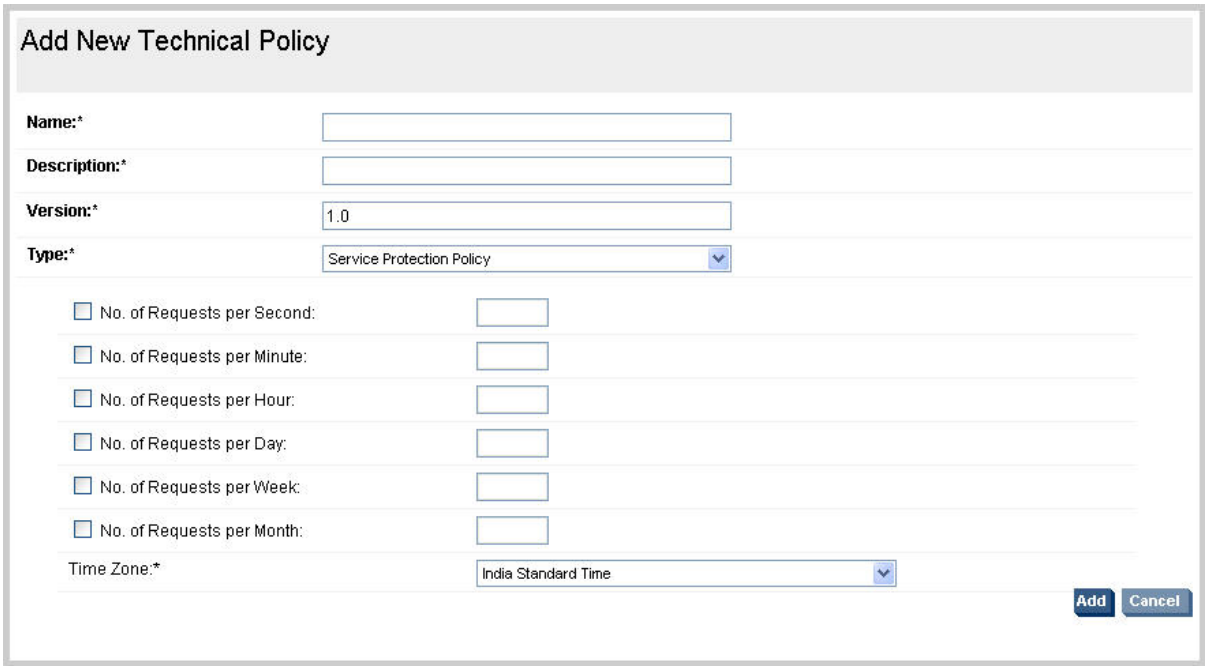
## Creating a Service Protection Policy

You can use a service protection policy to limit access to endpoints being managed using a policy enforcement intermediary. You can use this type of a policy to specify the number of service requests that an intermediary can accept. After the limit specified for the number of service requests that the intermediary can accept is exceeded, SOA PE rejects the subsequent service request messages by sending a SOAP fault which prevents the managed endpoint from crashing or denying service requests. For example, you can specify the number of requests that a managed endpoint can accept in a day, a week, or in a month.

SOA PE generates a SOAP fault for each service request rejected and updates the SOA PE Dashboard with the SOAP fault and the corresponding throughput information. For an XML service, SOA PE returns HTTP response with status code set as 500.

To create a service protection policy for enforcing access restrictions to the intermediary, perform step 1 through step 5 in the *Creating a New Technical Policy* section and then follow these steps:

1 Select **Service Protection Policy** from the **Type** drop-down list as shown in the following screen shot.




2 Select one or all of the following options based on your requirements and specify the number of requests in the box adjacent to each option you select:

- Requests per Second
- Requests per Minute
- Requests per Hour
- Requests per Day
- Requests per Week
- Requests per Month

SOA PE maintains the count only for Daily, Weekly, and Monthly attributes in the event of a restart of the intermediary.

3 Select the time zone to which this policy must be associated with from the **Time Zone:** drop-down list.

SOA PE maintains the time zone information only for the Day, Week, and Month attributes.

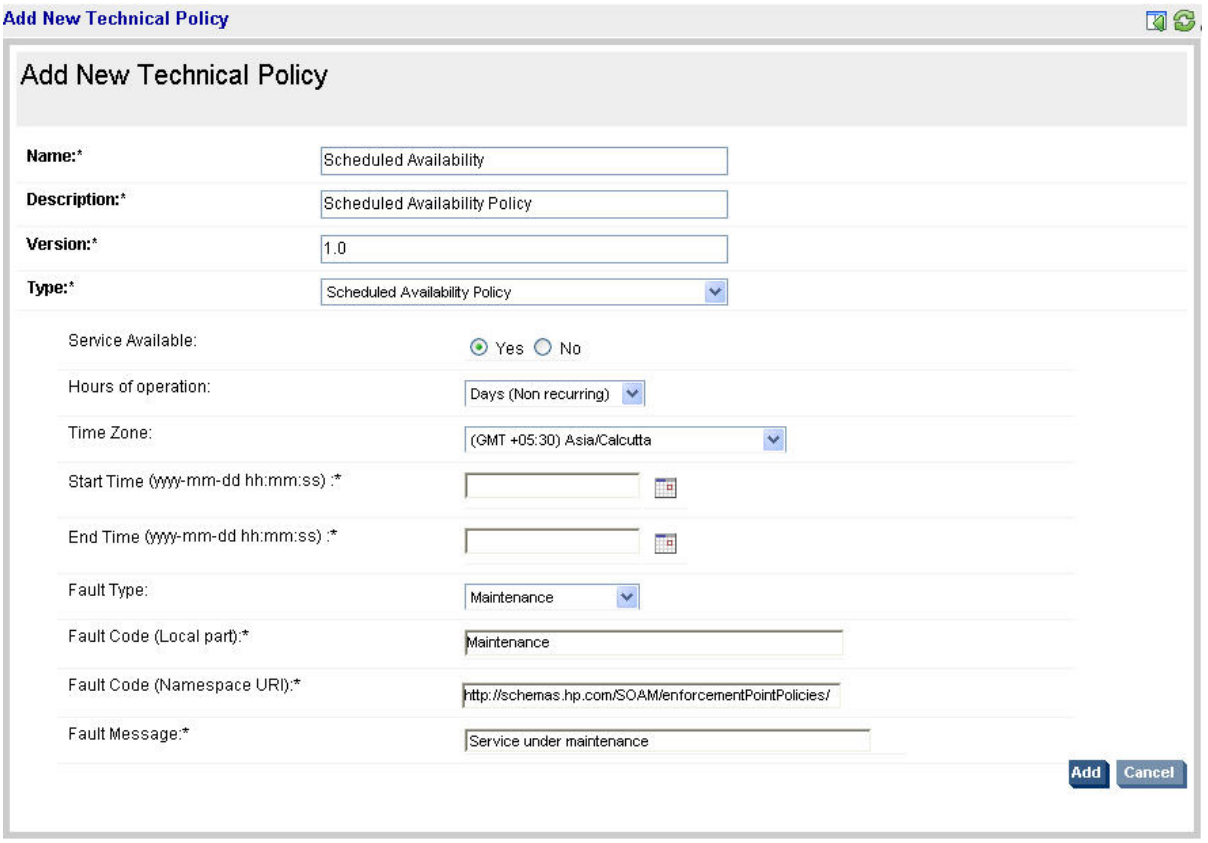
4 Click **Add** to complete the creation of the Service Protection policy.

## Creating a Scheduled Availability Policy

You can use a scheduled availability policy to allow or deny access to a service based on the scheduled availability time period specified for that service. The Intermediary uses this policy to verify the availability of a service at that specific time. If the service is specified to be available, the Intermediary forwards the message from the client to the endpoint. If the service is specified to be unavailable, the Intermediary rejects the message and sends a SOAP fault to the client. The audit handler logs all messages and fault information to the audit logs. SOA PE displays the status of the service requests received by the Intermediary on the SOA PE dashboard.

To create a scheduled availability policy, perform step1 through step 5 in the **Creating a New Technical Policy** section and then follow these steps:

1 Select **Scheduled Availability Policy** from the **Type** drop-down list. This displays the parameters that you can specify for this policy as shown below.




2 Select **Yes** (to specify that the service must be available) or **No** (to specify that the service must be unavailable) from the **Service Available** option.

3 Select one of the following options from the **Hours of operation** drop-down list:

- **Days (Non recurring)**: specifies that the policy is applicable on a non recurring basis on all days.
- **Workdays (recurring)**: specifies that the policy is applicable on a recurring basis on workdays.
- **Weekends (recurring)**: specifies that the policy is applicable on a recurring basis on weekends.

4 Select the time zone from the **Time Zone** drop-down list.

5 Specify the **Start Time** and the **End Time** for the service availability or unavailability in the respective boxes. You must specify the time in the mm/dd/yyyy hh:mm:ss (month/day/year hour:minute:second) format. You can alternatively click the (calendar) icon below the respective boxes to display the calendar where you can specify the start time or the end time.

6 Select one of the following options from the **Fault Type** drop-down list to specify the reason for the non availability of the service:

- **Maintenance:** to specify that the service is under maintenance
- **Scheduled downtime**: to specify that the service is under scheduled downtime
- **Backup**: to specify that the service is undergoing a backup procedure
- **Other**: to specify any other reason other than the reasons listed above for the unavailability of the service

7 Type the fault code identifier of your choice in the **Fault Code (Local Part)** box.

8 Type the URI to the schema you want to use if a fault code is generated in the **Fault Code (Namespace URI)** box.

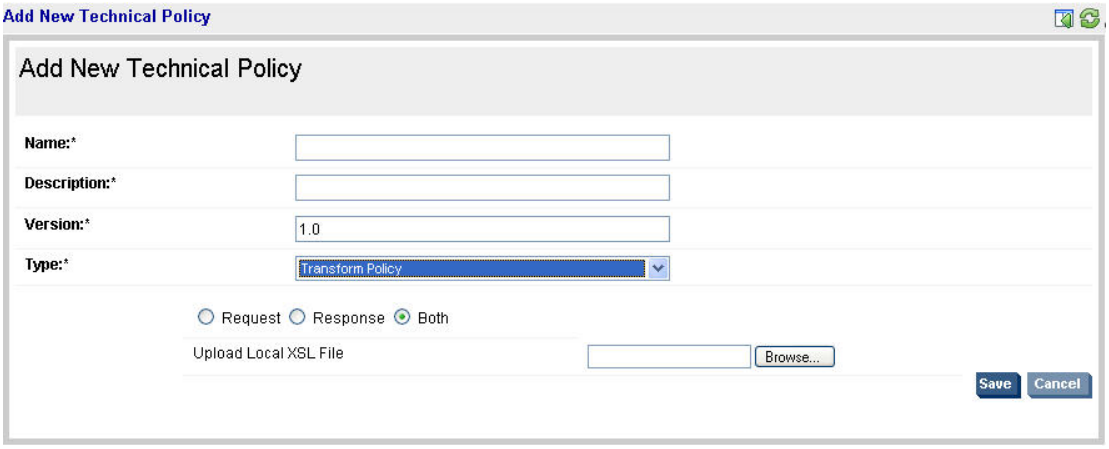
9 Type a message in the **Fault Message** box that you want the SOA PE Dashboard to display if a message is rejected by the Intermediary.

After you attach a scheduled availability policy to a service during service provisioning, you can monitor the Web service status to verify if the scheduled availability policy rejects messages that are sent by clients during a non availability period specified for the service.

## Creating a Transform Policy

You can use a transform policy to transform request messages or response messages based on the request or response template that you specify. To create a new transform policy, perform step1 through step 5 in the **Creating a New Technical Policy** section and then follow these steps:

1 Select **Transform Policy** from the **Type** drop-down list.




2 Select one of the following to specify the type of message on which the policy must be applied:

- **Request**- Indicates that the policy must be applied on request messages sent to the intermediary.
- **Response**- Indicates that the policy must be applied to response messages from the intermediary.
- **Both**- Indicates that the policy must be applied to both request and response messages.

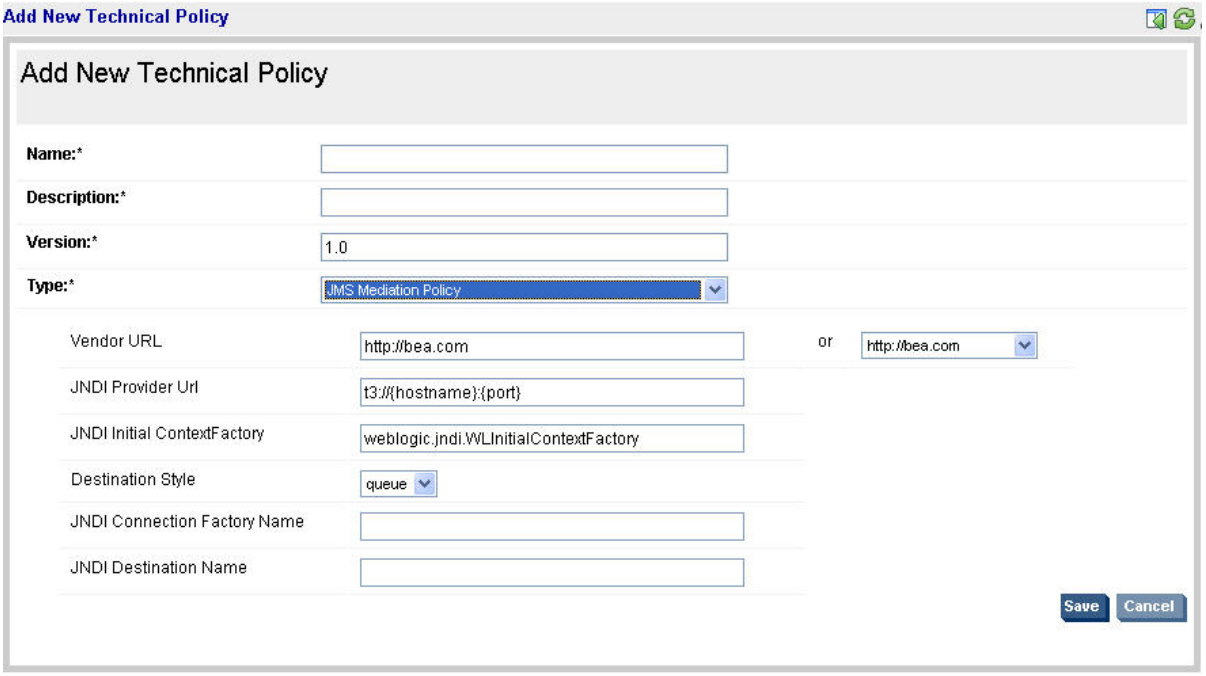
3 Type the path to the message transformation template file in the **Upload Local XSL File** box or click **Browse** to select the transformation template.

4 Perform *step 6* in the *Creating a New Technical Policy* section.

## Creating a JMS Mediation Policy

You can use a JMS mediation policy to support SOAP/XML over JMS at inbound. To create a JMS mediation policy, perform step 1 through step 5 in the *Create a New Technical Policy* section and then follow these steps:

1 Select **JMS Mediation Policy** from the **Type** drop-down list.




2 Specify the URL of the vendor who provides JNDI in the **Vendor URL** box or select a vendor from the drop-down box adjacent to the Vendor URL box.

3 Specify the URL of the JNDI server in the **JNDI Provider Url** box. For example when the vendor used is weblogic the URL is in the following format: *t3://{hostname}:{port}*

4 Select **queue** from the **Destination Style** drop-down list. This specifies the destination type for the JMS transport model.

5 Type the JNDI lookup name for the connection factory in the **JNDI Connection Factory Name** box.

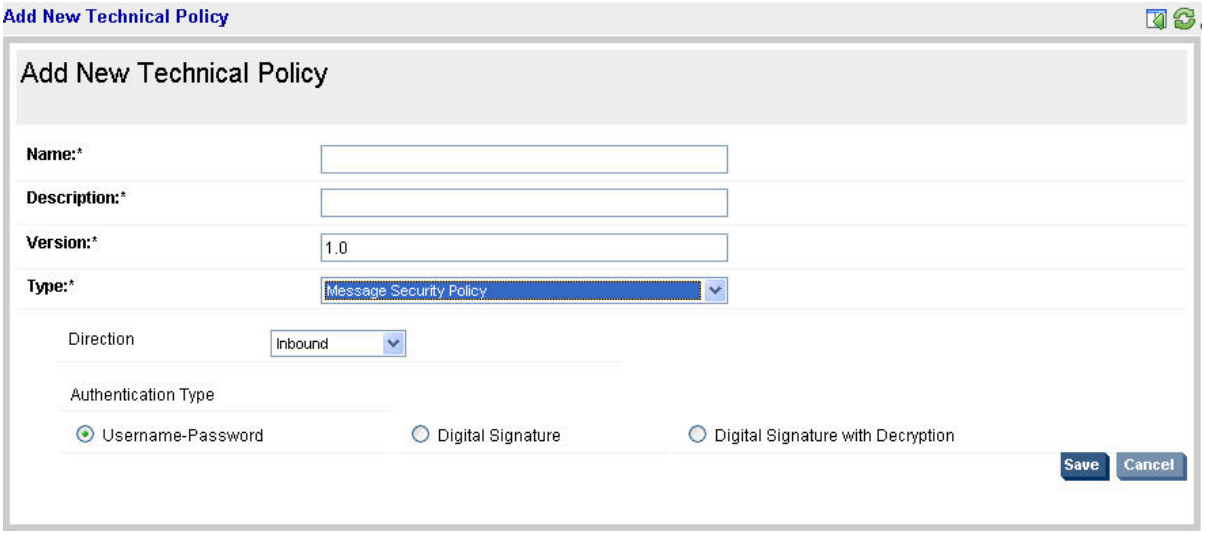
6 Type the JNDI lookup name for the destination in the **JNDI Destination Name** box.

7 Perform *step 6* in the *Creating a New Technical Policy* section.

## Creating a Message Security Policy

You can make inbound and outbound messages secure by using a message security policy. To create a message security policy, perform step 1 through step 5 in the **Creating a New Technical Policy** section and then follow these steps:

1 Select **Message Security Policy** from the **Type** drop-down list.




2 Select **Inbound** or **Outbound** from the **Direction** drop-down list. This specifies if the message security policy must be implemented on inbound or outbound messages.

3 Select one of the following to specify the type of authentication you prefer to implement for the message security policy:

- **Username-Password Authentication**- This option indicates that the WS-Security username token profile to be used. The intermediary performs authentication using the security provider configured or the default security provider in SOA PE. You are prompted for a user name and password if you selected **Outbound** in step 2 of this procedure.
- **Digital Signature Authentication**- This option uses WS-Security X.509 certificate (public-key certificate)-based digital signature for authentication.
- **Digital Signature with Decryption**- This option uses an X.509 certificate-based digital signature along XML encryption

4 For outbound messages, you can specify the alias corresponding to the end point in the **Endpoint Server Certificate Alias** box. The intermediary encrypts the outbound message by using the X.509 certificate corresponding to the end point server alias you specified in this step.

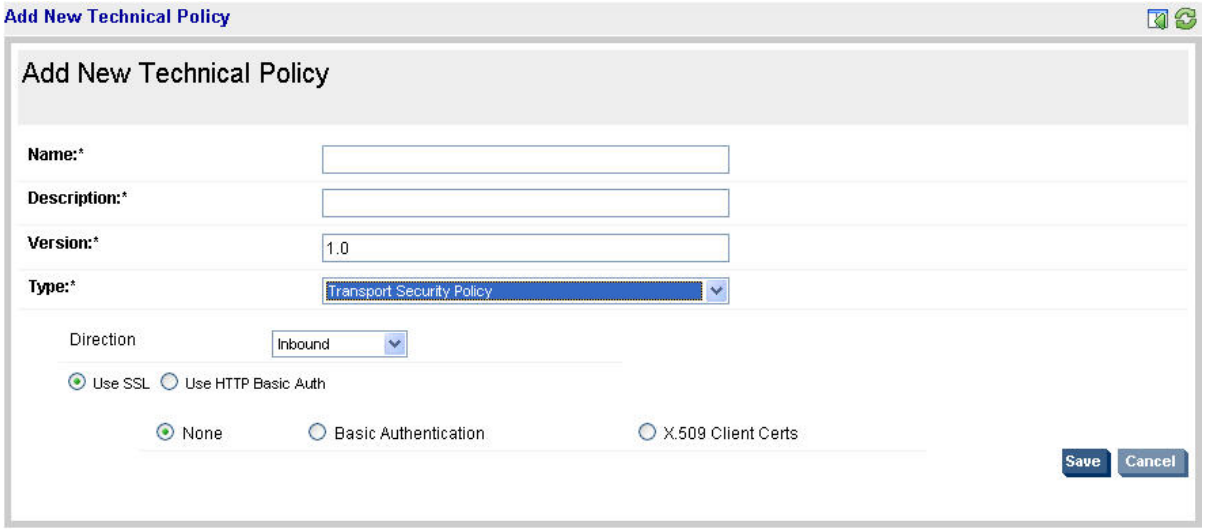
5 Select **No Digital Signature or Encryption in Response** to specify that the response messages from the intermediary must not be encrypted and must not contain a digital signature when it is sent.

6 Perform *step 6* in the *Creating a New Technical Policy* section.

## Creating a Transport Security Policy

You can use a transport security policy to implement security at the transport level for a message. This policy is not applicable when the transport used is JMS.To create a transport security policy, perform step 1 through step 5 in the *Creating a New Technical Policy* section and then follow these steps:

1 Select **Transport Security Policy** from the **Type** drop-down list.




2 Select **Inbound** or **Outbound** from the **Direction** drop-down list. This specifies if the policy must be implemented on incoming communication or outgoing communication.

3 For inbound communication, perform steps a through c. If you selected outbound communication, proceed to step 4:

   a Select one of the following to specify the protocol you want to implement:

      – **Use SSL**- Uses the Secure Socket Layer (SSL) protocol to implement secure communication. This protocol is selected by default.

      – **Use HTTP Basic Auth**- Uses the HTTP Basic Auth for authentication

4 If you select SSL as the protocol for secure communication, you must specify the details as follows:

   — **None**- Uses SSL protocol without any client authentication.

   — **Basic Authentication**- Indicates Basic auth to be used for authentication.

   — **X.509 Certs**- Indicates cert based authentication to be used.

5 Select **Authentication Only** to specify that the policy must be used only for authentication and not authorization.

6 When the direction selected is inbound and if you select **Basic Authentication**, **X.509 Client Certs while using SSL**, or use HTTP Basic Auth, you must specify the following if an alert needs to be generated on authentication failure:

   — **Enable Alert**- You can select this option to enable alerts.

— **Authentication Only**- You can select this option to specify that the policy must be used only for authentication and not authorization.

7 When the direction selected is outbound type the **Username** and **Password** in the respective boxes under **Basic Auth Parameters**. Make sure that the typed user name and password are configured in the security provider or in the default security provider for SOA PE.

8 Perform step 6 in the Creating a New Technical Policy section.

Security Alerts are generated only when the transport security fails.

## Modifying a Technical Policy

To modify a technical policy, follow these steps:

1 From the **View** drop-down menu, click **Technical Policies**. This lists the available policiesTechnical Policies page.

2 Click the policy that you want to modify. This displays the policy and its details.

3 Click **Edit** to modify the policy. This displays the Edit Technical Policy page.

4 Make the changes that you require and click **Save** to save the modified policy. This displays the WebService List page, which lists the services that uses this policy.

5 Click **Save** to confirm that the listed Web service must be redeployed. This saves the modified policy.

## Deleting a Technical Policy

To delete a technical policy, follow these steps:

1 From the **View** drop-down menu, click **Technical Policies**. This lists the available policies in the Technical Policies page.

2 Click the policy that you want to delete. This displays the policy and its details.

3 Click **Remove** to delete the policy. This displays the WebService List page, which lists the services that uses this policy

4 Click **Save** to confirm the listed Web service must be redeployed. This deletes the policy from the service model.

## Exporting Technical Policies

You can use the export technical policies feature to group all existing policies into a single archive file. You can import these policies to another computer where SOA PE is running by using the import policy feature discussed in the next section.

To export a technical policy, perform the following steps:

1 Log into SOA PE user interface as an administrator.

2 Click **Technical Policies** from the **View** drop-down menu. This displays the Technical Policies page.

3 Click **Export**. This displays the Export All Technical Policy page.

4 Click **Download** to group the policies into a single archive file. You can specify the location where you want to save the archived policies file.

### Importing Technical Policies

You can use the import policy feature to import policies from a local computer or a remote computer to the computer on which you have installed SOA PE. To import policies, you must perform the following steps:

1 Log into SOA PE user interface as an administrator.

2 Click **Technical Policies** from the **View** drop-down menu. This displays the Technical Policies page.

3 Click **Import**. This displays the Import Technical Policies page.

4 Select **Ignore Routing and Loadbalancing policies** if you do not want to import routing and load balancing policies.

5 Click **Browse** to select the location of the .jar file that contains the technical policies that you want to import if the policies exist on the local computer. If the policies exist on a remote computer, you can specify the URL to the .jar file in the **Specify Remote Technical Policy Jar URL:** box.

6 Click **Import**. This imports the specified policies.

## Setting Up the Audit Components

The components of the audit feature must be set up before message trace information is collected and stored in the database and viewed using the SOA PE web interface. To set up the auditing components you must perform the steps in the following section:

- Enable the Audit Policy. Refer to the section “Creating a New Technical policy”
- Configure the Audit Publisher
- Configure the Database

### Configure the Audit Publisher

The policy enforcement intermediary contains an audit publisher that is responsible for sending trace information to the SOA PE’s audit service.

There are two properties you can configure for the audit publisher. The properties define the number of trace messages (bucket size) to send to the audit service and the interval (in milliseconds) to wait before sending trace messages. Trace messages are published based on whichever value is reached first.

A small bucket size or interval means trace messages are published very often and may produce unwanted overhead that affects performance. A large bucket size or interval means trace messages will not be available for a long time and could hinder you from detecting and correcting problems or security violations. These properties should be set according to your business and application requirements.

### Policy Enforcement Intermediary

To configure the audit publisher for the policy enforcement intermediary, follow these steps:

1. Stop the policy enforcement intermediary if it is currently started.
2. Use a text editor to open *<install_dir>*`\conf\broker\mipServer.xml`.
3. Edit the audit publisher entries for `interval` and `threshold`. The `interval` value is in milliseconds and the `threshold` value is the total number of trace messages:

```
<entry name="com.hp.audit.publisher.interval">100000</entry>
<entry name="com.hp.audit.publisher.threshold">10</entry>
```

4. Save and close the file.
5. Restart the policy enforcement intermediary for the changes to take effect.

## Configure the Database

Message trace information is sent to the SOA PE audit service and stored in a database. SOA PE includes an embedded instance of the HSQL database (http://hsqldb.sourceforge.net/) that is enabled by default. This database can be used for testing. However, for production environments, a database schema for creating the data tables in Oracle 10g is provided. See the Oracle 10g documentation if you are not familiar with creating data tables using a schema file. As with all databases, you must monitor the database and periodically do maintenance. For the auditing feature, the number of trace messages will continue to grow in size. You should periodically retire old data before it becomes unmanageable.

### Configuring the HSQL Database

The default installation of SOA PE is configured to use the embedded HSQL database. This is reflected in: *<install_dir>*`\conf\networkservices\mipServer.xml`.

```
<entry name="com.hp.db.demo">true</entry>
```

Once this entry is set to `demo=true` remaining values related to JDBC URL, user name, and so on are ignored and will use the following default values. These default values are not reflected in the xml file but are hard-coded in SOA PE:

```
<entry name="com.hp.db.driver">org.hsqldb.jdbcDriver</entry>
<entry name="com.hp.db.url">
   jdbc:hsqldb:E:\<install_dir>\data\sn</entry>
<entry name="com.hp.db.user">sa</entry>
<entry name="com.hp.db.password"></entry>
```

HSQL comes with a swing-based GUI Database Manager that can be used to view trace information in the Audit tables and perform routine maintenance. The class for starting the database manager is located in the <*install_dir*>/lib/ext/hsqldb.jar. The full class name is org.hsqldb.util.DatabaseManager and can be started from the command line.

### Configuring an Oracle 10g Database

A schema for creating the audit tables in Oracle 10g is located at <*install_dir*>\data\oracle\ Create-Tables-Oracle.SQL. After you create the database and create the schema, configure SOA PE to use the database.

You must copy the Oracle 10g version of the Oracle thin JDBC driver (oracle_ojdbc14.jar and oracle_nls_charset12.jar) into the <*install_dir*>/lib directory.

To configure SOA PE to use the Oracle 10g database, follow these steps:

1 Stop SOA PE if it is currently started.

2 Open <*install_dir*>\conf\networkservices\mipServer.xml.

3 Add your database information in the DB Properties section. For example:

```
<entry name="com.hp.db.demo">false</entry>

<!-- The demo entry must be set to false. -->

<entry name="com.hp.db.driver">
   oracle.jdbc.driver.OracleDriver</entry>
<entry name="com.hp.db.url">
   jdbc:oracle:thin:@host:1521:DB1</entry>
<entry name="com.hp.db.user">admin</entry>
<entry name="com.hp.db.password">admin</entry>
```

4 Save and close the file.

5 Restart SOA PE.

# Viewing Audit Information

The SOA PE web interface lets you query trace messages that are stored in the SOA PE database. You can query successful messages and failed messages. For each trace message, you can see detailed trace information.

To view audit information, follow these steps:

1 Click **Business Services** under the **View** drop-down menu to view the Business Services List screen.

2 From the Business Services List screen, expand a business service to view its contained configurations and Web services configurations.

3 Click the Web service configuration you want to view or expand the configuration and click a specific operation. The appropriate view screen opens.

4 From the 1 hour summary table, click the success value (to query trace messages for successful requests) or failure value (to query trace messages for failed requests). The View Failures or View Successes screen opens depending on the value selected.

5 In the Query section, configure the following query fields:

— **Search For:** Select the **Success** or the **Failure** check boxes.

— **Service:** Use the drop-down lists to constrain the query by business service, PEP or policy enforcement intermediary service.

— **Start Date:** Use the fields to enter a specific start date and start time for the query to match.

— **End Date:** Use the fields to enter a specific end date and end time for the query to match.

— **User:** Enter a user in the field if you want query the trace messages based on a specific authenticated security principal (authenticated user) that made the request.

6 Click **Query**. The results of the query are listed in the Results section.

7 Click a trace message's Timestamp to view trace information details as well as Profile Data.

# Service Provisioning

The service provisioning feature in SOA PE helps you define a Web service (SOAP or XML), associate technical policies, end points, and a Policy Enforcement Point (PEP) intermediary to the Web service. This feature also lets you associate the Web service with an existing business service or create a new business service. You can also deploy (provision) the defined Web service to a PEP by using this wizard-based feature. You can provision a Web service in one of the following scenarios:

- Define a Web service in SOA PE and publish the service to the registry
- Provision the discovered services from Systinet registry, after updating the service using SOA PE, and publish the service back to the registry.

The wizard guides you through a seven-step procedure involved in provisioning a service as follows:

1 Specify implementation service details and PEP types

2 Associate technical policies

3 Specify service details

4 Specify end point related configuration for load balancing and routing

5 Associate the Web service with a business service

6 Specify the provisioning option

## Prerequisites

To use the service provisioning feature, you must make sure that the PEP Intermediary is running and registered with SOA PE.

For provisioning services from Systinet registry, you must make sure that you have configured the Systinet settings for SOA PE to discover Systinet services.

### Configuring Systinet

A Systinet Repository must be configured with SOA PE before you can use the SOA PE web interface to publish SOA PE's assets to the repository. You must have administrative rights to configure the Systinet Repository.

### To configure a Systinet Registry:

1. From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.
2. Click the **Systinet Settings** tab.
3. Enter the Registry settings:
   - **Username**: A user's name used to access the registry.
   - **Password**: The user's password.
   - **Inquiry URL**: Specify the URL that is used to connect to the registry and make inquiries. For example, `http://<host>:<port>/uddi/inquiry`.
   - **Publish URL**: Specify the URL that is used to connect and publish to the registry. For example, `http://<host>:<port>/uddi/publishing`.
   - **Security URL**: Specify the URL that is used to connect to the registry and retrieve the authentication token. For example, `http://<host>:<port>/uddi/security`
   - **Subscription URL**: Specify the URL that is used to connect to the registry and subscribe to Systinet notifications. For example, http://<host>:<port>/uddi/subscription
   - **Subscription Notification Interval**: Specify the interval (in milliseconds) at which the notifications must be fetched from the registry. For example, if you want to configure the notification interval as 1 minute, set this value as 60000.
   - **Subscription Validity Period**: Specify the validity (in days) to be set for the subscription when you register for notification.

### To configure the Systinet Platform

4. Enter the Platform Settings:
   - **URL**: Specify the URL to the Systinet Repository. For example, http://<hostname>:<port>/soa/systinet/platform/
   - **Use Basic Auth**: Select this option if you want to use basic authentication when users connect to the Systinet Repository. Selecting this option prompts you to enter the **Username** and **Password** that must be used for basic authentication.

5 Click **S**ave.

6 Restart SOA PE.

If the registry is configured to use SSL, you can do as follows to add the registry certificate as a trusted certificate:

Copy the file clientconf.xml from <systinet_install_dir>\Registry\conf\ to <soape_install_dir>\lib\systinet\conf\clientconf.xml.

<systinet_install_dir> signifies the directory in which you have installed HP Systinet and <soape_install_dir> signifies the directory in which you have installed SOA PE.

Systinet Repository settings can also be manually entered in *<install_dir>*`\conf\networkservices\mipServer.xml`.

## Specify Implementation Service Details and PEP Types

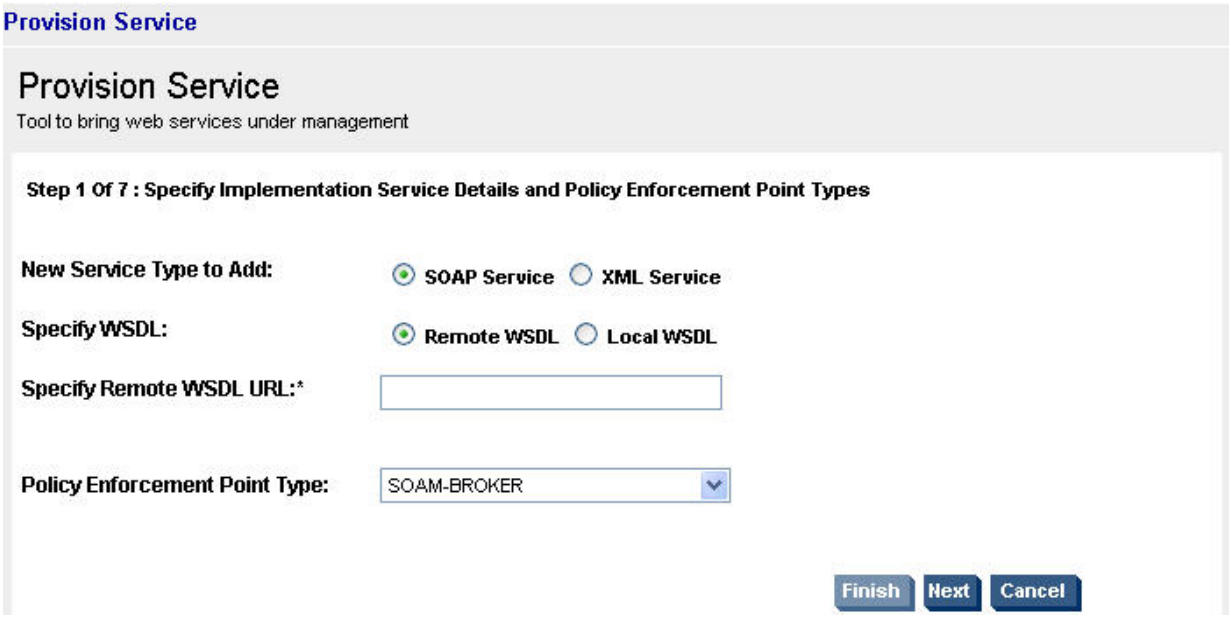
1 Log in to SOA PE web interface as an administrator

2 Click **Provision Service** from the **Actions** drop-down menu on the left pane. This displays the Specify Implementation Service Details and Policy Enforcement Point Types screen of the Service Provisioning Wizard as the following graphic shows.

If you are provisioning Web services from Systinet using SOA PE, the Lifecycle status page displays the services discovered from Systinet. The source of the session displays SOA Systinet and state of the session could be one of the following:

- Partially configured: indicates that the service is not yet provisioned in SOA PE
- Modified: indicates that the service that is already provisioned in SOA PE is modified in registry.
- Marked for delete: indicates that the service provisioned in SOA PE is deleted from the registry.

If the Web service state displays that it is marked for provisioning, you can select the service that you want to provision and click **Provision** to start provisioning the Web service. For more information about the Lifecycle Status page, refer to the Lifecycle Management section of this chapter.




3 Select **SOAP Service** or **XML Service** from the **New Service Type to Add** option.

4 If you selected SOAP Service, proceed to step 6 in this procedure.

5 For an XML service, you must provide the following additional details:

   a Type the XML service name in the **Service Name** box.

   b Type the namespace for the XML service in the **Namespace** box.

   c Select the transport protocol that must be used from the Transport Protocol option. You can select either JMS or HTTP. HTTP is the protocol enabled by default.

   d If you selected HTTP as the protocol, proceed to step 7 in this procedure.

   e If you select JMS as the protocol, you must specify additional information such as URL of the JNDI security provider, the destination style and so on. Refer to steps 2 through 6 in the section *Creating a JMS Mediation Policy* for more information about the details that you must provide when using JMS as the protocol.

   f Proceed to step 8 in this procedure.

6 Select either **Remote WSDL** or **Local WSDL** option to specify the **WSDL URL**. You can click **Browse** to specify the location of a local WSDL URL. For a web service from Systinet, SOA PE retrieves all the information regarding the service from the Systinet registry, except for the Policy Enforcement Point Type, which you can specify.

You cannot modify the existing details of a SOAP Web service from Systinet. For an XML Web service, you can modify the namespace for the Web service.

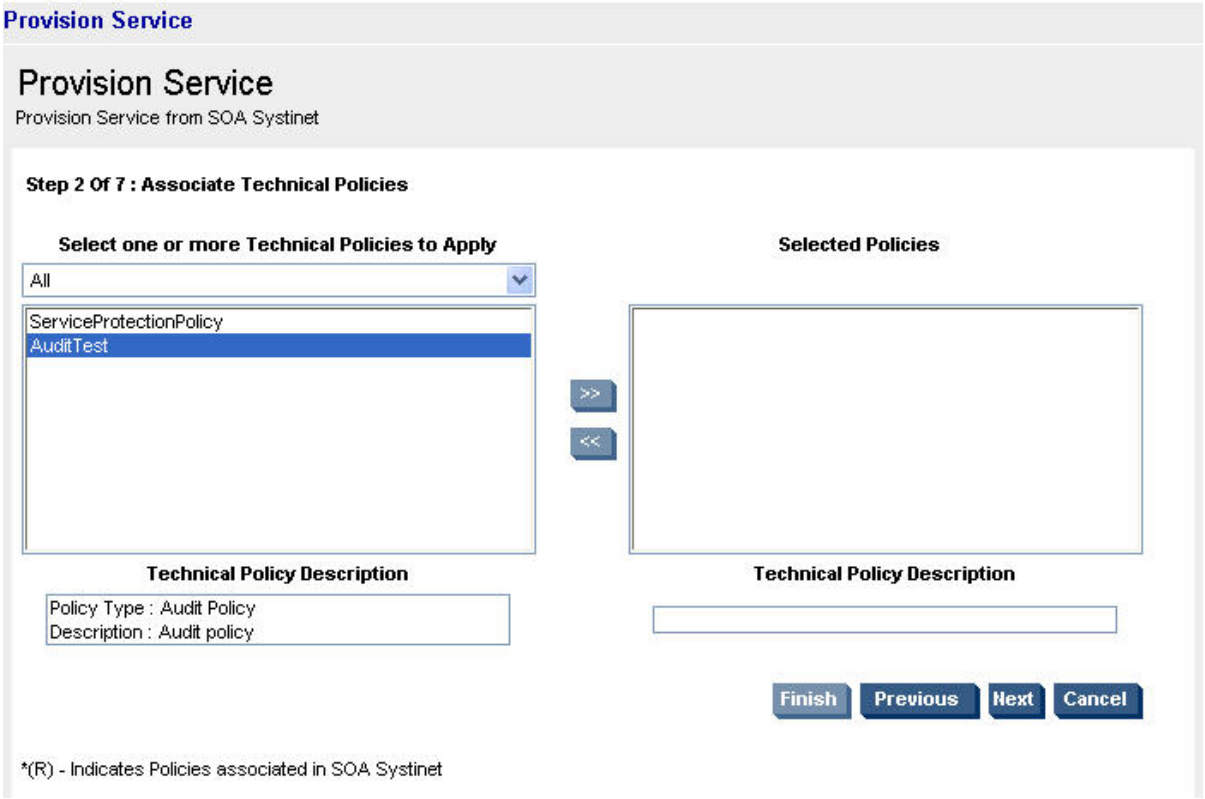
7 Select the PEP from the **Policy Enforcement Point Type** drop-down list.

8 Click **Next**. This displays the Associate Technical Policies screen as shown in the following section.

## Associate Technical Policies

1 Select the technical policies from SOA PE that you want to associate with the Web service from the **Select one or more Technical Policies to apply** list.

For a Web service from Systinet, this screen lists the technical policies associated with the service in the registry in the **Selected Policies** list. SOA PE prefixes the letter *R* to denote policies from Systinet in the **Selected Policies** list.




2 Click the >> icon to move the selected policies to the Selected Policies list. Make sure that you select the policies appropriate to the type of service you have selected (SOAP or XML) and that the policies are not conflicting. Refer Appendix E for more information on policy conflicts.

For a Web service from Systinet, you cannot disassociate the policies that are already associated with the service in the registry.

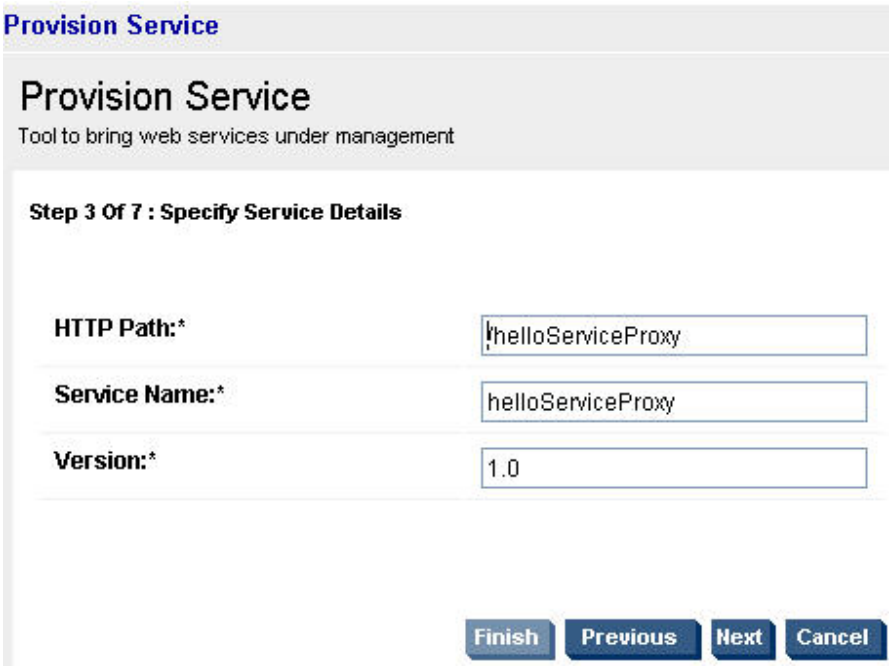
3 Click **Next**. This displays the Specify Service Details screen as shown in the following section.

## Specify Service Details

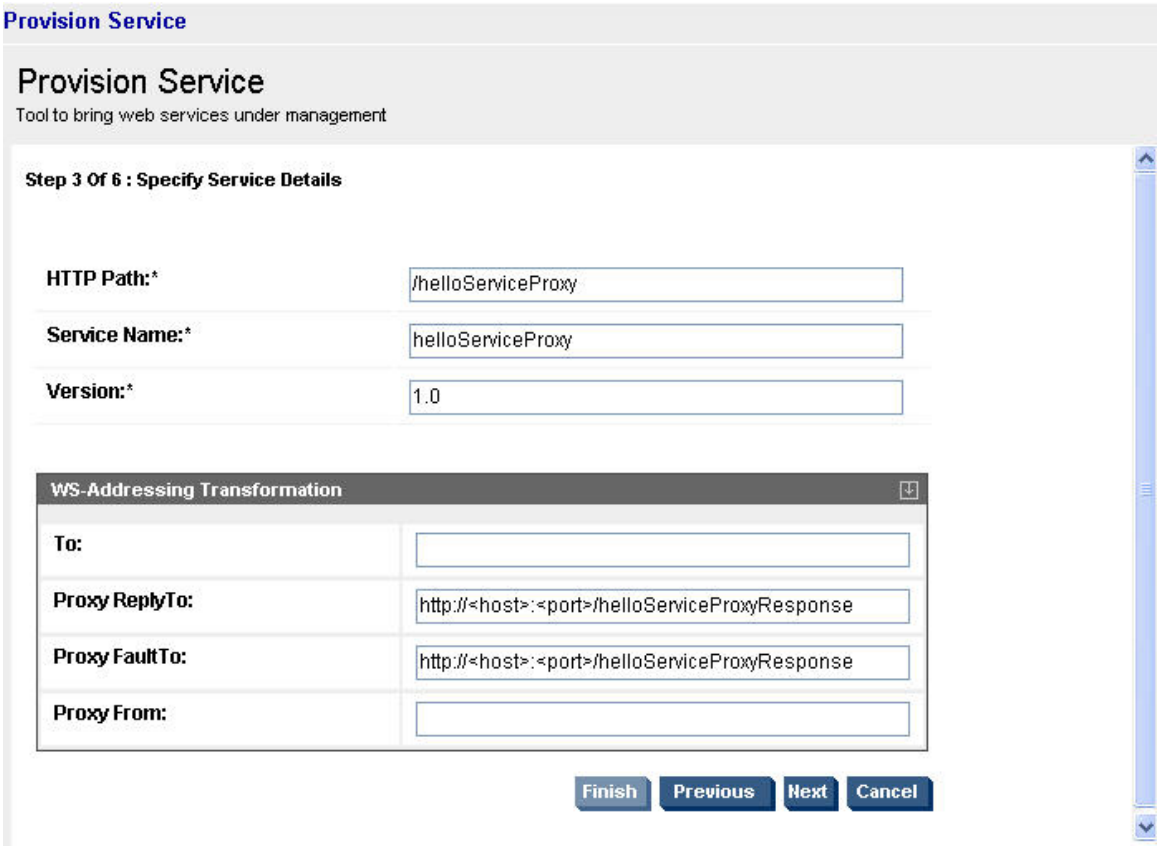
1 Specify the **HTTP Path**, **Service Name** and **Version** of the service in the respective boxes shown in the Service Details screen shown below.

You must only include alphanumeric characters when specifying the service name.

You must specify the HTTP context path in the HTTP Path box for an XML service if you had not selected JMS as the transport protocol for the service.




2 If you had associated a WS-Addressing policy in the previous screen, the provisioning wizard displays the following additional details on this page as shown below where you can specify the WS-Addressing transformation information. The parameters for this section are as follows:




— **To**: Use this parameter to specify the URI of the end point to which the request must be sent.

— **ProxyReplyTo**: Use this parameter to specify that the Broker must receive the successful response from the end point.

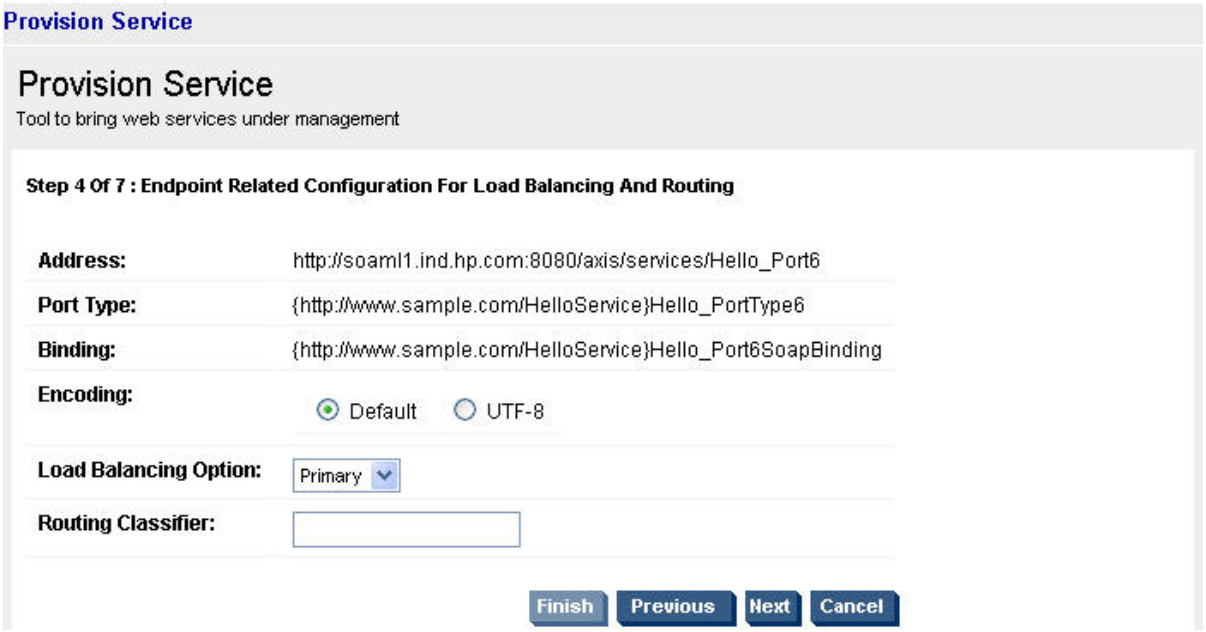
— **ProxyFaultTo**: Use this parameter to specify that the Broker must receive the fault response from the end point in the event of a fault generated at the end point.

— **ProxyFrom**: Use this parameter to specify the Broker URL address.

3 Click **Next** to go to the End Point Related Configuration for Load Balancing and Routing screen shown in the following section.

## Specify End Point Related Configuration for Load Balancing and Routing

1 Select **Primary** or **Backup** from the **Load Balancing Option** to specify if the specified end point is a primary load balancing end point or a backup load balancing end point for service requests.




2 Select **UTF-8** from the Encoding section if you want to enable UTF-8 encoding while communicating with the endpoint. The Default option is selected by default.

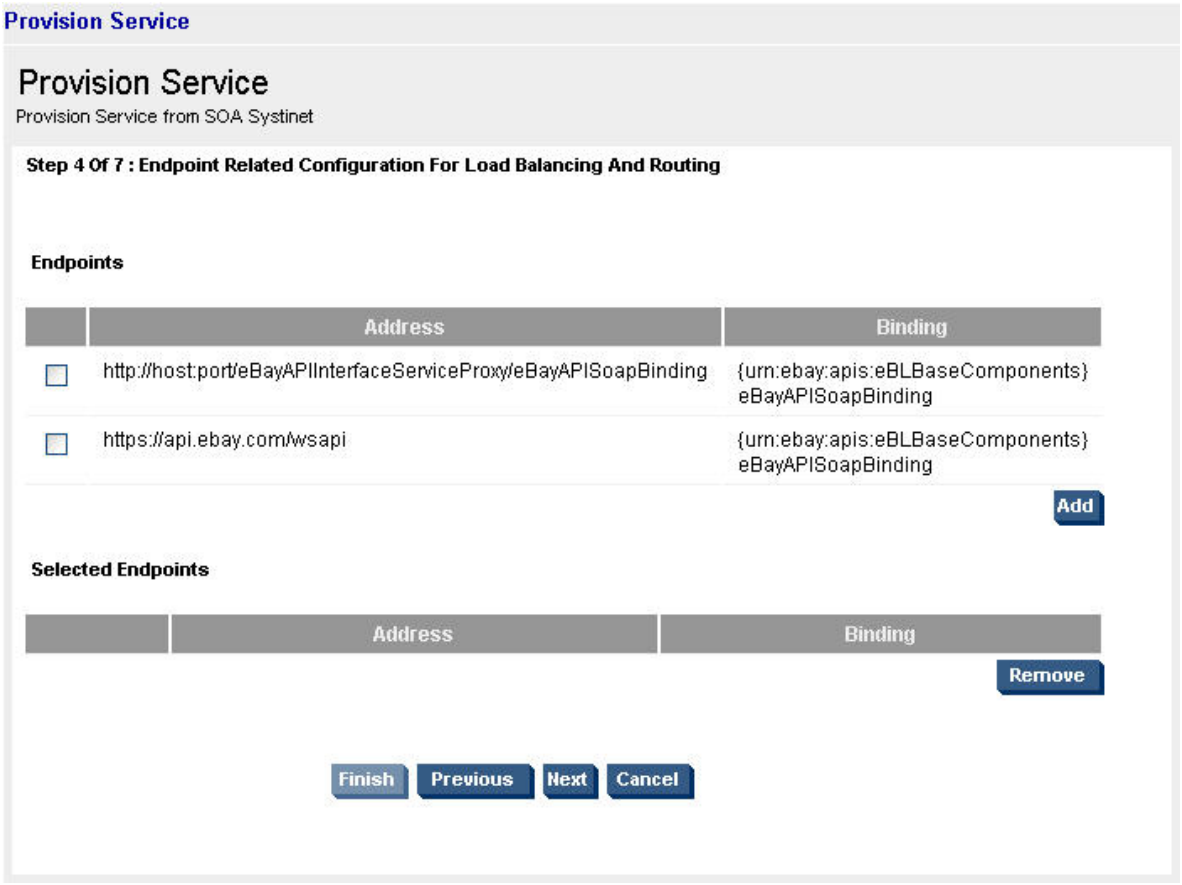
3 Specify the **Routing Classifier** in the corresponding box. This step is optional. Routing classifier indicates a classification for the endpoint. If you specify a routing classifier, you must provide the following details in the Routing Policy Definition page:

— Specify the XPath Expression to the routing classifier

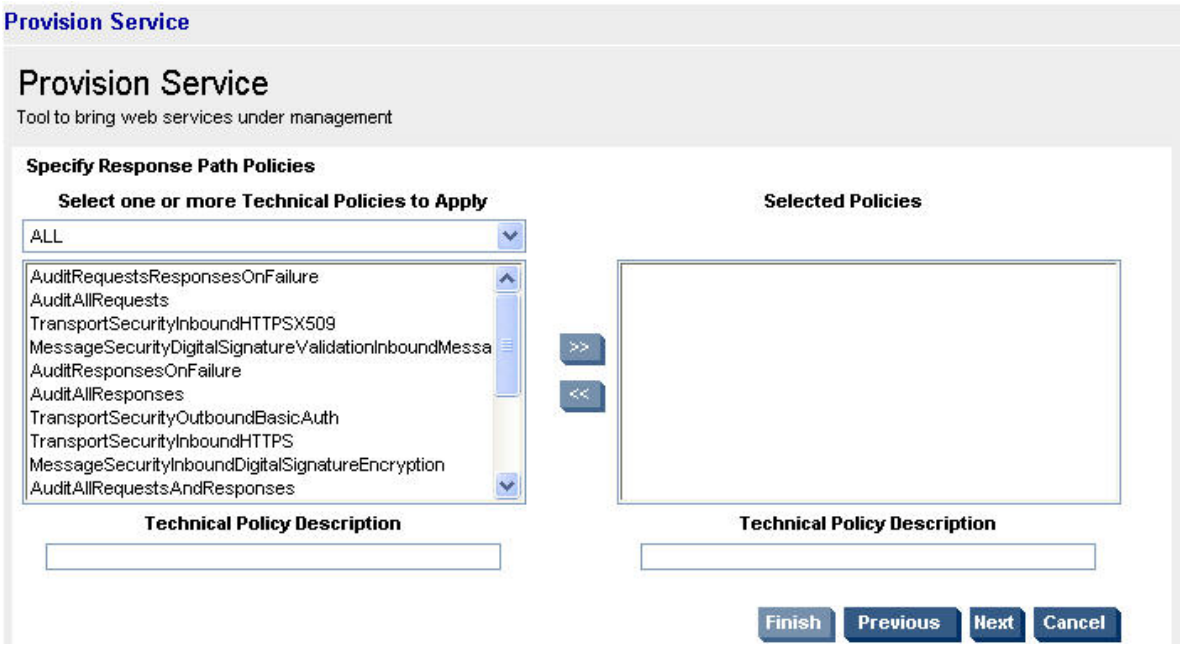
— Specify if the routing classifier is applicable for a Message (response or request) or Transport (transfer level security) Context option.

— Specify the name spaces and the corresponding URIs in the respective boxes. You can click Edit to edit the URI corresponding to a name space.

For Systinet sessions, select the end points and click **Add** to specify the end points associated with the service from the registry to be used for provisioning as shown in the following screen shot. You can edit the end point details.




4 Click **Next**. This displays the Specify Response Path Policies screen as shown below if you had associated a WS-Addressing policy with the service in the previous steps.

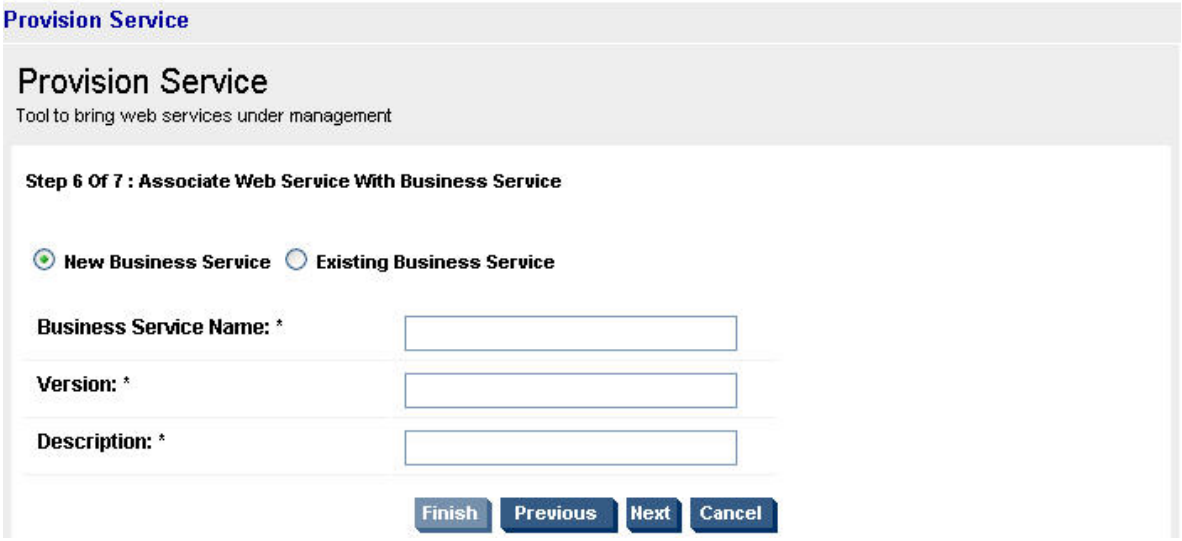


5 You can select the policies that you want to apply for the response messages from the **Select one or more Technical policies to Apply** drop-down list and click >> to move the policy to the **Selected Policies** list.

6 Click **Next**. This displays the Associate Web Service with Business Service screen as shown in the following section.

## Associate Web Service with Business Service

1 Specify the **Business Service Name**, **Version** of the business service, and a **Description** for the business service in the corresponding boxes. If you selected **Existing Business Service**, the wizard prompts you to select an existing business service from the **Business Service Name** drop-down list. If you are provisioning Web services that originate from Systinet, the service provisioning wizard displays the name and description of the business service from the registry.

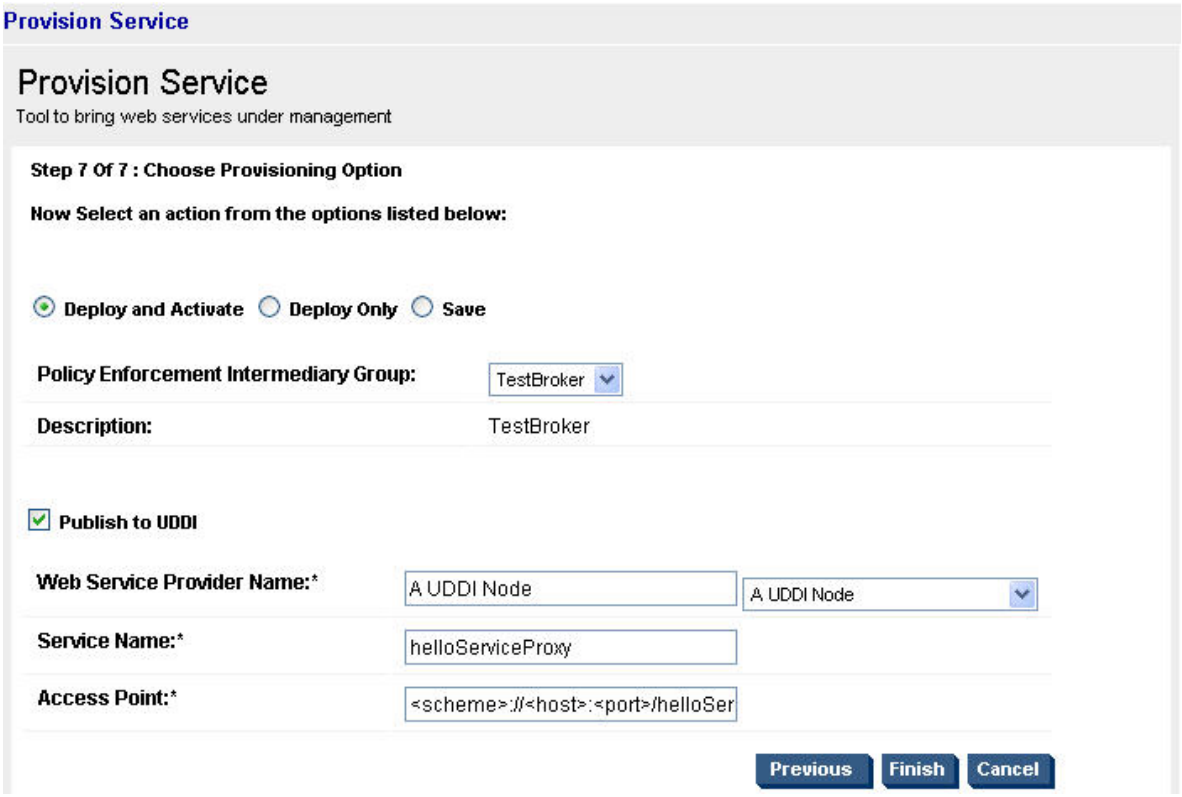



You must only include alphanumeric characters and whitespace character when you specify the business service name.

2 Click **Next**. This displays the Choose Provisioning option screen as shown in the following section.

## Choose Provisioning Option

1 Select one of the following options to provision the service from the screen shown below:




— **Deploy and Activate**- This option schedules the deployment and activation of the Web service on the PEP intermediary.

— **Deploy**- This option only deploys the Web service, but does not activate the service.

— **Save**- This option saves the Web service configuration that you can deploy at a later point of time.

2 Select the PEP from the **Policy Enforcement Intermediary Group** drop-down list

3 Select **Publish to UDDI** to publish the service to UDDI. This step is optional and is required only if you want to publish the Web service to UDDI. If you are publishing the service to UDDI, you must provide the **Web Service Provider Name**, the **Service Name** (Web service name), and the URL to the UDDI **Access Point** in the respective boxes.
For services from Systinet, this option is selected by default and SOA PE displays the web service provider name associated with the service in the registry. You can modify the access point.

4 Click **Finish** to complete the task of provisioning a Web service.

5 Click **OK** for the confirmation message that you receive.

# Life Cycle Management

SOA PE provides a life cycle management feature for monitoring the status (life cycle) of a provisioned Web service. After logging in as an administrator in SOA PE, you can click **Life Cycle Status** under the **View** drop-down menu on the left pane to view the Life Cycle Status screen.

SOA PE lists the provisioned Web services under the Life Cycle Status: drop-down list. You can view the following details about a provisioned Web service:

- Current State- This column denotes the current state of the provisioned Web service. The state can be one of the following:
  - Saved: indicates that the service is saved and is ready for deployment.
  - Ready to Deploy: indicates that the service is in the deployment state.
  - Deployed: indicates that the service is deployed and ready to be activated.

For sessions from Systinet, the state displays as follows:

  - Partially configured: indicates that the service is not yet provisioned in SOA PE
  - Modified: indicates that the service that is already provisioned in SOA PE is modified in registry.
  - Marked for delete: indicates that the service provisioned in SOA PE is deleted from the registry.
- Service Name and Description- This column denotes the name and description of the provisioned Web service.
- Source- This column denotes the source that provisioned the Web service for deployment. This can be SOA PE or Systinet.
- Since (Date and Time) - This column denotes the date and time when the service was last modified.
- Last Action Status- This column indicates the status (success or failure) of the last action on the Web service. The status can be as follows:
  - In progress: indicates that the last action is in progress.
  - Succeeded- indicates that the provisioned session is saved or deployed.
  - Failure- indicates that the service provisioning has failed.
  - Not initiated- indicates that the session is about to be deployed.
- Actions- This drop-down list provides the following actions you can use on a provisioned Web service:
  - Remove- This action deletes the provisioned Web service from the provisioned Web services list on the Life Cycle Status screen.
  - Provision - This action provisions a saved web service configuration to the PEP or activates a service that is already deployed on PEP. You can also use this option to re-provision a failed session after resolving the cause for failure.
  - Delete service- SOA PE displays this option only for the sessions from Systinet. This option undeploys the service that was removed from the registry.

SOA PE generates an alert when deployment to a PEP is a success or failure. The alert can be one of the following for a provisioning session:

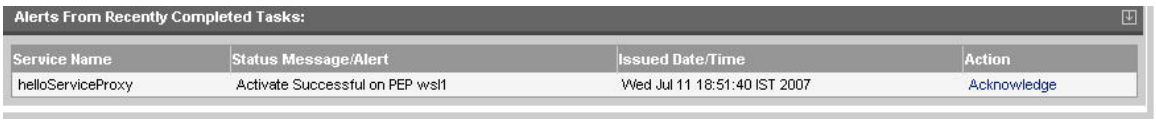
- Deploy success: indicated that the deployment was completed successfully.
- Deploy failure: indicates that the deployment has failed.
- Publish: indicates that the publish to UDDI registry is successful. SOA PE logs an event only if the publish to UDDI fails. You must note that SOA PE does not mark a session as a failure even if the publish to UDDI is unsuccessful.

You can also refer to the graphical representation to identify the status of a provisioned Web service.


SOA PE displays a plus mark on the SOA PE figure as soon as you provision a Web service. This indicates that the Web service is present on SOA PE and is yet to be deployed on the PEP. After the Web service is deployed to the PEP, a plus mark appears on the PEP figure indicating the deployment of the provisioned Web service on the PEP. Similarly, if you provision a Systinet Web service, the plus mark appears first on the SOA Systinet figure before moving to SOA PE and then to the PEP figure to indicate that the web service from Systinet is provisioned on the PEP.

After a successful deployment to the PEP, SOA PE does not display the details of the provisioned Web service on the Life Cycle Status: screen. You can click **Business Services** in the **View** drop-down menu on the left pane to see the provisioned Web service that was associated with a business service during Web service provisioning.

SOA PE also generates alerts for each action performed on a provisioned Web service in the Alerts from Recently Completed Tasks: drop-down list as shown in the following figure.




You can click **Acknowledge** to remove an alert from the list.

## Provisioning New services from Systinet

For information on how to provision new services created in Systinet, see the section “Service provisioning”.

## Provisioning Modified Sessions from Systinet

If a provisioned service from Systinet has a change of policy association from Systinet, SOA PE displays the state of the service in the Lifecycle Status page to `Modified`. You must provision the service again from SOA PE. SOA PE allows you to modify only the following details for a modified session:

- End point information
- Policy association
- Access point information

If you modify the end points associated with a service from Systinet, you must manually add the end point in SOA PE using the Add option in the routing table section for the service configuration.

## Re-provisioning Failed Provision Sessions

The lifecycle page alerts you about the status of a provisioned service. You can always re-provision services that failed a provisioning attempt after resolving the issues. Service provisioning might fail if one of the PEPs is not reachable.

If you receive an alert from the lifecycle status page indicating a service provisioning failure, you can re-provision the service by clicking the **Provision** option from the lifecycle page without having to provision the service using the service provisioning wizard. SOA PE deploys the service to the PEPs on which the deployment failed.

A failure while publishing a service to the registry generates an event that is logged. You can republish the service using the publish link of the corresponding business service.

## Removing Services Deleted from Systinet

After you provisioned a service from Systinet using SOA PE and if the service gets removed from Systinet, SOA PE creates a session with the state `Marked For Delete` in the Lifecycle status page. You can use the **Delete Service** option in the lifecycle status page to undeploy the service from the PEP.

Alternatively you can remove the provisioned session corresponding to that service from SOA PE using the **Remove** link for that service, if the service is not available in registry.

# Dashboard

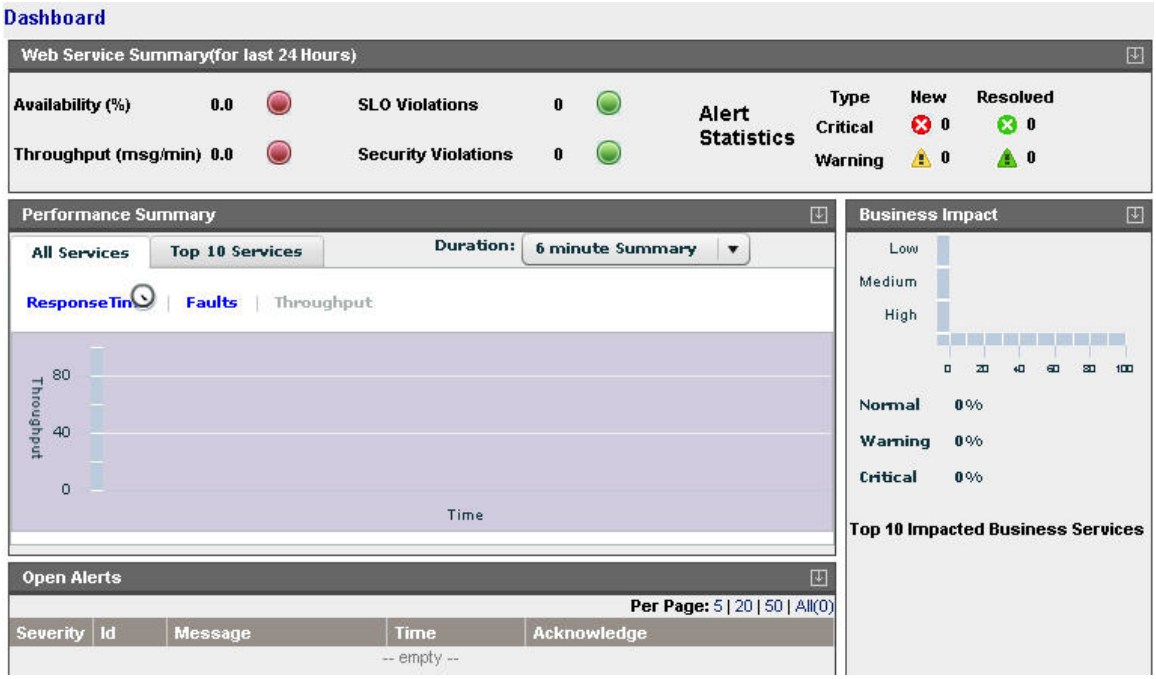
The SOA PE Dashboard provides a graphical view of the performance of services. You can use the dashboard to monitor the following:

- Monitor performance metrics of Web services
- Manage alerts

- Identify services that impact business
- Isolate and identify the cause of a problem

## Accessing the Dashboard

Log in to SOA PE as an administrator and click **Dashboard** from the **View** drop-down menu. This displays the Dashboard as the following graphic shows.




The dashboard is classified into the following sections. Refer to the sections below for more information about each section on the dashboard:

- Web Service Summary (for last 24 hours)
- Alert Statistics
- Performance Summary
  - All Services
  - Top 10 Services
- Business Impact
- Top 10 Impacted Business Services
- Open Alerts

## Web Service Summary

This section displays the status of all the Web services graphically in the past 24 hours based on the following performance metrics. A green circular display denotes that the performance metric is within the threshold limits. A yellow circular display denotes that the performance metrics has degraded and has surpassed the warning threshold value set for it. A red circular display next to the metric denotes that the performance metrics have degraded and has surpassed the critical threshold value set for it or the metric is not available:

- Availability (%)- Signifies the availability of the Web services in the last 24 hours.
- Throughput- Signifies the total number of messages received by the Web services in a minute in the last 24 hours.
- Security Violations- Signifies the security violations encountered by the Web services in the last 24 hours.

## Alert Statistics

This section represents the alerts aggregated in the last 24 hours for the following types of alerts:

- Critical New
- Critical Resolved
- Warning New
- Warning Resolved

The number next to each type of alert represents the number of corresponding alerts received in the past 24 hours.

## Performance Summary

You can use this section to list the performance summary for all the Web services or the top 10 Web services.

To view the performance summary of all the Web services, click **All Services** tab in the Performance Summary drop-down list. This displays the performance of all the services based on the following metrics. Click on each tab to view the summary graph based on the corresponding metric:

- Response Time
- Faults
- Throughput

To view the performance of the top 10 services, click **Top 10 Services** tab. This displays the graph for the performance of the top 10 Web services. You can view the top 10 services based on one the following performance metrics you select from the **Criteria:** drop-down list:

- Success Count

- Failure Count
- Total Request
- Security Violations
- Average Response Time (ms)
- Maximum Response Time (ms)
- Minimum Response Time (ms)
- Availability (%)
- Uptime (%)

The Performance Summary: drop-down list also lists the metric values based on which you identified the top 10 services.

## Web Service Performance Metrics

The Performance section of a Web service's view screen gives an overall view of how the Web service is performing. In addition, if the service model contains specific operations for a Web service, a Performance section also displays on each Web service operation's view screen. This allows you to view performance down to the operation level. In such cases, the Availability and Uptime metrics for Web service operations have the same values as the Availability and Uptime of the operation's Web service.

The following table defines each of the metrics that are collected for a Web service.

**Table 4-1: Web service Performance Metrics**

| Metric | Value |
|---|---|
| Availability (%) | The percentage of successful Web service requests sent during the configured interval. If there is traffic (requests are going through), Availability % = successful requests / total request (that is, if 5 requests go through, and 4 succeed, availability is 80%).<br>If no requests are sent, the field is left blank. If a policy enforcement intermediary group goes down, the Uptime percentage gradually goes down to zero. The value gradually goes to zero because the SOA PE server intermittently tries to contact a policy enforcement intermediary group and assumes the policy enforcement intermediary group will recover. |
| Average Response Time (ms) | The average amount of time in milliseconds for a successful Web service response. If no requests are sent during an interval, this field is left blank. |
| Failure Count | The total number of failed Web service invocations. |
| Maximum Response Time (ms) | The maximum amount of time in milliseconds for a successful Web service response. If no requests are sent during an interval, this field is left blank. |

| Metric | Value |
|---|---|
| Minimum Response Time (ms) | The minimum amount of time in milliseconds for a successful Web service response. If no requests are sent during an interval, this field is left blank. |
| Security Violations | The total number of times a security violation occurred. |
| Success Count | The total number of successful Web service invocations. |
| Total Requests | The total number of Web service requests. |
| Uptime (%) | The percentage over time that a Web Service has been available. It is the availability of the service that is being measured and does not depend on any traffic/messages.<br>At every poll interval, statistics for a service are gathered. If the service returns the statistics, it is considered available. To change the poll interval, see "Changing the Service Polling Interval" below. |

You can sort the performance metrics based on the following options present on the dashboard in the Performance Summary: section:

- **Best Performing**- This option lists the best performing Web services based on the specified metric.
- **Worst Performing**- This option lists the worst performing Web services based on the specified metric.

## Changing Performance Summary Interval

From the dashboard, click the **Duration:** drop-down list to change the performance summary collection interval from the default value (6 Minute Summary) to any of the following values:

- 1 Hour Summary
- 1 Day Summary

### Performance Graph

The Web services performance graph provides a visual view of the performance metrics based on the current monitoring interval. The graph includes the following elements:

- **Green line**: Represents the average response time in milliseconds during a given time interval
- **Red line**: Represents the total number of successful requests during a given time interval.
- **Orange bars**: Represents the total number of faults during a given time interval.

- The Throughput fields are calculated for the currently selected monitoring interval. For the Six Minute One Hour and One Day intervals, throughput is success/Minute. The lifetime interval does not provide any metrics currently.

### Changing the Service Polling Interval

The SOA PE periodically polls services to ensure their availability and update their performance metric values.

To change the service polling interval, follow these steps:

1. Stop the SOA PE server if it is currently started.
2. Use a text editor to open <*install_dir*>`\conf\networkservices\mipServer.xml`.
3. Add an entry for `com.hp.service.polling.interval` and enter a value in milliseconds. For example:

   ```
   <entry name="com.hp.service.polling.interval">60000</entry>
   ```

4. Save and close the file.
5. Restart the SOA PE server.

## Business Impact

The Business Impact drop-down list displays the total number of business services (in percentage values) and the status according to severity levels. The severity levels are as follows:

- Critical
- Warning
- Normal

The Y-axis (vertical portion) of the graph displayed in the Business Impact section indicates the following details:

- High signifies the number of business services designated with a high impact on business.
- Medium signifies the number of business services designated with a medium impact on business.
- Low signifies the number of business services designated with a low impact on business.

The X-axis (horizontal portion) of the graph indicates the number of business services in the high, medium, or low category.

Click the Problem Analysis button on the Business Impact drop-down list to view the business services that cause the problem. This helps you to isolate and identify a specific problem.

## Open Alerts

You can view a list of Open Alerts sorted by severity and time from the Open Alerts drop-down list on the dashboard. The dashboard displays the following details for a critical or a warning type of an alert:

- Severity- Signifies the severity of the alert.
- Message- Displays the alert message.
- Time- Displays the time of the alert generation.

You can acknowledge alerts by clicking the **Acknowledge** link displayed for each alert.

You can also choose from the following options on the Open Alerts drop-down list to view the number of alerts in a page:

- 5
- 20
- 50
- All

# Viewing Reports

The SOA PE web interface lets you to query the SOA PE database and produce business service reports. The following reports can be generated for any business service over any specified period of time:

- Audit Message Traces Reports
- Web Service Metrics

## Web Service Metrics Reports

The Web service metrics report provides the following statistics:

- Request Count
- Success Count
- Failure Count
- Availability Percentage
- Average Response Time
- Max Response Time
- Min Response Time.

To view Web service metrics reports, follow these steps:

1 From the **View** drop-down menu, select **Reports**. The Reports screen opens.

2 Click **Web Service Metrics**. The Web Service Metrics screen opens.

3 Complete the following fields:

- **Consumer:** Use the drop-down list to select a specific consumer of the business service to include in the query. To include all the consumers of a service, select **ALL**.
- **Service:** Use the drop-down list to select the business service on which to constrain the report.
- **Start Date:** Use the fields to enter a specific start date and start time for the query to match.
- **End Date:** Use the fields to enter a specific end date and end time for the query to match.
- **Interval:** Use the drop-down list to select a predefined interval of time on which to constrain the report. For example, if you select `1/2/05 10 AM PST` as the start date, and `1/3/05 10 AM PST` as the end date, and interval as an hour, the report will contain around 24 rows, one for each hour between 1/2/05 10 AM and 1/3/05 10 AM. Each row is labeled with the timestamp for the hour it represents.

4 Click **Query**. The results of the query are listed under the Service section.

## Audit Message Traces Reports

This report allows you to view audit message trace information.

To view Audit Message Traces reports, follow these steps:

1 From the **View** drop-down menu, select **Reports**. The Reports screen opens.

2 Click **Audit Message Traces**. The View Success and Failures screen opens.

3 In the Query section, configure the following query fields:

- **Search For:** Select the **Success** or the **Failure** check boxes.
- **Service:** Use the drop-down lists to filter the report by business service, or policy enforcement point, or existing service. The service model must exist for you to filter the report. You must select a business service from the **Filter by Business Service** drop-down list to filter the report by policy enforcement point or an existing service based on the business service. If you want to filter the report on the basis of an existing service, you must select a policy enforcement point from the **Policy Enforcement Point** drop-down list. This lists the existing service details based on the PEP specified. If the Service is not listed in the Existing Service drop-down, use the Specify a Service text box to specify the Service.
- **Start Date:** Use the fields to enter a specific start date and start time for the query to match.
- **End Date:** Use the fields to enter a specific end date and end time for the query to match.
- **Stakeholder:** Enter a user in the field if you want query the trace messages based on a specific authenticated security principal (authenticated user) that made the request.

4 Click **Query**. The results of the query are listed under the Service section.

5 Click on a trace message's Timestamp to view trace information details as well as Profile Data.

The report provides a summary of the request envelope to the end point and the transformed response envelope from the end point. The *Results* section of the report provides the following information:

- **Timestamp**: provides information about the time at which the audit trace was generated.
- **Duration**: provides the duration for which the audit trace was generated for the specified service.
- **User**: provides information about the user who owns the service.
- **Endpoint**: provides endpoint related information for the service.
- **Reason for Failure**: displays the reason for failure

5

# Using Alert Notifications

This chapter describes how to configure and use the alert notification features in SOA PE. The instructions include creating alert recipients and alert recipient categories. This chapter provides an overview and conceptual architecture of the alert notification feature.

## Overview

The alert notification feature is used to notify recipients when events occur that may impact network or business operation performance. When events occur, alerts are automatically sent to any number of alert recipients so that appropriate actions can be taken. In general, alerts help maintain efficient applications and help stop problems before they impact performance or breach business rules. Specifically, alerts are useful for:

- Troubleshooting – Alerts provide event details that can be used to see why a business service or its contained Web service may be failing.
- Content Monitoring – Alerts can notify recipients when a specific value (for example, order > $25,000.00) is found in a SOAP message.
- SNMP Management – Alerts in SOA PE can be integrated with SNMP management solutions.

## Security Alerts

Security alerts notify you of security violations at the transport level.

# Business Content Alerts

Business content alerts notify alert recipients when specific content is contained in a SOAP message. Business content alerts are useful because they allow you to react to events that can potentially have an impact on business operations. For example, if you are managing an order process service, you could receive an alert when:

- An important client is using the service
- An order total is greater than $25,000.00
- A specific product is ordered
- A specific product is shipped

Business content alerts display on three screens in the SOA PE user interface: a business service view screen, a Web service configuration view screen, and the alert list screen. A business content alert is generated for the Web service configuration associated with the Web service that sends the alert to the SOA PE Server. This alert has a severity level of `normal`. Another alert is generated for the business service that contains the Web service configuration. This alert has a severity level of `informational`.

A special alert category is used for business content alerts, the `Business Content Alert Category`. This category is set for the business service and the Web service configuration. If the category is not set for the Web service configuration, the business service setting is used for the Web service configuration alerts. For more information on alert categories, see the “Setting Up Alert Recipients” section later in this chapter.

## Defining a Business Content Alert

You must define and enforce an event policy on a service to generate alerts. Refer to the section *Creating an Event Policy* in *Chapter 4 SOA PE Administration* for information about creating an event policy. The policy enforcement intermediaries contain a business content alert handler that is used to define a business content alert. Business content alerts are defined differently for the policy enforcement intermediaries. Before you can view a business content alert for a Web service, the Web service must be contained in a business service.

Defining a business content alert requires knowledge of the W3C XPath expression language. It is beyond the scope of this documentation to cover the details of XPath. You can find books on XPath and you can visit the W3C website for details. If you are not familiar with XPath, you should consult a developer before defining a business content alert.

# Troubleshooting Business Content Alerts

The following steps can help troubleshoot configuration issues related to business content alerts.

## SOA PE Setup

- Ensure the policy enforcement intermediary that should be raising the Business Content Alerts is registered with SOA PE and is reachable.

  In the SOA PE web interface, select the PEP that contains the policy enforcement intermediary group and ensure that the Availability field displays the value **Operational** (also indicated by a green check).

- Ensure that SOA PE subscribes to the policy enforcement intermediary for Business Content Alerts.

  Edit the `xpllogging.properties` file in the <*install_dir*>/`conf/networkservices` directory and increase the log level for the logger: `com.hp.ov.mip.wsm.sn.monitoring.notification.BusinessContentMonitoringService.level=FINE`.

  Restart SOA PE. When the intermediary is re-added at startup, there should be log messages for each intermediary indicating whether or not it believes the intermediary supports Business Content Alerts, and if so, showing that SOA PE has subscribed for Business Metric `raiseAlert` events.

## Invocations

- Check that the invocations that should be triggering the Business Content Alert are actually reaching the configured policy enforcement intermediary.

  — When using the policy enforcement intermediary:

  View the Service Details page and ensure that the Logging option is selected for the Service.

  Edit the `xpllogging.properties` file in the <*install_dir*>/`conf/broker` directory and set the logger `service.`<*service name*> to `INFO`. For example, for a Service named FinanceServiceProxy, add the following line:

  `com.hp.ov.mip.service.FinanceServiceProxy.level=INFO`

  Restart the intermediary.

  Send an invocation through the intermediary. The request and response messages should display in the Broker Configurator (if the intermediary is not running as a win32 service) and in the intermediary log file.

- Confirm that the message body (request or response, depending on the Business Content Alert configuration) contains the necessary data to trigger the configured Business Content Alert.

  — When using the policy enforcement intermediary:

  Confirm that the operation name specified in the Operation field of the Business Content Alert configuration matches the Request Operation name in the log file.

Confirm that the XPath expression will select the correct node in the request or response body (depending on whether Request Message or Response Message was selected in the alert configuration).

Confirm that the namespace prefixes used in the XPath expression are correctly defined in the alert configuration.

- Check that the policy enforcement intermediary is raising the alert.
  - When using the policy enforcement intermediary:

    Edit the `xpllogging.properties` file in the <*install_dir*>/`conf/broker` directory and set the logger:

    `com.hp.ov.mip.wsm.sn.router.xml.bizmetrichandler.level=WARNING`

    Restart the policy enforcement intermediary.

    Send an invocation through the intermediary. A log message should appear in the log file indicating that a `BusinessMetricAlert` for metric <*metric name*> is being sent.
- Check that SOA PE is receiving the alert.

  If the alert is received, the `BusinessContentMonitoringService` and the `AlertDispatcher` will log any problems that occur processing the alert. Otherwise, the alert should display in the Alert List.

No positive debug logs exist in the `BusinessContentMonitoringService` to indicate normal processing of Business Content Alerts.

# Customizing Alert Messages

Customizing alert messages provides a greater level of granularity when describing the reasons for an alert and can help create more meaningful messages that are specific to an enterprise. Detailed and familiar alert messages can improve issue resolution as well as maintain overall performance.

Alert messages are created using a default message that contains information about the alert (alert severity, source, timestamp, and so on). You can customize any alert message to include additional information. The information can be text that you add to the message and can also include dynamic properties that are exposed by the Alert Service.

Alert messages can be customized only after an alert is generated for the first time. After the message is customized, all subsequent messages of the same alert type will contain the custom message.

To customize and alert, follow these steps:

1. From the **View** drop-down menu click **Alerts**. The Active Alert List screen opens.
2. Click the Alert Details for the alert type whose message you want to customize. The Basic Details screen opens. Basic details as well as specific properties of the alert message are listed.

3 From the Short Message row, click **format**. The Edit Alert Message screen opens.

4 In the message text box, customize the default message. You can use text as well as any dynamic properties that are listed in the Dynamic Values table. Dynamic properties must be entered using the format ${*property_name*}.

5 To preview the message, click **Test**.

6 Click **Save**. The next time an alert of this type is generated, it will contain the custom message.

7 Click **Done**.

8 Repeat this procedure to customize additional alert messages for an alert type.

# Acknowledging Alerts

Alerts that are resolved must be acknowledged and removed from the SOA PE user interface. If the alert is listed on multiple View screens, acknowledging an alert removes it from the View screens as well.

To acknowledge alerts, follow these steps:

1 From the **View** drop-down menu, click the **Alerts** tab. The Active Alerts List screen opens.

2 Use the option boxes to select the alerts you want to acknowledge, or select the option box in the table head to remove all alerts.

3 Click **Acknowledge Selected**. All the selected alerts are removed from the Alerts List as well as the alert section of a view screen.

Acknowledging alerts from SOA PE web interface does not remove alerts from the SOA PE database.

# Querying Alerts

All alerts are stored in the SOA PE database. The alerts remain in the database even after they are removed from the SOA PE web interface. The query link allows the user to find audit traces in the database that may be related to an alert.

This feature is the same as the audit feature. Using this feature returns audit traces, which include alerts.

To query an alert, follow these steps:

1 From the **View** drop-down menu, click **Alerts**. The Active Alert List screen opens.

2 Click the Alert Id number for the alert type you want to query. The Basic Details screen opens. Basic details as well as specific properties of the alert message are listed.

3 From the Message row, click **query**. The Query screen opens.

4 Use the **Start Date** fields to enter the query's start date and start time.

5 Use the **End Date** fields to enter the query's end date and end time.

6 Use the Service drop-down lists to select the service to query.

7 To query the alerts based on a specific authenticated security principal (authenticated user), enter the user name in the **User** text box.

8 Click **Query**. The results of the query are listed in the Results section.

9 Click on a timestamp to view audit details.

# Setting Up Alert Recipients

When an alert is generated, it is sent to recipients that are part of a recipient category. Alert recipients include the following:

- SOA PE Web Interface – Alerts are sent to the web interface. Depending on the source of the alert, the alert listed on the Alerts is also viewable on the Business Service View screen, Configuration View screen, and Resource View screens. All alerts are listed on the Alert List screen.
- SNMP – Alerts are sent to an SNMP log category that is configured to send the log message to an SNMP TRAP.
- SMTP – Alerts are sent as an email message to any number of email addresses.
- Log File – Alerts are sent to a log category and published using the output method defined by the category.

Recipient categories are used to organize recipients because they provide an efficient method of supporting multiple recipients for an alert. Several default categories are provided that you can customize. In addition, you can create your own recipient categories.

## Modifying an Existing Recipient Category

To modify an existing recipient category, follow these steps:

1 From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.

2 Click the **Alert Settings** tab. The Alert Settings screen opens.

3 Under the Service alerts assigned to *category* ... section, click the recipient category you want to modify. The Edit Alert Category screen opens.

4 From the list of targets, select the targets to be included in the category.

5 Click **Update Alert Targets**.

## Creating Recipient Categories

To create a new recipient category, follow these steps:

1. From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.
2. Click the **Alert Settings** tab. The Alert Settings screen opens.
3. Under the Service alerts assigned to *category* ... section, enter a name for the new alert category.
4. Click **Add Category**. The new category is listed in the list of available categories.

## Adding Alert Recipients to a Recipient Category

To add alert recipients to an alert category, follow these steps:

1. From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.
2. Click the **Alert Settings** tab. The Alert Settings screen opens.
3. Under the Service alerts assigned to *category* ... section, click the recipient category you want to modify. The Edit Alert Category screen opens.
4. Select the targets to be included in the category.
5. Click **Update Alert Targets**. The Alert Settings screen opens and lists the recipients associated with the recipient category.

## Creating Email Recipients

The SMTP feature uses the server's native SMTP service to send emails to an email recipient. If the SMTP service is not activated, you must activate the service before emails can be sent. See your operating system's documentation for instructions on enabling the SMTP service.

To create an email recipient, follow these steps:

1. From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.
2. Click the **Alert Settings** tab. The Alert Settings screen opens.
3. Under the Assigned alerts can be sent to *target* ... section, click **Add New Target**. The Add Alert Target screen opens.
4. From the drop-down list, select **email**.
5. In the text field, enter a name for the recipient.
6. Click **Add Target**. The Alert Settings screen opens and the new recipient is listed in the list of available recipients.
7. If the email settings have not been configured, click the **Email Settings** tab. The Email Settings screen opens. Enter the email properties using the following fields:

- **Email Support**: Select Enable.
- **SMTP Host**: The server's host name.
- **Port**: The port on which the SMTP service is running.
- **User**: The administrator's user name that has access rights to use the SMTP service on the server. Any user that has access to the SMTP service can be used.
- **Password**: The administrator password that has access rights to use the SMTP service on the server. Any user that has access to the SMTP service can be used.
- **Sender**: The email sender.

8. Click **Save**.
9. Click the **Alert Settings** tab.
10. Under the Assigned alerts can be sent to *target* ... section, click on the new email recipient to edit its properties. The Edit Target screen opens.
11. Enter the email properties using the following fields:
    - **To**: The recipient's email address.
    - **Subject**: The subject of the email.
    - **Body**: A message to be displayed in the body of the email message. The body can use any dynamic values listed in the Dynamic Values section.
12. Click **Test** to test if the configuration you entered is valid and works correctly.
13. Click **Save**. The Alert Settings screen opens.
14. For the new email recipient, click **Start** to activate the recipient.

## Creating Log Recipients

The log feature uses the Log4j logging implementation to send an alert to a log category that publishes the alert to the output specified by the log category. Log categories are configured in the `logging.properties` file in the <*install_dir*>`\conf\networkservices` directory.

To create a log recipient, follow these steps:

1. From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.
2. Click the **Alert Settings** tab. The Alert Settings screen opens.
3. Under the Assigned alerts can be sent to *target* ... section, click **Add New Target**. The Add Alert Target screen opens.
4. From the drop-down list, select **log4j**.
5. In the text field, enter a name for the recipient.
6. Click **Add Target**. The Alert Settings Screen opens and the new recipient is listed in the list of available recipients.
7. Under the Assigned alerts can be sent to *target* ... section, click on the new log recipient to edit its properties. The Edit Target screen opens.

8 Enter the log properties using the following fields:

- **category**: The log category that the alert is sent to. Any category that is in the logging.properties file can be used. The `default` category publishes outputs to the SOA PE console.
- **level**: The logging level to use. The log levels are `DEBUG`, `INFO`, `WARN`, and `ERROR`. By default the log level is set to `WARN`. To use a different level, assign the category's level appropriately in the `logging.properties` file.
- **message**: A message to be displayed in the log. The message can use any dynamic values listed in the Dynamic Values section.

If you change `logging.properties`, you must restart the SOA PE server for the changes to take effect.

9 Click **Test** to test if the configuration you entered is valid and works correctly.

10 Click **Save**. The Alert Settings screen opens.

11 For the new log recipient, click **Start** to activate the recipient.

## Creating SNMP Recipients

The SNMP feature uses the Log4J logging implementation to send an alert to a special SNMP log category (log4j.category.com.hp.wsm.sn.notification.target.snmp). The SNMP log category is set to publish the alert message to an SNMP TRAP. You can configure the location of the SNMP TRAP in *<mip_installation_dir>*`/conf/networkservices/logging.properties`. See the "Getting Started" chapter for more information on logging.

Before configuring an SNMP recipient, you must configure your SNMP TRAP settings. The SNMP TRAP settings indicate the location and configuration of your SNMP TRAP. These settings are located in *<install_dir>*`/conf/networkservices/logging.properties`. You must restart the SOA PE server for the changes to take effect.

To create an SNMP recipient, follow these steps:

1 From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.

2 Click the **Alert Settings** tab. The Alert Settings screen opens.

3 Under the Assigned alerts can be sent to *target* ... section, click **Add New Target**. The Add Alert Target screen opens.

4 From the drop-down list, select **log4j**.

5 In the text field, enter a name for the recipient.

6 Click **Add Target**. The Alert Settings Screen opens and the new recipient is listed in the list of available recipients.

7 Under the Assigned alerts can be sent to *target* ... section, click on the new log recipient to edit its properties. The Edit Target screen opens.

8 Enter the log properties using the following fields:

- **category**: Enter the SNMP log category `log4j.category.com.hp.wsm.sn.notification.target.snmp.`
- **level**: Enter `INFO` for the level.
- **message**: A message to be displayed in the log. The message can use any dynamic values listed in the Dynamic Values section.

9 Click **Test** to test if the configuration you entered is valid and works correctly.

10 Click **Save**. The Alert Settings screen opens.

11 For the new recipient, click **Start** to activate the recipient.

6

# Using Business Services

This chapter describes how to construct service models from the context of business services. Business services are an essential part of the service model definition and are the main context from which a service model is constructed and viewed. The overview introduces the business service concept and other service model conventions.

## Overview

A business service is the virtualization of some business application that is offered by a business manager to either internal or external customers. Currently, the SOA PE only implements one type of business service, which is a Web service. This chapter only covers business services as they relate to the management of Web services.

The benefit of managing Web services using a service model is

- A business service provides different views of a Web service that are relevant to all stakeholders. The stakeholders collaborate in the complete lifecycle of Web services that are delivered and managed as business services.
- Repetitive tasks such as deploying software and configuring connectivity between underlying PEPs are automated by leveraging the meta-data captured in the service model.

## Defining Business Services

Business services are defined using the SOA PE user interface. When you define a business service, you create the business service and then add a configuration for the business service. The configuration is bound to a policy enforcement point that contains the resources that are being managed. The definition process consists of the following tasks.

## Task 1: Create a Business Service

To create a business service, follow these steps:

1. From the **View** drop-down menu, click **Business Services**. The Business Service List screen opens.
2. Click **Add**. The Create Business Service screen opens.
3. Complete the following fields:
   - **Name**: Enter a user-friendly name for this business service.
   - **Version**: Enter a version number for this business service.
   - **Description**: Enter a description for this business service.
   - **Owner**: Use the Owner drop-down list to select an owner of the configuration. Check the checkbox to send email alerts to the selected owner.
   - **Support**: Use the Support the drop-down list to select a support person of the configuration. Check the checkbox to send email alerts to the selected support person.

     Before you can assign an owner or support person, the person must be added to SOA PE. The Send email alert works only if the Email Settings are configured.

   - **Route Propagated Alerts to Category**: Use the drop-down list to select a default alert category to be used for this business service. If you are not sure which category to use, keep the `Default` category.
   - **Route Business Content Alerts to Category**: Use the drop-down list to select a default alert category to be used for business content alerts for this business service. If you are not sure which category to use, keep the Default Business Content category.

     Before you can assign any category, it must be added to SOA PE.

4. Specify the business impact (LOW, MEDIUM, or HIGH) from the **Business Impact** drop-down list.
5. Click **Add**. The Business Service List screen reopens and lists the business service.
6. Repeat this procedure to create additional business services as required.

## Task 2: Import Existing Policy Enforcement Points

Web service intermediary container configurations are used to link a business service with a policy enforcement point. A PEP group contains the resources that are managed within a business service. Any number of Web service configurations can be included in a business service.

To select any subset of the existing PEPs and create corresponding Web service configurations, and the resource configurations all in one screen, follow these steps:

1 From the Business Services List screen, click a business service. The Business Service View screen opens for the selected business service.

2 From the Model section, use the Edit drop-down list and select **Link Policy Enforcement Points**.

3 The Link to existing Policy Enforcement Points screen opens. Select the PEP and resources to be managed within this business service.

4 Click **Link**. The Business Service View screen reopens and the Model section lists the new configurations as dependencies for this business service.

Task 3 “Add a Web service configuration” and Task 4 “Add a Resource Configuration” provide an alternate approach for what Task 2 accomplishes. Continue with Step 5.

## Task 3: Add a Web Service Configuration

A Web service configuration links a business service with a Web service.

To add a Web service intermediary or a container configuration and add a new Web service configuration to a business service, follow these steps:

1 From the **View** drop-down menu, click **Business Services**. This opens the Business Services List screen.

2 Click a business service. The Business Service View screen opens for the selected business service.

3 Click the **Configuration** tab. From the **Model** section, use the **Edit** drop-down list and select **Add New Web Service Intermediary Configuration** or **Add New Web Service Container Configuration** depending on where the resource to be managed within this business service exists. The Add New Configuration screen displays for the selected configuration type.

4 Complete the following fields:

— **Name**: Enter a user-friendly name for this configuration.

— **Version**: Enter a version number for this configuration.

— **Description**: Enter a description for this configuration.

— **Owner**: Use the Owner drop-down list to select an owner for the configuration.

— **Support**: Use the Support the drop-down list to select a support person for the configuration.

Before you can assign an owner or support person, the person must be added to SOA PE.

— **Route propagated Alerts to Category**: Use the drop-down list to select a default alert category to be used for this configuration. If you are not sure which category to use, keep the `Default` category.

Click **Edit Categories** to edit an alert category or create additional alert categories. For more information on creating alert categories and alert recipients, see chapter 6 “Using Alert Notifications.”

— **Bind to Policy Enforcement Point**: Use the drop-down list to select the policy enforcement point that contains the resources to be managed in this business service.

5 Click **Save**. The Business Service View screen reopens and the Model section lists the new configuration as a dependency for this business service.

6 Repeat this procedure to add additional configurations as required.

## Task 4: Add a Web Service Configuration

You can add a Web service configuration that contains the resources that are to be brought under governance within the business service. Resources are added in the context of the configuration type that corresponds to the type of resource being managed.

Intermediary Web services and Web services can also be added to a configuration by importing a WSDL. See “Importing a WSDL” section below.

### Web Service

To add a Web service resource configuration to a PEP configuration, follow these steps:

1 From the Business Services List screen, click a business service. The Business Service View screen opens for the selected business service.

2 From the Model section, use a policy enforcement intermediary configuration’s **Edit** drop-down list and select **Add New Web Service Intermediary Configuration**. The Add New Configuration screen opens.

3 Complete the following fields:

— **Name**: Enter a user-friendly name for this configuration.

— **Version**: Enter a version number for this configuration.

— **Description**: Enter a description for this configuration.

— **Owner**: Use the Owner drop-down list to select an owner for the configuration.

— **Support**: Use the Support the drop-down list to select a support person for the configuration.

Before you can assign an owner or support person, the person must be added to SOA PE.

— **Default Alert Categories**: Use the drop-down list to select a default alert category to be used for this configuration. If you are not sure which category to use, keep the `Default` category.

Click **Edit Categories** to edit an alert category or create additional alert categories. For more information on creating alert categories and alert recipients, see chapter 5 “Using Alert Notifications.”

— **Route Business Content Alerts to Category**: Use the drop-down list to select a default alert category to be used for business content alerts for Web/brokered Service that are contained in this configuration. If you are not sure which category to use, keep the `Default Business Content` category.

— **Deployment**: Click the check box if you would like to enable the deployment feature. This feature lets you deploy a Web/brokered service to a managed policy enforcement intermediary or undeploy the service from a managed policy enforcement intermediary. Disregard this field if the Web/brokered service for this business service is already deployed to a managed policy enforcement intermediary.

If you select the Deployment check box, additional fields are displayed that allow you to deploy or undeploy or select a deployment unit.

— **Resource Discovery**: Use the drop-down list to select the Web/brokered service to be contained in this configuration. The list contains all services that are discovered when a managed policy enforcement intermediary is registered as a PEP.

Or:

Use the text box to enter the namespace and local name of the Web/brokered service in the form *{namespace}localname* (for example, `{http://mycompany.com}MyService`). You can find the values to use in the pattern by inspecting a Web service's WSDL. The *namespace* corresponds to the Web service's targetNamespace, and the *localname* refers to the service name.

A discovery pattern is typically used when adding a Web/brokered service to the service model before the service is actually deployed to a container/intermediary that is registered as a Policy enforcement point. Once the service is deployed, the pattern is used to automatically discover and add the service to this configuration.

4 Click **Save**. The Business Service View screen reopens and lists the Web/brokered service as part of the configuration.

5 Repeat this procedure to add additional resource configurations.

## Importing a WSDL

You can add a Web service intermediary configuration to a configuration based on a WSDL file. If the WSDL file defines a service that currently exists in a registered Policy enforcement point, it is automatically mapped to this configuration. If the service is not currently deployed, you can still import the WSDL. Once the service is deployed, it will automatically be discovered and added to the appropriate configuration.

To import a WSDL, follow these steps:

1 From the Business Services List screen, click a business service. The Business Service View screen opens for the selected business service.

2 From the Model section, use a policy enforcement intermediary configuration's **Edit** drop-down list and select **Import WSDL**. The Import Web Service WSDL screen opens.

3 In the Browse Local WSDL file field, enter the location of the WSDL or click the **Browse...** button to locate the WSDL.

Or:

In the Specify Remote WSDL URL field, enter the URL to the WSDL.

If there is no service defined in the WSDL file, the operation fails without any error in the SOA PE web interface. The WSDL files cannot contain external links.

4 Click **Import**. The Business Services View screen opens and the model section is updated. All operations discovered in the WSDL are also listed.

## Manually Adding Operations

Web service operations can be added to the service model allowing for fine grained manageability at the operation level. Web service operations are automatically discovered and added to the service model when the import WSDL feature is used. You can also manually add any operations:

To manually add operations, follow these steps:

1 From the Business Services List screen, click a business service. The Business Service View screen opens for the selected business service.

2 From the Model section, use a Web service configuration's **Edit** drop-down list and select **Add New Web Service Operation Configuration**. The Add New web Service Operation Configuration screen opens.

3 Complete the following fields:

- **Name**: Enter a user-friendly name for this configuration.
- **Version**: Enter a version number for this configuration.
- **Description**: Enter a description for this configuration.
- **Owner**: Use the Owner drop-down list to select an owner for the configuration.
- **Support**: Use the Support the drop-down list to select a support person for the configuration.

Before you can assign an owner or support person, the person must be added to SOA PE.

- **Route Propagated Alerts to Category**: Use the drop-down list to select a default alert category to be used for this configuration. If you are not sure which category to use, keep the `Default` category.

  Click **Edit Categories** to edit an alert category or create additional alert categories. For more information on creating alert categories and alert recipients, see chapter 6 "Using Alert Notifications."

- **Operation Name**: Use the drop-down list to select an operation that appears in the Web service WSDL file or use the text box to enter the operation name as it appears in the WSDL file.

4 Click **Save**. The Business Service View screen opens and the operation is listed within the model section.

### Task 5: Designate the Entrypoint

A business service can contain several different IT configurations and resource configurations. Any of the resource configurations can be designated as the entrypoint. Entrypoints are used in SOA PE to designate the resource configuration that is the most important. After an entrypoint is assigned, the user can set policies on the business service, and SOA PE is able to filter and propagate alerts accordingly.

A service model can become very complex depending on the number of assets that are defined in the model. By designating a resource configuration as an entrypoint, all relevant alerts are propagated to the Business Service. In other words, an entrypoint acts as a designated alert filter mechanism. It is important to note that in a given business service only one resource configuration can be designated as the entrypoint.

To designate or change the entrypoint for a business service, follow these steps:

1. From the business service view screen, expand the **Edit** drop-down list next to the name of the business service in the Model section. Select **Select Entrypoint**.
2. On the next screen, select the radio button for the resource configuration that should be the entrypoint.
3. Click **Save**.

## Selecting Dependencies for a Business Service

This section describes how to add or remove explicit dependencies from a business service's model definition. The dependencies include other business services, configurations, and resources that have already been added as part of the business service definition process. Dependencies allow alerts to be propagated from a dependency to its business service. In the absence of explicit dependencies, for example, alerts are propagated from service configurations to their contained policy enforcement intermediary instances configurations, and not vice-versa. Therefore explicit dependencies are needed for a business service to receive alerts from the contained policy enforcement intermediary instances.

A business service can use, or be used by, any number of other business services. The relationship between business services can be expressed as A uses B and B is used by A. This relationship has to be known and declared in the business service model and represents a dependency relationship between the Web services in one business service to that in another. This dependency relationship is used for impact analysis and root cause analysis.

Monitoring a business service that uses other business services lets you perform root cause analysis to determine which related business services are degrading. Conversely, monitoring a business service that is used by other business services lets you perform impact analysis to determine how a business service's performance affects related business services.

To add or remove dependencies from a business service's model definition:

1. From the Business Service List screen, select a business service to view its details. The Business Services View screen opens for the selected business service.

2 From the Model section, use the **Edit** drop-down menu next to the business service's name and click **Select Dependencies**.

3 From the list of resources, click the check box to add or remove a resource. A check mark next to the resource indicates that it is currently a dependency of the business service.

4 Click **Save**. The Business Services View screen opens and the model section is updated to display the explicit dependencies.

# Adding Routing Targets

You can add additional endpoints to a brokered service. The endpoints must first be deployed to a policy enforcement intermediary that is registered as a PEP and bound to a business service.

Routing targets are automatically added to the intermediary's list of available endpoints able to service a request at runtime. When an intermediary service contains multiple endpoints, requests are dispatched to the endpoints using a round robin load balancing scheme.

To add a routing target, follow these steps:

1 From the Business Services List screen, expand a business service to view its contained configurations.

2 From a Web Service Intermediary configuration, click the Web service configuration to which you want to add additional routing targets. The View Web Service Configuration screen opens.

3 From the Web Service Configuration section, click **Edit**. The Edit Web Service Configuration screen opens.

4 Click to select the Endpoint Update Policy check box. A check indicates that the routing feature is enabled.

5 Click **Save**. The View Web Service Configuration screen reopens.

6 From the Routing Table section, click **Edit**. The Select Resources screen opens. The screen lists all the Web services that are in the business service. The Web services are organized by type.

7 From the list of Web services, click the check box to add the Web services as a routing target. A check mark indicates an active routing target. You can also select routing targets from Systinet from the Routing Table dialog box and then click **Provision** to start provisioning the modified service. This opens the *Specify End Point Related Configuration for Load Balancing and Routing* screen of the service provisioning wizard. SOA PE allows you to only update this page of the provisioning wizard during service provisioning.

8 Click **Save**. The Web Service View screen opens and the Routing Table lists all routing targets.

# Assigning Owner and Support Roles

Business services, configurations, and resources can be assigned to an owner or a support person. Once assigned, you can filter business services and configurations based on the owner or support person.

Before you can assign an owner or support person, the person must be added to SOA PE. See the “Adding People” section in Chapter 2.

## Business Service Roles

To assign owner and support roles for a business service, follow these steps:

1. From the Business Service List screen, select a business service to view its details. The Business Services View screen opens for the selected business service.
2. Click the **Edit** link in the Business Service section. The Edit Business Service screen opens.
3. Use the Owner drop-down list to select an owner for the business service. The owner is generally responsible for lifecycle management. If needed, click the check box for sending email alerts to this person.
4. Use the Support the drop-down list to select a support person for the business service. The person or group is responsible for supporting deployed instances of the service. If needed, click the check box for sending email alerts to this person.
5. Click **Save**. The Business Services View screen reopens.
6. From the SOA PE web interface, click **Business Services**. The Business Services List screen opens.
7. Use the **Filter ‘By Person’ and ‘By Role’** drop-down lists to filter the list based on business service owners and roles.

# Publishing Business Services to a Registry

The SOA PE web interface can publish web services associated to a business service to a registry. To use this feature, you must have a Systinet registry and configure the registry with the SOA PE server.

To publish a business service to the registry:

1. From the Business Service List screen, select a business service to view its details. The Business Service’s View screen opens.
2. Click the **Publish** link next to the Business Service section. The Publish Business Service screen opens.
3. Complete the following fields:

— **Web Service Provider**: Select the provider for the Web service. This is the Web service managed by the SOA PE web interface business service. Select a Business Entity name from the drop-down list of Business Entities in the registry.

— **Web Service Name**: The name of the Web service.

4 Click **Publish**. The Business Service View screen opens. If an error occurs, the error is shown in red at the top of the Publish Business Service screen.

The registry entities corresponding to Web services are not deleted when the Web service is deleted.

### JMS Support

The UDDI feature also supports business services that include JMS resources. The following support is included for JMS:

- A `TModel` for the JMS transport is published with the name `hp-com:jms`.
- A functional business service binding template contains the following:
  - An access point with the following attributes:

    `destinationStyle`, `initialContextFactory`, `jmsVendorURI`, `jndiConnectionFactoryName`, `jndiDestinationName`, `jndiProviderUrl`
  - A binding `TModel` with a keyed reference for the JMS transport in the category bag.

## Re-using a Business Service

The SOA PE web interface lets you import and export business services. This simplifies and saves time when moving business services between environments (for example, development to production).

SOA PE supports importing a business service that was created using the previous version. The business service is automatically updated to the new version compliant business service. Refer to the *SOA PE Installation Guide* for more information about importing business services from the previous release. In particular, the import updates the following:

- Business services and their contained intermediary configurations, container configurations, database service configurations
- Business service relationships

### Exporting a Business Service

To export a business service, follow these steps:

1 From the Business Service List screen, select a business service to view its details. The Business Services View screen opens for the selected business service.

2 Click the **Export** link in the Business Service section. The Export Business Service screen opens.

3 Click **Download**.

4 The file download dialog box for your browser opens.

5 Use the download dialog box to save the business service.

## Importing a Business Service

To import a business service:

1 From the Business Service List screen, click the **Import** link. The Import Business Services screen opens

2 Use the **Browse** Local Business Service Jar field to enter the location to a business service JAR file.

Or:

Use the Specify Remote Business Service Jar URL field to enter the URL to a business service JAR file.

3 Click **Import**. The Business Service List screen opens and the business service is listed. It may take several seconds for the business service to be deployed and displayed on the list.

# Deleting a Configuration

You can delete configurations without deleting the business service. When you delete a configuration, all pending alerts for the configuration are acknowledged, UDDI entries in the UDDI registry are deleted. Any configuration contained in this configuration is also removed.

To delete a business service configuration, follow these steps:

1 Click the **Business Services** tab to view the Business Services List screen.

2 From the Business Service List screen, expand a business service to view its configurations.

3 Click the **Configuration View** tab to display configuration for the selected configuration.

4 From the Configuration View screen, click **Remove** next to the Configuration section. The Remove screen opens.

5 Click **Remove**. The Business Service View screen opens and the configuration is no longer listed as part of the business service.

# Deleting a Business Service

You can delete a business service. When you delete the business service, business service configurations are deleted, pending alerts for this business service are acknowledged, dependencies on the business services are removed. If the business service was published to a UDDI registry, it is deleted from the registry. The functional business service UDDI registry entries are not deleted.

To delete a business service, follow these steps:

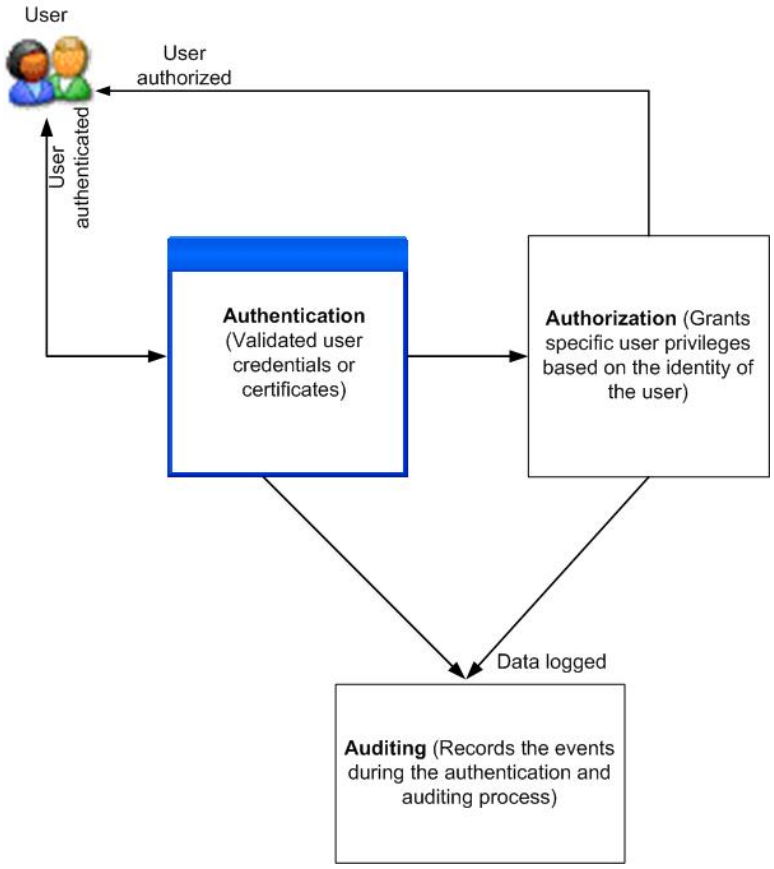
1. Click the **Business Services** tab to view the Business Services List screen.
2. Click the **Configuration View** tab to display configuration for the selected business service.
3. From the Business Services View screen, click **Remove** next to the Business Service section. The Delete Business Service screen opens.
4. Click **Remove**. The Business Service List screen opens and the business service is no longer listed.

7

# Security in SOA PE

This section discusses about the security implementation architecture in SOA PE. This section also provides information about configuring a third-party security provider for SOA PE.

The security implementation architecture in SOA PE includes the processes of authentication, authorization, and audit (widely referred to as AAA) for a user as shown in the following diagram.



Authentication is the process of verifying the identity and authenticity of a user. This process is usually performed either through a user name and password that the user provides or through digital certificates exchanged between the server and client.

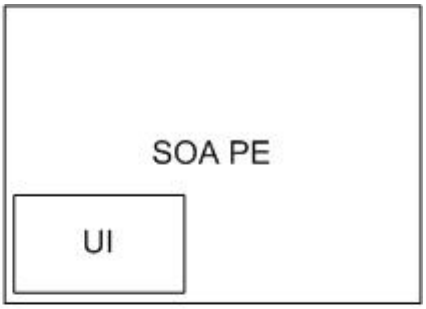
Authorization is the process of granting user privileges to authenticated users.

Auditing is the process of recording the details during the authentication and authorization process.

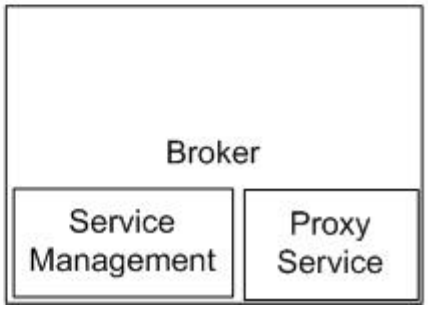
# SOA PE Components and Interfaces

This section provides you a broad-level overview of the various components that comprise SOA PE. The following diagram illustrates the components.

**SOA PE Components**




**SOA PE Broker Components**




The SOA PE consists of the following components:

- SOA PE: Allows you to implement run-time policy enforcement in a SOA environment.
- SOA PE Broker: Performs the role of a gateway between the service provider and the service consumer.

The SOA PE includes the following interface:

- SOA PE user interface: this graphical user interface allows you to log into the SOA PE and perform actions such as configure and view policy enforcement, view the lifecycle state of a web service, generate reports, and so on. You can log on to this interface using a valid user name and password.

The Broker includes the following interfaces:

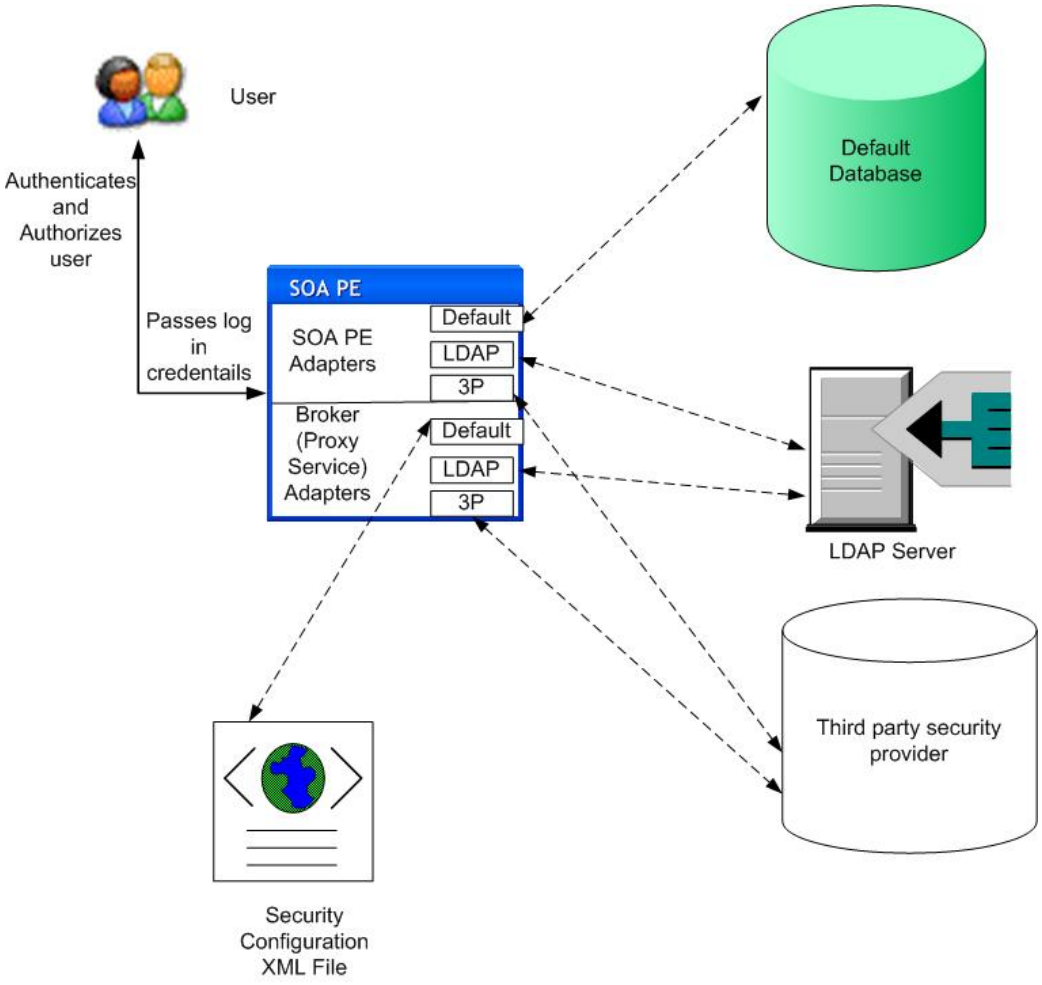
- Service management interface: this interface provides operations that you can use to expose policy enforcement information, deploy policies for enforcement, maintain information about service proxies, and so on.
- Service proxy: this interface (service) on the Broker, as the name denotes, acts as a proxy service (gateway) between the service provider and the service consumer.

## Security Providers Supported by SOA PE

SOA PE supports the following security providers. You can configure any one of the following security providers:

- Default security provider: This is the default configuration in SOA PE for authenticating and authorizing users. You can also configure XACML with the default security provider for authorization.
- Authentication and authorization using LDAP: You can configure an LDAP server to be the security provider and register it with the security module of SOA PE.
- Authentication and authorization using a third party security provider: You can configure any third party security provider to be the security provider and register it with the security module of SOA PE.

The following diagram shows you the adapters present for each of these security providers.




Refer to the following sections for more information about the security mechanisms and to know the interfaces that support these security mechanisms.

## Default Security Provider

SOA PE Server uses the local configuration and the local database to provide authentication. Refer to the *Security* section in *Chapter 1* of the *SOA PE User Guide* for more information. SOA PE Server authorizes users based on the role-based access control list present on the SOA PE Server.

The SOA PE Broker uses the security configuration file to authenticate users. The SOA PE Broker does not support authorization of users.

Do as follows to configure XACML for authorization with the default security provider:

Open the file mipServer.xml present under the <install_dir>\conf\networkservices directory using any text editor and make the following changes:

<entry name="com.hp.mip.security.providers">default</entry>

<entry name="com.hp.mip.security.provider.authorization"> default </entry>

<entry name="com.hp.mip.security.provider.authentication"> default </entry>

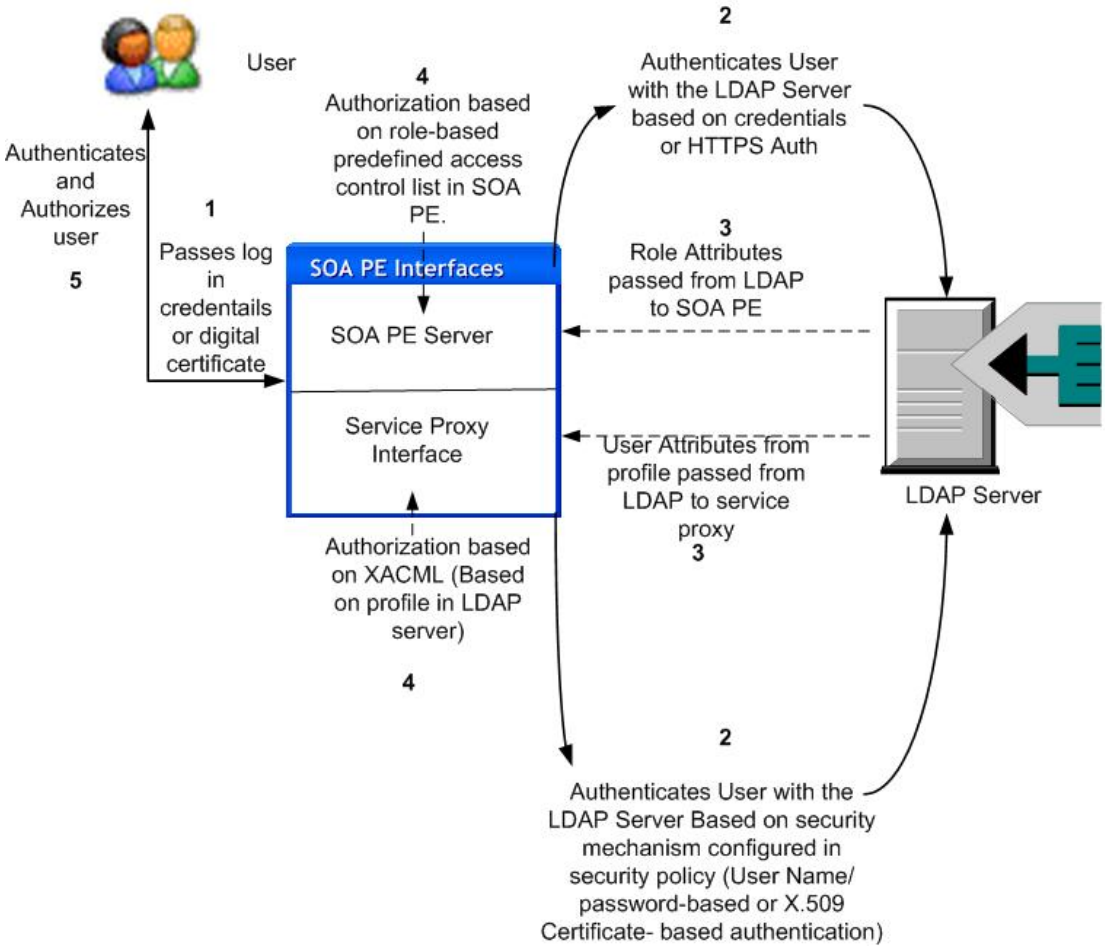
<entry name="com.hp.mip.security.provider.console"> default </entry>

<entry name="com.hp.mip.security.provider.auditing"> default </entry>

<entry name="com.hp.mip.security.provider.default.authorizationPolicy">XACML</entry>

## Using the LDAP Security Provider

Refer to the following diagram based on the numbers provided to understand how an LDAP security provider works for the SOA PE interfaces.




SOA PE accepts a digital certificate or user credentials from the user and passes it to the LDAP adapter for verification with the LDAP server. The LDAP server verifies the user credentials or the certificate and allows or denies access based on the authenticity of the certificate or the user credentials. SOA PE currently supports one-way and two-way HTTPS certificate-based authentication.

For certificate-based authentication, the authentication is based on X.509 certificates (SSL-based authentication). For SSL-based authentication, SOA PE passes the X.509 certificate from the client to the LDAP adapter or the security provider adapter. The adapter passes the certificate to the respective security provider configured for certificate validation to authenticate the user. The adapter compares the certificate passed with the certificate present on the LDAP server for authenticity. SOA PE supports one-way SSL authentication and two-way SSL authentication. In two-way SSL authentication, after validating the certificate from the client, the security provider sends a valid certificate to the client for validation.

After a user passes the credentials and is authenticated, SOA PE authorizes the user based on the role-based access control list in SOA PE. You must make sure that the user name you use to log into SOA PE user interface is the same as the user name specified in with the security provider for role-based access.

For the Service Proxy on the Broker, authorization is performed using the Extended Access Control Markup Language (XACML) policies deployed at the Broker. This is based on the user profile on the LDAP server and the resources being accessed.

The Broker implements security in the service proxy at the transport and message levels. Apart from authentication and authorization, the service proxy also supports auditing. The proxy service logs the audit messages to the security provider. Refer to *Appendix F Configuring LDAP Authorization* for more information.

## Configuring SOA PE to Use LDAP Security Provider and use XACML for Authorization

Perform the steps listed under the following sections to configure SOA PE Server and Service Proxy (Broker) to use LDAP as the security provider and use XACML as the authorization provider.

### *SOA PE Server*

1 Open the file mipServer.xml present under the <install_dir>\conf\networkservices directory using any text editor.

2 Replace the following lines in the file as shown in the table below

| **Existing Lines** | **New Lines** |
|---|---|
| <entry name="com.hp.mip.security.providers">default</entry> | <entry name="com.hp.mip.security.providers">default;ldap</entry> |
| <entry name="com.hp.mip.security.provider.console">default</entry> | <entry name="com.hp.mip.security.provider.console">ldap</entry> |

3 Add the following lines to the file if the file does not include these lines

```
<entry
name="com.hp.mip.security.provider.ldap.class">com.hp.wsm.sn.common.security.provider.WsLdapXacmlSecurityProvider</entry>

<entry name="com.hp.mip.security.provider.ldap">ns</entry>
```

4 Save and close the mipServer.xml file.

### SOA PE Broker

1 Open the file mipServer.xml present under the <install_dir>\conf\broker directory using any text editor.

2 Replace the following lines in the file as shown in the table below

| Existing Lines | New Lines |
|---|---|
| <entry name="com.hp.mip.security.providers">default</entry> | <entry name="com.hp.mip.security.providers">default;ldap</entry> |
| <entry name="com.hp.mip.security.provider.authorization">default</entry> | <entry name="com.hp.mip.security.provider.authorization"> ldap </entry> |
| <entry name="com.hp.mip.security.provider.authentication">default</entry> | <entry name="com.hp.mip.security.provider.authentication"> ldap </entry> |
| <entry name="com.hp.mip.security.provider.console">default</entry> | <entry name="com.hp.mip.security.provider.console"> ldap </entry> |
| <entry name="com.hp.mip.security.provider.auditing">default</entry> | <entry name="com.hp.mip.security.provider.auditing"> ldap </entry> |

3 Add the following lines to the file if the file does not include these lines

```
<entry
name="com.hp.mip.security.provider.default.class">com.hp.wsm.sn.commo
n.security.provider.BrokerDefaultSecurityProvider</entry>

<entry
name="com.hp.mip.security.provider.ldap.class">com.hp.wsm.sn.common.s
ecurity.provider.WsLdapXacmlSecurityProvider</entry>

<entry name="com.hp.mip.security.provider.ldap">broker</entry>
```

4 Save and close the mipServer.xml file.

## Using XACML-based Authorization

You must complete this task before performing the configuration task listed in this section. In this configuration task, you can specify the access permissions for a provisioned Web service or operations. You can author XACML policies at the SOA PE UI and deploy these policies to the Broker instances. If an instance is not available, you can redeploy the policy to the instance. These policies are deployed at all instances of the Intermediary group. During runtime, these policies are used for authorization. You can grant permissions based on the user, group, or host as follows:

## Adding a XACML-based Authorization Configuration

1 Log in to the SOA PE UI as an administrator

2 Click **Policy Enforcement Points** from the **View** drop-down menu on the left pane. This displays the Policy Enforcement Point Summary page.

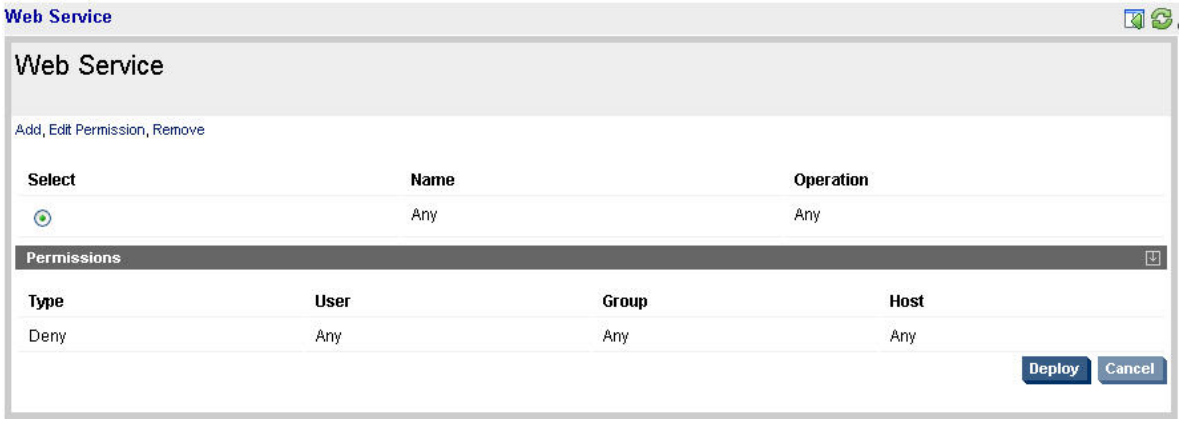
3 Click the Service Proxy that you want to configure. This displays the Policy Enforcement Intermediary Group page for the Service Proxy.

4 Click **View Authorization Policies**. This displays the Web Service page as shown below.

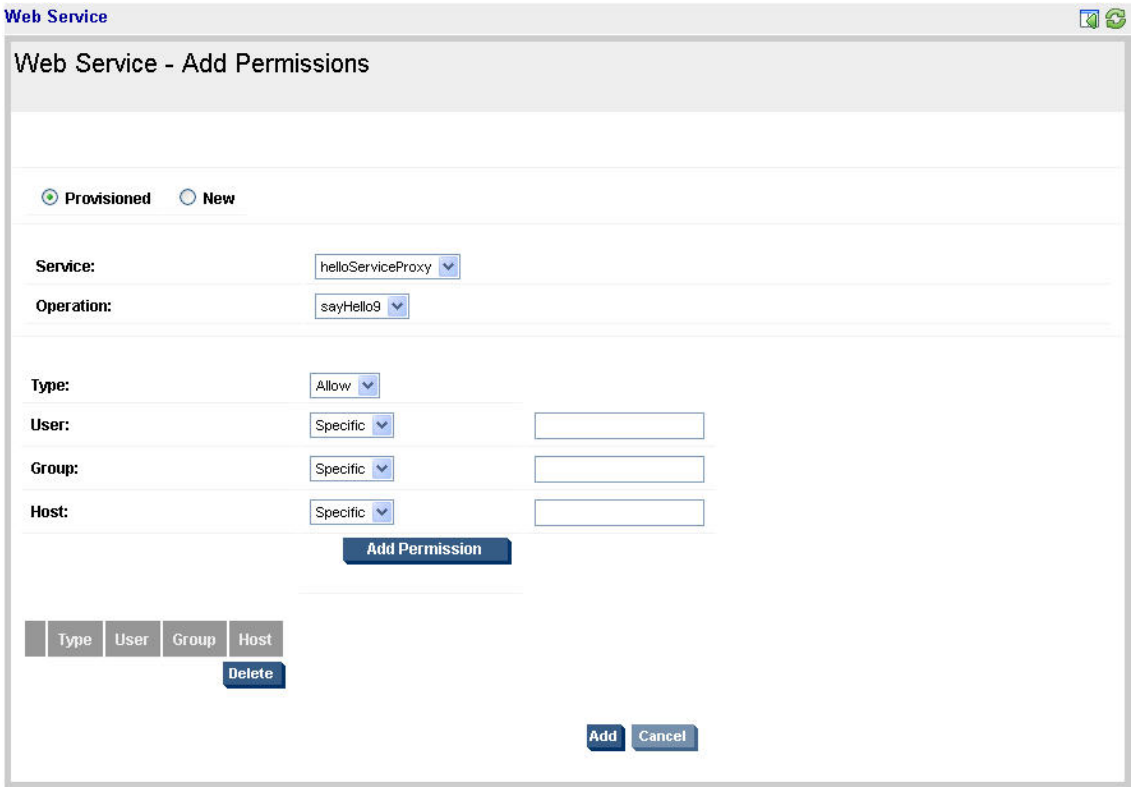
**Note**:

i. When the transport level security is used, authorization policies will not be supported at operations level.

ii. Authorization will always fail if the password is blank




5 Click **Add** to add a new authorization configuration for a service. This displays the Web Service – Add Permissions page as shown below.




6 Select one of the following options:

- **Provisioned**: to configure authorization for a provisioned service
- **New**: to configure authorization for a new service

7 Select the configuration from the drop-down list provided for the following options. In this step it is assumed that you have selected Provisioned in the previous step:

- **Service**: the provisioned service for which the configuration is applicable
- **Operation**: the operation to which this configuration is applicable. You can select **Any** if you want this configuration to be applicable to any of the operations associated with this service.

> For a New service, you must specify the **Service Name**, **Namespace**, and **Operation Name** in the respective boxes.

- **Type**: Select **Allow** or **Deny** to specify the type of permission for the service.
- **User**: Select **Specific** to configure a specific user for whom you want to configure authorization. Select **Any** if you want to configure authorization for any of the configured users. You must make sure that the user names match with the user names in the profile on the LDAP server.
- **Group**: Select **Specific** to configure a specific group of users whom you want to configure authorization. Select **Any** if you want to configure authorization for any of the configured groups of users. You must make sure that the group names match with the group names in the profile on the LDAP server.

— **Host**: Select **Specific** to configure a specific host for whom you want to configure authorization. The specified host name is matched with the host from where the request originates. Select **Any** if you want to configure authorization for any of the configured hosts. You must make sure that the host name matches with the host name in the profile on the LDAP server.

8 Click **Add Permission**. This displays the configuration at the left-hand corner of the page. You can click **Delete** if you want to remove this configuration.

9 Click **Add**. This adds the configuration for the service and displays the Web Service page.

10 Click **Deploy** to deploy the XACML policy on the Service Proxy instance. Click **OK** on the confirmation message that you get on a successful deployment.

### Modifying a XACML-based Authorization Configuration

Perform step 1 through step 4 in the previous section and do as follows to modify an existing XACML-based authorization configuration:

1 Select the service configuration that you want to modify from the Web Service page and click **Edit Permissions** from the Web Service page. This displays the Web Service Edit Permissions page.

2 Modify the **Type**, **User**, **Group**, and **Host** options for the service.

3 Click **Add Permission** and click **Modify**. This displays the Web Service page.

4 Click **Deploy** to deploy the modified XACML policy on the Service Proxy instance.

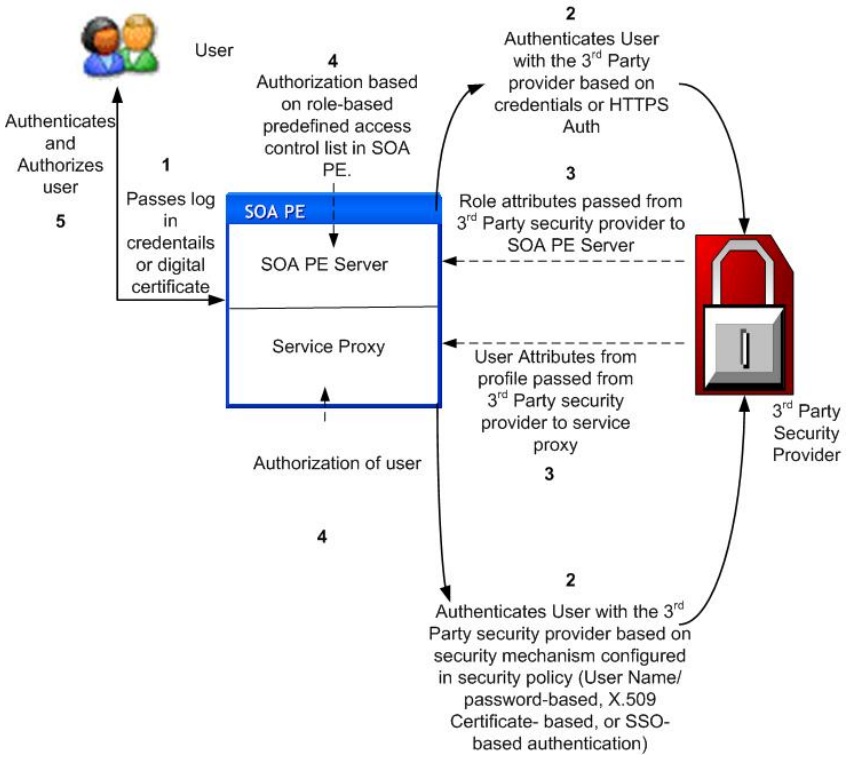
### Removing a XACML-based Authorization Configuration

Perform step 1 through step 4 in the previous section and do as follows to remove an existing XACML-based authorization configuration:

Select the service configuration that you want to remove from the Web Service page and click **Remove**.

## Using a Third Party Security Provider for SOA PE

You can integrate any third party security provider with SOA PE. Refer to the following diagram to know how SOA PE works with a third party security provider.




As shown in the diagram above, you can use the SOA PE Server UI to authenticate the user. After authentication at the third party security provider, SOA PE Server performs authorization through the role-based access control list at the SOA PE Server.

For service proxy, SOA PE Broker authenticates the user with the third party security provider using the user name and password, digital certificates, or Single Sign On (SSO) methods. After authenticating the user, the service proxy authorizes the user.

Refer to *Appendix D Creating a Third Party Security Provider* for information about creating a third party security provider.

The following section provides information about configuring security using SSL and digital certificates for SOA PE.

# Using SSL for the Management Channel

This section describes on how to secure the management channel and the management components that are used in SOA PE. You should be familiar with general security principles and SSL security before attempting any of the tasks in this section. In particular, you should be familiar with Key Stores and you should have SSL certificates, including Certificate Authority (CA) root certificates, for the servers being used to implement the SOA PE solution.

This section does not include instructions for securing the application channel.

# Overview

The SOA PE management channel contains sensitive data about Web services that are being managed. The data includes performance data, auditing data, and business content data. More importantly, the management channel exposes interfaces that are used to interact with a Policy Enforcement Intermediary and its deployed services. The potential for security violations and malicious attacks does exist and should be considered when setting up the WSM solution.

The management channel is secured at the transport layer (HTTP) using SSL. SSL provides the means to implement authentication, confidentiality, and data integrity. SSL is used to secure the management communication between SOA PE and WSM Intermediary and is also used to secure communication to SOA PE web interface and Broker Configurator.

# Setting Up SSL

This section provides instructions that are used to implement SSL security between the management components of the WSM solution. Using SSL ensures that management data is secured and that the SOA PE web interface and Broker Configurator are accessed in a secure manner.

## Assign Key Stores and Trust Stores

The steps in this section detail how to assign Key Stores and Trust Stores for the various management servers used in the WSM solution. Before you complete the instructions in this section, make sure that each server participating in the WSM solution contains an SSL certificate which has been verified by a Certificate Authority (CA).

See Appendix A for information on creating Java Key Stores and server certificates.

## SOA PE

The steps below detail how to assign a Key Store and Trust Store for use by SOA PE. SOA PE acts as an HTTP client. Therefore, its Trust Store must contain the CA root certificate for each server participating in the WSM solution. If each server is verified by the same CA, only a single CA root certificate is required.

To configure a Key Store and Trust Store for the SOA PE server, follow these steps:

5 Stop SOA PE if it is currently started.

6 Use a text editor to open <*install_dir*>`\conf\mipServer.xml`.

7 Use the following example and enter the properties for your Key Store and Trust Store. Each property is described following the example.

```
<entry name="com.hp.mip.security.server.keystore.type">
   jks</entry>
<entry name="com.hp.mip.security.server.keystore.location">
   C:\\temp\\MyKeystore.jks</entry>
<entry name="com.hp.mip.security.server.keystore.password">
   MyPassword</entry>
<entry name="com.hp.mip.security.server.privatekey.alias">
   MyAlias</entry>
<entry name="com.hp.mip.security.server.privatekey.password">
   MyPassword</entry>
<entry name="com.hp.mip.security.server.truststore.type">
   jks</entry>
<entry name="com.hp.mip.security.server.truststore.location">
   <jdk_install>/jre/lib/security/cacerts</entry>
<entry name="com.hp.mip.security.server.truststore.password">
   MyPassword</entry>
```

- **Keystore Type**: The entry can either be a Java Key Store (`jks`) or a PKCS12 Key Store (`pks`).
- **Keystore Location**: Enter the full path to the Key Store.
- **Keystore Password**: Enter the password for the Key Store.
- **Private Key Alias**: Enter the private key alias for the Key Store.
- **Private Key Password**: Enter the private key password for the Key Store.
- **Truststore Location**: Enter the full path to the Trust Store.
- **Truststore Password**: Enter the password for the Trust Store.
- **Truststore Type**: The entry can either be a Java Key Store (`jks`) or a PKCS12 Key Store (`pks`).

If your CA trusted roots certificates are stored together with the server certificate in the Key Store, enter the same Key Store values for the Trust Store. In such scenarios, the Key Store is considered the Trust Store.

8 Save and Close the file.

## Policy Enforcement Intermediary

The steps below detail how to assign a Key Store and Trust Store for use by the Intermediary. If the Intermediary is co-located with the SOA PE server, they share the same Key Store and Trust Store. Assigning a Key Store and Trust Store for the SOA PE also assigns the Key Store and Trust Store for the Intermediary.

To assign a Key Store and Trust Store for the Intermediary, follow these steps:

1 Start the Broker Configurator.

2 From the Configurator's main tool bar, click **SSL Settings**. The SSL Settings screen opens.

3 Set the following properties:

- **Keystore Location**: Enter the full path to the Key Store (for example, *C:\\temp\\MyKeystore.jks*).

- **Keystore Password**: Enter the password for the Key Store.
- **Keystore Type**: The entry can either be a Java Key Store (`jks`) or a PKCS12 Key Store (`pks`).
- **Private Key Alias**: Enter the private key alias for the Key Store.
- **Private Key Password**: Enter the private key password for the Key Store.
- **Truststore Location**: Enter the full path to the Trust Store (for example, *<jdk_install>/jre/lib/security/cacerts*).
- **Truststore Password**: Enter the password for the Trust Store.
- **Truststore Type**: The entry can either be a Java Key Store (`jks`) or a PKCS12 Key Store (`pks`).

If your CA certificates are stored together with the server certificate in the Key Store, enter the same Key Store values for the Trust Store. In such scenarios, the Key Store is considered the Trust Store.

4 From the bottom of the screen, click **Save**.

## Configure SSL Settings

The steps in this section detail how to configure SSL on the management servers that are participating in the WSM solution. This typically includes enabling an SSL implementation and defining an HTTPS port.

### SOA PE

To configure SSL settings in SOA PE, follow these steps:

1 Use a text editor to open *<install_dir>*`\conf\networkservices\mipServer.xml`.

2 Enter the following properties:

```
<entry name="com.hp.http.server.securePort">port_number</entry>
<entry name="com.hp.mip.security.server.webapps.secure">
   true</entry>
<entry
name="com.hp.mip.security.server.management.webapps.secure">true</
entry>
```

- **Secure Port**: SOA PE secure port that is used to accept HTTPS requests from the SOA PE web interface. Any open port can be used.
- **Webapps Secure**: Enables SSL on the SOA PE server. Valid entries are **true** and **false**.

3 Save and close the file.

4 Start SOA PE.

### Policy Enforcement Intermediary Management Channel

To configure management channel SSL settings in the policy enforcement intermediary, follow these steps:

1. Stop the policy enforcement intermediary if it is currently started.
2. Use a text editor to open *<install_dir>*`\conf\broker\mipServer.xml`.
3. Set the `com.hp.mip.security.server.management.webapps.secure` element to `true`.

   ```
   <entry name="com.hp.mip.security.server.management.webapps.
      secure">true</entry>
   ```

4. Specify a port value for the `com.hp.http.server.secureManagementPort` element. Make sure the port is not being used by any other application on your system.

   ```
   <entry name="com.hp.http.server.secureManagementPort">443</entry>
   ```

5. Save and close `mipserver.xml`.
6. Start the policy enforcement intermediary.

### Broker Configurator

To configure the Broker Configurator to use SSL, follow these steps:

1. Stop the policy enforcement intermediary if it is currently started.
2. Use a text editor to open *<install_dir>*`\conf\broker\mipServer.xml`.
3. Set the `com.hp.mip.security.server.webapps.secure` element to `true`.

   ```
   <entry name="com.hp.mip.security.server.webapps.secure">
      true</entry>
   ```

4. Specify a port value for the `com.hp.http.server.securePort` element. Make sure the port is not being used by any other application on your system.

   ```
   <entry name="com.hp.http.server.securePort">-1</entry>
   ```

5. Save and close `mipserver.xml`.
6. Start the policy enforcement intermediary.

## Registering a Secure Policy Enforcement Intermediary

Managed policy enforcement intermediaries that run on a secure server are registered with SOA PE by using the SOA PE web interface in the same manner as non-secure managed policy enforcement intermediaries. However, because a managed policy enforcement intermediary runs on a secure server, the server's secure port must be used.

As SOA PE acts as an HTTP Client, its Trust Store must contain the CA root certificate for each managed policy enforcement intermediaries server participating in the WSM solution. If each server is verified by the same CA, only a single CA root certificate is required.

To register a secure managed policy enforcement intermediary, follow these steps:

1. Make sure the managed policy enforcement intermediary that you want to register is started.

2 From the SOA PE web interface, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.

3 Select the PEP you want to contain the Policy Enforcement Intermediary. The PEP View screen opens for the selected PEP.

4 From the Contained Policy Enforcement Intermediary instances section, click the **Add** link. The Add Policy Enforcement Intermediary screen opens.

5 From the **Type** drop-down box, select the type of resource you want to register.

6 Using the fields provided, enter the host and secure port where the managed Policy Enforcement Intermediary is installed.

7 Click to select the **SSL** check box.

8 Click **Add**. The Add Policy Enforcement Intermediary screen reopens and lists the Web services that were discovered in the managed Policy Enforcement or Intermediary.

9 Click **Add**. The Policy Enforcement Intermediary screen opens and lists the resources that are now registered in the SOA PE. The Management Interface (WSDL) field indicates an HTTPS URL.

10 Repeat this procedure to register additional secured managed Policy Enforcement Intermediary.

If secure mode is turned on, then Broker allows access only through the secure channel. SOA PE does not deploy web applications on non secure servers if the secure mode is turned on.

## Accessing the SOA PE User Interface

When using SSL, the SOA PE user interface is accessed through the secure port of SOA PE (see “Configuring SSL Settings” above). Any browser used to access the web interface must contain a CA root certificate from the CA that was used to verify the SOA PE server’s SSL certificate. See your browser’s documentation for information on installing a CA’s trusted root certificate.

To access the SOA PE user interface:

1 Open a Browser.

2 Enter the following URL and substitute <*host*> with the DNS host name where the SOA PE server is running and <*secure_port*> with the server’s secure port:

`https://<host>:<secure_port>/bse`

8

# Using Custom Intermediary Services

This chapter explains how to use custom intermediary services. The instructions include tasks for creating and configuring a custom intermediary service definition as well as adding handlers to a custom intermediary service. In most situations, a simple intermediary service provides enough functionality to manage a Web service. However, there are situations when a custom intermediary service can be used to allow greater control of the service definition and access to custom WSM functionality.

## Overview

Custom intermediary services are similar to simple intermediary services in that they act as proxies to a Web service endpoint and provide WSM capabilities in the form of handlers that are organized in a handler chain. Any handler available for a simple intermediary service is also available for a custom intermediary service. Simple intermediary services use a predefined set of handlers, while custom intermediary services are boundless. The handler chain can be customized to include a broad range of handlers (including custom handlers). The ordering of the handlers in the handler chain can be configured.

The benefits of using a custom intermediary service include the following:

- Maximum control when assigning handlers and creating the handler chain
- Support for a broad range of handlers
- Support for custom handlers
- Reuse of handlers within a handler chain (that is, multiple business metric handlers)

## Convert a Simple Intermediary Service

Custom intermediary services are created by first creating a simple intermediary service (see Chapter 10) and then converting the simple intermediary service to a custom intermediary service.  You can convert SOAP/HTTP and XML/HTTP simple services to custom services.

To convert a simple intermediary service to a custom intermediary service, follow these steps:

1 From the Intermediary Services list, find the intermediary service that you want to convert.

2 From the Action column, click **edit**. The Edit Service screen opens.

3 Click **Convert**. The Edit Custom Service screen opens and lists the handlers for the custom service. Any handlers that were configured for the simple intermediary service are also configured for the custom service. Several default handlers, which were part of the simple intermediary service but not previously visible, are listed.

4 Click **Save**. The Intermediary Service screen opens and the intermediary service is automatically deployed. The deployment is complete when the status changes to `Operational`. The Style field indicates that the intermediary service is `Custom`.

# Adding Handlers

Using custom intermediary services provides greater control when adding handlers for an intermediary service. Handlers are assigned to a custom intermediary service using the Broker Configurator's Edit Custom Service screen.

To add handlers to a custom intermediary service, follow these steps:

1 From the Service list, find the custom intermediary service that you want to edit. The Style field indicates that the intermediary service is `Custom`.

2 From the Action column, click **edit**. The Edit Custom Service screen opens and displays the handlers currently assigned to the intermediary service.

3 Use the Add a new handler drop-down list to add a handler. The handler is added to the list of handlers. Repeat this step to add additional handlers. Please contact support to get additional details regarding Handlers.

4 Click **Save**. The Intermediary Services screen opens and the intermediary service is automatically deployed. The deployment is complete when the status changes to `Operational`.

## Adding Custom Handlers

Custom intermediary services let you add your own custom handlers to an intermediary service's handler chain. To add a custom handler, you must first create the custom intermediary service and then edit the service's definition file located in the intermediary service jar file.

To add a custom handler, follow these steps:

1 Uncompress <*install_dir*>`\conf\broker\`<*intermediary_service_name*>`.jar`.

2 Using a text (or XML) editor, open `service.xml`.

3 Under the `<service>` element, add a `<handler>` element and include the fully qualified class name. For example:

```
<handler classname="com.company.HandlerClass" />
```

4 If the handler requires any properties, add them as elements under the handler class. For example:

```
<handler classname="com.company.HandlerClass" >
   <property1>foo</property1>
   <property2>
      <property name="foo" value="bar" />
   </property2>
   <ns1:property3>foo</ns1:property3>
</handler>
```

If the property uses a namespace, you must declare the namespace as an attribute of the `<service>` element before using the namespace (for example, `xmlns:ns1="com.company"`).

5 Save and close `service.xml`.

6 Place the custom handler class and any dependent classes in the same directory as `service.xml`.

7 Re-jar the intermediary service including the custom handler class and any dependent classes.

8 Place the jar in *<install_dir>*`\conf\broker\`. The intermediary service is automatically deployed. You can use the Broker Configurator to verify that the jar has been deployed. The intermediary service is listed on the Service List and its status is `Operational`.

# Defining Service Providers for Custom Web Services

The intermediary allows you to route a SOAP request to an appropriate endpoint based on the context or content of the message. Intermediary can be configured to do this routing as follows:

- When you create an intermediary Web service, if the WSDL used contains multiple end points, the intermediary lets you classify these endpoints.
- The definition for this classification is provided as properties of the Classifier handler. The following properties must be specified:
  - XPath expression – The XPath expression that should be evaluated on the incoming request
  - Context – This field indicates whether the expression should be evaluated on the transport context or content of the message. When transport is selected, the XPath expression is evaluated using XML in the following format:
    - <header>
    - <header-name1>value</header-name1>
    - <header-name1>value</header-name1>
    - </header>

The variables <header-name1> and <header-name2> represent the HTTP headers when HTTP transport is used or JMS headers when JMS transport is used. When HTTP is used as transport, the XML file also contains the following details:

- <TCP_HOST>source_host</TCP_HOST>
- TCP_PORT>source_port</TCTP_PORT>

When a request is sent to an intermediary Web service, based on the content or context of the message, the intermediary can route the request to the appropriate endpoint. The definition for this classification is provided by using classification handlers. An incoming request can be classified.

If you enable XSLT transformation, the intermediary transforms the classified message. See the XSLT Transformation section for additional information about XSLT transformation. The intermediary then forwards the request based on the specifications in the classifier to the corresponding endpoint. You must perform the steps in the following section to enable content-based routing. See Enabling Content-based Routing for Intermediary Web Services for information on configuring content-based routing for intermediary Web services. Refer to the following scenario for additional information.

Consider a banking Web service where you must administer requests from customers belonging to the following classifications:

- High loan request ($25,000 and above)
- Medium loan request (up to $25,000)

The banking Web service must forward requests from these two types of loan requests automatically to the corresponding endpoints that handle specific types of loan requests. For example, according to the bank loan guidelines, a medium loan request does not need approval from the higher authorities in the bank.  A high loan request needs approval from the manager and senior management staff. For this scenario, you can configure the banking Web service to automatically forward loan requests to the corresponding endpoints based on the loan amount requested by the customer. You can perform this configuration using the content-based routing feature that SOA Policy Enforcer provides.

Content Based Routing feature is supported only for XML service types.

## Enabling Content-based Routing

To enable content-based routing for the example scenario, follow these steps:

1. Start SOA Policy Enforcer Intermediary and log in to the Broker Configurator.
2. Click **Add New Intermediary Web Service**. The Step1: Import WSDL screen of the Add New Broker Service page opens.
3. Import the desired WSDL in the **Browse local WSDL file:** box.
4. Click **Next**. The Step 2: Configure Endpoints screen of the Add New Broker Service page opens.

5 Type **high_loan** and **medium_loan** in the Classifier boxes available for each endpoint. Special characters such as %, &, and +, and so on are nut supported for classifier names.

6 Click **Next**. The Step 3: Configure Broker Service screen of the Add New Broker Service page opens.

7 Change the name of the service and HTTP Path if a similar service is already deployed.

8 Select **Classifier Handler** from the **Features** section. The Classifier Handler section opens.

9 Type `high_loan` and click **Add** in the **Enter New Classifier** box.

10 Type `medium_loan` and click **Add** in the **Enter New Classifier** box

11 Select **high_loan** from the **Classifier** drop-down list and provide the following details:

   a Specify an XPath expression for the endpoint of the classifier in the **Expression** box. For example, number(`//ns1:LoanAmount/text()) >= 25000` where ns1 is the prefix of the namespace and it should have a valid URI associated with it

   b Select **Message** from the **Context** option

   c Type the Prefix and the URIs for the Namespaces in the corresponding boxes

   d Click **Save**.

12 Repeat steps a through d for the medium_loan classifier and click **Save**. Specify the XPath expression as follows for the medium_loan classifier number(//ns1:LoanAmount/text()) < 25000

13 Click **Save**.

14 Click **Finish**.

10

# Using the Intermediary's Security Features

This chapter provides instructions for securing the Web services application channel when using a WSM Intermediary deployment scenario. An overview section has been included that introduces many of the fundamentals of the security implementation. Users should be familiar with general security principals and Web services-based security before completing the instructions in this chapter.

The use of the security implementation is dependent on the use of the WSM Intermediary. However, you can use such deployment scenarios in conjunction with the WSM Intermediary and thus leverage the security features that are provided with the Intermediary and discussed in this chapter.

## Overview

While emerging trends in Web services architecture indicate that the future of Web services is loosely coupled, multi-hop, document exchange style message oriented interactions; most current implementations are point-to-point request-response HTTP based. Most enterprise security groups have existing security infrastructure and products established in house. The Intermediary security architecture takes this into consideration and provides a comprehensive set of options for securing Web services either at the (HTTP) transport layer or (SOAP) messaging layer.

## Feature Matrix

The following table lists the support technology that is included with the Intermediary security solution.

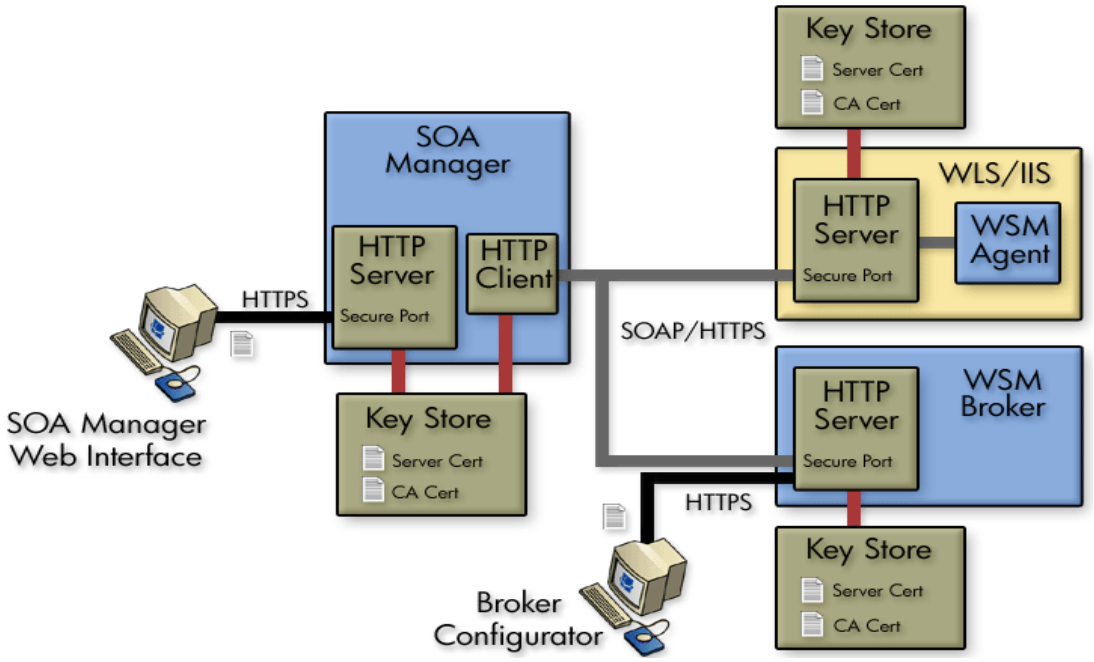
| Security Concern | Transport Level | Message Level |
|---|---|---|
| Authentication | HTTP/S: basic auth<br>HTTPS: X.509 certificates<br>HTTP/S: SSO tokens | WS-Security: User password<br>WS-Security: X.509 certificates<br>WS-Security: SSO tokens |
| Confidentiality | SSL | WS-Security: XML-Encryption |
| Integrity | SSL | WS-Security: D-Sig |
| Non-Repudiation | SOA PE Audit Service (using D-Sig) | SOA PE Audit Service (using D-Sig) |

- All User Identity Management – authentication, authorization, and administration is deferred to enterprise security products.
- WS-Security implementation in the Intermediary (D-Sig, Encryption) is done using Verisign TSIK toolkit.
- Java Key Store and PKCS12 Key Stores can be used for PKI support – except that covered by the security products.

## Supported Security Scenarios

This section describes end-to-end security scenarios supported by the Intermediary security implementation. There are three basic security scenarios discussed:

- Scenario 1: Intermediary is the Entry Point for External Consumers.
- Scenario 2: Web Application is the Entry Point for External Consumers.
- Scenario 3: Intermediary is the Exit Point for External Providers.




## Scenario 1: Intermediary is the Entry Point for External Consumers

In this scenario, incoming HTTP/S traffic through the firewall is front-ended by the Intermediary. The Intermediary supports HTTP/S basic authentication and X.509 client certificate authentication over SSL. Alternately, the intermediary can also be configured to decrypt incoming message payload and use X.509 certificates embedded in the digital signature of the payload to authenticate the message. The actual authentication/authorization is delegated to security products.

Authentication/Authorization failures are tracked and sent to the SOA PE so that alerts can be raised if the failures exceed SLO threshold values.

The security provider typically returns a security token (referred to as SSO token) as a result of successful authentication. This token can be propagated further to the back end Web service implementation either as an HTTP header or embedded in a WS-Security header in the payload. Obviously, for this to be meaningful, the back end Web service container platform must be integrated with the SSO security provider.

In case the back-end Web service container platform is not participating in the SSO, there are three options:

- Once authentication/authorization is done at the intermediary, no subsequent security authentication/authorization is done at the back end Web service implementation. In this case, firewalls may be configured to ensure that all traffic entering the Web service implementation is coming authenticated and authorized through the intermediary. The shortcoming of this approach is that business logic requiring security principal information cannot be written unless such information is also present in the message payload.

- A variation of the above option is that all actual authentication/authorization is done at the intermediary, but the intermediary presents some normalized identity to the back end Web service implementation. For example, some things like user, intermediary, password, and secret such that the back end application can be secured without having to configure firewalls. This too has the shortcoming that original security principal information is lost in the transition between intermediary and Web service implementation. However, it does make the back end implementation secure. The Intermediary (dispatcher) can be configured with credentials for basic authentication or x.509 client certificates that it can present while authenticating against back end Web service implementations. This can be done at the HTTP layer or embedded as WS-Security headers in the payload.
- If it is technically not feasible to integrate the SSO solution to the back end Web service container environment, the SSO problem can potentially be solved at the Intermediary. The Intermediary would have to know how to present credentials for represented principals in the back end Web service container realm. Some mapping must be made between incoming security principals and those known to the Web service container realm. Intermediary security does not natively support identity mapping features.

### Scenario 2: Web Application is the Entry Point for External Consumers

Incoming traffic such as regular Web application requests (i.e. non-SOAP) is authenticated at the Web Server/Web Application Server layer. If this layer is already integrated with the SSO provider, it can make requests against the Intermediary by propagating the SSO security token over SSL. The tokens can be presented either as HTTP headers or embedded in the WS-Security header. The Intermediary supports both styles for re-authentication against the SSO security provider.

Alternately, the internal Web service consumer may present some other authentication credentials via HTTP/S basic authentication, X.509 certificates over SSL or WS-Security D-Sig. The Intermediary can be configured to use any of these for authentication against the security provider. In this case, the Intermediary behavior is no different than that specified in Scenario 1, where it accepted calls from external consumers.

When the Intermediary forwards the request on to its final destination, it can support all the options described in Scenario 1.

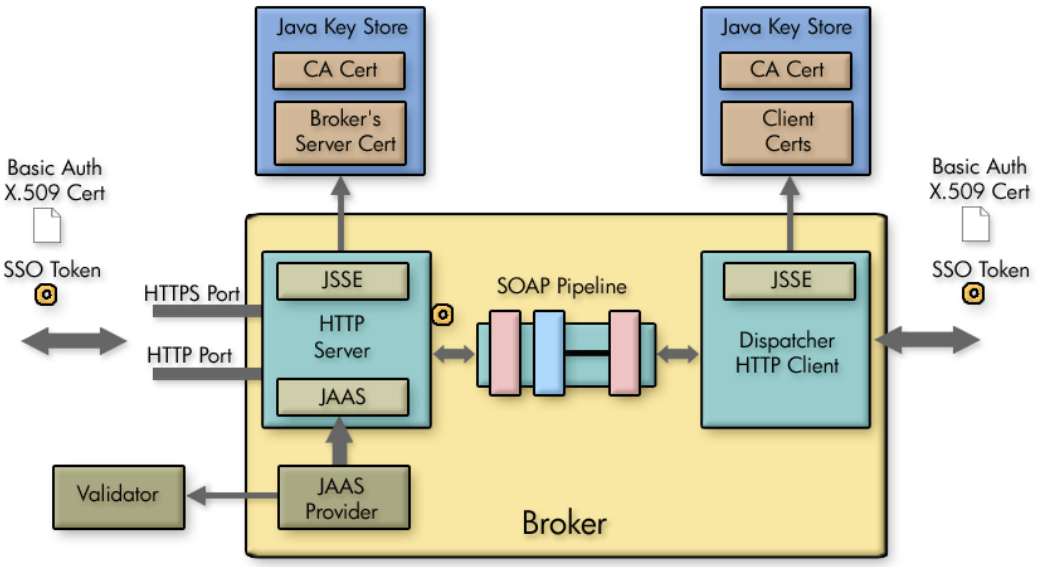
### Scenario 3: Intermediary is the Exit Point for External Providers

This scenario is covered between Scenario 1 and Scenario 2 and does not require any different explanation. In addition, Intermediary security does not support SAML. However, future releases of SOA PE will provide SAML support.

## Transport Level Security

HTTP/S serving is done by the Intermediary. HTTP/S client side (known as the Dispatcher) is implemented using a performance enhanced version of Jakarta commons HTTP Client that further uses JSSE for its SSL implementation.

Each intermediary service can be configured with transport security options for inbound traffic.




## Message Level Security

Message level security is offered using SOAP handlers. Figure 10-1 shows a common view of message level security.

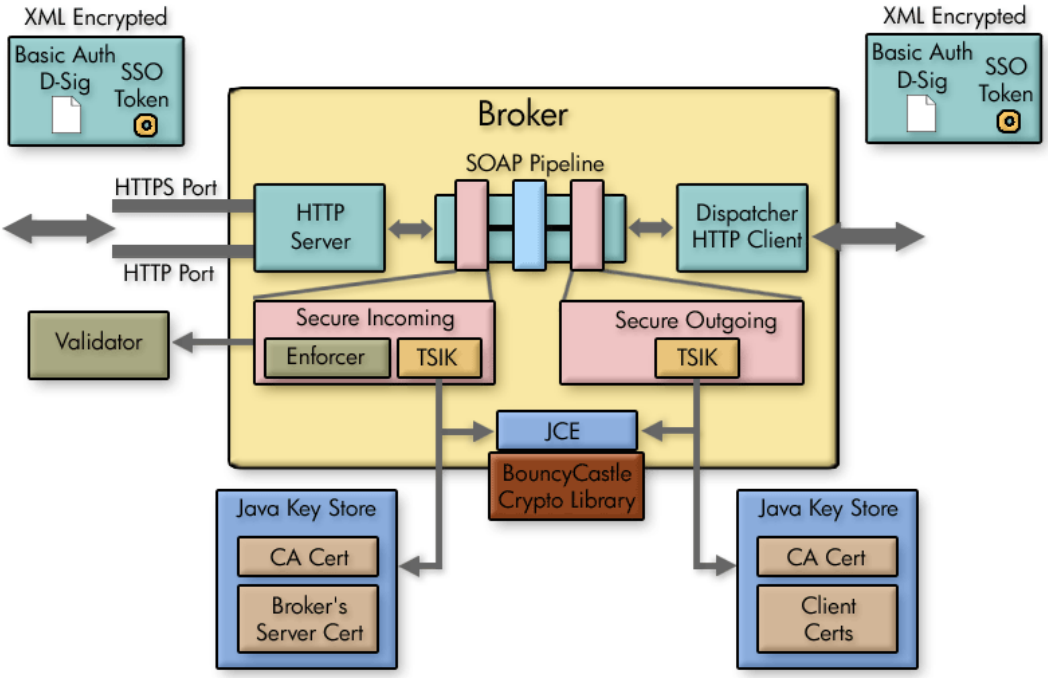



**Figure 10-1: Message Level Security**

### Inbound Message Processing

Inbound request payload can be decrypted using the Intermediary's server certificate. This assumes that the public key for this certificate was exchanged a priori (exactly how is out of the scope of this documentation) with the caller of the message and was used to encrypt the message. Once decrypted, the digital signature of the message is validated to ensure that the message integrity has not been tampered with. The digital signature contains the clients X.509 certificate (or chain leading to CA certificate). This certificate can be used to authenticate the message sender. The message processing handler also saves this certificate in case it needs to be used to encrypt the response before returning the response to the caller.

Meta-data required for XML Encryption and D-Sig behavior is extracted from WS-Security headers. Actual underlying implementation is provided by Verisign's TSIK toolkit. This toolkit uses JCE to provide crypto algorithms. SOA PE includes BouncyCastle JCE provider by default. We do not provide any PKI maintenance and customers are expected to use the Java Key Store.

Three types of WS-Security header credentials can be used for authentication:

- plain user:password, X.509 certificates
- incoming SSO token
- authentication/authorization is delegated to security provider APIs

### Outbound Message Processing

Outbound payload can be digitally signed using the Intermediary's server certificate configured in the Java Key Store. This digital signature embeds the Intermediary's X.509 certificate into a WS-Security header. It can be used by the receiver to authenticate the intermediary. Alternately, we can also embed a WS-Security user:password or WS-Security SSO token that either entered the Intermediary or that was created by authenticating against the security provider.

Once signed, it can be encrypted using the receiver's public key. This must have been entered into the Java Key Store a priori. The key alias is then specified in the configuration.

The returned response can be decrypted using the Intermediary's server certificate and payload integrity can be validated by checking against the embedded D-Sig.

# Setting Up the Security Components

As discussed in the "Overview" section, the Intermediary utilizes several external security components in order to secure communication on the application channel. The components must be configured as discussed in this section prior to implementing a security scenario. In addition, The Intermediary must be configured to use the various security components.

If you do not require the security features provided by a particular security component, you may skip the setup instruction for that component. However, if you are unsure of which security components you require or if you are testing different security capabilities, it is suggested that you setup all the security components.

This section does not cover the security configuration at the WS Container or in the consumers (applications) that are using the Web services. Refer to your vendor's documentation for instructions on setting up security.

## Configure a Key Store

The steps below detail how to use the Broker Configurator to configure a Key Store for use by the Intermediary.

A Key Store is required in the following steps. The Intermediary security solution supports both Java Key Stores and PKCS12 Key Stores. The steps below outline the configuration for use with a Java Key Store. For information on creating a Java Key Store, see Appendix A "Creating a Java Key Store."

To configure a Java Key Store:

1. Start the Broker Configurator.
2. From the Configurator's main tool bar, click **SSL Settings**. The SSL Settings screen displays.
3. Set the following properties:
   - **Keystore Location**: The location of your Java Key Store (i.e., `C:\\crypto\\scream.jks`).
   - **Keystore Password**: The password for your Java Key Store.
   - **Keystore Type**: Because we are using a Java Key Store this property is set to "jks".
   - **Private Key Alias**: The alias of the Java Key Store private key.
   - **Private Key Password**: The private key password in the Key Store.
4. From the bottom of the screen, click **Save**.

## Configure a CA Trust Store

A CA Trust Store is used to store certificates from Certificate Authorities (CA) that are to be considered trusted. In these instructions, the Trust Store is a Java Key Store populated with certificates from trusted CA's. The Java Developers Kit includes Java Secure Socket Extension (JSSE) which provides a populated Trust Store and is located in <*jdk_install*>`/jre/lib/security/cacerts`.

A Key Store is required in the following steps. The Intermediary security solution supports both Java Key Stores and PKCS12 Key Stores. The steps below outline the configuration for use with a Java Key Store. For information on creating a Java Key Store, see Appendix A "Creating a Java Key Store."

To configure the intermediary to use a CA Trust Store:

1 From the Configurator's main tool bar, click **SSL Settings**. The SSL Settings screen displays.

2 Set the following properties:

— **Truststore Location**: Trust Store location (i.e., <*jdk_install*>`/jre/lib/security/cacerts`).

— **Truststore Password**: Trust Store password. By default, the Trust Store password is `changeit`.

— **Truststore Type**: Because we are using a Java Key Store, this property is set to `jks`.

If you have changed any defaults associated with this Trust Store, the above entries will not work. Ensure settings are configured to match that of your environment.

3 From the bottom of the screen, click **Save**.

## Configure the Intermediary's SSL Port

The Intermediary's SSL port is used to accept HTTPS requests and is used to implement transport-level security. You must define which port you want to use to accept HTTPS requests.

To configure the Intermediary's SSL Port:

1 From the Configurator's main tool bar, click **HTTP Settings**. The HTTP Settings screen displays.

2 In the HTTPS Server Port field, enter the port you want the Intermediary to use for SSL connections.

3 From the bottom of the screen, click **Save**.

## Setting Up Authentication and Authorization

This section provides details on how to provide authentication and authorization. The intermediary supports basic authentication and authorization using basic authentication and x.509 client certificates. For either scenario, you can implement authentication and authorization for all intermediary services, specific intermediary services, or for specific operations within an intermediary service.

By applying authentication and authorization services to your Web services, you can confidently ensure that only selected consumers gain access to identified resources. The Intermediary security solution provides authentication and authorization services on a best of breed approach by integrating to well known and proven enterprise security products.

# Implementing a Security Scenario

This section provides instructions for implementing security scenarios. There are scenarios for both transport-level security and message-level security. The security scenarios include options for securing inbound communication from a consumer to the Intermediary and outbound communication from the Intermediary to a consumer.

Before implementing a security scenario, you must configure the security components that are used by the Intermediary (see "Setting Up the Security Components" above).

The security scenarios discussed in this section are not mutually exclusive. You may choose to implement a single scenario, or you may choose to combine several scenarios together. The scenarios you choose to implement depend on the security requirements of your environment and the security requirements of your applications. Refer to the "Overview" section above for detailed information about the Intermediary's security capabilities.

The scenarios discussed in this section include:

- Inbound Transport Security
- Outbound Transport Security
- Inbound Message Security
- Outbound Message Security

## Inbound Transport Security

In this scenario, the Intermediary accepts requests from consumers using SSL and authenticates/authorizes the user using a security provider. This is a typical scenario where an enterprise needs to secure inbound communications but does not need to secure the channel when calling the actual endpoints. An example of this could be providing a service externally; once the messages are received and through the firewall, the secure channel is not needed as the messages are traveling across a private network.

### Enabling SSL

The Broker Configurator is used to configure an intermediary service and enable inbound SSL connections. You can configure SSL when you create an intermediary service or you can edit an existing intermediary service.

To enable inbound SSL:

1. From the Broker Configurator, create a new or edit an existing intermediary service.
2. From the Service Configuration screen, check the **Use SSL** option located in the Inbound Transport section.
3. At the bottom of the screen, click **Save Changes**. The Brokered Services screen opens. The service you just configured has a Service Interface URL that indicates HTTPS. This is the URL your clients should use to access the service.

If the Key Store was configured with a signed server certificate from a Certificate Authority (CA) which is not commonly known, you may see an error message indicating that a trust relationship could not be established. If this is the case, you will need to obtain the CA's certificate and install that in the Trust Store for all clients who will access this service.

### Enabling Authentication

The Broker Configurator is used to configure an intermediary service and enable authentication for inbound transport security. Users are authorized using a security provider. You can enable authentication when you create an intermediary service or you can edit an existing intermediary service.

To enable authentication:

1. From the Broker Configurator, create a new or edit an existing intermediary service.
2. From the Service Configuration screen's Inbound Transport section, check the type of authentication you want to enable:
   - **Basic Authentication**: All requests to the Intermediary need to be authenticated using a user name and password.
   - **X.509 Client Certs**: All requests to the Intermediary need to be authenticated using an X.509 certificate.
3. At the bottom of the screen, click **Save Changes**. Once this service is deployed, the Intermediary will communicate with the security provider for all inbound requests to ensure that the consumer has supplied the proper credentials to gain access to the service. If the user is not authenticated and/or authorized, the Intermediary will return a 404 Not Authorized error.

## Outbound Transport Security

In this scenario, the Intermediary accepts requests from consumers and then forward that request to the provider using an SSL channel. This scenario can be combined with the inbound transport scenario to provide end-to-end transport-level security.

### Enabling Outbound SSL

The Broker Console is used to configure an intermediary service and enable outbound SSL connections. You must enable SSL when you create an intermediary service. You cannot edit an existing intermediary service to use outbound SSL.

To enable outbound SSL:

1. From the Configurator's main toolbar, click **Create Brokered Web Service**. Step 1 of the Create Brokered Service wizard displays (Step 1: Import WSDL).
2. In the text box, specify the WSDL with HTTPS if your server will dynamically create port bindings based off of the WSDL URL. For example:

```
https://company.com/finance?wsdl
```

Or,

Click **browse** to locate a Web service's WSDL.

3 Click **next** to move to Step 2 of the wizard (Step 2: Configure Endpoints). A binding is created for the Web service and displays in the Select Endpoints screen. If a Web service definition contains multiple endpoints, a binding for each endpoint is listed.

4 From the Authentication field, click to select the **Send Credentials** check box.

5 Complete creating the intermediary service by following the prompts. The intermediary service is configured to use outbound SSL when you have completed creating the intermediary service and it is deployed.

If the endpoint has a server certificate signed by a CA whose CA Certificate is not present within the trust store configured for the Intermediary, the SSL handshake will fail. Make sure the endpoint's CA's Certificate is located in the Intermediary's trust store.

## Inbound Message Security

In this scenario, a consumer must authenticate with the Intermediary before messages are accepted. In addition, the consumer may choose to encrypt messages before sending them to the Intermediary; in which case, the intermediary will decrypt the messages before they are dispatched to the final endpoint. Refer to Figure 10-1 for a conceptual architecture of message-level security.

The Broker Configurator is used to configure an inbound message security handler for an intermediary service. Users are authorized using a security provider and decryption is implemented through a Key Store (See "Configure a Key Store" above). You can enable message security when you create an intermediary service or you can edit an existing intermediary service.

To enable inbound message security:

1 From the Broker Configurator, create a new or edit an existing intermediary service.

2 From the Service Configuration screen's Feature section, click the **Inbound Message Security** option. The security options display.

3 Click the security option you want to enable:

- **Username-Password Authentication**: All messages to the Intermediary need to be authenticated using a user name and password.
- **Digital Signature Authentication**: All messages to the Intermediary need to be authenticated using a digital signature.
- **Digital Signature Authentication with Decryption**: All messages to the Intermediary need to be authenticated using a digital signature. In addition, the Intermediary's private key is used to decrypt the message.

4 Click to select the **No Digital Signature or Encryption in Response** option if you do not require the response message to be encrypted or have a digital signature. If you do not select this option, the intermediary expects the response message to be encrypted and have a digital signature.

5 At the bottom of the screen, click **Save Changes**. Once this service is deployed, the Intermediary will communicate with the security provider for all inbound requests to ensure that the consumer has supplied the proper credentials to gain access to the service. If the user is not authenticated and/or authorized, the Intermediary will return a 404 Not Authorized error.

The Intermediary will fail to recognize a Digital signature if the XML payload is changed after it has been signed. This typically happens during debugging when the XML payload is reformatted in “pretty print” for ease of reading. If the payload is reformatted, it should not be sent to the Intermediary.

## Outbound Message Security

The Broker Configurator is used to configure an outbound message security handler for a brokered service. You can enable message security when you create an intermediary service or you can edit an existing intermediary service.

To enable outbound message security:

1 From the Broker Configurator, create a new or edit an existing intermediary service.

2 From the Service Configuration screen’s Feature section, click the **Outbound Message Security** option. The security options displays.

3 Choose the security option you want to enable:

4 Click to select the **No Digital Signature or Encryption in Response** option if the response message does not have digital signature and is not encrypted. If you do not select this option, the intermediary expects the response message to have a digital signature and/or be encrypted.

5 At the bottom of the screen, click **Save Changes**.

# Management Channel HTTP Basic Authorization

HTTP basic authorization can be enabled to secure the intermediary management channel. This functionality is the same as securing the application channel.

To configure intermediary management channel security:

1 Stop the Intermediary if it is currently started.

2 Use a text editor to open <*install_dir*>`\conf\broker\mipServer.xml`.

3 Remove the comment tag and text (<!-- -->) from the following three property entries:

```
<entry name="com.hp.mip.security.provider.management">
   default</entry>

<entry name="com.hp.mip.security.sba.user">user</entry>

<entry name="com.hp.mip.security.sba.password">password</entry>
```

   — Specify the name of the security provider for management channel in the `com.hp.mip.security.provider.management` element.

   — Specify the user name for the user who is authorized to access the Web URL of the management channel in the `com.hp.mip.security.sba.user` element.

   — Specify the password for the user who is authorized to access the Web URL of the management channel in the `com.hp.mip.security.sba.password` element.

4 Save and close `mipserver.xml`.

5 Start the Intermediary server.

9

# Implementing Load Balancing and Failover

This chapter provides instructions for setting up the load balancing and failover features that are included with the WSM Intermediary. In addition, an overview and conceptual architecture for load balancing and failover is provided.

The load balancing and fault tolerance features included with the Intermediary are primarily designed for requests made between an intermediary service and its Web service endpoints. However, load balancing and failover can also be implemented between a client and an Intermediary. The final section “Using Multiple Intermediaries” explains this scenario and provides implementation instructions.

## Overview

The WSM Intermediary contains a load balancing and failover feature that automatically routes a Web service request that is made to an intermediary service to multiple endpoints. Should requests to a primary endpoint fail, a backup endpoint is automatically used instead. The endpoints are defined in a service’s definition (WSDL) file and are configured when an intermediary service is created using the Broker Configurator console. When a Web service with multiple endpoints is managed, the management information (success, response time, and so on) for each endpoint is aggregated.

Load balancing and failover is an important part of distributed applications and offers some key benefits. In particular, these features:

- Provide redundancy – Multiple instances of a Web service that are spread across different hosts means a service is always available for requests.
- Minimize downtime – Multiple instances of a Web service that are spread across different hosts allows an application to continue making requests even if one host fails or is being serviced.
- Increase reliability – Users never experience an unavailable application.
- Improve performance – Request loads are spread across different hosts, which prevents bottlenecks from occurring.

- Reduce single points of failure – Requests to an endpoint which is failing are automatically rerouted to working endpoints.

# Conceptual Architecture

All requests that are sent to an intermediary service are sent to a final endpoint using the Intermediary's dispatcher. A list of available endpoints is registered with the Intermediary and is used to find endpoints that can satisfy a request.

A WSDL file is used to define a service and the endpoints (SOAP addresses) available for the service. When an intermediary service is created from the WSDL file, these endpoints are discovered and registered by the Intermediary and configured as either an active endpoint or a backup endpoint.

## Load Balancing Scenario

Active endpoints are the primary addresses that are used to service a request. Multiple active endpoints can be used to share the load of servicing requests. Only after all active endpoints fail, will a backup endpoint be used. When a request is dispatched to an active endpoint, it is done using a round robin scheme. That is, an endpoint is used once and then moved to the bottom of the list of available endpoints. The next request goes to the next endpoint on the list and then that endpoint is moved to the bottom of the list and so on.

## Failover Scenario

Backup endpoints are only used when all active endpoints fail. A failure occurs when an HTTP Status code is returned that is greater than or equal to 300, less than 500, or equal to 503. While the backup endpoint is being used, the Intermediary continues to try an active endpoint at 30 second intervals. When an active endpoint becomes available, requests are again routed to it and the backup endpoint is no longer used.

If you have multiple backup endpoints, requests are sent using a round robin scheme.

# Setting Up Load Balancing and Failover

Load Balancing and failover is set up for each intermediary service that you create. When you create an intermediary service, each endpoint that is discovered can be configured as either an active endpoint or a backup endpoint. This section describes how to modify a WSDL file to include multiple endpoints and how to configure each endpoint as an active or backup endpoint.

## Defining Multiple Endpoints in a WSDL File

The load balancing and failover feature is dependent on a WSDL file that defines multiple endpoints for a Web service. For example, if two instances of the same Web service are running on two different hosts, then a single WSDL file can be used to define the Web service and each endpoint that is available. Endpoints are defined in the `<service>` node of a WSDL file as demonstrated below for the finance service:

```
<service name="FinanceService">
   <port name="FinanceServiceSoap" binding="tns:FinanceServiceSoap">
      <soap:address
        location="http://host1:7001/FinanceService/FinanceService" />
   </port>
   <port name="FinanceServiceSoap" binding="tns:FinanceServiceSoap">
      <soap:address
        location="http://host2:7001/FinanceService/FinanceService" />
   </port>
</service>
```

The `FinanceService` above contains two SOAP address endpoints. One endpoint is located on `host1` and the other is located on `host2`. Each endpoint must be defined within a `<port>` node that also defines the `PortType` and `binding`.

Before creating an intermediary service using the Broker Configurator, make sure you have modified a WSDL to include multiple endpoints as demonstrated above.

## Configuring Load Balancing and Failover

An intermediary service is created by using the Broker Configurator. The create service wizard steps you through the process of creating an intermediary service, including importing a WSDL file and configuring whether an endpoint should be an active endpoint or a backup endpoint.

To configure load balancing and failover:

1. Log in to the Broker Configurator.
2. Click on the **Create Brokered Web Service** link. Step 1 of the Create Brokered Service wizard displays (Step 1: Import WSDL).
3. Enter a WSDL that defines multiple endpoints for a Web service.
4. Click **next** to move to Step 2 of the wizard (Step 2: Configure Endpoints).
5. By default, an endpoint is configured to be the primary endpoint as indicated by the **Primary** option in the Options field. Click to select the **Backup** option if the endpoint is to be only used as a backup if a primary endpoint should fail.

Endpoints can only be configured when an intermediary service is initially created.

# Using Multiple Intermediaries

Multiple Intermediaries are used to provide an additional level of assurance that no single point of failure exists between clients and an Intermediary. In this scenario, a third party load balancer, such as Cisco's IP Director, is used to balance requests between two or more Intermediaries that are running on different hosts.

Each Intermediary contains an intermediary service for the same Web service. Loads are balanced between each intermediary service and if one Intermediary fails, additional intermediaries are available to continue servicing requests. Management information (i.e., success, response time, and so on) for each intermediary service is aggregated. In addition, each intermediary service can be viewed separately in a single business service when using the SOA PE user interface.

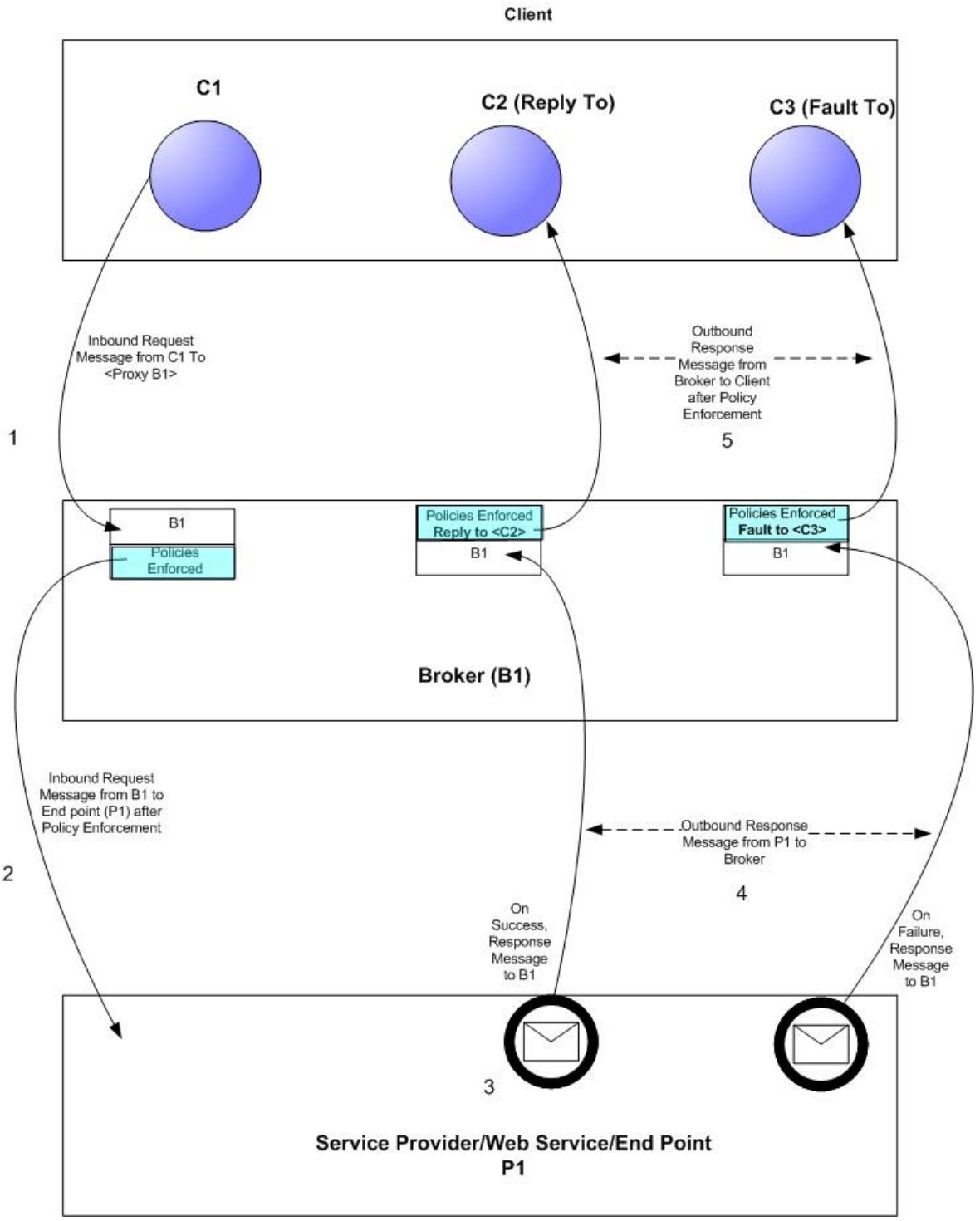
When implementing this scenario, use the instructions in the "Setting Up Load Balancing and Failover" section discussed previously for each installation of the WSM Intermediary.

It is beyond the scope of this documentation to detail installation and configuration of a third party load balancer. See the documentation that was included with your load balancer product for full installation and setup instructions.

11

# WS-Addressing Support in SOA PE

SOA PE supports WS-Addressing specifications. Refer to the following diagram to understand the implementation and benefits. For an overview of WS-Addressing, see the section “WS-Addressing- An Overview” in Chapter 1: HP SOA Policy Enforcer Overview.




In this implementation, the SOA PE Broker performs the role of a proxy between the service provider and the service consumer. Refer to the numbers in the diagram for understanding the flow of the message:

- Number 1: Indicates client C1 sending a request message to the broker with the To field set to broker B1.

- Number 2: Indicates B1 forwarding the message to the end point P1 after attaching the policies associated with the service as configured during service definition.
- Number 3: Indicates the end point processing the request message.
- Number 4: Indicates the end point forwarding the response message to the broker B1. The end point forwards the success or failure message to the Broker.
- Number 5: For a successful message, the Broker enforces the policies associated with the Web service for a response message and forwards the message to client <C2> set as <Reply To> recipient in the Broker for successful messages. For a failed message, the Broker enforces the policies associated with the Web service for a fault and forwards the message to client <C3> set as <Fault To> recipient in the Broker for failed messages.

The advantage Broker offers is that of policy enforcement on request and response messages. SOA PE Broker enforces the policies (you define and attach to a Web Service) to the request messages to the Web Service and the response messages from the Web Service. The Broker also performs the role of a gateway to the end point to allow or deny access to the end point and alert the client through a fault if the end point is not accessible.

For example, if you had defined a scheduled availability policy for a Web Service, the Broker as soon as it receives a request message for the concerned Web Service, enforces the policy on this request message. Based on the availability period specified for this Web Service in the scheduled availability policy, the Broker either sends a successful reply or a fault message to the service consumer according to the configuration in the scheduled availability policy.

## Prerequisites for WS-Addressing Support in SOA PE

You must make sure that both the client and the end point support WS-Addressing specifications as mentioned in the WS-Address specification listed in http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509

## Configuring SOA PE for WS-Addressing

You can use WS-Addressing if the service provider and service consumer supports WS-Addressing. You can configure WS-Addressing in SOA PE using one of the following methods:

- Using the WS-Addressing policy and service provisioning feature in the SOA PE Server UI. Refer to the WS-Addressing policy framework specifications at www.w3.org/Submission/WS-Policy/ to know more about the WS-Addressing specifications implemented by this policy
- Manually configure the WS-Addressing handlers provided by the SOA PE Broker

# Appendix A Creating a Java Key Store

Java Key Stores are used to create secure communication channels based on the SSL standard. Key Stores are created using the Java Keytool utility, which is a key and certificate management tool provided by Sun Microsystems and distributed with the Java Development Kit. This appendix is designed to act as a fast track to creating a Java Key Store intended to be used with the security scenarios introduced in Chapter 15 "Using the Intermediary's Security Features" for the specific purpose of securing the communication channel to and from the Intermediary. This tutorial is NOT intended to be a replacement to the documentation provided by Sun as it pertains to the Java Keytool. For detailed information on the Java Keytool, see Sun's documentation. To complete the tasks in this appendix, you must have:

- A general understanding of the Public Key Infrastructure
- A general knowledge of the Sun JDK
- Installed the JDK and insured it is located on the computer's PATH

## Step 1: Create a Private Key and the Initial Java Key Store File (JKS file)

Open a command prompt and execute the following command:

```
keytool -genkey -keystore scream2.jks -alias scream -keyalg RSA
```

This will start the creation of the private key and Key Store. Provide answers to the prompts as they appear. You will receive prompts similar to the following:

*Enter Key Store password:*

This is the password you will use to access the Key Store in the future.

*What is your first and last name?*

This is the identity of the owner of the Key Store. Enter the fully qualified DNS name of the server.

When asked for your first and last name, use the fully qualified domain name of the system you will use this Key Store on. Failure to do this will result in a failure of the SSL connection.

*What is the name of your organizational unit?*

Enter any departmental information you want associated with the Key Store.

*What is the name of your organization?*

Enter the name of the organization this Key Store will be associated with.

*What is the name of your City or Locality?*

Enter you city name.

*What is the name of your State or Province?*

Enter state or province.

*What is the two-letter country code for this unit?*

Enter country code. (for example, US, UK)

The next prompt is a review prompt displaying the information you just entered. If everything is correct, type `y` and press Enter. If you need to make corrections, press Enter and follow the prompts.

If you typed `y` above to continue, you will be asked for a key password. This is a password that will be associated with the private key only. You can use the same password you provided to for the Key Store if desired. It is a matter of personal preference.

You should know have a Key Store created.

# Step 2: Generate a CSR request

Execute the following command:

```
keytool -certreq -keystore scream2.jks -alias scream -file
scream2.csr
```

Enter the Key Store password when prompted. When completed, you should have a CSR to send to a certificate authority.

# Step 3: Obtain a Signed Certificate from a Certificate Authority

Using the CSR file created in step 2, contact a Certificate Authority to obtain a digital certificate for your server.

If you are in a test environment, you can obtain a test certificate. Test certificates are generally easier and quicker to obtain. If using a test certificate, ensure that the test root certificate is also obtained from the CA you are using.

Below are some URLs to Certificate Authorities:

**VeriSign**

http://www.verisign.com

**Thawte**

http://www.thawte.com

**GlobalSign**

http://www.globalsign.net/

http://www.belsign.be/

# Step 4: Import Signed Server Certificate to Key Store

Once you obtain a signed certificate from a Certificate Authority, you need to import that certificate to the Java Key Store. Copy the certificate file you received from the authority to same directory as your Java Key Store.

Edit the file so that all of the text above `-----BEGIN CERTIFICATE-----` is removed. This is approximately 45 lines of text which you will remove. When completed, you should be left with something with looks like the following:

```
-----BEGIN CERTIFICATE-----
MIIDtDCCAx2gAwIBAgIBETANBgkqhkiG9w0BAQQFADCBmzELMAkGA1UEBhMCVVMxEzARB
gNVBAgTCk5ldyBKZXJzZXkxFTATBgNVBAcTDE1vdW50IExhdXJlbDEgMB4GA1UEChMXSG
V3bGV0dC1QYWNrYXJkIENvbXBhbnkxDTALBgNVBAsTBERSUEUxEjAQBgNVBAMTCUplZmY
gVHVjazEbMBkGCSqGSIb3DQEJARYManP2332tAaHAuY29tMB4XDTA0MDMzMDE3MTUzM1o
XDTA1MDMzMDE3MTUzM1oweTELMAkGA1UEBhMCVVMxEzARBgNVBAgTCk5ldyBKZXJzZXkx
IDAeBgNVBAoTF0hld2xldHQtUGFja2FyZCBDb21wYW55MQ0wCwYDVQQLEwREUlBFMSQwI
gYDVQQDExtzY3JlYW0uYW1lcmljYXMuaHBxY29ycC5uZXQwgZ8wDQYJKoZIhvcNBBEEBQ
ADgY0AMIGJAoGBANeRXdQBWOXbVjbmsL0kmf9QlOnq9mvJh7ehZsbNZQN2wspcLYrfb1v
4769BxbegdW/uY9LnNP0a3vVLg2hLWWx8L703SLd7S/ztZF8QU/RAE1w6pxDT+KsHwtfn
OAlBj2FEcChrIEQI2PVUIcw8PpQ/HAMNRj7DVEvR2Po9B5wHAgMBAAGjggEnMIIBIzAJB
gNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2
F0ZTAdBgNVHQ4EFgQUNpLKwfcQM0GP219/V5I3H81smGwwgcgGA1UdIwSBwDCBvYAU9Uj
i64FvhWKvwHlB0LdGi+Q+FKhgaGkgZ4wgZsxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpOZ
XcgSmVyc2V5MRUwEwYDVQQHEwxNb3VudCBMYXVyZWwxIDAeBgNVBAoTF0hld2xldHQtUG
Fja2FyZCBDb21wYW55MQ0wCwYDVQQLEwREUlBFMRIwEAYDVQQDEwlGDTWmIFR1Y2sxGzA
ZBgkqhkiG9w0BCQEWDGp0dWNrQGhwLmNvbYIBADANBgkqhkiG9w0BAQQFAAOBgQBXKfZ2
oQCqf5mWyiqJ3bZpcFamNmHtoXlBkZmgIx5D9ITD0PJ+eQaeRzFS1Pphv2rrYvddpsAs7
sjXjTSNXNjNNCnAZTsvFB7j8wKFQObPT6XmgevJ2kVwEIFoYxpNKGoZYPyCBkopEHR5KX
z+ClPA/z7+iqnB9iAmV/Pgib9Obg==
-----END CERTIFICATE-----
```

To insure you do not overwrite the original file, save this file with a slightly different file name and close the file.

Execute the following command:

```
keytool -import -keystore scream2.jks -alias scream -file
scream2_import.crt -trustcacerts
```

Ensure that the alias name is the same as the private key name.

This should result in a response from the keytool similar to the following:

```
"Certificate was added to keystore"
```

You now have a successfully configured Key Store.

If you have a test certificate or a certificate issued by an authority which is a commonly known public authority, you will need to ensure that the Certificate is installed on all client trust stores in order for the connection to be created. In addition, the CA root certificate needs to be installed in the server Trust Store as well.

B

# Appendix B Troubleshooting SOA PE

This chapter provides common troubleshooting tasks when using the SOA PE.

## Troubleshooting Tips/FAQ

- After installing SOA PE, configuring Systinet Repository, and restarting SOA PE, I cannot find the policies in SOA PE

  Make sure that the policies are published to the registry configured with the repository.

- How do I enable logging of deadlock detection messages for the Smart Business Agent (SBA)?
  Perform the following steps:

1 Stop SOA PE and SOA PE Intermediary if they are running.

2 If you are running SOA PE on UNIX, perform the following steps or go to step 5 in this procedure:

3 Open <install_dir>/bin/unix/mipserver, where <install_dir> represents the directory in which you installed SOA PE.

4 Add the line `JAVA_OPTS="-Dwsm.deadlock-detection.debug=true $JAVA_OPTS"` above the line `exec "${MIP_JAVA_HOME}/bin/java" \` in the file mipserver,

5 Perform the following steps if you are running SOA PE on Microsoft Windows:

6 Open <install_dir>/bin/win32/mipserver.bat, where <install_dir> represents the directory in which you installed SOA PE.

7 Add the line `set JAVA_OPTS=-Dwsm.deadlock-detection.debug=true %JAVA_OPTS%` above the line `"%JAVA% %JAVA_OPTS% -classpath %MIP_CLASSPATH% com.hp.wsm.sn.bootstrap.Bootstrap -bootscript %ARGS%"` in the file mipserver.bat. Setting the `wsm.deadlock-detection.debug` system property to `true` as shown in this procedure enables deadlock detection debugging for SBA.

8 Start SOA PE and SOA PE Intermediary.

- How do I change the temporary directory location used by SOA PE before startup?
  - SOA PE, during start up, creates a temporary director under which run time files that are essential for SOA PE and SOA PE Broker to run are stored. You can change the temporary directory location used by SOA PE by adding the line `Djava.io.tmpdir=<temp_directory_full_path>`, in the file mipserver.bat (for Microsoft Windows platforms) or mipserver (for UNIX platforms) under JAVA_OPTS. <temp_directory_full_path> represents the full path to the new location where you want Jetty to create the temporary directory. The mipserver files are present at the following locations:
    - mipserver.bat: <install_dir>\win32
    - mipserver: <install_dir>/unix
      <install_dir> represents the directory in which you have installed SOA PE.
- If I send an invalid JMS message, SOA PE Intermediary continuously processes the same message repeatedly. What do I do?
  - When an invalid message is sent to SOA PE Intermediary, SOA PE does not commit the transaction. You must configure JMS provider's redelivery mechanism to avoid this message getting redelivered continuously to the intermediary.
- How can I move a provisioned web service from one Business Service to another?
  - Remove the web service or web service intermediary configuration without undeploying the service. Provision the service again specifying same set of parameters that you used during provisioning and the new business service.
- In JMS->HTTP protocol switching scenario, the request sent to endpoint does not have SOAP Action header set. What do I do?
  - Any properties that must be sent as part of transport header must be sent as JMS properties in the JMS message sent at inbound. Also configure HTTPPassThrough handler to pass these properties from inbound to outbound.
- Why do I see multiple authentication failure alerts even if I sent a single request to SOA PE Intermediary?
  - When using Basic Authentication, on receiving authentication failure some clients send the requests again. As a result you can see an alert corresponding to each retry request sent by the client.
- Can I specify the same JMS queue at inbound for different service during provisioning?
  - No. Each service must be configured with its own Queue at inbound during provisioning.
- Even after changing the policy association in Web service configuration page, I am still seeing old policies. Why?
  - SOA PE updates the policy association only after successful deployment to intermediaries. Using Lifecycle Management link, check the status of the deployment.
- I added a new endpoint to the routing table, but still web service configuration page doesn't show the new endpoint in routing table. Why?

  - SOA PE updates the routing table only after successful deployment. Using Lifecycle Management link, check the status of the deployment.

- Some messages are not getting audited even after I attach audit policy to the service.
  - When a request message sent to the broker fails before it reaches the message pipeline, it is not audited. For example when the request fails because of inbound transport security failure, the message is not audited.

# Installation and Configuration Problems

## Errors occurred during installation

Receive an error message at the end of the installation:

```
The installation of SOA PE is finished, but some errors occurred
during the install. Please see the installation log for details.
```

**Solution:**

1. Check the `<SOAPE dir>/HP_SOA_Policy_Enforcer_3.00_InstallLog.xml` log file for errors.
2. If you see install file errors, `<action name="Install File" status="error" />`, it means you only copied the `HPSOAPolicyEnforceInstaller3_00.bin` file from the SOA PE installation CD to the system. You need to copy all of the files that are on the CD in the `../Installation` directory to the system where you're trying to install SOA PE.

## AutoPass fails to install

Receive an error dialog during installation:

```
AutoPass, the HP Software licensing tool, failed to install properly.
This installation will abort.  Please refer to the <temp
dir>\AutoPass_install.log log file for more details.
```

**Solution:**

1. Check to see if the `<temp dir>\AutoPass_install.log` log file exists.
2. If the log file exists, check for errors.
3. If the log file doesn't exist, check to see if there are non-English characters in the `<temp dir>` name. AutoPass has a bug where it doesn't allow non-English characters in path names. If there are non-English characters in the `<temp dir>` name:
   a. Uninstall the SOA PE.
   b. Save the value of the `TMP` environment variable.
   c. Change the `TMP` environment variable to a directory with all English characters.
   d. Install the SOA PE.
   e. Change the value of the `TMP` environment variable back to its original value.

# Runtime Problems

## Could not start monarch-sba

When trying to start SOA PE, receive a message:

```
[WARN] unable to locate tools.jar, possible non-sun jvm?
```

and later

```
[SEVERE]; Could not start monarch-sba: java.lang.Exception: Monarch
did not initialize
```

**Solution:**

Verify that the environment variable `MIP_JAVA_HOME` is assigned to the Java 1.4 SDK and not the JRE.

When trying to start SOA PE, receive a message:

```
[SEVERE]; Could not start monarch-sba: java.lang.Exception: Monarch
did not initialize.
```

**Solution:**

1. Turn on logging for the Smart Business Agent (SBA) to get more details about the problem.
   a. Change directories to `<install_dir>/conf/networkservices`.
   b. Edit the `logging.properties` file.
      – Change `log4j.category.com.hp.wsm.impact=OFF` to `log4j.category.com.hp.wsm.impact=INFO, ROLL_FILE`
      – Add the following to the end of the file

```
# ROLL_FILE - rolling file appender that writes the logs to
the file system
#
log4j.appender.ROLL_FILE=org.apache.log4j.
   RollingFileAppender
log4j.appender.ROLL_FILE.File=C:\\temp\\soam-ns-sba.log
log4j.appender.ROLL_FILE.MaxFileSize=512KB
log4j.appender.ROLL_FILE.MaxBackupIndex=1
log4j.appender.ROLL_FILE.layout=org.apache.log4j.
   PatternLayout
log4j.appender.ROLL_FILE.layout.ConversionPattern=-->
   %d{yyyyMMdd|HH:mm:ss}|%p|%t|%c{5}|%m%n
```

2. Restart SOA PE.
3. Look for errors in the `C:\temp\soam-ns-sba.log` file.

## Failed to initialize listener

When trying to start SOA PE, receive a message:

```
…;SEVERE;An error occurred while initializing the MIP Server: … :
failed to initialize listener
```

**Solution:**

1. Check to see if the SOA PE is already running. If you are running on Windows and selected to install SOA PE as a service during the installation process, SOA PE is automatically started when you reboot the system.
2. If SOA PE is not running, then another application must be using the port. By default, the SOA PE uses port 5002. Change the SOA PE to use a different port.
   a. Change directories to `<soape_home>/conf/networkservices`.
   b. Edit the `mipServer.xml` file. Change the `<entry name="com.hp.http.server.port">5002</entry>` property.
   c. Start the SOA PE

## Timezone error when using Oracle 9i

Receive the message when starting the SOA PE:

```
java.sql.SQLException: ORA-01882: timezone region not found
```

**Solution:**

1. Verify that the Oracle JDBC driver version is 9.2.0.5.0. The SOA PE prints out the JDBC driver information at startup to `stdout`.
2. Make sure your timezone is in the timezone file Oracle is using. The following is from Chapter 2 "Creating an Oracle Database" in the *Oracle 9i Database Administrator's Guide Release 2 (9.2)*:

   "Oracle uses a time zone file, located in the Oracle home directory, as the source of valid time zones. If you determine that you need to use a time zone that is not in the default time zone file (`timezone.dat`), but that is present in the larger time zone file (`timezlrg.dat`), then you must set the `ORA_TZFILE` environment variable to point to the larger file."

## Performance graph error on HP-UX and Linux

The performance graph located in the Performance section of a view screen does not display when the SOA PE Server is installed on HP-UX or Linux.

The performance graph is implemented using Java Swing libraries. The libraries require that the server have an X server display defined. If the display is not defined, the performance graph fails.

To define an X server display:

1 On the SOA PE server, create a `DISPLAY` environment variable that contains the X server's display name of the form *hostname:displaynumber.screennumber*. For example:

```
export DISPLAY=Myserver.com:0.0
```

This variable defines that the display is located on `Myserver.com` and that the default display and screen number will be used.

2 In addition to the `DISPLAY` variable, you must give clients the ability to access the X server's display. This can be done using X host. For example:

```
xhost +
```

3 Restart the SOA PE Server process.

## Intermediary audit traces not showing up in User Interface

When querying for audit messages in the user interface, there are no audit messages returned in the query.

**Solution:**

Verify the clock synchronization. If the intermediary is running on a different system than the SOA PE, verify that the clocks are synchronized.

Verify auditing is enabled for the intermediary service:

1 In the Broker Configurator, click on the Intermediary Service for which you are not seeing audit messages.

2 In the Features section on the Service Details page for the service, confirm that **Auditing** is checked. If this is not checked, then edit the Service settings and check the **Auditing** feature to enable auditing. Check to see if Audit messages are now displaying.

Verify the audit message is being received by the SOA PE:

1 In the `<install_dir>\conf\networkservices\xpllogging.properties` file, set the logging level for SOA PE to `fine`:

```
com.hp.ov.mip.level = FINE
```

2 Restart the SOA PE.

3 Confirm that SOA PE subscribes for audit messages. Look for the following message in the trace file:

```
Jul 5, 2005 9:56:35
AM;157;13;com.hp.wsm.sn.monitoring.collectionservice.CollectionSer
vice;wseeAdded;com.hp.ov.mip.Auditing;INFO;><Subscribing to wsee
http://<Service-Host>:9032/wsmf/services/Runtime$service=Wsee?wsdl
listening for audit events
```

4 Send a request to the service so an audit message is generated. Confirm that trace message is being received by SOA PE. Look for the following message in the trace file:

```
Jul 5, 2005 9:58:04
AM;581;19;com.hp.wsm.sn.monitoring.collectionservice.CollectionSer
vice;handleNotify;com.hp.ov.mip.Auditing;FINE;>!<Received audit
messages

Source: http://<Service-
Host>:9032/wsmf/services/Runtime$service=Wsee?wsdl

Event:
http://schemas.hp.com/mip/2004/WsExecutionEnvironment/Event/Messag
eTraceNotificaTion
```

If the message is there, then you know the SOA PE has received the audit message. Go through the rest of the trace messages to pinpoint the problem.

If the message is not there, then you know the SOA PE has not received the audit message. Read the next section.

Verify audit message is being sent by the Intermediary:

1. In the `<install_dir>\conf\broker\xpllogging.properties`, set the logging level for the Broker to `FINE`:

   ```
   com.hp.ov.mip.level = FINE
   ```

2. Delete the `9032MipNotificationManager.xml` file to clean up the subscriptions. On Windows, it's in the `\tmp` directory. On UNIX, it's in the `/var/tmp` directory.

3. Restart the Intermediary.

4. Restart the SOA PE to make sure that the SOA PE subscribes to the Intermediary. Wait until you see the following message in the SOA PE log file:

   ```
   Jul 5, 2005 9:56:35
   AM;157;13;com.hp.wsm.sn.monitoring.collectionservice.CollectionSer
   vice;wseeAdded;com.hp.ov.mip.Auditing;INFO;><Subscribing to wsee
   http://<Service-Host>:9032/wsmf/services/Runtime$service=Wsee?wsdl
   listening for audit events
   ```

5. Send a request to the service so that an audit message is generated.

6. Verify that the message was dispatched from the Intermediary. You should see the following log messages in the Intermediary's log file:

   ```
   Jul 5, 2005 9:58:03
   AM;269;16;com.hp.wsm.sn.router.server.audit.MessageTraceBuffer$Que
   ueThread;run;com.hp.ov.mip.wsm.sn.router.server.audit.MessageTrace
   Buffer;FINE;>!<Dispatched 1 traces.
   ```

7. Verify that the SOA PE is subscribed for audit messages. Find the following message in the log file, which contains a list of the services subscribed for audit messages. Confirm that there is a `tuple` for the SOA PE that is subscribed to the `http://schemas.hp.com/mip/2004/WsExecutionEnvironment/Event/MessageTraceNotification` event type.

   ```
   Jul 5, 2005 9:58:03
   AM;270;16;com.hp.wsm.sn.router.server.audit.WSMFPublisher;dispatch
   ;com.hp.ov.mip.wsm.sn.router.server.audit.WSMFPublisher;FINE;>!<Cu
   rrent services subscribed for audit traces:
   ```

```
<SubscriptionTableList>
  <SubscriptionTable>
     <ManagedObject>Endpoint:id=e4f85099f4ab9246c0595be76856c2d3
     </ManagedObject>
     <SubscriptionList>
      <PushSubscriptions />
      <PullSubscriptions />
     </SubscriptionList>
  </SubscriptionTable>
<SubscriptionTable>

     <ManagedObject>SoapDispatcher:serviceId=financeServiceProxy
     </ManagedObject>
     <SubscriptionList>
       <PushSubscriptions />
       <PullSubscriptions />
     </SubscriptionList>
   </SubscriptionTable>
<SubscriptionTable>

     <ManagedObject>SmartBusinessAgent:service=Weuser
interfacerviceDirectory
     </ManagedObject>
     <SubscriptionList>
       <PushSubscriptions />
       <PullSubscriptions />
     </SubscriptionList>
   </SubscriptionTable>
<SubscriptionTable>

     <ManagedObject>Runtime:service=Service,id=financeServiceProxy
     </ManagedObject>
     <SubscriptionList>
       <PushSubscriptions />
       <PullSubscriptions />
     </SubscriptionList>
  </SubscriptionTable>
<SubscriptionTable>

     <ManagedObject>Runtime:service=Wsee</ManagedObject>
     <SubscriptionList>
      <PushSubscriptions>
        <EventType name="http://schemas.hp.com/mip/2004/
           WsExecutionEnvironment/Event/MessageTraceNotification">

          <Tuple>urn:subscription-push-2|Tue Jul 05 10:56:35 PDT
             2005|http://<NetworkServices_Host>:5002/
             _collectionServiceCallback
          </Tuple>
        </EventType>
      </PushSubscriptions>
      <PullSubscriptions />
    </SubscriptionList>
  </SubscriptionTable>
</SubscriptionTableList>
```

The date displayed in the `tuple` is the subscription expiration time. By default, the SOA PE sets the expiration time to the current time + 1 hour. If there is not an entry for the SOA PE and you are running the SOA PE and the Intermediary on different systems, it could be that the times on the systems aren't synchronized. Either synchronize the clocks or increase the SOA PE subscription expiration time in the `<install_dir>\conf\networkservices\mipServer.xml` file:

```
<entry name="com.hp.mip.event.subscriptionInterval">1440</entry>
```

8 Restart the SOA PE.

## Out of Memory

Receive an error that ran out of memory when running SOA PE as a service.

**Solution:**

Increase the stack and heap sizes.

1 Modify the <*install_dir*>\bin\win32\services\service-manager.bat file. Add the stack and heap parameters to the system properties (`@set SYS_PROPS=`**-Xms64m –Xmx256m** `-Dcom.hp.mip.autopass.home...`).

2 Run the bat file to remove the SOA PE service (`service-manager.bat -remove networkservices`).

3 Run the bat file again to add SOA PE as a service with the new parameters (`service-manager.bat -install networkservices`).

4 Check that the new parameters are configured by looking in the registry under `HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/networkservices<version num>`.

Receive an error that ran out of memory when running SOA PE from the command line.

**Solution:**

Increase the stack and heap sizes.

1 Modify the <*install_dir*>\bin\<unix | win32>\mipserver[.bat] file. Increase the sizes for `-Xms` and `-Xmx`.

2 Restart SOA PE.

## WSDL with JMS and HTTP Port Binding Fails

When multiple endpoints (JMS and HTTP) are configured, the request are not sent to the correct endpoint.

**Workaround:** You must remove the unused binding (either HTTP or JMS) from the WSDL before importing the WSDL for the endpoint.

## Broker Logs a Message till a Web Service is Undeployed

During a protocol switching scenario, when an invalid SOAP message is encountered, the Broker repeatedly logs fatal error messages that fill the outbound queue.

**Workaround**: You must undeploy the Web service.

## JMS-JMS Mediation Generates NULL Value Attributes in Security Audit Log File

For the JMS->JMS protocol switching scenario, in the security audit log file (sa.log), the start time, end time, and process name are listed as NULL.

**Workaround**: You must enable transport security to resolve this issue.

## Unable to Generate Web Service Metrics Report

Generation of Web Service Metrics report fails.

**Workaround**: The Reports link in SOA PE uses the audit traces collected from PEP to generate reports. You must make sure that you have attached an audit policy with the Web service configuration for which you are generating the Web Service Metrics report.

C

# Appendix C Technical Policies

The following section contains the guidelines to be followed while associating technical policies to a web service:

## WS-Addressing Policy

You can associate one WS-Addressing policy to a service. This policy is applicable only for a SOAP Service.

## Audit Policy

The Audit Policy in the Broker collects trace information on messages sent to Web services. The auditing feature can collect a message's SOAP payload and transport headers. The information collected is sent to HP SOA Policy Enforcer Server and is stored in a database. The HP SOA PE web interface (server) is used to query the database to retrieve audit information. Any management application can be extended to access the audit data.

Apart from a regular audit policy you can associate a security audit policy by enabling the security option on the audit policy. This policy is applicable for both XML service and SOAP Service. The Security Audit policy is used to collect security trace information (used for non-repudiation, and so on) and sends the payload to a security provider. When using the Security Audit policy, you must configure the security provider where security trace information will be sent.

If more than one regular audit policy or security audit policy is associated to a web service, SOA PE Broker uses one of the policies

**Note:**

Request:

SOA PE does not audit inbound request from a consumer to the SOA Policy Enforcer Broker. SOA PE audits the outbound requests from SOA Policy Enforcer Broker to the functional endpoint.

Response:

SOA PE audits the response sent from the SOA Policy Enforcer Broker to the consumer. SOA PE does not audit the response from the functional endpoint to the SOA Policy Enforcer Broker.

## JMS Mediation Policy

You can associate only one JMS mediation policy for a web service. This policy is applicable for both XML service and SOAP Service.

If there is more than one JMS mediation policy associated, SOA PE Broker uses one of the policies in the list.

## Transport Security Policy

You can associate only two transport security policies to a web service, one for the inbound transport and the other for outbound transport. This policy is applicable for both XML and SOAP Service where the transport used is HTTP.

If there are multiple inbound or outbound policies, SOA PE Broker uses one inbound transport security policy and one outbound transport security policy.

## Message Security Policy

You can associate only two message security policies to a web service, one for inbound message and other for outbound message. This policy is applicable for SOAP service only.

If there are multiple inbound or outbound message policies, SOA PE Broker uses one of the policies for inbound and one for outbound message security.

**Note:** When using both transport level and message level security with same authentication mechanism, the credentials used must be same.

## Schema Validation Policy

Schema validation ensures that SOAP requests conform to a Web service's WSDL. If the schema validation feature is enabled, requests that do not strictly conform to the WSDL are not dispatched to the service endpoint and an HTTP 500 error is returned by the Intermediary. If the schema validation feature is disabled, SOAP requests are not validated before being dispatched to the service endpoint. Depending on the level of nonconformity, a SOAP request may or may not be successful.

Schema validation is only applied to services implemented using document literal SOAP operations to the SOAP body only.

Schema validation is only applied to services implemented using document literal SOAP operations to the SOAP body only.

You can associate only one schema validation policy for a web service. This policy is applicable for SOAP Service only.

SOA PE Broker uses one of the policies if more than one policy is associated to the service.

## Event Policy

The alerts generated by the event policy at the broker are sent to the SOA PE server which notifies alert recipients (email, HP SOA PE Server console, and so on). You can associate only one event policy for a web service. This policy is applicable for SOAP Service and XML service. SOA PE Broker uses one of the policies if more than one policy is associated to the service.

## Transform Policy

You can associate one transform policy with direction set as request and another transform policy with direction set as response. Alternatively you can associate only one transform with direction set as 'both'. This policy is applicable for both XML service and SOAP Service.

If transform policies are defined with direction set as 'request','response' and 'both' SOA PE Broker uses those transform policies for which direction is set as 'request'/'response'. If multiple policies with the same direction are specified, SOA PE Broker uses one of the policies associated.

## Service Protection Policy

You can map only one service protection policy for a web service. This policy is applicable for both XML and SOAP services.

SOA PE Broker uses one of the policies associated to the service if multiple policies of this type are associated.

## Scheduled Availability Policy

You can map only one scheduled availability policy for a web service. This policy is applicable for both SOAP and XML services. SOA PE Broker uses one of the policies associated to the service if multiple policies of this type are associated.

## Content Detection Policy

You can map only one content detection policy for a web service. This policy is applicable for SOAP and XML services. SOA PE Broker uses one of the policies associated to the service if multiple policies of this type are associated.

## Load Balancing Policy

SOA PE creates this policy using the information entered while adding endpoints in the route table. Only one policy of this type can be associated to the service. SOA PE Broker uses one of the policies associated to the service.

## Route Policy

SOA PE associates route policies to a Web service when entries are added in the route table. SOA PE associates multiple policies of this type to the service depending on the endpoint information set in route table.

SOA PE Broker uses all the route policies while mapping the policies to the Intermediary. If a route policy is present with a JMS endpoint then SOA PE uses that route policy for a service. SOA PE ignores all the HTTP end point route policies are ignored while mapping.

## HTTP Pass-Through Transport Header Handler – Broker behavior

The HTTP Pass-Through Transport Header Handler copies transport headers from either side of the intermediary (request or response). This handler must be used in conjunction with, and before, the Dispatch Handler.

The headers are configured in <install_dir>/conf/broker/mipServer.xml. There is a property for both a request (SOAPAction is the default) and a response (no default):

```
<entry name="com.hp.transport.headers.pass.request">SOAPAction
</entry>
<entry name="com.hp.transport.headers.pass.response"></entry>
```

This handler copies JMS properties when the transport used is JMS.

When you set a request property, the handler copies properties in the request message from the broker to the service. For a response property, the handler copies properties from the service to the client.

D

# Appendix D Creating a Third Party Security Provider

SOA PE provides Java security interfaces that you can use to create your own third party security adapter. You can use these interfaces to configure your security adapter to support AAA.

You can integrate any third party security provider with SOA PE Broker. This involves the following steps:

- Create the adapter for integrating with the third party security provider.(SOA PE defines interfaces for initialization, authentication, authorization, and auditing of interfaces) by defining the interfaces and compiling and configuring the adapter.
- Register the security provider adapter with SOA PE Broker and configure SOA PE Broker to use the adapter.

## Create the Security Provider Adapter

During the creation of the security provider adapter, you must make sure that the security provider performs the required configuration tasks. This phase is known as the initialization phase. You can use the `com.hp.wsm.sn.common.security.provider.WsSecurityProvider` interface to make the security provider implement the initialization phase. After the initialization phase, SOA PE Broker allows you to implement the rest of the security interfaces as you may need. The Broker identifies each security provider with a unique name and registers the security provider using the configuration settings in Broker.

You can configure the Broker to use different security providers for authentication, authorization, and auditing. The `WsSecurityProvider` interface includes the following interfaces that are defined in the com.hp.wsm.sn.common.security.provider.iface package.

| **Interface Name** | **Description** |
|---|---|
| SecurityProvider | Use for initialization of the interface |
| SecurityAuthenticationProvider | Use for authentication at the transport level |
| SecurityAuthoriuzationProvider | Use for authentication at the transport and message level |

| | |
|---|---|
| WsSecurityProviderAuditor | Use for auditing |
| WebSecurityProvider | Use for authorizing the console login. Alternatively, you can also use the com.hp.wsm.sn.common.security.provider.WsAbstractSecurityProviderImpl for monitoring security violations. |

You can also use the `com.hp.wsm.sn.common.security.provider.WsAbstractSecurityProviderImpl` interface in place of the `com.hp.wsm.sn.common.security.provider.WsSecurityProvider` interface to implement the initialization phase. You can use this initialization interface to implement security violations monitoring.

Refer to the following sections for a description about the list of interfaces.

## Common Interfaces for SOA PE Server and Service Proxy

### *WsSecurityProvider*

***Description:*** The security provider must implement this interface. This interface allows you to specify the name of the security provider using the following operation.

String getName()

### *SecurityProvider*

***Description:*** The security provider must implement this interface. You can perform the initialization phase for the security provider using this interface that includes the following operation.

`Init(String Configuration)`:The configuration in the <broker_install_directory>\bin\conf\mipServer.xml file is passed to this operation for initialization.

### *WebSecurityProvider*

***Description:*** Use this interface to configure the security provider to authorize the UI login. This interface provides the following operations. You may also leave the operations blank.

- boolean authorize(Subject subject, SimpleHttpRequest httpRequest) throws AuthorizationException: Use this operation to authenticate a user logging in to the UI.
- public boolean isUserInRole(Principal principal, String roleName): Use this operation to authorize a valid user. You can use this operation to check if the user belongs to a specified role. SOA PE Server uses role-based access mechanism.

## Interfaces for Service Proxy

### SecurityAuthenticationProvider

***Description:*** You can use this interface if you want to use the security provider as the authentication provider. This is an optional interface and you can leave the interfaces blank to specify that this option must not be used. This interface includes the following operations:

- Boolean authenticate(Subject subject): Use this operation to specify that the security provider must be used for authentication at the transport level for the service proxy.
- StringgetSsoCookieName(): Use this operation to specify the cookie name that must be used for the Single Sign On (SSO) method.

### SecurityAuthorizationProvider

***Description:*** You can use this interface to configure the security provider as the authorization provider. This is an optional interface and you can leave the interfaces blank to specify that this option must not be used. This interface includes the following operations:

- boolean authorize(Subject subject, ServiceRequest webService, AuthorizationContext context) throws AuthorizationException: Use this operation to specify that the security provider must be used for authentication and authorization at the message level for the service proxy.
- Authorized getCumulativeAuthorize( Subject pSubject, ServiceRequest webService, AuthorizationContext context) throws AuthorizationException: Use this operation to specify that the security provider must be used for authorization at the transport level for the service proxy.
- String getSsoCookieName(): Use this to specify the cookie name that must be used for the Single Sign On (SSO) method.
- void setServiceSecViolation(String serviceId): Use this operation to set the security count violation for a service ID.
- int getServiceSecViolationCount(String serviceId): Use this operation to get the service security violation count for a service ID.

### WsSecurityProviderAuditor

***Description:*** You can use this interface to configure the security provider to log security information. This is an optional interface and you can leave the interfaces blank to specify that this option must not be used. This interface includes the following operation.

void auditMessage(SecurityAuditMessage message): Use this operation to specify the security audit message for logging information.

After defining the interfaces that you require you must perform the following steps to compile and configure the security provider for the Broker.

## Compile the Interfaces

Perform the following steps to compile the interfaces you defined:

1. Create a jar file by compiling the security provider interfaces
2. Use the following jar files as dependent jar files for the compilation:
   - <INSTALL_DIR>/lib/wsm-broker.jar
   - <INSTALL_DIR>/lib/mip-commons.jar

<INSTALL_DIR> signifies the directory in which you installed Broker.

## Register the Security Provider

Perform the following steps to configure the Broker to use the security provider:

1. Copy the jar file that you created in the section above to the <INSTALL_DIR>/lib/addons directory.
2. Update the <INSTALL_DIR>/conf/broker/mipServer.xml file with the required configuration changes to register the new security provider with the Broker. Refer to the following section for more details about updating the mipServer.xml file for configuration changes.
3. Start Broker.

## Updating mipServer.xml File

| Parameter Name | Description |
|---|---|
| com.hp.mip.security.providers*** | The values for this parameter indicate the security providers configured for the Broker. If you have multiple security providers, you must separate each entry with a semicolon (;) in this parameter. As each entry signifies a security provider, you must make sure that the entry names are unique. |
| com.hp.mip.security.provider.<name>.class*** | This parameter signifies the name of the security provider. You must make sure that the name specified in this parameter is the same as the name configured in com.hp.mip.security.providers parameter. The value for this parameter must be the fully qualified name of the security provider class that implements the com.hp.wsm.sn.common.security.provider.WsSecurityProvider interface. |
| com.hp.mip.security.provider.<name>*** | The value for this parameter signifies the configuration to be passed from the mipServer.xml file while initializing the security provider adapter. |
| com.hp.mip.security.provider.authorization* | This parameter signifies the name of the security provider to be used for authorization at both transport and message level. |
| com.hp.mip.security.provider.console*** | This parameter signifies the name of the security provider to be used for UI login |
| com.hp.mip.security.provider.authentication* | This parameter signifies the name of the security provider to be used for authentication. |
| com.hp.mip.security.provider.auditing* | This parameter signifies the name of the security provider to be used for security auditing |
| com.hp.mip.security.provider.management** | This parameter signifies the name of the security provider to be used by the SOA PE Server. |

The mipServer.xml file includes the parameters listed in the following table that you can use to configure the Broker to use the security provider.

In the table displayed above, the notations are as follows:

- The presence of * next to the parameter name signifies that the parameter is for a service proxy only.
- The presence of ** next to the parameter name signifies that the parameter is for SOA PE Server only.
- The presence of ***next to the parameter name signifies that the parameter is applicable for the service proxy and SOA PE Server.

## A Sample Configuration File

The following sample mipServer.xml configuration file configures the Broker to use default security provider for authorization, `customauth` for authentication, and `customconsole` for the user interface security.

```
<!-- security provider settings -->
<entry
name="com.hp.mip.security.providers">default;customconsole;customauth
</entry>
<entry
name="com.hp.mip.security.provider.authorization">default</entry>
<entry
name="com.hp.mip.security.provider.authentication">customauth</entry>
<entry
name="com.hp.mip.security.provider.console">customconsole</entry>
<entry name="com.hp.mip.security.provider.auditing">default</entry>
<entry
name="com.hp.mip.security.provider.default.class">com.hp.wsm.sn.commo
n.security.provider.BrokerDefaultSecurityProvider</entry>
<entry name="com.hp.mip.security.provider.
customauth.class">com.hp.auth.CustomAuthSecurityProvider</entry>
<entry name="com.hp.mip.security.provider. customauth">ldap</entry>
<entry name="com.hp.mip.security.provider.
customconsole.class">com.hp.auth.CustomConsoleSecurityProvider</entry
>
```

<entry name="com.hp.mip.security.provider. customconsole">userList</entry>

# Appendix E Service Modeling

This chapter provides conceptual information about the SOA Policy Enforcer's service modeling capabilities. The information includes:

- **Overview**: This section provides a basic overview of a service model and the importance of using service models.
- **Conceptual View**: This section provides a description of each element in the service model and how the service model relates to different individuals in an organization.
- **Defining Service Models**: This section provides a summary of the steps that are used to define service models when using the SOA PE Web Interface.

## Overview

A ***Service Model*** is the virtual representation of managed SOA resources. Currently, these resources include: Web services, intermediary services, Web service containers and Web service intermediaries,.

The service model's structure provides an organized view of the managed SOA resources and their relationships to each other. The structural elements that make up the service model are:

- Business services
- Configurations
- Policy Enforcement Point Group
- Application resources
- These structural elements are detailed in the "Conceptual View" section below. The section primarily focuses on an end user's view of the service model. However, at the code level, the service model is represented as a management information model, which is exposed externally in order to create integrated management solutions.

# Conceptual View

The service model is comprised of two main structural elements: Policy Enforcement Points and business services. This section describes these structural elements in both abstract terms as well as their specific application in the area of SOA. The description also includes the roles various people in an organization play in relation to these service model elements.

## Policy Enforcement Point Group

A ***Policy Enforcement Point*** as captured in the SOA Policy Enforcer represents the virtualization of management information or capabilities of a group of resources of a certain type that are associated with a set of stakeholders. The concept of virtualization of IT resources for the purpose of consumption is prevalent and well understood– examples of these include virtual networks, storage and blade systems, web server farms, application server clusters, database clusters, and many more. However, the virtualization of management of IT resources is relatively unprecedented.

The idea behind virtualization of management is to take various management capabilities (such as provisioning and configuration, performance and availability monitoring) that are typically well understood when applied to individual resources, and apply them to a new virtual but clearly identifiable and addressable entity called policy enforcement point group.

The virtualization of management capabilities is governed by a set of user configurable policies.

The management capabilities of Policy Enforcement Points are offered externally using a set of Web services. These Web service interfaces are documented in the *SOA PE Integration Guide*. Opening up the management interfaces in an open and standards, compliant manner provides the fundamental ability to use SOA Policy Enforcer to create integrated management solutions.

### Policy Enforcement Point Types

While this section described the concept of a Policy Enforcement Point in the abstract, the current version of the SOA Policy Enforcer's service model implements and understands two types of Policy Enforcement Points.

#### *Policy Enforcement Intermediary Group*

This type of Policy Enforcement Point captures the management of Policy Enforcement Intermediary Group and their hosted intermediary services. The Policy Enforcement Intermediary Group supports the deployment and discovery of an intermediary service. The WSM Intermediary is a Policy Enforcement Intermediary Group and is the only intermediary currently supported in the SOA Policy Enforcer.

### Policy Enforcement Point Stakeholders

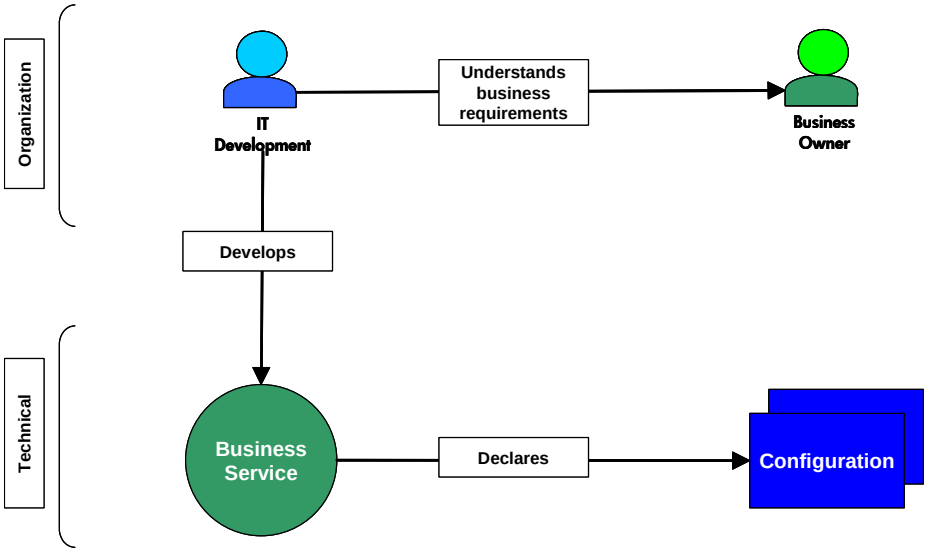
Policy Enforcement Point resources are typically co-located in the Data Center and have some Business Owner that pays for the provisioning and maintenance of these resources to run their Business Applications.

Various groups in IT have expertise in these different types of resources and are responsible for the various activities related to managing these IT resources. These activities include: provisioning, deployment, configuration, monitoring, problem management, change management, control, automation, versioning and upgrades. Moreover, each type of resource typically has some IT Operations and Support Contact. The ***Owner*** and ***Support*** contacts are examples of stakeholders or ***People*** that are involved in IT.

## Business Service

A ***Business Services*** is the virtualization of some business application that is offered by a business manager to either internal or external customers. Business applications are created and maintained by the IT department and are typically initiated and sponsored by business managers. Business applications are created to meet the needs of internal or external customers and typically represent some business product to the business manager. The figure below shows the relationship between a business manager, the IT department, and a business service.

SOA Policy Enforcer only models business services representing Web services. Because of this one-to-one relationship, the term business service is often used interchangeably with an offered or consumed Web service.




### Conceptual Architecture

As part of the business services definition, a business service configuration is created and bound to a PEP and its managed resources. For each PEP type, a corresponding business service configuration type is available.

The configuration types include Web Service Intermediary configurations

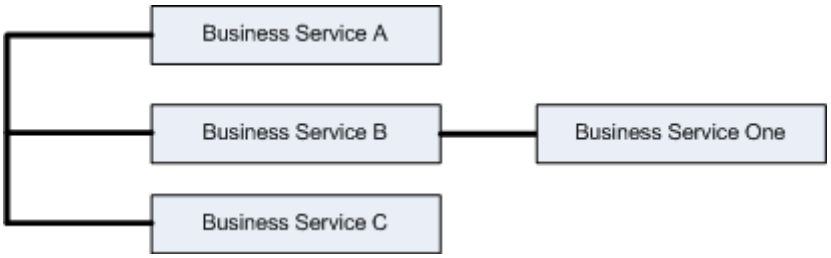
The use of configurations allows the model to provide automation features such as automatic resource discovery, automatic resource deployment, and automatic endpoint routing.

## Service Models

This section discusses some basic service model use cases that are supported by the SOA Policy Enforcer. The examples do not include every potential service model use case and should be considered a starting point for understanding service models.

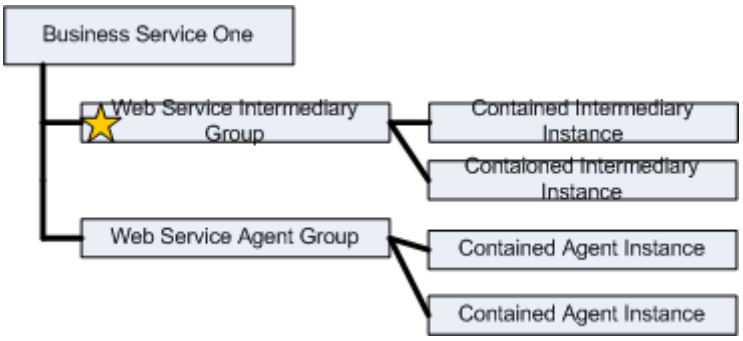
### Model – Business Service

A service model can contain multiple business services. A business service can also be related to other business services. The relationship between business services must be explicitly defined. The figure below shows a service model that contains three business services and one business service relationship.




### Model – Web Services Only

Business services contain the representation of Web services that are being managed. A business service can contain multiple Web services. Any Web service can be designated as an entry point to the model (see “Step 5: Designate the Entrypoint” later in this chapter). The figure below shows a basic service model. The star indicates entrypoints to the model.



## Business Service Configurations

Every business service contains one or more configurations. Business service configurations exist in the service model in order to link business services with the Policy Enforcement Points that contain managed resources (e.g., Web services). Once an Policy Enforcement Point is linked to a configuration, the managed resources can be added to the configuration and are ultimately managed through the business service.

Configurations are specific for each Policy Enforcement Point type. The configuration types include:

- Web Service Container configuration
- Web Service Intermediary  configuration

The use of configurations also allows the SOA Policy Enforcer to provide automation features such as automatic resource discovery, automatic resource deployment, and automatic endpoint routing.

# Defining Service Models

The SOA PE Web Interface provides a graphical way of creating, editing, and viewing service models. The service model functionality is spread across different screens; each screen is specific for the structural element of the model being defined.

In addition to the SOA PE Web Interface, much of the service model can be created, edited and viewed by using integration interfaces. Integration is discussed in the next chapter. This section only discusses the use of the SOA PE Web Interface.

The following tasks outline the typical manner in which a service model is defined. This section does not provide detailed procedural steps. Detailed procedures for these and many other tasks are included in the *SOA PE User Guide*.

In general there are two ways of creating service model

- Using Provisioning Wizard
  - **Create a Policy Enforcement Point** – This step involves the creation of the Policy Enforcement Points that are required to deliver a business application.
  - **Create Policies** – This step involves creation of  policies that must be associated with  service. This is optional.
  - **Provision Service** – This step involves creation of web service, association of policies to the web service, definition of Business service, association of web service to Business service and deployment of web service to a Policy Enforcement Point.
- Using manual steps
  - **Create a Policy Enforcement Point** – This step involves the creation of the Policy Enforcement Points that are required to deliver a business application.

- **Create a Business Service** – These steps involve creation of a business service that you can associate with the Web service to be brought under runtime governance.
- **Add a configuration** – These steps involve adding Policy Enforcement Points configuration that is required to register a PEP.
- **Add Resources** – These steps involve configuring the resources that are required to be brought under runtime governance with the business service.

F

# Appendix F Configuring LDAP for Authentication and Authorization

## Integrating LDAP with SOA PE

During database installation or by employing the Setup tool, you may choose to use accounts from external repositories. This chapter describes how to integrate accounts from an LDAP server and from non-LDAP user stores into SOA PE.

An LDAP server can be integrated with SOA PE with these scenarios:

- LDAP with a single search base - The scenario is very simple. There is only one LDAP server in this scenario. All identities are stored under a single search base
- LDAP with multiple search bases - In this scenario there is also only one LDAP server, but it has multiple search bases mapped to a domain. The domain is a specified part of the user's login name (that is, DOMAIN/USERNAME). All users must specify the domain name in the login dialog. When managing accounts or groups, we recommend using the DOMAIN/USERNAME format for performance reasons. If no domain is set, searches are performed across all domains
  - Multiple LDAP services - More than one LDAP service is used in this scenario. The correct LDAP service is chosen via DNS. As in the previous scenario, users must specify a domain name during login. When managing accounts or groups, users have to set domain name. If the domain name is not specified, then no domain is processed.
  - On Sun One LDAP above configurations are supported and on MS Active Directory only Single search base is supported

SOA PE treats external stores as read-only. User account properties stored in these external stores cannot be modified by SOA PE.

On Sun One LDAP above configurations are supported and on MS Active Directory only Single search base is supported.

You can provide the following details for LDAP configuration in the directory.xml file present at the following location <install_dir>\conf\networkservices. The default properties are noted in parentheses.

- Java naming provider URL.

  Provider URL and the port where the directory is running. (<property name="java.naming.provider.url" value="ldap://example.net:53371"/>).

- Initial Naming Factory

  LDAP Factory to be used. (<property name="java.naming.factory.initial" value="com.sun.jndi.ldap.LdapCtxFactory"/>).

- Security Principal

  Principal details which will be used to connect to the directory. (<property name="java.naming.security.principal" value="uid=admin,ou=People,dc=asiapacific,dc=hpqcorp,dc=net"/>)

- Password

  Password of security principal. (<propertyCoded name="java.naming.security.credentials" value_coded="password"/>)

- Security Protocol

  Name of the security protocol. (<property name="java.naming.security.authentication" value="simple"/>). You can also specify none instead of simple.

You can select the following LDAP usage scenarios:

- LDAP with a single search base: The scenario is very simple. There is only one LDAP server in this scenario. All identities are stored under a single search base.
- LDAP with multiple search bases: In this scenario there is also only one LDAP server, but it has multiple search bases mapped to a domain. The domain is a specified part of user's login name (that is, DOMAIN/USERNAME). All users must specify the domain name in the login dialog. During the managing with accounts or groups it is recommended to use DOMAIN/USERNAME because of performance. If no domain is set then search is performed across all domains.

Domains can be specified dynamically or statically. For dynamic settings it is necessary to specify, for example, a domain prefix or postfix. Static domains are set during the installation directly and so they must be known in time of installation.

## Multiple LDAP services

More than one LDAP service are used in this scenario. The correct LDAP service is chosen via DNS. As in the previous scenario, users must specify a domain name during login. When managing accounts or groups users have to set domain name. If domain name is not specified then no domain is processed.

For multiple LDAP the Java naming provider url will be as below:

Provider URL and the port where the directory is running (<property name=""java.naming.provider.url"" value=""ldap://example.net:53371 ldap://example2.net:53372"" />)."

**Note:**

Automatic discovery of the LDAP service using the URL's distinguished name is supported only in Java 2 SDK, versions 1.4.1 and later, so be sure of the Java version you are using.

The automatic discovery of LDAP servers allows you not to hardwire the URL and port of the LDAP server.

For example, you can use ldap:///o=JNDITutorial,dc=example,dc=com as a URL and the real URL will be deduced from the distinguished name o=JNDITutorial,dc=example,dc=com.

Systinet Registry integration with LDAP uses the JNDI API. For more information, see http://java.sun.- com/products/jndi/tutorial/ldap/connect/create.html and http://java.sun.com/j2se/1.4.2/docs/guide/jndi/jndidns.

html#URL

## Single LDAP with a Single Search Base

For single LDAP single search base the search base is same as what is set in soape.ldap.searchbase.user as follows

<scenario name="single.ldap.single.searchbase">  </scenario>

## Single LDAP with Multiple Search Base

If the domain is present in enable then the dn is the search base. That is, if the domain is asiapacific then the domain is ou=People,dc=asiapacific,dc=hpqcorp,dc=net. If there is no domain then the search base is as follows.

search base  = soape.ldap.domain.prefix  + domain  + ',' + (domainPostfix.length() > 0 ? domainPostfix + ',' : "") + soape.ldap.searchbase.user;

If any domain has to be disabled then it can be added to the disable domain as follows.

```
<scenario name="single.ldap.multiple.searchbase">
        <property name="soape.ldap.domain.delimiter" value="/"/>
        <property name="soape.ldap.domain.prefix" value="ou=People"/>
        <property name="soape.ldap.domain.postfix" value=""/>
        <domains>
          <enable>
            <domain name="asiapacific"
dn="ou=People,dc=asiapacific,dc=hpqcorp,dc=net"/>
            <domain name="others" dn="ou=org,dc=asiapacific,dc=hpqcorp,dc=net"/>
          </enable>
          <disable>
            <domain dn=""/>
```

```
            <domain dn=""/>
        </disable>
    </domains>
</scenario>
```

## User Properties Mapping

You can specify mapping between SOA PE user account properties and LDAP properties for the following attributes:

<property soapeName="loginName" directoryName="uid"/>

<property soapeName="email" directoryName="mail"/>

All other attributes, if any, can be mapped to a string.

## LDAP over SSL/TLS

It is only a matter of configuration to setup LDAP over SSL (or TLS) with a directory server of your choice. We recommend that you first install SOA PE with a connection to LDAP that does not use SSL. You can then verify the configuration by logging in as a user defined in this directory before configuring use of SSL. The configuration procedure assumes that you have already installed SOA PE with an LDAP account provider. Systinet Registry must not be running.

## LDAP over SSL without Client Authentication

In this case only LDAP server authentication is required. This is usually the case.

Edit the <install_dir>\conf\networkservices\directory.xml file (in case of SOA PE Server) or <install_dir>\conf\broker\directory.xml (for broker) as the case maybe, in one of the following ways depending on the version of Java used to run SOA PE:

If SOA PE will always be running with Java 1.4.2 or later, change the java.naming.provider.url property to use the LDAP protocol and the port on which the directory server accepts SSL/TLS connections. For example ldaps://sranka.in.idoox.com:636

## LDAP over SSL with Mutual Authentication

SOA Policy Enforcer does not support LDAP over SSL with mutual authentication.

# Appendix G List of Attributes to Configure XACML

Refer to the following table for the list of attributes you can use to configure XACML.

## Attribute Mapping

The following lists the attributes that are used during authorization by HP SOA Policy Enforcer

The XACML authorization request contains following 4 entities,

- Subject
- Resource
- Action

Subject

Represents information about who is accessing the resource.

| AttributeId | Type | Description |
|---|---|---|
| urn:oasis:names:tc:xacml:1.0:subject:subject-id | If UserId contains Domain name(Ex: user@xyz.com) then<br>Type is: urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name<br>Else<br>DataType is: http://www.w3.org/2001/X | Name of the user accessing the resource |

| | MLSchema#string | |
|---|---|---|
| group | http://www.w3.org/2001/XMLSchema#string | Group to which the user belongs. This is used only in case of LDAP. If the user belongs to multiple groups only the first in the search list is taken. |
| remoteHost | http://www.w3.org/2001/XMLSchema#string | IP Address/host name of the machine from where the request is received. Will be set only if HTTP is used as transport. |
| ldap:<attributename> | http://www.w3.org/2001/XMLSchema#string | If there are multiple values only the first one is taken. |

Resource

Represents information about what is being accessed

| AttributeId | Type | Description |
|---|---|---|
| urn:oasis:names:tc:xacml :1.0:resource:resource-id | http://www.w3.org/2001/XMLSchema#anyURI | Service URL being accessed. This is set only for http(s) transport. |
| service-id | http://www.w3.org/2001/XMLSchema#string | The service that is being accessed |

Action

Represents information about what is the action being performed on the resource. Only one action element need to be present.

| AttributeId | Type | Description |
|---|---|---|
| urn:oasis:names:tc:xacml :1.0:action:action-id | http://www.w3.org/2001/XMLSchema#string | The local name of the operation that is being accessed. This is available only for message level authorization |

# Appendix H Standards Support for Policy Enforcer 3.0

| Standard | Version |
|---|---|
| UDDI | *UDDI V3, GIF (Governance Integration Framework)* |
| WSDL | *WSDL 1.1, WSDL 1.2* |
| SOAP | *SOAP 1.1* |
| XML | *XML 1.0, XML 1.1* |
| XSLT | *XSLT 1.0* |
| XSD | *XML Schema 1.0* |
| WS-I Basic Profile compliance | *Web Services Interoperability (WS-I) Basic Profile 1.1, WS-I Attachments Profile 1.0* |
| Basic Security Profile compliance | *Covered under WS-Security* |
| WS-Security | *WS-Security 1.0, WS-UsernameToken Profile 1.0, WS-X.509 Certificate Token Profile* |
| WS-Addressing | *WS-Addressing 1.0* |
| WS-Policy | *WS-Policy 1.2, WS-PolicyAttachment 1.2* |
| HTTP transport | *HTTP 1.1* |
| SSL transport | *SSL 3.0* |
| JMS transport | *JMS 1.0* |
| SMTP transport | *Supported only for sending mail* |
| XACML | *XACML 1.1* |
| LDAP for authentication | *LDAP V3 supported* |
| SNMP TRAP | *SNMP V1* |

# Glossary

### Application Channel

Application channel refers to the request/response communication between an application client, such as a browser, and an application component such as a Web service.

### Auditing

Auditing is a management feature that captures trace information for all Web service requests and responses.

### Availability Monitoring

Availability monitoring is a management feature that is used to monitor the availability of SOA resources such as Web services.

### Brokered Services

A brokered service is a proxy to a final Web service endpoint and is used to enable the management of a Web service.

### Business Services

A business service is the virtualization of some business application that is offered by a business manager to either internal or external customers.

### Distributed Management

Distributed management is an approach to managing resources that are deployed and distributed across an enterprise network environment.

### Enterprise Management Integration

Enterprise management integration is the ability to leverage and/or customize the SOA Manger in order to create custom management solutions.

### Impact Analysis

Impact analysis is the ability to discover how the performance of a service affects other related services.

### Integration Points

Integration points provide the ability to either extract information from the SOA PE or add additional management data to the SOA PE.

### Interposed Manageability

Interposed manageability means inserting management policies in the request/response path of Web services.

### Logging

Logging in the SOA PE captures the local standard output for Web service containers and Web service intermediaries so that the output can be analyzed from a remote central location.

### Managed Object (MO)

An MO is a representation of a managed element such as a Web service. An MO can be related to either a logical or physical piece of the IT infrastructure. In the SOA PE, MOs are exposed as Web services that provide attributes and operations that can be invoked.

### Managed Service

A managed service is a Web service which is being managed by the SOA PE.

### Management Information Model

The management information model is a set of Web services (based on various standards such as WSDL, WSDM, etc.) consumable on the wire, and discoverable through meta-data populated in a UDDI registry.

### Management Policies

Management policies contain the management logic that is used to interpose visibility and controls on Web services. Management policies are implemented in the WSM Broker.

### Management Proxies

Management proxies are software components that get installed on a computer and are responsible for gathering management data for computers that do not have a native management agent available for them. The WSM Broker is an example of a management proxy.

### Management Server

A Management server is a centralized software component that aggregates the data that is gathered by any number of management agents. The SOA PE is an example of a management server.

### Management Web Service

A management Web service is a Web service that exposes management information using standard Web services management protocols. The WSM Broker expose their management information as management Web services.

### Public Key Infrastructure (PKI)

A PKI enables users of a basically unsecure public network such as the Internet to securely and privately exchange data through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. The public key infrastructure provides for a digital certificate that can identify an individual

or an organization and directory services that can store and, when necessary, revoke the certificates.

### Resource Management

Resource management is the act of managing the SOA resources that are being used by business applications.

### Root Cause Analysis

Root cause analysis is the ability to discover which Web service is causing a group of related Web services to degrade.

### Secure Sockets Layer (SSL)

SSL is a commonly-used protocol for managing the security of a message transmission over the Internet

### Service

A service is a self contained collection of functionality that promotes a high degree of isolation from internal details while at the same time offering its functionality to other services.

### Service Consumer

A service consumer is a participant in a service-based application that uses a service based on the functionality and value that the service provides.

### Service Level Agreement (SLA)

An SLA is an agreement between a service consumer and a service provider about the expected level of availability and performance of a service.

### Service Oriented Architecture (SOA)

An SOA is a set of principles that define an architecture that is loosely coupled and comprised of service providers and service consumers that interact according to a negotiated contract or interface.

### Service Model

A service model is the virtual representation of managed SOA resources.

### Service Producer

A service producer is a participant in a service-based application that focuses on how the service provides functionality and value and which resources provide the service.

### Simple Object Access Protocol (SOAP)

SOAP is an XML-based protocol that is typically used over HTTP to send messages (commonly referred to as SOAP messages) between application clients and servers. SOAP is the standard for Web services messages and is one of the foundation standards of Web services.

### Solution

A set of features and capabilities delivering business value to a customer through a combination of hardware, software, and services.

### Trend Analysis

Trend analysis allows operators and administrators to analyze changes in Web service performance over time.

### Universal Description, Discovery, and Integration (UDDI)

UDDI is a specification that defines a registry service for Web services that allows Web services to be discovered. UDDI is often referred to as a Yellow Pages of Web services.

### Web Service

A Web service is a service that is built using the SOAP and WSDL standards.

### Web Services (WS) Container

A WS container represents a SOAP container or environment that can host Web services. AXIS, IIS, and WebLogic Server are examples of WS containers.

### Web Service Management (WSM)

WSM is the act of managing the Web services that are being used by business applications. WSM in the SOA PE software goes beyond managing just Web services to include a range of SOA resources that are equally vital to the success of Web services.

### Web Services Description Language (WSDL)

WSDL is an XML-based language that is used to describe a software component. A WSDL definition describes how to access a Web service and what operations it can perform.

### Web Services Distributed Management (WSDM)

WSDM is an OASIS standard that has been formed to define web services management, including using web service architecture and technology to manage distributed resources.

### Web Services Intermediary

A Web services intermediary represents a proxy to a WS Container. The WSM Broker is considered a Web service intermediary. The Web services intermediary is also referred to as the policy enforcement point in the documentation.

### WSM Broker

The WSM Broker is a flexible, configurable, high performance Java-based Web services intermediary process. The WSM Broker is used to manage Web services that are hosted in containers that do not provide native management for Web services. The WSM Broker is an implementation of a Management Proxy and does not need to be co-located with the Web services being managed. The WSM Broker is also referred to as SOA PE Broker in the documentation.

### XML (Extensible Markup Language)

XML is a markup language used to describe data and does not include any presentation logic for the data.

# Index

## N



## O



## P



## Q



## R

## X