

HP Service Oriented Architecture Manager

User Guide

Version: 2.50

Windows®, HP-UX, Linux



October 2007

© Copyright 2004-2007 Hewlett-Packard Development Company, L.P.

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

© Copyright 2004- 2007 Hewlett-Packard Development Company, L.P., all rights reserved.

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

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

Table of Contents

Service Modeling	1-1
Overview	1-1
Conceptual View	1-2
Policy Enforcement Point Group	1-2
Policy Enforcement Point Types	1-2
Policy Enforcement Point Stakeholders	1-3
Business Service	1-3
Conceptual Architecture	1-4
Service Models	1-4
Model – Business Service	1-4
Model – Web Services Only	1-5
Business Service Configurations	1-5
Defining Service Models	1-6
Resource Management.....	2-1
Overview	2-1
WSM Features	2-2
Availability Monitoring	2-2
Performance Monitoring.....	2-2
Impact Analysis.....	2-3
Root Cause Analysis.....	2-3
SLO Monitoring	2-3
Auditing	2-3
Content Monitoring.....	2-4
Logging	2-4
Provisioning	2-4
Deployment.....	2-4

Security	2-4
Enabling WSM Manageability	2-5
Enablement Architecture.....	2-5
Deploying WSM Manageability	2-6
WSM Agents.....	2-7
WSM Intermediary	2-8
Supported Handlers for Agents and Intermediaries	2-9
Management Server.....	2-10
SOA Manager	2-10
SOA Manager Web Interface.....	2-11
Introduction	3-1
Document Overview	3-1
Audience.....	3-1
Prerequisites	3-1
Component Overview.....	3-2
HP SOA Manager	3-2
Service Model Definition	3-2
Performance and Availability Monitoring.....	3-3
SLO Management.....	3-3
Alerting	3-3
Auditing.....	3-4
Deployment.....	3-4
UDDI Publishing	3-4
Policy Enforcement Agents and Policy Enforcement Intermediary Components	3-5
Agent Handlers	3-5
WS-Based Management Enablement.....	3-6
Deployment Service.....	3-6
Audit Publisher.....	3-6
Dispatcher.....	3-6
Security	3-6
Integration Points	3-7
SOA Manager Roles	3-7

Line of Business Perspective	3-7
Development Team Perspective	3-8
IT Operations and Support Perspective	3-8
Improving Business IT Alignment and Providing IT Automation	3-8
Life Cycle Stages	3-9
Model Definition	3-9
Resource Discovery	3-10
SLO Monitoring	3-10
Problem Resolution.....	3-10
Deployment and Configuration	3-10
WSM Deployment Scenarios	3-10
Intermediary-Based Scenario	3-11
WSM Agent-Based Scenario	3-11
Intermediary vs. Agent Deployment.....	3-11
Getting Started.....	4-1
Finance Example Overview.....	4-1
Setting Up the Finance Application	4-2
Starting HP SOA Manager	4-2
Installing the HP SOA Manager as a Windows Service	4-3
Stopping HP SOA Manager	4-3
Windows	4-3
UNIX	4-4
Starting the HP SOA Manager Web Interface.....	4-4
Assigning Access to the Web Interface.....	4-4
Configuring HTTP Settings.....	4-5
Configuring the HTTP Server Port Number	4-5
Configuring HTTP Server Thread Settings	4-5
Configuring the Refresh Setting	4-6
Configuring the Threshold for Web Service Summary	4-6
Configuring an Oracle 10g Database	4-7
Configuring Time Zones.....	4-8
Performing Database Maintenance.....	4-8

Migrating an SOA Manager Database	4-8
Configure a UDDI Registry	4-9
Adding User Roles	4-9
Adding User Roles	4-10
Modifying Existing User	4-10
Removing Existing User.....	4-11
Using XPL Logging	4-11
Installing XPL Logging	4-11
XPL Tools	4-11
Configuring XPL.....	4-11
Configuring Log Levels	4-12
Viewing Logs	4-13
Using XPL Tracing	4-13
Installation.....	4-14
Windows	4-14
HP-UX	4-14
Linux	4-14
Example Configuration Entries	4-14
Managing Resources Using PEP Groups.....	5-1
Overview	5-1
Creating a PEP	5-2
Create a Policy Enforcement Intermediary Group	5-2
Create a Policy Enforcement Agent Group.....	5-2
Registering Resources	5-3
Registering Policy Enforcement Agent and Intermediary Resources.....	5-3
Managing Web Service Resources	5-4
Viewing Registered Resources	5-4
Viewing Log Traces	5-5
Editing and Querying Log Levels	5-5
Enabling Availability Notifications.....	5-6
Deleting a Resource	5-7
Managing Policy Enforcement Intermediaries or Agents.....	5-7

Viewing a Policy Enforcement Agent/Intermediary	5-8
Adding Resources.....	5-8
Removing Resources.....	5-8
Enabling Availability Notifications.....	5-9
Deleting a Policy Enforcement Agent/Intermediary PEP.....	5-10
HP SOA Manager	6-1
Technical Policies	6-1
Bundled Policies	6-1
Policy Enforcement Point.....	6-2
Creating a New Technical Policy	6-2
Creating an Audit Policy	6-2
Creating an Event Policy	6-3
Creating a Schema Validate Policy.....	6-5
Creating a Transform Policy	6-5
Creating a JMS Mediation Policy	6-6
Creating a Message Security Policy	6-6
Creating a Transport Security Policy	6-7
Modifying a Technical Policy.....	6-9
Deleting a Technical Policy	6-9
Exporting a Policy	6-9
Importing a Policy	6-10
Setting Up the Audit Components	6-10
Enable the Audit Handler	6-10
Policy Enforcement Intermediary Group.....	6-10
WSM J2EE Agent	6-11
WSM .NET Agent	6-11
Configure the Audit Publisher	6-12
Policy Enforcement Intermediary	6-12
WSM J2EE Agent	6-13
WSM .NET Agent	6-13
Configure the Database	6-14
Configuring the HSQL Database	6-14
Configuring an Oracle 10g Database.....	6-15
Viewing Audit Information.....	6-15

Viewing Reports	6-16
Web Service Metrics Reports	6-16
Audit Message Traces Reports.....	6-17
Service Provisioning.....	6-18
Prerequisites	6-18
Specify Service Details and PEP Types	6-18
Associate Technical Policies.....	6-20
Specify Service Details	6-20
Specify End Point Related Configuration for Load Balancing and Routing .	6-21
Set SLO Thresholds.....	6-22
Associate Web Service with Business Service	6-23
Choose Provisioning Option	6-24
Life Cycle Management	6-25
Dashboard.....	6-26
Accessing the Dashboard	6-26
Web Service Summary	6-27
Alert Statistics	6-28
Performance Summary	6-28
Web Service Performance Metrics	6-29
Changing Performance Summary Interval	6-30
Performance Graph	6-30
Changing the Service Polling Interval	6-30
Business Impact.....	6-31
Open Alerts.....	6-31
Using Alert Notifications	7-1
Overview	7-1
SLO Alerts.....	7-1
Assigning an SLO Alert to an Alert Category	7-2
Configuring the SLO Alert Polling Interval.....	7-2
Business Content Alerts	7-3
Defining a Business Content Alert	7-3
Policy Enforcement Intermediary	7-4
WSM J2EE Agent	7-5

WSM .Net Agent	7-6
Troubleshooting Business Content Alerts	7-8
SOA Manager Setup.....	7-8
Service Setup	7-8
Invocations.....	7-9
Customizing Alert Messages.....	7-11
Acknowledging Alerts.....	7-11
Querying Alerts	7-12
Setting Up Alert Recipients	7-12
Modifying an Existing Recipient Category.....	7-13
Creating Recipient Categories	7-13
Adding Alert Recipients to a Recipient Category	7-13
Creating Email Recipients.....	7-14
Creating Log Recipients.....	7-15
Creating SNMP Recipients	7-15
Using Business Services	8-1
Overview	8-1
Defining Business Services.....	8-2
Task 1: Create a Business Service	8-2
Task 2: Import Existing Policy Enforcement Points.....	8-3
Task 3: Add a PEP Configuration	8-3
Task 4: Add a Resource Configuration	8-4
Web Service	8-4
Importing a WSDL	8-6
Manually Adding Operations.....	8-6
Task 5: Designate the Entrypoint.....	8-7
Selecting Dependencies for a Business Service	8-7
Adding Routing Targets	8-8
Assigning Owner and Support Roles	8-9
Business Service Roles	8-9
Publishing Business Services to a UDDI Registry.....	8-10
JMS Support	8-10

Reusing a Business Service.....	8-11
Exporting a Business Service	8-11
Importing a Business Service	8-11
Deleting a Configuration.....	8-12
Deleting a Business Service.....	8-12
Using SSL for the Management Channel.....	9-1
Overview	9-1
Setting Up SSL.....	9-2
Assign Key Stores and Trust Stores	9-2
SOA Manager	9-2
Policy Enforcement Intermediary	9-3
WSM Agents.....	9-4
Configure SSL Settings.....	9-4
SOA Manager	9-4
Policy Enforcement Intermediary Management Channel.....	9-4
Broker Configurator	9-5
WSM Agents.....	9-5
Registering a Secure Policy Enforcement Agent /Intermediary	9-5
Accessing the SOA Manager Web Interface.....	9-6
Accessing the Broker Configurator	9-6
Integrating with Select Access.....	10-1
Overview	10-1
Setting Up the Select Access Integration	10-2
Install the Select Access Servlet Enforcer	10-2
Copy the Required Jars	10-3
Configure SOA Manager to Use Select Access.....	10-3
Modify Security Provider Settings for Select Access	10-4
Modify the Select Access Enforcer Properties File	10-5
Authenticating SOA Manager Web Interface and Broker Configurator Login..	10-5
Configuring Identities for SOA Manager Web Interface Role-based Authentication	10-5
Define a Select Access Resource Server for the SOA Manager Web Interface	10-8

Configure the Intermediary to Use Select Access.....	10-12
Modify Security Provider Settings for Select Access	10-12
Modify the Select Access Enforcer Properties File	10-13
Modify Select Access Settings for Auto Configuration of Services	10-14
Adding SA Certificates to Broker Keystore.	10-14
Enable Auto Configuration for SA.....	10-14
Configure Select Access for SOA Manager Auto Configuration	10-16
Define a Select Access Service for the Broker Configurator.....	10-17

WSM Intermediary An Overview..... 11-1

Prerequisites	11-1
Contextual Overview	11-2
Broker Configurator.....	11-2
Common Handlers	11-2
Monitoring Handler.....	11-2
Logging Handler.....	11-3
Auditing Handler	11-3
Schema Validation Handler	11-3
Business Content Alerting Handler	11-3
Security Handlers.....	11-3

Using Intermediary Services..... 12-1

Overview	12-1
Viewing Intermediary Service Details	12-2
Performance Metrics.....	12-2
Undeploying an Intermediary Service	12-2
Deploying an Intermediary Service	12-2
Editing an Intermediary Service	12-3
Changing an Intermediary Service's Version	12-3
Configuring an Intermediary Service's HTTP Path.....	12-4
Removing an Intermediary Service	12-4
Enabling Protocol Switching at the Intermediary	12-4
Prerequisites	12-5

Enabling JMS-to-JMS-Two-Way Protocol Switching.....	12-5
Enabling HTTP-to-JMS-One-Way Protocol Switching	12-6
Enabling JMS-to-HTTP-One-Way Protocol Switching	12-7
Configuring Handlers	13-1
Audit Handler	13-1
Fields	13-1
Configuring the Audit Publisher	13-2
Business Metric Alerts Handler	13-2
Fields	13-3
Generic SOAP Contract Handler.....	13-3
Fields	13-3
HTTP Pass-Through Transport Header Handler.....	13-4
Invocation Handler	13-4
Fields	13-4
Log Handler.....	13-4
Fields	13-4
Schema Validation Handler.....	13-5
Security Auditing	13-5
Field.....	13-5
Configuring Security Auditing.....	13-5
Service Security Inbound Handler.....	13-5
SOAP Contract Handler	13-6
SOAP Dispatch Handler.....	13-6
SOAP Monitoring Handler.....	13-6
Fields	13-6
WS Security Outbound Handler	13-6
Fields	13-7
WS Security Message Processing Inbound Handler.....	13-7
Fields	13-7
XML Contract Handler.....	13-8

XML Dispatch Handler	13-8
XPath Monitoring.....	13-8
Fields	13-8
XSLT Handler.....	13-9
Fields	13-9
Classifier Handler.....	13-9
Fields	13-9
Using Custom Intermediary Services	14-1
Overview	14-1
Convert a Simple Intermediary Service	14-2
Adding Handlers.....	14-2
Adding Custom Handlers	14-3
Defining Service Providers for Custom Web Services	14-3
Enabling Content-based Routing	14-5
Getting Started	15-1
Starting the WSM Broker.....	15-1
Stopping the WSM Broker.....	15-2
Windows	15-2
UNIX	15-2
Starting the Broker Configurator Console	15-2
Installing the Broker as a Windows Service	15-3
Configuring HTTP Settings.....	15-3
Configuring the HTTP Server Port Number	15-4
Configuring the Broker's Management Channel Port.....	15-4
Configuring HTTP Server Thread Settings	15-5
Configuring HTTP Client Settings	15-5
Configuring HTTP Proxy Settings	15-6
Assigning Access to the Console	15-6
Using XPL Logging	15-7
Installing XPL Logging	15-7

XPL Tools	15-7
Configuring XPL.....	15-7
Configuring Log Levels	15-8
Viewing Logs	15-9
Using XPL Tracing	15-9
Installation.....	15-9
Windows	15-9
HP-UX	15-10
Linux	15-10
Example Configuration Entries	15-10
Implementing Load Balancing and Failover.....	16-1
Overview	16-1
Conceptual Architecture	16-2
Load Balancing Scenario	16-2
Failover Scenario	16-2
Setting Up Load Balancing and Failover	16-3
Defining Multiple Endpoints in a WSDL File.....	16-3
Configuring Load Balancing and Failover	16-3
Using Multiple Intermediaries	16-4
Using the Intermediary's Security Features.....	17-1
Overview	17-1
Feature Matrix.....	17-2
Supported Security Scenarios	17-3
Scenario 1: Intermediary is the Entry Point for External Consumers	17-4
Scenario 2: Web Application is the Entry Point for External Consumers ..	17-5
Scenario 3: Intermediary is the Exit Point for External Providers.....	17-5
Transport Level Security	17-5
Message Level Security	17-6
Inbound Message Processing.....	17-7
Outbound Message Processing.....	17-7
Setting Up the Security Components	17-7
Configure a Key Store.....	17-8

Configure a CA Trust Store.....	17-8
Configure the Intermediary's SSL Port.....	17-9
Setting Up Authentication and Authorization.....	17-9
Using Select Access	17-10
Setting Up Basic Authentication Only	17-10
Setting up Basic Authorization	17-12
Setting Up X.509 Authorization.....	17-13
Enable Select Auth for Basic Authorization and X.509 Certificate Authorization.....	17-15
Mapping Resources in Select Access.....	17-16
Implementing a Security Scenario.....	17-17
Inbound Transport Security.....	17-18
Enabling SSL.....	17-18
Enabling Authentication	17-18
Outbound Transport Security	17-19
Enabling Outbound SSL	17-19
Inbound Message Security	17-20
Outbound Message Security.....	17-21
Management Channel HTTP Basic Authorization	17-22
Introduction to WSM .NET Agent	18-1
Prerequisites	18-1
Contextual Overview	18-1
Control Point	18-2
WS Management Web Services	18-2
Runtime Description	18-2
Installing the WSM .NET Agent.....	19-1
Software Requirements.....	19-1
Pre-Installation Steps	19-1
Running the Installer	19-2
Verifying the Installation	19-4
Uninstalling the WSM .NET Agent	19-5

Configuring SOAP Extensions	20-1
Overview	20-1
Creating a Service Configuration	20-1
Registering Configuration Handlers	20-2
Using Service Specific Configurations	20-3
Using a Default Service Configuration	20-3
Verifying Extension Types	20-4
Logging Extension.....	20-4
Example	20-5
Audit Extension	20-5
Example	20-5
Business Metric Extension	20-6
Example	20-7
Introduction- WSM J2EE Agent	21-1
Prerequisites	21-1
Contextual Overview	21-1
Common Handlers	21-2
Monitoring Handler.....	21-2
Audit Handler	21-2
Business Metric Handler	21-3
Deploy the WSM J2EE Agent for WebLogic	22-1
Software Requirements.....	22-1
Pre-Deployment Steps	22-2
ANT 1.5.x Setup.....	22-2
WebLogic Server 8.1.x Setup	22-2
Deploy to the Administration Server.....	22-2
Extract the WSM J2EE Agent for WLS	22-3
Modify the Environment Setup Script.....	22-3
Modify the Agent-Setup Script	22-4
Run the Agent-Setup Script	22-4
Modify the WebLogic Startup Script.....	22-5

Modify the WebLogic Policy File	22-5
Start the WebLogic Administration Server	22-5
Verify the Deployment.....	22-6
Verify the Automatic Discovery of Web Services	22-7
Deploy to Managed Servers.....	22-8
Extract the WSM J2EE Agent for WLS	22-9
Modify the Environment Setup Script.....	22-10
Modify the BEA Node Manager.....	22-10
Modify the Node Manager Windows Service	22-10
Modify the Node Manager Standalone Process.....	22-11
Start the Managed Server	22-11
Verify the Agent Deployment to a Managed Server	22-12
Verify the Automatic Discovery of Web Services	22-13
Un-Deploying on the Administration Server	22-14
Un-Deploying on Managed Servers	22-15
Deploying the MOM Agent for WLS	22-16
Appendix A Creating a Java Key Store.....	A-1
Step 1: Create a Private Key and the Initial Java Key Store File (JKS file)	A-1
Step 2: Generate a CSR request	A-2
Step 3: Obtain a Signed Certificate from a Certificate Authority.....	A-2
Step 4: Import Signed Server Certificate to Key Store	A-3
Troubleshooting SOA Manager	B-1
Troubleshooting Tips/FAQ	B-1
Installation and Configuration Problems.....	B-2
Errors occurred during installation	B-2
AutoPass fails to install	B-2
Unable to add Intermediary to Policy Enforcement Agent Group.....	B-3
Runtime Problems.....	B-3
Could not start monarch-sba.....	B-3
Failed to initialize listener.....	B-4
Timezone error when using Oracle 9i	B-5

Performance data not showing up in Business Service	B-5
Performance graph error on HP-UX and Linux	B-6
Intermediary audit traces not showing up in BSE.....	B-7
Out of Memory	B-10
WSDL with JMS and HTTP Port Binding Fails.....	B-10
WebLogic Server WSDL with JMS Topics Fail	B-10
Broker Logs a Message till a Web Service is Undeployed.....	B-11
JMS-JMS Protocol Switching Generates NULL Value Attributes in Security Audit Log File	B-11
Troubleshooting Intermediary	C-1
Installation and Configuration Problems.....	C-1
Errors occurred during installation	C-1
AutoPass fails to install	C-1
Runtime Problems.....	C-2
Could not start monarch-sba.....	C-2
Failed to initialize listener.....	C-3
Unable to determine binding from message element.....	C-3
Authentication header not progressed to backend.....	C-3
Select Access enforcer cannot connect to validator.....	C-4
XML message not being passed to Select Access	C-4
Out of Memory	C-4
Troubleshooting J2EE Agent.....	D-1
Runtime Problems.....	D-1
Remote Services is not Discovered	D-1
Performance Data not Available from Cluster.....	D-2
Non-Existent Services in deployed services list.....	D-2
WsmfEventHandlerChain could not be loaded	D-2
Appendix A Technical Policies.....	E-1
Technical Policies- Policy Mapping.....	E-1
Route Policy.....	E-1
Audit Policy	E-1
JMS Mediation Policy.....	E-1
Message Security Policy.....	E-2

Transport Security Policy	E-2
Log Policy	E-2
Schema Validation Policy	E-2
Event Policy	E-2
Transform Policy	E-2
Load Balancing Policy.....	E-2
Notes.....	E-3
Glossary	G-1
Index	I-1

Service Modeling

This chapter provides conceptual information about the SOA Manager'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 HP SOA Manager 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 Manager 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. Performance and availability of the underlying resources may be specified on a resource contained in a Policy Enforcement Point group using a set of SLOs. At runtime, any violations of these SLOs may generate management events that may be used for human or machine consumption.

The management capabilities of Policy Enforcement Points are offered externally using a set of Web services. These Web service interfaces are documented in the *HP SOA Manager Integration Guide*. Opening up the management interfaces in an open and standards, compliant manner provides the fundamental ability to use SOA Manager 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 Manager's service model implements and understands two types of Policy Enforcement Points.

Policy Enforcement Agent Group

This type of Policy Enforcement Point captures the management of Policy Enforcement Agent Groups and their hosted Web services. The Policy Enforcement Agent Group supports the deployment, discovery, and SLO monitoring of a Web service implementation deployed to multiple Web services containers. A Policy Enforcement Agent Group must expose its manageability using a WSM Agent (J2EE or .NET).

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, discovery, and SLO monitoring of a intermediary service. The WSM Intermediary is a Policy Enforcement Intermediary Group and is the only intermediary currently supported in the SOA Manager.

Policy Enforcement Point Stakeholders

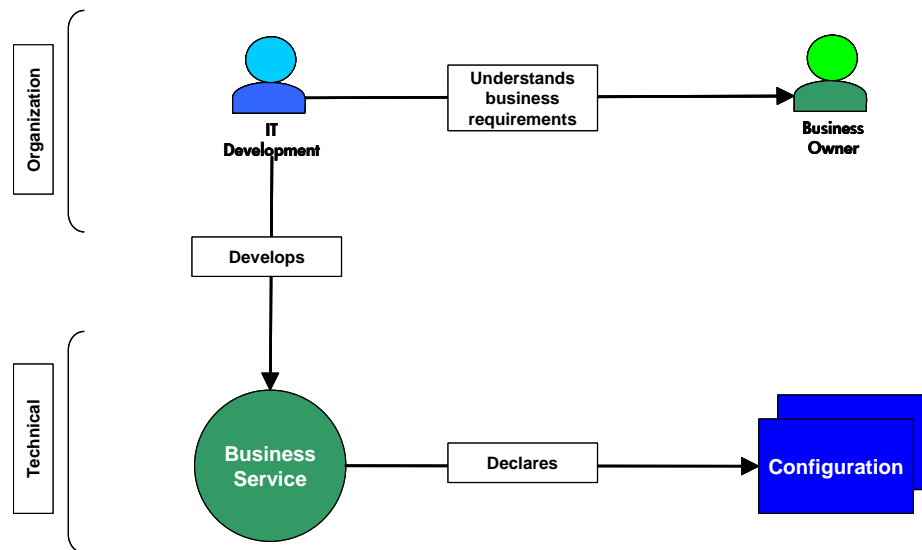
Policy Enforcement Point resources are typically co-located in the Data Center and have some Business Owner that pays for the provisioning and maintenance of these resources to run their Business Applications. **Error! Reference source not found.** below shows the relationship between a resource owner, the resource administrators, and a Policy Enforcement Point.

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 Manager 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 Agent configurations
- Web Service Intermediary configurations

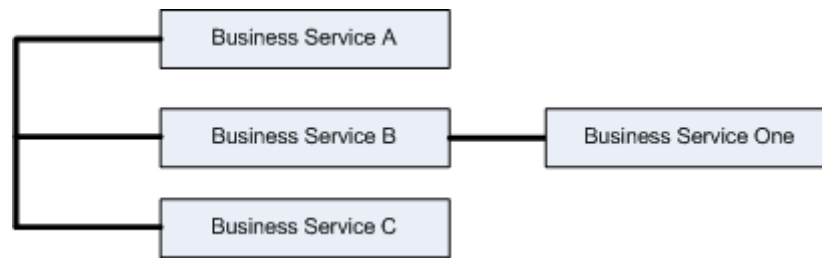
The use of configurations allows the model to provide automation features such as automatic resource discovery, automatic resource deployment, and automatic endpoint routing.

Service Models

This section discusses some basic service model use cases that are supported by the SOA Manager. The examples do not include every potential service model use case and should be considered a starting point for understanding service models.

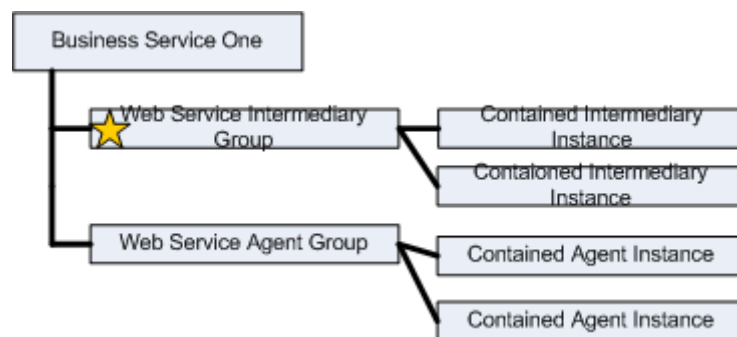
Model – Business Service

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 Manager to provide automation features such as automatic resource discovery, automatic resource deployment, and automatic endpoint routing.

Defining Service Models

The HP SOA Manager 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 HP SOA Manager 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 HP SOA Manager 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 *HP SOA Manager User Guide*.

In general there are two ways of creating service model

Using Provisioning Wizard

- **Create a Policy Enforcement Point** – These steps are often performed by IT administrators or operators who are aware of the Policy Enforcement Points that are required to deliver a business application.
- **Create Policies** – These steps are often completed by IT Administrator which involves creation of policies that should 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
- Steps for creating the service model include:
 - **Create a Policy Enforcement Point** – These steps are often performed by IT administrators or operators who are aware of the Policy Enforcement Points that are required to deliver a business application.
 - **Create a Business Service** – These steps are often completed by a business manager who is responsible for providing a business application to internal or external customers.
 - **Add a configuration** – These steps are often completed by IT Development members who are aware of the Policy Enforcement Points that are required to deliver a business service.
 - **Add Managed Resources** – These steps are often completed by IT Development members who are aware of the resources (e.g., Web services) that are required to deliver a business service.
 - **Set SLO Policies** – These steps are often completed by IT administrators or operators who are responsible for monitoring the health of the resources. However, SLOs are typically defined by business managers.

Resource Management

This chapter provides conceptual information about the SOA Manager's Resource management capabilities. In particular, the content focuses on Web services management. The chapter includes:

- **Overview:** This section provides a basic definition of resource management and why it is important.
- **Service Management Features:** This section provides a description of each service management feature and the benefits that each provides.
- **Enabling Manageability:** This section provides basic concepts that are needed to understand how the SOA Manager implements manageability for Web services and SOA resources. A generic architectural view is provided.
- **Deploying Manageability:** This section provides a description of the SOA Manager's Web Services Management (WSM) agents. These agents are installed in SOA environments in order to manage Web services and SOA resources.
- **Management Server:** This section provides a description of the default management server that is included as part of the SOA Manager's service management capabilities.

Overview

Resource Management is the act of managing the resources in an SOA that are being used by business applications. The SOA Manager includes a range of SOA resources that are vital to the success of a business application. These resources include: Web services, Intermediary Web services, Web service agents and WSM Intermediaries (a Web service proxy also referred to as a Web service intermediary). This chapter only focuses on Web services management.

Operators and administrators utilize resource management to ensure SOA resources are always available and functioning within acceptable operating limits. When problems occur, resource management allows them to quickly identify possible causes and initiate appropriate resolution procedures.

Web services are prevalent within current SOA environment implementations and their management is essential to the overall success of an SOA. Web service management is important because the health and well-being of a single Web service may impact the overall health and well-being of multiple business applications.

WSM Features

There are some basic management objectives that the SOA Manager software provides in the service management space. These objectives include:

- Ensuring the high-availability of service-based applications
- Ensuring the optimum performance of service-based applications
- Decrease response and resolution times
- Maintaining a record of Web service usage
- Troubleshooting and diagnosing problems
- Ensuring the secure usage of Web services

The topics in this section describe the features that allow administrators and operators to achieve these objectives and more.

Availability Monitoring

Availability Monitoring determines when Web services and SOA resources are no longer operational and then generates an alert notification. This feature allows administrators to quickly react to errors and mitigate application downtime. Availability alerts are typically the first indication that a problem has occurred with a Web service or an SOA resource.

Performance Monitoring

Performance Monitoring captures a set of real-time performance metrics that clearly indicate the health, availability, and performance of Web services. The metrics include:

- Availability %
- Average Response Time
- Failure Count
- Maximum Response Time
- Minimum Response Time
- Security Violations
- Success Count
- Total Requests
- Uptime %
- Uptime

The SOA Manager software captures the performance and availability experience of real consumers, computed by monitoring real transactions, without doing externally probed synthetic transactions.

An important part of this feature is that the metrics are calculated over time. This allows operators and administrators to analyze changes in Web service performance. This is commonly referred to as **Trend analysis**.

Impact Analysis

Impact Analysis is the ability to discover how the performance of a service affects other related services. When performing impact analysis in the SOA Manager, a relationship between services must be explicitly defined. For example, *Service A* is related to *Service B*. In this example, if *Service B* depends on *Service A*, then any performance problems for *Service A* can also affect *Service B*. Dependencies are defined in the service model.

Administrators and operators use this feature to quickly visualize the impact a poorly performing service may have on other services and business applications.

Root Cause Analysis

Root Cause Analysis is the ability to discover which services are causing a group of related services to degrade. Like impact analysis, a relationship between services must be explicitly defined. For example, *Service A* is related to *Service B*. In this example, if *Service A* depends on *Service B*, then any performance problems for *Service A* may be the result of a problem with *Service B*. Dependencies are defined in the service model.

Administrators and operators use this feature to quickly troubleshoot which service is causing an overall group of related services to degrade. A considerable amount of time can be saved by pinpointing a problem without having to manually complete a process of elimination.

SLO Monitoring

SLO Monitoring evaluates a Web service's performance metric values (described above) against pre-defined service level objectives (SLO). **SLOs** are the preferred operating limits for a Web service.

For example, an SLO may stipulate that a Web service's Availability % be greater than 90% and have an Average Response Time of less than 200 Milliseconds. At runtime, if these SLO values are violated, an alert notification is generated that indicates the breach and also provides the actual values of the performance metrics (in this case, the actual Availability % and Average Response Time). An additional alert is generated when the service levels return to normal.

Administrators and operators use this feature to stay informed about changes in a Web service's performance before the changes can affect a user's experience or break a service level agreement (SLA). **SLA** is an agreement between a service consumer and a service provider about an expected level of availability and performance of a service.

Auditing

Auditing captures trace information for all Web service requests and responses. Trace information provides a historical record of a Web service's performance, access history, security, size, source and destination endpoints, successes, failures, and can also include the SOAP request-response payloads and profile data. Trace information can be persisted to a database at regular intervals. The information is used to generate audit reports or can be used by other auditing applications.

Administrators and operators use auditing for a number of reasons. The reasons may include repudiation (i.e., evidence or proof that an SLA has either been maintained or broken), billing or metering, or to validate unauthorized access to a Web service.

Content Monitoring

Content Monitoring searches Web service request and/or response messages for specific content. An alert notification is generated when the content is found.

Administrators and operators use this feature to react to events that can potentially have an impact on business operations. For example, when managing an order process service, an alert notification could be generated 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

Logging

Logging captures the local standard output for SOA components so that the output can be analyzed from a remote central location. Administrators and operators use this feature to view the current log messages as well as change log output levels in order to view more detailed log messages. Logging is particularly useful when troubleshooting problems.

Provisioning

Provisioning allows configuration of web service proxies and associated policies from remote central location using a wizard. Administrators and operators can use this feature to deploy and activate a proxy service, deploy a service for later activation, or save the service for deploying or activating at a later time.

Deployment

Deploying installs Web services or Web service proxies from a remote central location. Administrators and operators use this feature to install new or updated versions of a Web service. The feature is also used to install additional instances of a Web service or Web service proxies to compensate for increased demand. This feature can save administrators and operators a considerable amount of time and allow them to quickly adapt to changes in business applications.

Security

Security ensures that access to Web services is secure. The security features are implemented using several industry standard security technologies and Select Access. These include:

- Transport Level Security: HTTP/S, X.509 Client Certificates
- Message Level Security: XML Digital Signature, XML Encryption - WS-Security
- Authorization/Authentication: AAA Security Integration with HP Select Access

Administrators and operators typically use these features to secure communication to and from the SOA Manager's WSM Intermediary. The WSM Intermediary is discussed later in this chapter.

Enabling WSM Manageability

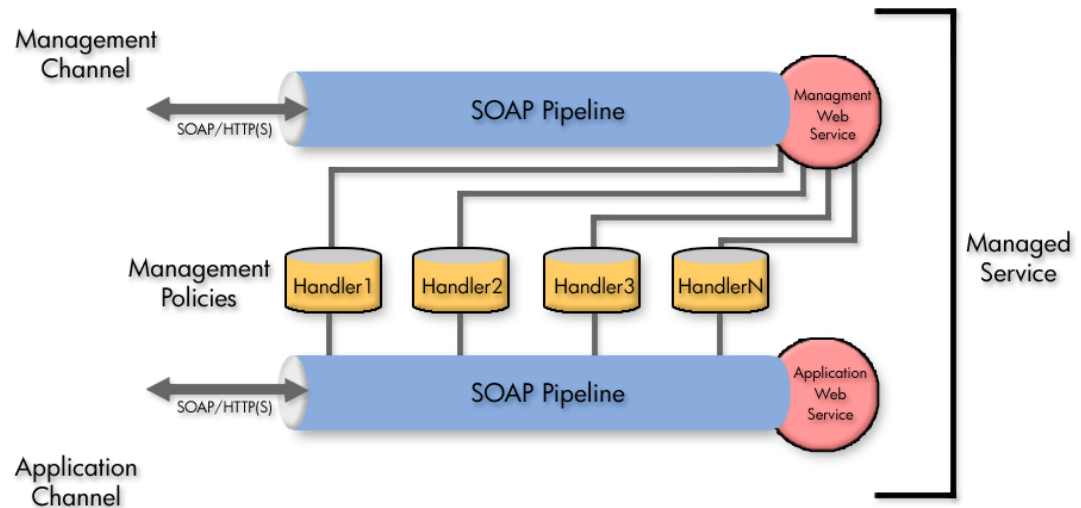
Most enterprises have standardized on a Web services container to host their Web services. Web services containers are included with standard J2EE platforms (such as BEA WebLogic, IBM WebSphere, Apache Axis, JBoss, etc...). Microsoft uses the .NET platform and the Internet Information Server (IIS) to host Web services. Lastly, there may be proprietary server environments that provide a means of hosting Web services.

None of these containers expose adequate instrumentation and manageability information about their hosted Web services. Moreover, the specifications used to implement Web services (such as WSDL and SOAP) do not provide any built-in manageability. Web services management vendors are left to their own devices to implement appropriate manageability instrumentation hooks into Web services.

The SOA Manager software uses *Interposed Manageability* to manage Web services. Interposed manageability means inserting management policies in the request/response path of Web services. This allows Web services to be managed in a standard and consistent manner and also allows the management of Web services that were not designed with manageability in mind.

Enablement Architecture

Management Policies contain the management logic that is used to interpose visibility and controls on Web services. The actual implementation of the management policies is done using *Policy Handlers* (also referred to as simply *Handlers*). Handlers are inserted in the HTTP or SOAP pipeline that is responsible for processing request and response messages. Multiple handlers can be linked together in a *Handler Chain*.



The management information that is obtained from the handlers is then exposed through a **Management Web Service** that utilizes standard Web services management protocols. Management clients (such as the SOA Manager) use the management Web services to construct views of management data. Management Web services provide both operations and event exchange patterns.

- Management Web services are also used to provide manageability for WS containers such as discovery and deployment of Web services to the WS container.

Management Web services can be published to a separate **Management Channel** instead of the **Application Channel**, which typically only contains application related traffic. For example, an application could invoke Web services that are deployed in a Web services container which is accessed using port 8080. The SOA Manager invokes the Web service's management Web service using port 8090.

Independent channels provide a good separation of management and application traffic to ensure that management traffic does not adversely affect the performance for users of the application.

Deploying WSM Manageability

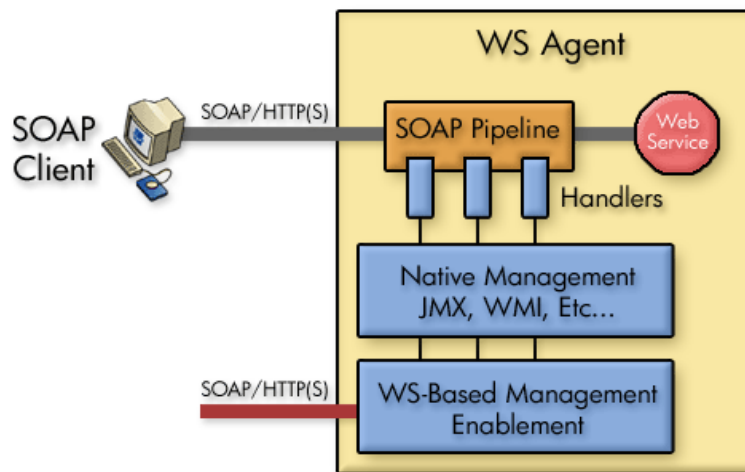
There are two ways to manage Web services: directly in a Web services container (using a WSM Agent), and through the use of a Web services intermediary process (using the WSM Intermediary). Both of these deployments are typical of distributed management solutions. In both cases, the underlying enablement architecture described in the previous section remains relatively the same. This section describes each of these deployment options. Service Management setup typically begins by setting up one of these options into a service environment.

WSM Agents

A **WSM Agent** is an enablement component that is installed in a WS Container in order to manage the Web services in the container as well as the container itself. WSM Agents are implemented using specific technologies and programming languages that are native to the WS Container's platform. There are currently two agents that can be deployed:

- The **WSM J2EE Agent** – an agent for the WebLogic Server Web services Container. The J2EE agent is written leveraging Java J2EE specifications and technologies such as JMX, JAXRPC, RMI, and Servlets.
- The **WSM .Net Agent** – an agent for the .NET Web services Container. The .Net agent is written as a .Net C# application that leverages Windows/.Net technologies such as WMI instrumentation, global HTTP pipeline and the WSE pipeline offered by the .Net programming model.

Agents are often referred to as **Platform Native WSM Agents** because they are specific to a certain platform. The figure below shows a high-level architecture of a WSM Agent.



The agents have two core components:

- A set of policy handlers that must be incorporated into the Application Web service path. The handlers update JMX or WMI instrumentation internally. SOA Manager ships with the implementation of these handlers. However, the handlers must be manually enabled by modifying XML files.
- A management-related Web services application that converts native JMX or WMI information into standard management Web services. This application is provided by SOA Manager but must be installed into a WS Container. This is an initial one time installation and configuration process.

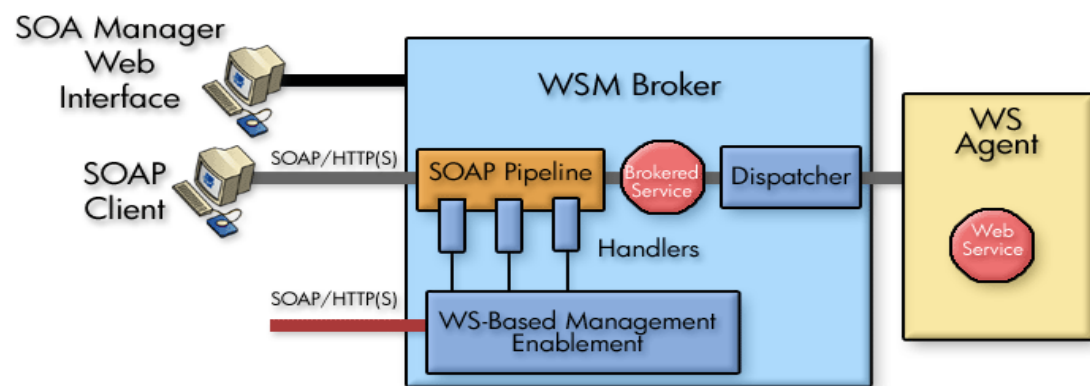
Installation and configuration instructions for the WSM J2EE Agent and the WSM .NET Agent are located in the *HP SOA Manager User Guide*. This guide is located in the /documentation directory of the distribution.

WSM Intermediary

The **WSM Intermediary**, also referred as Policy Enforcement Point intermediary, is a flexible, configurable, high performance Java-based Web services intermediary process. It is a self-contained application that runs in a single JVM process and does not require an existing Application Server. The WSM Intermediary is not specific to any particular Web Services container; therefore, it can be used to add manageability to any container.

The WSM Intermediary hosts **Intermediary Services**, which are Web service proxies to the Web service being managed.

An intermediary service must be created for each Web service that you want to manage. The SOA Manager web interface provides the ability to configure and deploy intermediary services.



Advanced solutions can create custom handlers and include them for execution in the Intermediary. Advanced solutions may directly create configurations using the documented XML configuration format. The *SOA Manager Integration Guide* contains detailed information about the Java-based APIs for creating custom handlers. The custom handlers cannot build additional information such as new metrics or any behavior that reflects in the SOA Manager web interface. However, custom handlers can do some local processing of messages.

The WSM Intermediary generates a WSDL for each intermediary service, replacing the application Web service endpoint with the brokered Web service endpoint. Web service consumers send requests to this brokered Web service. Steps must be taken to ensure that consumers do not directly go to the Web service implementations and are always routed through the Intermediary. This can be done using manual processes and best practices or enforced using IP firewall policies.

Runtime requests that are sent to the intermediary service are processed for management and security purposes, and then forwarded (dispatched) to the actual service's endpoint(s). The extra network hop introduces a small latency that is less than 20 ms for each transaction for very basic monitoring. The latency increases proportionally as you add other features such as security authentication and authorization. Due to the considerable flexibility and separation of concerns they provide, Web service intermediaries have gained considerable mainstream acceptance as a preferred means for managing Web services.

Supported Handlers for Agents and Intermediaries

As previously mentioned, several pre-defined Handlers are provided for both Agents and Intermediaries, including:

- **Performance and Fault Monitoring:** Capture response times and faults for messages to generate metrics for SLO monitoring.
- **Logging:** Log detailed diagnostic messages to local log files. Some other vendors describe Logging as the ability to log information to a central database. We describe this feature of capturing messages to a database as Auditing and refer to Logging as the capability used for SOA component troubleshooting rather than Web services application troubleshooting.
- **Auditing:** Collects context and payload of SOAP messages sent to Web services. The information collected is sent to the SOA Manager and is stored in a database.
- **Business Content Alerting:** Lets you “watch” SOAP message payload (content) and raise alerts when certain conditions are met. For example, a business manager might want to be informed if someone with a credit rating of less than 5 applies for a loan. This is information that could be detected using Business Content Alerting.

Additional pre-defined security handlers are provided for the Intermediary, including:

- **Inbound Message Security:** Provides authorization using the principal and credentials associated with an operation. The authorization is done using a configured security provider such as Select Access.
- **Outbound Message Security:** Provides support for WS-Security on outbound messages (i.e., from the Intermediary to a Web Services container). This includes user name/password, signing, and encryption.
- **Schema Validation:** Validate that SOAP requests conform to a Web service's WSDL or reject the message and return a SOAP fault. This stops malformed messages from reaching the Web service implementation. This handler is not provided for Agents.
- **Security Auditing:** Used to collect security trace information (used for non-repudiation) and to send the payload to a security provider. For example, when using Select Access to control authorization, the traces can be viewed using the Select Access Audit Report Viewer.
- **Transformation:** Used to transform inbound message or outbound message according to a specified xsl template.
- **Routing:** Used to route requests to endpoints based on the content/context of the message.

No security Handlers are provided for the Agents because security is handled by modern Web Service containers for J2EE and .Net. Additionally most Identity and Access Management solutions offer native integrations into these platforms.

In addition to the pre-defined Handlers described above, you can add your own custom handlers to a handler chain. Custom Handlers are created using the SOA Manager APIs. For more details, refer to the *SOA Manager Integration Guide*.

Management Server

The previous sections described how the WSM Components interposed manageability in order to gather management data about Web services and SOA resources. The final step of service management is to use a management server to collect the management data and make that data available in a meaningful context. There are three options that can be used to accomplish this:

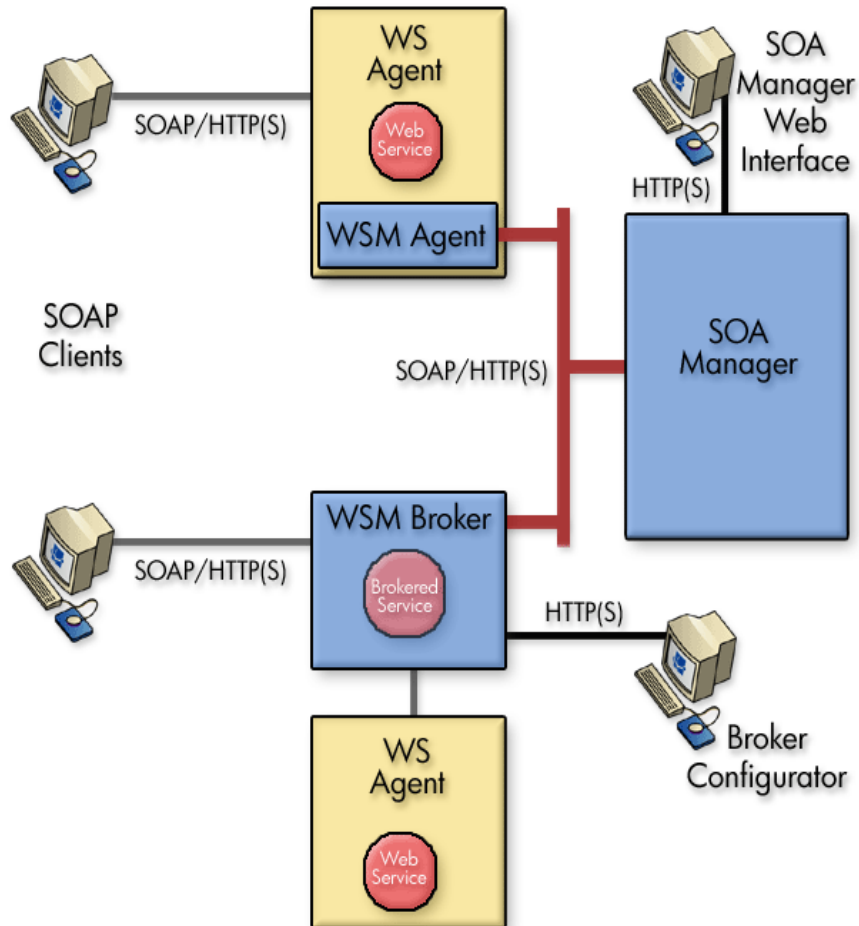
- Using the SOA Manager web interface (default implementation that is provided out-of-box)
- Using other enterprise management clients (requires integration)
- Creating your own management clients (requires integration)

This section discusses the first option.

SOA Manager

The SOA Manager is a central management server that runs in a single Java Virtual Machine (JVM) process. A typical SOA Manager installation includes a single SOA Manager that interacts with any number of WSM Agents and/or WSM Intermediaries that are deployed in an SOA environment. As previously discussed, the WSM components expose Web services and SOA resource management data as management Web services. This means the SOA Manager itself is a SOAP client that utilizes SOAP/HTTP(S) to invoke the management Web services.

The figure below shows the interaction and relationship between a WSM Agent, WSM Intermediary, and the SOA Manager.



SOA Manager Web Interface

The SOA Manager includes a management/administrative console called the HP SOA Manager Web Interface. This web interface, among other things, is used to construct a view of the management data that is being collected by the SOA Manager. The HP SOA Manager Web Interface is also used to interact with and configure many of the service management features that are discussed in the “Service Management Features” section above.

- ▶ The SOA Manager and the HP SOA Manager web interface are also used to model Web services and SOA Resources into the service models. The following chapter discusses the SOA Manager’s Service Model capabilities.

The HP SOA Manager Web Interface is automatically installed as part of the SOA Manager installation. Once the SOA Manager is started, the HP SOA Manager Web Interface can be accessed from a Web Browser. As the default, the HP SOA Manager Web Interface can be accessed on port 5002 of the SOA Manager’s host computer.

Introduction

This chapter covers general information about the *User Guide* as well as technical overview information about the HP Service Oriented Architecture (SOA) Manager software. The technical overview information is fundamental to understanding and using the software. Read this information carefully before setting up the software in a test or enterprise environment.

Document Overview

The *HP SOA Manager User Guide* provides instructions for setting up, configuring, and using the SOA Manager software. The chapters in the book are organized by feature with more generic features explained first. It is suggested that new users proceed through the chapters sequentially since the features become more complex in each successive chapter.

The guide does not detail the SOA Manager's implementation of standard WS-based management protocols. These standards define how to manage resources (including Web services) using Web services technology.

Audience

The *User Guide* is primarily intended for enterprise architects and operation managers who are responsible for integrating and enabling Web services-based applications and management solutions in their IT environments. In addition, the guide is intended for business managers who are responsible for monitoring the health of Web services-based business applications. In addition, customers, partners, and industry analysts can read this chapter to get a technical overview of the SOA Manager software.

Prerequisites

To use this guide, users must have a fundamental knowledge of Web services principles, Java platform technologies, and software management principles. In addition, users must have basic experience deploying Web services-based applications.

Component Overview

The SOA Manager software allows an organization to dynamically manage the SOA resources that are deployed in an enterprise. The software is composed of a set of core components that are distributed in an IT environment. The core components that are discussed in this section include:

- **HP SOA Manager** – A central management server that works to collect management data and present the data in a meaningful context. The data is collected from any number of WSM Agents and WSM Intermediaries.
- **Policy Enforcement Agent Services** – Web Services Management (WSM) agents that provide management capabilities for Web services containers and their hosted Web services. The agents are integrated with the Web services container.
- **Policy Enforcement Intermediary Services** – A proxy server process that provides management capabilities for Web services containers and their hosted Web services. The intermediary is a separate process from the policy enforcement agents group.

HP SOA Manager

HP SOA Manager is a central management server that runs in a single Java Virtual Machine (JVM) process and includes a J2EE-based administrative console called the HP SOA Manager web interface. A typical SOA Manager installation includes a single HP SOA Manager server that interacts with any number of policy enforcement agents and policy enforcement intermediaries. The communication between the software components is SOAP over HTTP(S). This communication channel is often referred to as the management channel.

The server's primary functions are outlined below and discussed throughout this guide.

Service Model Definition

HP SOA Manager maintains service model definitions. A service model definition allows a user to define an end-to-end model that encompasses business services and their relationships to actual IT resources. The service model is a distinguishing characteristic from other management solutions and is essential to understanding how SOA resources are managed using the SOA Manager software. The service model includes software assets (such as Web services) and also includes virtualized IT infrastructure components. A service model is defined using the HP SOA Manager web interface and contains the following elements:

- **Policy Enforcement Points** – The channel using which SOA Manager enforces policies to manage web services. Policy enforcement points can be either an agent or an intermediary service.

- **Business Service** – A business service represents some business application that is offered by a business manager to either internal or external customers. Currently, SOA Manager only implements one type of business service, which is a Web service. Multiple Web services can be part of a single business service. A business service definition is created and then bound to the Web services that are required to deliver the business application. The model definition also allows relationships between business services. Such relationships can be used to provide root cause analysis and impact analysis.

The service model allows business managers, developers, IT operations and IT support personnel to work together when defining, creating, deploying, and managing services. Ultimately, the service model definition links business operations with IT operations and automates many of the tasks that are required to deliver service-based applications.

Performance and Availability Monitoring

SOA Manager aggregates management data that is collected by agents and intermediary services that are registered as policy enforcement points with SOA Manager. The SOA Manager web interface is then used to view the collected data. Two types of management data are collected: performance metrics and availability status.

Performance metrics are collected over time and show the overall health and performance of the resources that are contained in a business service. Different metrics are collected depending on the type of resource being managed.

Availability status shows whether or not the Web services that are part of the overall service model are operational. The availability of a Web service typically provides the first indication that a business service is not operational.

SLO Management

SOA Manager allows Service Level Objectives (SLO) to be defined for performance metrics. The SLO definition includes acceptable warning-level and breach-level limits. When an SLO limit is reached, an alert is raised. The web interface is used to define SLO threshold values and also to view any SLO alerts. SLOs allow operators to react and adapt to degrading services.

Alerting

SOA Manager provides alerting capabilities for management events. There are generally three types of alerts: SLO alerts, availability alerts, and business content alerts. SLO alerts are raised whenever an SLO limit is reached. Availability alerts are raised whenever the status of a Web service changes. Lastly, business content alerts are raised when specific content is contained in the SOAP request or response message for a Web service.

Alert messages include information such as the origin, severity, and description of the alert. Alerts are viewed using the SOA Manager web interface and can be sent to email recipients, an SNMP TRAP, and to a log file. Alerts are persisted to SOA Manager's database. The web interface is also used to create alert recipients and manage alerts (that is, acknowledge alerts).

Auditing

SOA Manager aggregates Web service message trace information that is collected by policy enforcement agents and policy enforcement intermediaries. Trace information provides historical data related to a Web service's 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 SOA Manager's database at regular intervals. The web interface is used to view the trace information and generate reports.

Deployment

SOA Manager can deploy a deployment unit to remote policy enforcement agent services or intermediary services that are registered with SOA Manager as a policy enforcement point. In the current release, deployment units represent Web services that are packaged in an agent or an intermediary in a deployment format (for example, `.ear`, `.jar`, `.msi`). Any Web services that are contained in the deployment unit are automatically discovered at the time of deployment. Remote deployment saves time and allows business services to scale as service demands increase.

UDDI Publishing

HP SOA Manager can be configured to use a UDDI registry. The HP SOA Manager web interface is used to publish management Web services for Web services and business services to the registry. Using a registry allows the assets that are defined in the service model to be reused by other applications.

Policy Enforcement Agents and Policy Enforcement Intermediary Components

The policy enforcement agents and policy enforcement intermediary are responsible for providing manageability for Web services. The agents and intermediary are also responsible for exposing management data to SOA Manager using standard WS-based management Web services.

The SOA Manager software provides a policy enforcement agent (called the J2EE Agent) for the WebLogic server and a separate policy enforcement agent (called the .Net Agent) for the .NET server. The agents are integrated with their respective services and run in the same process as the server. Configuration files are used to configure the agents.

The policy enforcement intermediary is a separate server process that runs in a single JVM process and includes an administrative console called the Broker Configurator. The WSM Intermediary is a WS intermediary and can be used to add manageability to any WS container. The Intermediary provides manageability by using an intermediary service (a proxy to the service being managed). An intermediary service must be created for each Web service that you want to manage. Intermediary services are created using the Broker Configurator.

Agent Handlers

The J2EE Agent and Intermediary collect management data through the use of agent handlers that are arranged in a handler chain. The agents are preconfigured to use a set of handlers that include (but are not limited to) a monitoring handler, business metric handler, and auditing handler. Configuring the handlers is done differently for the J2EE Agent and the Intermediary.

Each SOAP message passes through the handler chain during a service request and response. Figure 3-1 is a common view of the handler chain. For detailed information about installing and configuring the J2EE Agent and Intermediary, see the *WSM J2EE Agent Administrator Guide* and the *HP SOA Manager User Guide* respectively. These guides are located in the /Documentation directory of the distribution.

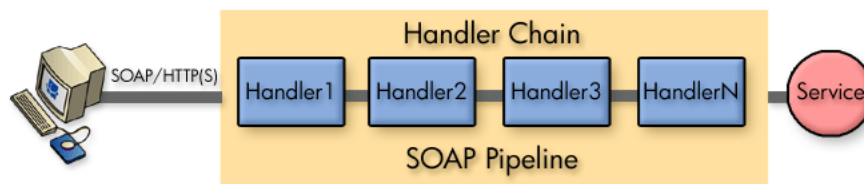


Figure 3-1: Management Handler Chain

The .NET Agent uses a Dynamic-Link Library (DLL) that monitors SOAP requests and responses. The DLL uses SOAP Extensions to gather management information for Web services. Figure 3-2 shows a common view of .NET management. For detailed information about installing and configuring the .NET Agent, see the *WSM .NET Agent Administrator Guide* located in the /Documentation directory of the distribution.

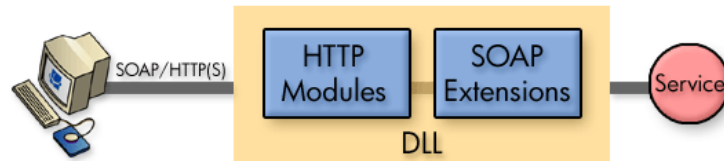


Figure 3-2: .NET Management

WS-Based Management Enablement

The WS-based management enablement layer is deployed into the Web service container or intermediary as a Web services application. The application is used by the SOA Manager server to get management data and also converts WS management interactions into interactions with the native management platforms (that is, JMX invocations for the J2EE platform, and WMI invocations for .NET platform.)



Second, the SOA Manager 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.

In both cases, identity management is handled by a tight integration with HP OpenView Select Access. This integration can also be used to provide authorized users access to the HP SOA Manager web interface and Broker Configurator.



This guide only covers management channel security, Select Access integration, and securing the HP SOA Manager web interface and Broker Configurator console using Select Access. For instructions on securing the application channel when using an Intermediary, refer to the *HP SOA Manager User Guide*.

Integration Points

The SOA Manager software provides many integration points that allow custom integrations with existing software assets in an IT environment. Integrations with the SOA Manager 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 Manager Integrator Guide*, located in the /Documentation directory of the distribution.

SOA Manager Roles

As described previously, SOA Manager's service model facilitates better alignment among three functional groups within an enterprise: line of business or business teams, application development teams, and application and operations support. This alignment is enabled by capturing the concept of a business service in the model and providing the three groups with interfaces to interact with the model.

Line of Business Perspective

A line of business manager (LOBM) drives the creation and functional definition of any business service. A business service represents an IT implementation of a business product offered by the LOBM to business consumers such as customers, partners and suppliers. Subsequently, the LOBM is also motivated to define appropriate Service Level Objectives (SLO) for the business service and ensure operational compliance of the business service with the desired SLO.

For example, a LOBM decides that they need to offer an Order Status Query service to customers. The service needs to be available to consumers between certain business hours and return responses within some defined response time.

Development Team Perspective

Once the LOBM defines the business service, an architect identifies how the architecture of the business service is to be broken up into different and distributed types of elements in the IT environment that execute in some coordinated manner. These different types of elements typically have different groups responsible for deploying and supporting them. The infrastructure for supporting these different IT element types is represented by a Policy enforcement point. An architect creates a model of the business service which also includes Policy enforcement points.

One such Policy enforcement point is the creation, deployment, and ongoing support of a Web service implementation. The development team is responsible for implementing the Web service. Once they create the implementation, they capture the deployment unit for that implementation as an asset of the Policy enforcement point representing the Web service implementation.

Another example of a Policy enforcement point is the configuration and support of an entry within the WSM Intermediary. A Web Services Management team within IT may be responsible for maintenance of such an Intermediary. They may create the configuration of this asset and save the configuration as an asset of the Policy enforcement point representing Web service intermediary.

IT Operations and Support Perspective

IT operations teams must deploy the implementation assets captured in the Policy enforcement point model by the architect onto appropriate IT resources to create running instances of Policy enforcement points. Some coordination is required to configure the connections between the Policy enforcement points. For example, someone must keep the Web Services Intermediary up-to-date to forward messages to one or more deployed Web service implementation endpoints.

Once the business service is made operational by connecting together the Policy enforcement points, IT operations and support teams must monitor and maintain underlying resources for the implementations to keep these Policy enforcement points running smoothly and ensure that the operation of the business service complies with the desired SLO. Additionally, they need to rapidly respond to changing needs of the business service by making appropriate changes to the underlying Policy enforcement points.

Improving Business IT Alignment and Providing IT Automation

The functional groups can achieve better communication through the shared context provided by the business service. When the LOBM requires changes, they drive them through the business service. Thus the involved IT operations and support teams can quickly affect the underlying Policy enforcement points. Implementation changes directed at development teams can be prioritized by understanding the affected business services.

Repetitive tasks, such as deploying software and configuring connectivity between underlying Policy enforcement points, can be automated by leveraging the metadata captured in the service model.

Life Cycle Stages

Web services, like all managed resources, have a life cycle. The SOA Manager software provides a clear life cycle definition for Web services. This definition provides an efficient and calculated method for managing Web services. The life cycle is composed of two areas:

- **A Permanent Management Model:** The structure, relationships, policies, and assets are captured in the service model definition and are applied at runtime for management functions.
- **Transient Managed Resources:** Resources such as Web service containers and Web services can be created, destroyed, and relocated anytime as required.

Figure 3-3 shows the stages of the life cycle and the relationship between the permanent management model and transient managed resources.

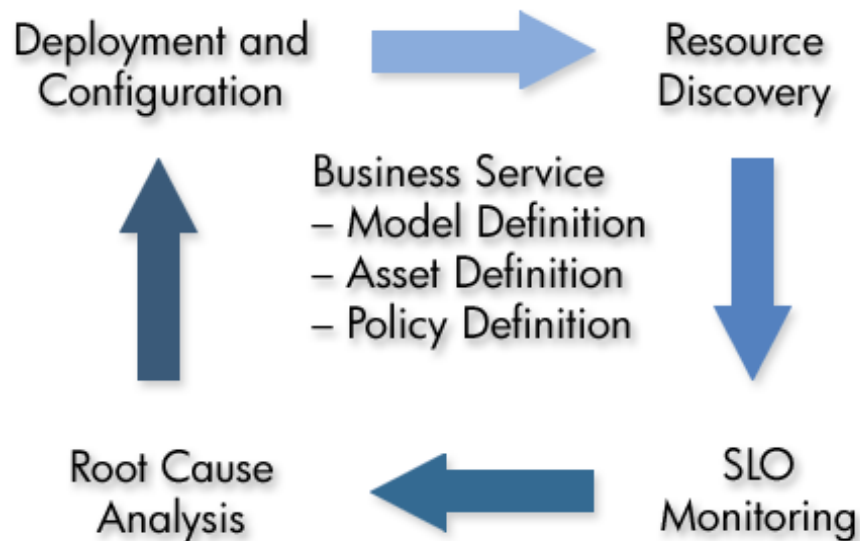


Figure 3-3: Life Cycle Stages

Model Definition

The structure, relationships, policies, and assets described in the service model are captured using the HP SOA Manager web interface. The HP SOA Manager web interface is not only used to create business service models, it is also used to view the business services, including their associated model. The underlying service model is architected to be interchangeable so that it can be created and manipulated by other tools or products (that is, other Service Management products from HP).

Resource Discovery

At runtime, all managed Web services and Web services containers in the environment are registered using the HP SOA Manager web interface. Once resources are registered with the SOA Manager server, various Policy enforcement points then identify resources contained in them by using identity matching patterns. This allows the Policy enforcement points map to automatically reflect deployed resources in the Policy enforcement points.

SLO Monitoring

Once Policy enforcement points identify underlying resources, an SLO monitoring engine uses SLO values to monitor the performance of the underlying resources. SLO warning and breaches are detected and such events can be mapped to different types of responses. The most common response today is to send an email to the configured recipient, but this mechanism is flexible and can be mapped to any required automatic execution.

Problem Resolution

Once SLO warning and breach alerts are raised, the HP SOA Manager web interface can be used to navigate the virtual service and resource relationships to troubleshoot and isolate problems using root cause analysis or impact analysis.

Deployment and Configuration

Deployment assets captured in Policy enforcement points can be deployed against target Web services containers. A corresponding undeploy mechanism is provided as well.

WSM Deployment Scenarios

This section describes two basic deployment scenarios when setting up SOA Manager's software components. The first scenario is a intermediary-based scenario and the second is WSM agent-based scenarios. The deployment scenario you select is ultimately based on your business requirements and your services environment. Because the enterprise is increasingly becoming heterogeneous, it is likely that your environment will include a mix of the deployment scenarios discussed in this section.



The two scenarios discussed in this section are not meant to be inclusive of all possible deployment scenarios and are only meant to establish a foundation for understanding how SOA Manager's software components are used together.

Intermediary-Based Scenario

An Intermediary-based scenario uses one or more WSM Intermediaries to collect management data for Web services that are deployed in a WS Container. The intermediary acts as a proxy to the container and does not need to be co-located with the container. The intermediary provides manageability by using an intermediary service (a proxy to the service being managed). An Intermediary service must be created for each Web service that you want to manage. For more information on installing and configuring the WSM Intermediary, see the *HP SOA Manager User Guide*.

Runtime requests are sent through an intermediary service and dispatched to the actual requested service implementation running in a WS Container. Management data is collected by handlers during the requests and responses that are sent through the intermediary service.

A single instance of the WSM Intermediary can manage multiple intermediary services. In addition, multiple Intermediaries can be used on a single host or distributed across hosts. In scenarios where a single service is replicated across multiple machines, management data and metrics are aggregated. The intermediary can also be used for services that are being managed by WSM Agents to leverage the security capabilities of the intermediary.

WSM Agent-Based Scenario

A WSM Agent-based scenario uses agents that are integrated into a WS Container. The WSM Agents run in the same process as the WS Container. There is an agent for the WebLogic Server WS Container and the Microsoft .NET WS Container.

Any number of WSM Agents can be used in a production environment. In scenarios where a single service is replicated across multiple machines, management data and metrics are aggregated.

Intermediary vs. Agent Deployment

The following table shows a quick overview of when to consider using the different WSM deployment methods.

Table 3-1: Deployment Method

Intermediary Web services	Agents
You do not have control of the deployment environment or the Web service itself (for example, if you want to monitor an external Web service that is part of a composite application).	You control the deployment environment and are planning to deploy to either a WLS or IIS container (for example, a Web service that is developed internally). You must keep your platform version in synch with SOA Manager Agent releases.

Intermediary Web services	Agents
You have multiple instances of your Web service running on different containers and you want the intermediary to determine which end point to route to at run-time (for example, load balancing and fail-over routing).	You want to monitor the health of the WLS or IIS container as well as the Web service itself.
You want to secure an individual Web service using an Authentication, Authorization, Audit (AAA) product like HP Select Access and the container it is running in does not provide native security.	You are already securing the WLS or IIS container the Web service is running in using an Authentication, Authorization, Audit (AAA) product like HP OpenView Select Access.
	You prefer not to separately deploy and manage extra servers (ntermediary) for managing Web services and are concerned about the latency imposed by using intermediary Web services. Agents impose less latency than Intermediaries because there is no extra process hop.

Getting Started

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

- Finance Example Overview
- Starting and Stopping the SOA Manager
- Starting HP SOA Manager
- Configuring HTTP Settings
- Configuring an Oracle Database
- Configuring a UDDI Registry
- 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 Manager 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 Manager 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 HP SOA Manager

A script for both Windows and UNIX is provided to start HP SOA Manager. 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 Manager installation. If you accepted the default program group during installation, you can start the SOA Manager server by clicking **Start | Program Files | SOA Manager 2.5 | Network Services**.




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

To start SOA Manager:

- 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 Manager starts.

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

Installing the HP SOA Manager as a Windows Service

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


To install HP SOA Manager 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 "HP SOA Manager v2.5 networkservices" installed.
```

 The script configures SOA Manager 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 HP SOA Manager

HP SOA Manager 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 Manager 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 Manager 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 HP SOA Manager process and issue the `kill` command to stop the process. For example:

```
kill <process number>
```

Starting the HP SOA Manager Web Interface

Hp SOA Manager is administered through the HP SOA Manager web interface. The web 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 Manager as described previously.
- 2 Open a browser.
- 3 Enter the following URL and substitute `<host>` with the host name where the SOA Manager 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 Manager 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 Manager software.



SOA Manager also integrates with Select Access, which can be used to secure access to the web interface. See Chapter 8 "Integrating with Select Access" for more information. Select Access is the preferred method for securing access to the web interface in production installations.

To add access rights for a user:

- 6 Stop SOA Manager if it is currently started.
- 7 Use a text editor to open `<install_dir>\conf\networkservices\mipServer.xml`.
- 8 Comment out the following entry:


```
<entry name="com.hp.mip.security.provider.console">default</entry>
```
- 9 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>
```
- 10 Save and close the file.
- 11 Restart HP SOA Manager.

Configuring HTTP Settings

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

Configuring the HTTP Server Port Number

HP SOA Manager contains a Java HTTP Server that listens for HTTP messages and is used by the HP SOA Manager 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 Manager will not start.

To configure the port number:

- 1 Stop SOA Manager 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 Manager.

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 Manager 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 Manager.

Configuring the Refresh Setting

HP SOA Manager'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. This feature can also be enabled/disabled by clicking on the refresh image.

To configure the refresh setting:

- 1 Log in to the HP SOA Manager 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 the Threshold for Web Service Summary

HP SOA Manager Web interface Dashboard contains a Web Service Summary (for last 24 Hours) section. This section displays the status of all the web services graphically in the past 24 hours based on the following performance metrics. The feature can be configured to change the color of the circular display based on the user defined threshold values.

To configure the Threshold setting:

- 1 Log in to HP SOA Manager web interface as an administrator.
- 2 Click **Change Settings** under **Actions** drop-down menu. The Settings screen opens.

- 3 From the **Threshold Settings** tab, enter the **Critical Threshold** and **Warning Threshold** values for all the four metrics.
- 4 Click **Save**.

Configuring an Oracle 10g Database

SOA Manager persists service messages, service trace messages, and alerts to a database. The SOA Manager 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 Manager 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 Manager to use the Oracle 10g database:

- 1 Stop SOA Manager 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 Manager.

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 Manager 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 (`timez1rg.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/timez1rg.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/timez1rg.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 Manager 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 Manager Database

The SOA Manager's distribution includes a script to migrate a SOA Manager 2.1 database to a 2.50 database. The script preserves current 2.0 data in the database.

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

Configure a UDDI Registry

A UDDI Registry must be configured with HP SOA Manager before you can use the HP SOA Manager web interface to publish SOA Manager's assets to the registry.



If your UDDI Registry is accessed using SSL, you must configure the SOA Manager server's SSL settings. See Chapter 9, "Using SSL for the Management Channel."

To register a UDDI Registry:

- 1 From the **Actions** drop-down menu, click **Change Settings**. The Settings screen opens.
- 2 Click the **UDDI Settings** tab.
- 3 Enter the UDDI Registry settings:
 - **Username for UDDI Publishing:** A user's name used to access the registry.
 - **Password for UDDI Publishing:** The user's password.
 - **UDDI Registry Inquiry URL:** The URL used to connect to the registry and make inquiries. For example, `http://<host>:<port>/uddi/inquiry`.
 - **UDDI Registry Publish URL:** The URL used to connect and publish to the registry. For example, `http://<host>:<port>/uddi/publishing`.
 - **Maximum Rows Returned for Provider Query:** Constrains the number of providers that are returned when the UDDI Registry is queried.
- 4 Click **Save**.
- 5 Restart SOA Manager.



UDDI Settings can also be manually entered in `<install_dir>\conf\networkservices\mipServer.xml`.

Adding User Roles

SOA Manager let you to assign roles for users. You must have administrative privileges to create new users. SOA Manager 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 Manager performs authentication or authorization using the security provider configured. A default security provider is bundled along with SOA Manager. 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 Manager web interface.

You can also configure SOA Manager to use a different provider. To use Select Access as the security provider, you must perform the configuration steps listed in *Configuring SOA Manager and Select Access for Security*.

When using a provider other than the default user authentication is accomplished by using users configured in the security provider for example, Select Access (SA).

You must configure users configured in other third party providers (such as Select Access), in SOA Manager for assigning services and resources to the user.

Adding User Roles

To add users, follow these steps:

- 1 Log in to the HP SOA Manager web interface as an administrator.
- 2 Click **StakeHolder Roles** in the **View** drop-down list present on the left pane of HP SOA Manager 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

NOTE: 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.

- 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 HP SOA Manager 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.

NOTE: 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 User

To remove existing user roles, follow these steps:

- 1 Log in to the HP SOA Manager 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 **Remove** to confirm the removal of the selected user role.

Using XPL Logging

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

Installing XPL Logging

During the SOA Manager 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 OpenView on Windows and /opt/OV on UNIX. The default value for the data directory is C:\Program Files\HP OpenView\data on Windows and /var/opt/OV on UNIX. The HP SOA Manager log files are created in the log subdirectory of the data directory. If you do not run SOA Manager 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 Manager CD.

Configuring XPL

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

networkservices[unique].sequence.locale

For example:

networkservices0.0.en_US

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

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

SOA Manager 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 HP SOA Manager 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 Manager log levels using the SOA Manager 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 Manager Web Interface

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

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 SOAM 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 Manager. 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 Manager 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 Manager, the logger is created on SOA Manager 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 Manager by editing the `logging.properties` file in the `JRE /lib` directory. You must restart SOA Manager 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 Manager to `FINE`.

Using the XPL Properties File

You can change the log level for SOA Manager by editing the `xpllogging.properties` in the `<install_dir>/conf/networkservices` directory. You must restart SOA Manager 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 Manager to `FINE`.

Viewing Logs

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

Using XPL Tracing

SOA Manager 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 Manager 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 OpenView\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 Manager 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 Manager 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
```


Managing Resources Using PEP Groups

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

Overview

PEP is an abstract concept that can mean different things. Within the scope of the SOA Manager'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. For example, a PEP could be used to organize all the managed policy enforcement agent groups for a specific application.

The PEP groups that are supported include the following:

- **Policy Enforcement Agent Groups:** This type of PEP captures the management of policy enforcement agents and their hosted Web services. The policy enforcement agent group supports the deployment, discovery, and SLO monitoring of a Web service implementation deployed to multiple Web services containers. A policy enforcement agent group must expose its manageability using a J2EE or .NET agent.
- **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, discovery, and SLO monitoring 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.

Create a Policy Enforcement Agent Group

To create a Policy enforcement agent group, follow these steps:

- 1 From the **View** drop-down menu, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.
- 2 From the Policy enforcement agent group section, click **Add**. The Add Policy enforcement agent 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 agent that is contained in the policy enforcement agent group 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 agent groups or refer to the “Registering Resource” 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 Agent and Intermediary Resources

When a managed policy enforcement agent or 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 agent or intermediary, they are automatically added and removed from the PEP.

To register an agent or intermediary resource, follow these steps:

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

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



Some SOA Manager features may not work as expected when using agent or intermediary versions that are different than the SOA Manager version. It is recommended that the SOA Manager version and the policy enforcement agents and 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 agent or 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. For WLS, you must supply the Standalone Server name or Cluster name where the WSM J2EE Agent is installed.



The policy enforcement agent or intermediaries publish their management interface (WSDL) to a URL. The SOA Manager 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 Agent or Intermediary screen reopens and the Contained Web Services section lists the Web services that are discovered in the managed Policy enforcement agent or intermediary.
- 9 Click **Add**. The Policy Enforcement Agent or 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 agent/intermediary from within the SOA Manager 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 agent/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 agent/intermediary. Different log levels and loggers provide varying levels of log details that can help identify process events.

A policy enforcement agent/intermediary contains a predefined set of loggers. For .NET, two categories (Catalog and libraries) are used. For policy enforcement agents and intermediaries, loggers are defined in each agent's XPL configuration file. In addition, any custom loggers that are implemented in a policy enforcement agent/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 agent or 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 agent/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 agent/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 agent/intermediary is not available. Enabling this feature will quickly notify individuals when a policy enforcement agent/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 agent/intermediary for which you enabled availability alerts. Refresh the screen. An alert message opens in the Alerts section and indicates that the policy enforcement agent/intermediary is unavailable. Restart the agent/intermediary. When the agent/intermediary becomes available, an alert message opens in the Alerts section and indicates that the agent/intermediary is available.

Deleting a Resource

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

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

To deregister a policy enforcement agent or 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 or Agents

The View PEP screen provides details about a PEP. The screen is a convenient way to view managed policy enforcement agents/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 agent/intermediary PEP

Viewing a Policy Enforcement Agent/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 agent/intermediary resources that are contained in the PEP. The Contained Policy Enforcement Intermediary Instances section also lists the current version of the policy enforcement agent/intermediary resource as well as the resource's management WSDL.



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

Adding Resources

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

To add a policy enforcement agent/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 or Agent Policy Enforcement Points screen opens.
- 4 From the list of agents or intermediaries, click the **Contains** check box for each resource you want to add to this PEP. A check mark indicates that the policy enforcement agent/intermediary is selected.
- 5 Click **Save**.

Removing Resources

To remove a managed policy enforcement agent/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 or Agent Policy Enforcement Point screen opens.
- 4 From the list of policy enforcement agents or 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 agent/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 (agent or intermediary). The View PEP screen provides the details of the alert and also indicates which managed policy enforcement agent/intermediary caused the alert.



It is good practice to enable availability notifications for a managed policy enforcement agent/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 or Agent 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 Agent/Intermediary PEP

You can delete a PEP at any time. When you delete a PEP, its alerts are removed. However, any managed policy enforcement agents/intermediaries that are contained in the PEP are not removed from SOA Manager 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.

HP SOA Manager

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

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

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 Manager 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 are WS Policy documents.

Bundled Policies

After installation, SOA Manager 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 HP SOA Manager web 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:

- 3 Log in to HP SOA Manager web interface as an administrator.
- 4 From the **Actions** panel on the left pane, click **Add Policies**. The Technical Policies page opens.
- 5 Click **Add**. The Add New Technical Policy page opens.
- 6 Type the Name, Description, and Version in the respective boxes on the Add New Technical Policies page.
NOTE: Make sure that you type meaningful names and description for the policies. This will help in easy identification of the policies.
- 7 Based on the type of policy that you want to create, perform the steps listed in the following sections to create the specific policy.
 - Create an audit policy
 - Create an event 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
- 8 Click **Save** to create the new Technical Policy and save the policy in the service model. SOA Manager saves the policy as Web service (WS) policy document.

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.

The screenshot shows a configuration window for an audit policy. At the top, there is a 'Type:' label and a dropdown menu currently showing 'Audit Policy'. Below this, there are two radio buttons: 'Auditing' (which is selected) and 'Security Auditing'. Underneath, there is a checkbox labeled 'Include Detailed Traces' which is currently unchecked. Below the checkbox is a 'Payload Option:' label and a dropdown menu showing 'NONE'. At the bottom of the configuration area, there are two radio buttons: 'Payload Log All' (selected) and 'Payload Log on Failure' (unselected). In the bottom right corner of the window, there are two buttons: 'Save' and 'Cancel'.

- 2 Select **Auditing** if you want audit traces to be recorded in SOA Manager or select **Security Auditing** if you want to log audit traces to security validator.
- 3 Select **Include Detailed Traces** to enable all tracing details.
- 4 Select **REQUEST**, **RESPONSE**, or **REQUEST-RESPONSE** from the **Payload Option:** drop-down list. This option signifies whether the payload for the SOAP 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.
- 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 Perform *step 6* in the *Creating a New Technical Policy* section.

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:

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

- 2 Type the name of the event in the **Event Name** box.
- 3 Type the name of the operation that contains the business service you want to monitor in the **Select the operation to alert from Operation** box.
- 4 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.
- 5 Type the XPath expression in the **XPathExpression(ExpressionHelp)** 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.
- 6 Type a message that you want the alert to display in the **Message:(MessageHelp)** box.
- 7 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.
- 8 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.

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

Creating a Schema Validate Policy

You can use the schema validate policy to validate a schema of the incoming SOAP request. To create a schema validate policy, perform *step 1* 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 Perform *step 6* in the *Creating a New Technical Policy* section.

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 *step 1* 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.

The screenshot shows the 'Add New Technical Policy' dialog box. The fields are as follows:

- Name:***: Empty text box.
- Description:***: Empty text box.
- Version:***: Text box containing '1.0'.
- Type:***: Drop-down menu showing 'JMS Mediation Policy'.
- Vendor URL**: Text box containing 'http://bea.com'. To its right is an 'or' label and a drop-down menu also showing 'http://bea.com'.
- JNDI Provider Url**: Text box containing 't3://{hostname}:{port}'.
- JNDI Initial ContextFactory**: Text box containing 'weblogic.jndi.WLInitialContextFactory'.
- Destination Style**: Drop-down menu showing 'queue'.
- JNDI Connection Factory Name**: Empty text box.
- JNDI Destination Name**: Empty text box.

'Save' and 'Cancel' buttons are located at the bottom right of the dialog.

- 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 either **queue** or **topic** from the **Destination Style** drop-down list. This lets you specify 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 which could be either Select Access or the default security provider in SOA Manager. 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 transport level security. 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 Select Access or in the default security provider for SOA Manager.
- 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 policies Technical 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 a Policy

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 Manager 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 HP SOA Manager web 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 a Policy

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 Manager. To import policies, you must perform the following steps:

- 1 Log into HP SOA Manager web 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 Manager web interface. To set up the auditing components you must perform the steps in the following section:

- Enable the Audit Handler
- Configure the Audit Publisher
- Configure the Database

Enable the Audit Handler

The policy enforcement agents and the intermediary contain a group of handlers that are responsible for gathering management data for a Web service. Enabling an agent's audit handler is achieved differently for the intermediary and policy enforcement agents.



The audit handler for the J2EE Agent and the .NET Agent is enabled by default.

Policy Enforcement Intermediary Group

You can enable an intermediary group audit handler using the Broker Configurator. The audit handler must be enabled for each intermediary group that you create. You can enable auditing when you first create the intermediary group or you can edit an intermediary group at any time to enable auditing. The following procedure enables the audit handler for an intermediary group which has already been created. See the *Broker Configurator Online Help* or for additional information on creating intermediary groups.

To enable the audit handler, follow these steps:

- 1 Log in to the Broker Configurator.
- 2 From the Action column, click **edit** for the intermediary service you want to configure. The Service Configuration screen opens.
- 3 From the Features section, click the **Auditing** check box. A check indicates that the auditing is selected and that traces will be sent to the audit database.
- 4 If you want the audit handler to capture profile data, click to select the **Include detailed traces** check box. This feature captures the outcome of a Web service invocation as it passes through each handler in the handler chain for an intermediary service.
- 5 If you want to also collect a message's SOAP payload, use the Payload Option drop-down list to select whether you want the payload to be collected for requests or responses, or both requests and responses.
- 6 Use the options provided to select whether you want the payload for all messages (successful and failed) or just for failed messages.
- 7 At the bottom of the screen, click **Save Changes**. The Brokered Services screen opens and the intermediary service is automatically deployed. The deployment is complete when the status changes to operational. You may need to refresh the screen to see the status change.

WSM J2EE Agent

The WSM J2EE Agent automatically adds an audit handler to every Web service it discovers. The handler is enabled in a Web service's `web-services.xml` file. See the *HP SOA Manager User Guide* for additional information on configuring handlers when using a J2EE Agent.

WSM .NET Agent

The .NET Agent's Audit SOAP extension is enabled for either specific Web services or for all Web services. The following instructions are applicable when enabling Auditing for a specific Web service. See the *HP SOA Manager User Guide* for additional information on configuring SOAP extensions.

To enable the audit SOAP extension for a specific Web service, follow these steps:

- 1 Use a text editor to open an application's `Web.config` file.
- 2 Edit the file by adding a `<services>` node within the `<configuration>` node.
- 3 For each Web service, create a `<service>` node within the `<services>` node and include a name attribute that contains the Web service name. For example:

```
<services>
  <service name="FinanceService.asmx">
```

- 4 Within the `<service>` node add an `<audit>` node and include a `payload-option` and `payload-filter` attribute. For example:

```

<services>
  <service name="FinanceService.asmx">
    <audit payload-option="REQUEST-RESPONSE"
          payload-filter="ALL" />
  </service>
</services>

```

- **payload-option:** Defines what payloads to audit. Valid entries are REQUEST, RESPONSE, REQUEST-RESPONSE, or NONE.
- **Payload-filter:** Defines when to capture the payload. Valid entries are ALL or ERROR. Setting this attribute to ALL captures payloads that are successful and payloads that encountered errors.

- 5 Repeat Steps 3 and 4 to enable auditing for additional Web service in the application.
- 6 Save and close the file.

Configure the Audit Publisher

The policy enforcement agents and intermediary contain an audit publisher that is responsible for sending trace information to the SOA Manager's audit service. Configuring an agent's audit publisher is achieved differently for the intermediary and policy enforcement agents.

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.

WSM J2EE Agent

In addition to the audit publisher properties for interval and trace, the J2EE Agent's publisher also controls whether or not a message's SOAP payload is sent. These settings are also discussed in this procedure.

To configure the audit publisher, follow these steps:

- 1 Stop the WebLogic server, where the J2EE Agent is running, if it is currently started.
- 2 Use a text editor to open `<j2ee_agent_install_dir>\config\agent.xml`.
- 3 Under the `<components>` node, find the `<component>` element with the `id` attribute `Auditing`.
- 4 Edit the `publishInterval` and `traceThreshold` parameters. The `publishInterval` value is in milliseconds and the `traceThreshold` value is the total number of trace messages:

```
<parameters>
  <parameter name="traceThreshold" value="200" />
  <parameter name="publishInterval" value="3600000" />
</parameters>
```

- 5 Within the same node, you can configure whether you want a message's SOAP payload to be sent with the trace information. For example:

```
<parameter name="payloadOption" value="request-response" />
<parameter name="payloadFilter" value="all" />
```

- `payloadOption` – Valid values are `none`, `request`, `reponse`, and `request-response`.
- `payloadFilter` – Valid values are `all`, and `failure`. The `all` value sends payload for both failed and successful messages.

- 6 Save and close the file.
- 7 Restart the WebLogic server for the changes to take effect.

WSM .NET Agent

The audit publisher for the .NET Agent is configured using a Web service that comes with the .NET Agent. The Web service is called `RemoteConfig.asmx` and is only accessible from the computer that is hosting the .NET agent.

To configure the audit publisher:

- 1 From the computer that is hosting the .NET Agent, open a browser.
- 2 Go to the following URL, follow these steps:

```
http://<DotNetAgentHost>/hpwsm/RemoteConfig.asmx
```

Replace `<DotNetAgentHost>` with the with the fully qualified DNS name of the computer.

- 3 From the `RemoteConfig` Web service, click **SetAuditBucketSize**. The `SetAuditBucketSize` operation screen opens.

- 4 Using the `bucketSize` text box, enter the total number of trace messages as an integer.
- 5 Click **Invoke** to set the bucket size. The `SetAuditBucketSize` operation is run and a new browser window opens with a blank screen.
- 6 Close the blank browser window.
- 7 From the browser window for the `SetAuditBucketSize` operation screen, click the browser's **Back** button. The RemoteConfig Web service opens.
- 8 From the RemoteConfig Web service, click **SetAuditInterval**. The `SetAuditInterval` operation screen opens.
- 9 Using the `intervalSeconds` text box, enter the trace interval in Milliseconds as an integer.
- 10 Click **Invoke**. The `SetAuditInterval` operation is run and a new browser window opens with a blank screen.
- 11 Close all browser windows.

Configure the Database

Message trace information is sent to the SOA Manager audit service and stored in a database. SOA Manager 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 Manager 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 Manager:

```
<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 Manager 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 Manager to use the Oracle 10g database, follow these steps:

- 1 Stop SOA Manager 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 Manager.

Viewing Audit Information

The SOA Manager web interface lets you query trace messages that are stored in the SOA Manager 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 agent/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.

Viewing Reports

The HP SOA Manager web interface lets you to query the SOA Manager 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 constrain the report by business service, or policy enforcement point, or existing service. 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.

Service Provisioning

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

- 1 Specify 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 Set SLO threshold values
- 6 Associate the Web service with a business service
- 7 Specify the provisioning option

Prerequisites

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

Specify Service Details and PEP Types

- 1 Log in to HP SOA Manager 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.

Provision Service

Provision Service

Tool to bring web services under management

Step 1 Of 7 : Specify Implementation Service Details and Policy Enforcement Point Types

New Service Type to Add: SOAP Service XML ServiceSpecify WSDL: Remote WSDL Local WSDLSpecify Remote WSDL URL:* Policy Enforcement Point Type: 

- 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 Specify the WSDL file for the Web service. If the WSDL file is present in the local computer, you can click **Browse** and locate a local WSDL file and specify it in the **Browse Local WSDL File** box. If the WSDL file is present in a remote computer, you can specify the WSDL URL of the remote computer in the **Specify Remote WSDL URL** box. Proceed to step 8 in this procedure to complete the steps in this section if you selected a SOAP service.
- 7 Specify the endpoint in the **Endpoint** box
- 8 Select the PEP from the **Policy Enforcement Point Type** drop-down list.
- 9 Click **Next**. This displays the Associate Technical Policies screen as shown in the following section.

Associate Technical Policies

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

Provision Service

Provision Service

Tool to bring web services under management

Step 2 Of 7 : Associate Technical Policies

Select one or more Technical Policies to Apply


DefaultAuditRequestsResponsesOnFailure
 DefaultMessageSecurityDigitalSignatureValidationInbound
 DefaultTransportSecurityInboundHTTPS509
 DefaultAuditAllRequests
 DefaultAuditResponsesOnFailure
 DefaultSchemaValidation
 DefaultAuditAllResponses
 DefaultTransportSecurityOutboundBasicAuth
 DefaultTransportSecurityInboundHTTPS
 DefaultMessageSecurityInboundDigitalSignatureEncryptio

Technical Policy Description

Selected Policies

Technical Policy Description

Finish Previous Next Cancel

- 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.
- 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.

Provision Service

Provision Service
Tool to bring web services under management

Step 3 Of 7 : Specify Service Details

HTTP Path:*

Service Name:*

Version:*

Finish **Previous** **Next** **Cancel**

- 2 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.

Provision Service

Provision Service
Tool to bring web services under management

Step 4 Of 7 : Endpoint Related Configuration For Load Balancing And Routing

Address:

Port Type:

Binding:

Encoding: Default UTF-8

Load Balancing Option:

Routing Classifier:

Finish **Previous** **Next** **Cancel**

- 2 Select **UTF-8** from the Encoding section if you want to enable UTF-8 encoding for the endpoint. The Default option is selected by default.
- 3 Specify the **Routing Classifier** in the corresponding box. This step is optional. Routing classifier signifies the end point to which a service request must be forwarded automatically. 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.
- 4 Click **Next**. This displays the Set SLO Thresholds screen as shown in the following section.

Set SLO Thresholds

- 1 Specify the Warning Threshold values and the Critical Threshold values for the following service metrics. See the *HP SOA Manager User Guide* for more information about the service metrics:
 - Availability(%)
 - Average Response Time
 - Failure Time
 - Maximum Response Time (in milliseconds)
 - Minimum Response Time (in milliseconds)
 - Security Violations
 - Success Count
 - Total Requests
 - Up Time(%)

Provision Service

Provision Service

Tool to bring web services under management

Step 5 Of 7 : Set SLO Thresholds

SLO Threshold

Service Metric	Warning Threshold	Critical Threshold
Availability(%):	< <input type="text"/>	< <input type="text"/>
Average Response Time:	> <input type="text"/>	> <input type="text"/>
Failure Count:	> <input type="text"/>	> <input type="text"/>
Max. Response Time(ms):	> <input type="text"/>	> <input type="text"/>
Min. Response Time(ms):	< <input type="text"/>	< <input type="text"/>
Security Violations:	> <input type="text"/>	> <input type="text"/>
Success Count:	< <input type="text"/>	< <input type="text"/>
Total Requests:	> <input type="text"/>	> <input type="text"/>
Up Time(%):	< <input type="text"/>	< <input type="text"/>

 Alert if unavailable

- 2 Select **Alert if Unavailable** option to generate an alert every time any of the specified service metrics is unavailable.
- 3 Click **Next** to go to the Associate Web Service with Business Service screen as shown in the following section.

Associate Web Service with Business Service

- 1 Select **New Business Service** or **Existing Business Service** from the Associate Web Service with Business Service screen.

Provision Service

Provision Service

Tool to bring web services under management

Step 6 Of 7 : Associate Web Service With Business Service

 New Business Service **Existing Business Service**
Business Service Name: *
Version: *
Description: *

- 2 If you selected **New Business Service**, you must 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.
- 3 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.
- 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 Manager provides a life cycle management feature for monitoring the status (life cycle) of a provisioned Web service. After logging in as an administrator in HP SOA Manager, you can click **Life Cycle Status** under the **View** drop-down menu on the left pane to view the Life Cycle Status screen as the following graphic shows.

The screenshot shows the HP SOA Manager interface. The top navigation bar includes the HP logo, 'HP SOA Manager', and user information 'User: admin Logout'. The left sidebar contains a 'Quick Links' section with 'Provision Service', 'Business Services', and 'Life Cycle Status'. Below this is a 'View' section with a dropdown menu currently set to 'Life Cycle Status'. The main content area is titled 'Life Cycle Status' and features a 3D visualization of SOA components: 'SOA Synchronet' (green), 'SOA Manager' (purple), and 'Policy Enforcement Point' (yellow). Below the visualization is a table with columns: 'Current State', 'Service Name & Description', 'Source', 'Since (Date & Time)', 'Last Action Status', and 'Actions'. Below the table is an 'Alerts From Recently Completed Tasks' section with a table containing one alert for 'helloServiceProxy'.

Current State	Service Name & Description	Source	Since (Date & Time)	Last Action Status	Actions

Service Name	Status Message/Alert	Issued Date/Time	Action
helloServiceProxy	Activate Successful on PEP wsl1	Wed Jul 11 18:51:40 IST 2007	Acknowledge

SOA Manager 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.
- **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.
- **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.
- **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 a saved session**- This action provisions a saved web service configuration to the PEP or activates a service that is already deployed on PEP.

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



SOA Manager displays a circular highlight mark on the SOA Manager figure as soon as you provision a Web service. This indicates that the Web service is present on SOA Manager and is yet to be deployed on the PEP. After the Web service is deployed to the PEP, a circular highlight mark appears on the PEP figure indicating the deployment of the provisioned Web service on the PEP.

After a successful deployment to the PEP, SOA Manager 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 Manager 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.

Alerts From Recently Completed Tasks:			
Service Name	Status Message/Alert	Issued Date/Time	Action
helloServiceProxy	Activate Successful on PEP wsl1	Wed Jul 11 18:51:40 IST 2007	Acknowledge

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

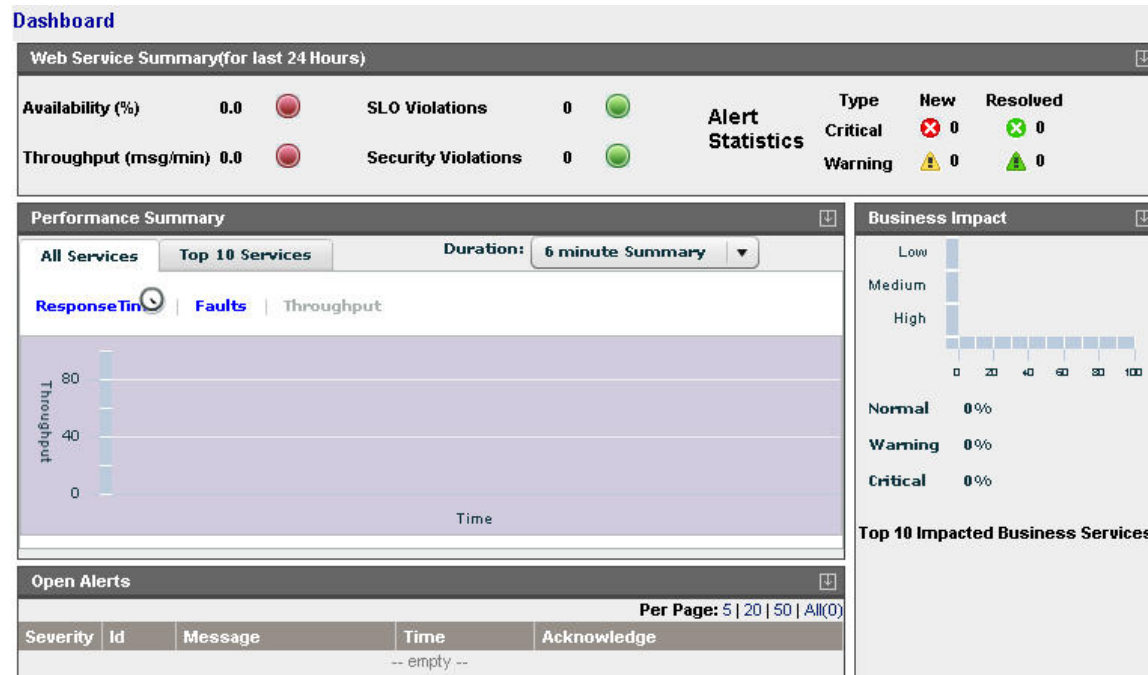
Dashboard

The HP SOA Manager 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 HP SOA Manager 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.
- SLO Violations - Signifies the SLO violations encountered by the Web services 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 6-1: Web service Performance Metrics

Metric	Value
Availability (%)	<p>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%).</p> <p>If no requests are sent, the field is left blank. If a policy enforcement intermediary/agent group goes down, the Uptime percentage gradually goes down to zero. The value gradually goes to zero because the SOA Manager server intermittently tries to contact a policy enforcement intermediary/agent group and assumes the policy enforcement intermediary/agent group will recover.</p>
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.
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 (%)	<p>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.</p> <p>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.</p>

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 Manager 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 Manager 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 Manager 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

Using Alert Notifications

This chapter describes how to configure and use the alert notification features in SOA Manager. The instructions include creating alert recipients and alert recipient categories. This chapter covers two alert types: SLO Alerts and Business Content Alerts, and 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.
- Monitoring – Alerts allow personnel to be notified of current performance metric values so they can react to unwanted situations before they escalate and degrade performance.
- Content Monitoring – Alerts can notify recipients when a specific value (for example, order > \$25,000.00) is found in a SOAP message.
- SLO Management – Alerts can safeguard against metric values that may be elevating past SLO levels.
- SNMP Management – Alerts in SOA Manager can be integrated with SNMP management solutions.

SLO Alerts

SLO alerts notify recipients when an SLO threshold value for a performance metric is exceeded. For more information on defining SLO threshold values, see Chapter 5 “Monitoring Performance and SLO”. There are four types of SLO alerts:

- SLO Normal (✅) – An alert that is generated when the SLO threshold value is within normal levels or when a threshold value has returned to an acceptable level. This is a low-level alert and typically no action is required.
- SLO Warning (⚠️) – An alert that is generated when the SLO Warning threshold value for a business metric is exceeded. This is a medium-level alert and should be used to indicate that a minor event has occurred that may impact an SLO.
- SLO Breach (🚨) – An alert that is generated when the SLO Breach threshold value for a business metric is exceeded. This is a high-level alert and should be used to indicate that a major or critical event has occurred and needs immediate attention.
- SLO Operational (❌) – An alert that is generated when a component in the service model is not operational or unavailable. Operational alerts can be enabled for Policy Enforcement Points, Business Services, Configurations, and Instances.

Assigning an SLO Alert to an Alert Category

When SLO alert threshold values are violated, an alert is generated and forwarded to all recipients in a recipient category that is assigned to an SLO alert type (warning or breach). See “Setting Up Alert Recipients” category below for creating alert recipients and recipient categories.

To assign an SLO Alert Type to an Alert Category, follow these steps:

- 1 Click **Business Services** from 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.
- 3 Click the host web service you want to view. The appropriate view screen opens.
- 4 Click the **Configuration** tab and click **Edit SLOs**. The Edit Configuration Performance screen displays.
- 5 In the Alert Recipients section, use the drop-down lists to select a recipient category for each SLO alert type.
- 6 Click **Save**.

Configuring the SLO Alert Polling Interval

The SLO Engine evaluates performance metric values against SLO threshold values. When the metric value violates the SLO threshold value, an alert is generated. The values are evaluated every minute by default.

To change the SLO polling interval, follow these steps:

- 1 Stop SOA Manager if it is currently started.
- 2 Use a text editor to open `<install_dir>\conf\networkservices\mipServer.xml`.
- 3 Add a `com.hp.mip.slo.windowSize` entry. Possible values are `OneMinute`, `FiveMinutes`, `FifteenMinutes`, `OneHour`, `EightHours`, and `Life`. The default is `OneMinute`. For example:

```
<entry name="com.hp.mip.slo.windowSize">FiveMinutes</entry>
```

- 4 Save and close the file.
- 5 Restart SOA Manager.

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 HP SOA Manager web 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 Network Services 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

The policy enforcement agents and 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 agents and 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.

Policy Enforcement Intermediary

You can enable a policy enforcement intermediary agent's business content handler using the Broker Configurator. The handler must be enabled for each intermediary service that you create. You can enable the handler when you first create the intermediary service or you can edit an intermediary service at any time and enable the handler. The following procedure enables the handler for an intermediary service and defines a business content alert using the Finance Service as an example.

To enable the handler and define a business content alert, follow these steps:

- 1 Start the Broker Configurator.
- 2 From the Action column, click **edit** for the intermediary service you want to configure. The Service Configuration screen opens.
- 3 From the Features section, click the Business Content Alerting check box. A check mark indicates that the option is selected and that business content alerting is enabled. The business content alerting parameters are displayed.
- 4 Define the business metric using the following parameter fields:
 - **Name:** Enter a user friendly name to identify the alert (for example, HPQ Alert).
 - **Operation:** Enter an operation in the service that contains the business content you want to monitor. The XPath expression is applied to the operation (for example, `getInfo`).
 - **Alert applies to:** Select when you want the intermediary to search for the operation. You can select to search during requests or responses.
 - **Expression:** Enter an XPath expression which selects the business content from the operation. For example, `//ns1:InfoRequest/ns1:symbol/text()`. This expression traverses the SOAP message for the InfoRequest node and selects the text found for the symbol child node.
 - **Message:** Enter a user friendly message that is sent with the alert (for example, A `{name}` alert has occurred).
 - **Dynamic Properties:** Enter a dynamic variable defined within the message. The Name field corresponds to the variable name. The XPath field corresponds to an XPath expression used to update the variable. For example, **Name:** name
Xpath: `//s:Envelope/s:Body/t:InfoRequest/t:symbol/text()`
 - **Namespace Prefixes:** Enter any name space prefixes that appear in the XPath expression for example, **prefix:** ns1 **URI:**
`http://wsm.hp.com/Finance/Request`
- 5 At the bottom of the screen, click **Save**. The Intermediary Services screen opens and the intermediary service is automatically deployed. The deployment is complete when the status changes to Operational.
- 6 Send the Web service a request that uses the operation that is being monitored by the BusinessMetricHandler handler. An alert is sent to the HP SOA Manager web interface.

- 7 From the HP SOA Manager web interface Business Services list, select the business service that contains the intermediary service for which the business content alert was defined. The Business Service View screen opens and the alert is listed in the Alerts section.

WSM J2EE Agent

You can enable a J2EE agent's business content alert handler using the Web service's `web-services.xml` file. See the relevant section in the guide for additional information on configuring handlers when using a J2EE agent.

To enable the handler and define a business content alert, follow these steps:

- 1 Stop the WebLogic Server (WLS), where the J2EE Agent is running, if it is currently started.
- 2 Use a text editor to open a Web service's `web-services.xml` file. The file is typically located in a web application's `/WEB-INF` directory.
- 3 Add the following XML content within the `<web-services>` root element.

```
<handler class-name=
    "com.hp.wsm.agent.bizmetrichandler.BusinessMetricHandler">
  <init-params>
    <init-param value="" name="" />
  </init-params>
</handler>
```

- 4 Configure the handler's parameters using name/value pairs as shown below. Finance Service is used as an example.

```
<handler class-name=
    "com.hp.wsm.agent.bizmetrichandler.BusinessMetricHandler">
  <init-params>
    <init-param value="FinanceService" name="catalog" />
    <init-param value="FinanceService" name="servicename" />
    <init-param value="FinanceServiceSoap"
      name="serviceporttype" />
    <init-param value="http://wsm.hp.com/finance"
      name="servicenamespace" />
    <init-param value="symbol" name="Name" />
    <init-param value="InfoRequest" name="OperationName" />
    <init-param value="//n1:InfoRequest/n1:symbol"
      name="Expression" />
    <init-param value="true" name="ProcessRequest" />
    <init-param value="false" name="ProcessResponse" />
    <init-param value="false" name="ProcessFault" />
    <init-param value="http://wsm.hp.com/Finance/Request"
      name="xmlns:n1" />
    <init-param value="An InfoRequest alert has occurred"
      name="message" />
    <init-param value="//s:Envelope/s:Body/t:InfoRequest/
      t:symbol/text()"
      name="dynamicprop:p1" />
  </init-params>
</handler>
```

— **catalog**: Enter the web application's name.

— **servicename**: Enter the Web service's name.

- **serviceporttype:** Enter the port type as defined in the WSDL file for the Web service.
 - **Name:** Enter a user friendly name to identify the alert.
 - **OperationName:** Enter an operation in the service that contains the business content you want to monitor. The XPath expression is applied to the operation.
 - **Expression:** Enter an XPath expression which selects the business content that will trigger an alert
 - **ProcessRequest:** Allows a request to be processed. Valid values are `true` or `false`.
 - **ProcessResponse:** Allows a response to be processed. Valid values are `true` or `false`.
 - **ProcessFault:** Allows a fault to be processed. Valid values are `true` or `false`.
 - **Namespace Prefixes:** Enter any namespace prefixes that appear in the XPath expression are entered as `prefix/namespaceURI` pairs.
 - **Message:** Enter a user friendly message that is sent with the alert.
 - **Dynamic Properties:** Enter a dynamic variable defined within the message pattern. The value is an XPath expression used to update the variable.
- 5 Save and close the file.
 - 6 Restart WLS for the changes to take effect.
 - 7 Send a Web service request that uses the operation that is being monitored by the BusinessMetricHandler handler. An alert is sent to the SOA Manager web interface.
 - 8 From the SOA Manager web interface Business Services list, select the business service that contains the managed Web service for which the business content alert was defined. The Business Service View screen opens and the alert is listed in the Alerts section.

WSM .Net Agent

The .NET Agent Business Metric SOAP extension is enabled by modifying a Web service application's `Web.config` file. The file can be edited using an XML editor or a text editor. The following instructions enable the extension and also demonstrate how to define a business content alert using Finance Service as an example.

To define a business content alert, follow these steps:

- 1 Using a text editor, open the FinanceService's `Web.config` file. For example:
`C:/Inetpub/wwwroot/FinanceService/Web.config`
- 2 Edit the file by adding a `<services>` node within the `<configuration>` node.
- 3 Create a `<service>` node within the `<services>` node and include a `name` attribute that contains the Web service name. For example:

```
<services>
  <service name="FinanceService.asmx">
```
- 4 Define the business content alert within a `<bizmetric>` node as shown below.


```

<bizmetric>
  <name>HPQ Info</name>
  <expression>
    //s:Envelope/s:Body/t:InfoRequest/t:symbol[text() = 'HPQ']
  </expression>
  <message>InfoRequest = ${company}</message>
  <operation>getInfo</operation>
  <direction>REQUEST</direction>
  <properties>
    <property>
      <name>company</name>
      <value>text()</value>
    </property>
  </properties>
  <namespaces>
    <property>
      <name>s</name>
      <value>http://schemas.xmlsoap.org/soap/envelope/</value>
    </property>
    <property>
      <name>t</name>
      <value>http://wsm.hp.com/Finance/Request</value>
    </property>
  </namespaces>
</bizmetric>

```

- 5 Use the fields to enter the alert policy:
 - **Name:** A user friendly name to identify the alert.
 - **Expression:** An XPath expression which selects the business content from the operation.
 - **Message:** A user friendly message that is sent with the alert. Any alert service variables can also be used in the message. Alert variables are described in the HP SOA Manager web interface.
 - **Operation:** The operation in the service that contains the business content you want to monitor.
 - **Direction:** When to search for the operation. A valid entry is REQUEST.
 - **Properties:** A dynamic variable defined within the message. The name attribute corresponds to the variable name. The value attribute corresponds to an XPath expression used to update the variable.
 - **Namespace:** Any namespace prefixes that appears in the XPath expression. The name attribute refers to namespace prefix. The value attribute refers to the namespace URI.
- 6 Save and close the file.
- 7 Send a Web service request that uses the operation that is being monitored by the Business Metric SOAP extension. An alert is sent to the SOA Manager web interface.
- 8 From the SOA Manager web interface Business Services list, select the business service that contains the managed Web service for which the business content alert was defined. The Business Service View screen opens and the alert is listed in the Alerts section.

Troubleshooting Business Content Alerts

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

SOA Manager Setup

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

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

- Ensure that SOA Manager subscribes to the policy enforcement agent or intermediary for Business Content Alerts.

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

Service Setup

- In the WSM Agent, ensure that the handler that should be raising Business Content Alerts is configured to raise Business Content Alerts.

— When using the policy enforcement intermediary:

View the Service Details page and ensure that the Business Content Alert section is selected and that the details are filled in correctly.

On the Services list, ensure that the Service is deployed and using the current configuration. The words “(changed on disk)” should not appear next to the Service in the Service list. If it does appear, undeploy and redeploy the Service.

— When using the WSM J2EE Agent:

View the web applications `web-services.xml` configuration file and ensure that the `BusinessMetricHandler` has been added to the handler chain and is properly configured. You must restart the WLS for any changes to `web-services.xml` to take effect.

— When using the WSM .NET Agent:

Make sure that the `BusinessMetricPolicy` is defined and a business content alert is configured in the `Web.config` file for the Web service application.

Invocations

- Check that the invocations that should be triggering the Business Content Alert are actually reaching the configured policy enforcement agent or 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.
 - When using the WSM J2EE Agent:

Edit the `logging.properties` file in the `JRE /lib` directory and add the logger:

```
com.hp.ov.mip.wsm.commons.net.level=WARNING
```

Restart WLS.

Send an invocation to the Web service. The request and response messages should appear in the domain log file (for example, `bea/user_projects/domains/<domain>/<domain>.log`).
 - .NET does not currently log request or response payloads.
- 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.
 - When using the WSM J2EE Agent:

Confirm that the operation name of the invoked method matches the value of the `OperationName` parameter in the `web-services.xml` file.

Confirm that the XPath expression will select the correct node in the request body, response body, or fault.

Confirm that the namespace prefixes used in the XPath expression are correctly defined in the `web-services.xml` file.
 - When using the WSM .NET Agent:

Confirm that the service being invoked is the service for which the Business Content Alert is configured.

Confirm that the XPath expression specified in the `Web.config` file selects the correct node in the request.

Confirm that the condition specified in the XPath expression exists in the selected node (if the alert specifies that the value must equal `foo`, the node text value should be `foo`).

Confirm that the namespace prefixes used in the XPath expression are correctly defined in the `Web.config` file for Web service application.

- Check that the policy enforcement agent or 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.

- When using the WSM J2EE Agent:

Edit the `logging.properties` file in the `JRE /lib` directory and add the category:

```
com.hp.ov.mip.wsm.impact.sba.controller.service.event=WARNING
```

Restart WLS.

Send an invocation through the intermediary. Log messages should appear in the domain log file (`bea/user_projects/domains/<domain>/<domain>.log`) indicating that a `BusinessMetricAlert` is being sent.

- .NET does not currently log the sending of `BusinessMetricAlerts`.

- Check that Network Services 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 HP SOA Manager web 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 Manager web interface does not remove alerts from the SOA Manager database.

Querying Alerts

All alerts are stored in the SOA Manager database. The alerts remain in the database even after they are removed from the SOA Manager 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 Manager 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 Manager 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 Manager 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 Manager 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.

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 Manager 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.

Business services are used to better align business managers, IT/operation administrators, and application developers. In this model, business managers define the business service, application developers architect and develop a services-based solution, and administrators deploy and manage the solution across the enterprise. This orchestration is captured in the service model and allows an organization to quickly react and adapt to business changes.

Some of the benefits of managing Web services using a service model are the following.

- 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.
- A business service includes various metrics, operations, and events that support the paradigm of assess, advise, and act.
- Repetitive tasks such as deploying software and configuring connectivity between underlying PEPs are automated by leveraging the meta-data captured in the service model.
- Business services are represented as a standards-based managed object that can be integrated into current management products or used as an integration point between management products that need to coordinate adaptive behavior.
- Business services can be published to a UDDI registry. The registry allows business services to be discovered and used by any management products that support UDDI.

Defining Business Services

Business services are defined using the HP SOA Manager web 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 Manager. 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 Manager.

- 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

PEP 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. A corresponding configuration type exists for each PEP type. Any number of PEP configurations can be included in a business service. Each PEP configuration can include resource configurations that contain the resources of the corresponding PEP that are to be managed within the business service.

To select any subset of the existing PEPs and create corresponding PEP 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 PEP 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 PEP Configuration

PEP configurations are used to link a business service with a PEP. A PEP contains the resources that are managed within a business service. A corresponding configuration type exists for each PEP type. Any number of PEP configurations can be included in a business service.

To add a PEP 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 Manager.

- **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 Resource Configuration

You can add a resource configuration that contains the resources that are to be managed 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/intermediary 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 agent or 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 Manager.

- **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 agent or intermediary or undeploy the service from a managed policy enforcement agent or intermediary. Disregard this field if the Web/brokered service for this business service is already deployed to a managed policy enforcement agent or 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 agent or 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 or brokered Web service 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 agent or 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 HP SOA Manager 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 HP SOA Manager.

- **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 Manager 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 Manager 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 or agent instances configurations, and not vice-versa. Therefore explicit dependencies are needed for a business service to receive alerts from the contained policy enforcement intermediary or agent 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 agent or 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.
- 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 the Network Services Server. 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 and publishing of the service. 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 Manager 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 UDDI Registry

The HP SOA Manager web interface can publish business services to a UDDI registry. Business services that are published to a UDDI registry can be reused by other applications. To use the UDDI feature, you must have a UDDI registry and configure the registry with the Network Services server. To configure a UDDI registry, see “Configure a UDDI Registry” in Chapter 2.



You must publish the business service’s dependencies before the business service. Publish consumed business services, policy enforcement intermediary group, and Policy enforcement points first. See chapter 3 for details on publishing these dependencies.

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:
 - **Management Web Service Provider:** Select the provider for the business service. Select a Business Entity name from the drop-down list of Business Entities in the UDDI registry.
 - **Web Service Provider** (optional): Select the provider for the Web service. This is the Web service managed by the HP SOA Manager web interface business service. Select a Business Entity name from the drop-down list of Business Entities in the UDDI registry. If the Web service is already in the UDDI registry, SOA Manager will not change the existing entries. If the service is already in the registry, the Web Service Provider parameter is not used.
 - **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.



When you delete a policy enforcement agent or intermediary configuration, its UDDI entries are deleted.

When you delete a business service, the business service and policy enforcement agent or intermediary configuration UDDI entries are deleted. The UDDI registry entities corresponding to Web services are not 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.

Reusing a Business Service

The HP SOA Manager 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).

SOAM version 2.1 supports importing a business service that was created using version 2.0. The business service is automatically updated to a 2.1 compliant business service. In particular, the import updates the following:

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

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, and SLO alerts are no longer triggered. 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, and SLOs and alerts are no longer triggered for this service. 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.

Using SSL for the Management Channel

This chapter describes how to secure the management channel and the management components that are used in SOA Manager. You should be familiar with general security principles and SSL security before attempting any of the tasks in this chapter. 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 Manager solution.



This chapter does not include instructions for securing the application channel. For more information about Application channel security for the WSM Intermediary, see the relevant information in this guide.

This chapter does not include instructions for using Select Access for identity management. See Chapter 8 “Integrating with Select Access” for instructions on using Select Access.

Overview

The SOA Manager management channel contains sensitive data about Web services that are being managed. The data include performance data, auditing data, and business content data. More importantly, the management channel exposes interfaces that are used to interact with a Policy Enforcement Agent or 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 Manager, WSM Agents, and WSM Intermediary and is also used to secure communication to SOA Manager 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 Manager 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 Manager

The steps below detail how to assign a Key Store and Trust Store for use by SOA Manager. SOA Manager 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 Manager server, follow these steps:

- 1 Stop SOA Manager if it is currently started.
- 2 Use a text editor to open `<install_dir>\conf\mipServer.xml`.
- 3 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.

- 4 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 Manager server, they share the same Key Store and Trust Store. Assigning a Key Store and Trust Store for the SOA Manager 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**.

WSM Agents

The WSM Agents are integrated with their respective WS Containers (IIS or WLS) and leverage their container's HTTP Server and SSL implementation. See the IIS or WLS SSL documentation for instructions on configuring a Key Store and Trust Store.

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 Manager

To configure SSL settings in SOA Manager, 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>
```

- **Secure Port:** SOA Manager secure port that is used to accept HTTPS requests from the SOA Manager web interface. Any open port can be used.
- **Webapps Secure:** Enables SSL on the SOA Manager server. Valid entries are **true** and **false**.

- 3 Save and close the file.
- 4 Start SOA Manager.

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.

WSM Agents

The WSM Agents are integrated with their respective policy enforcement agent (IIS or WLS) and leverage their container's HTTP Server and SSL implementation. See the IIS or WLS SSL documentation for instructions on enabling SSL and defining secure ports.

Registering a Secure Policy Enforcement Agent /Intermediary

Managed policy enforcement agents/intermediaries that run on a secure server are registered with SOA Manager by using the SOA Manager web interface in the same manner as non-secure managed policy enforcement agents/intermediaries. However, because a managed policy enforcement agent/intermediary runs on a secure server, the server's secure port must be used.



Because SOA Manager acts as an HTTP Client, its Trust Store must contain the CA root certificate for each managed policy enforcement agent/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 agent/intermediary, follow these steps:

- 1 Make sure the managed policy enforcement agent/intermediary that you want to register is started.
- 2 From the SOA Manager web interface, click **Policy Enforcement Points**. The Policy Enforcement Points Summary screen opens.
- 3 Select the PEP you want to contain the Policy Enforcement Agent or Intermediary. The PEP View screen opens for the selected PEP.
- 4 From the Contained Policy Enforcement Agent or Intermediary instances section, click the **Add** link. The Add Policy Enforcement Agent or 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 Agent or Intermediary is installed.
- 7 Click to select the **SSL** check box.
- 8 Click **Add**. The Add Policy Enforcement Agent or Intermediary screen reopens and lists the Web services that were discovered in the managed Policy Enforcement Agent or Intermediary.
- 9 Click **Add**. The Policy Enforcement Agent or Intermediary screen opens and lists the resources that are now registered in the SOA Manager. The Management Interface (WSDL) field indicates an HTTPS URL.
- 10 Repeat this procedure to register additional secured managed Policy Enforcement Agent or Intermediary.

Accessing the SOA Manager Web Interface

When using SSL, the SOA Manager web interface is accessed through the secure port of SOA Manager (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 Manager server’s SSL certificate. See your browser’s documentation for information on installing a CA’s trusted root certificate.

To access the SOA Manager web interface:

- 1 Open a Browser.
- 2 Enter the following URL and substitute *<host>* with the DNS host name where the SOA Manager server is running and *<secure_port>* with the server’s secure port:

```
https://<host>:<secure_port>/bse
```

Accessing the Broker Configurator

When using SSL, the Broker Configurator is accessed through the Intermediary’s secure port (see “Configuring SSL Settings” above). Any browser used to access the Broker Configurator must contain a CA root certificate from the CA that was used to verify the Intermediary’s SSL certificate. See your browser’s documentation for information on installing a CA’s trusted root certificate.

To access the Broker Configurator:

- 1 Open a Browser.
- 2 Enter the following URL and substitute *<host>* with the DNS host name where the SOA Manager server is running and *<secure_port>* with the server’s secure port:

```
https://<host>:<secure_port>/console
```

Integrating with Select Access

This chapter describes how to set up and configure the Select Access integration for the SOA Manager and SOA Manager Intermediary. To complete the instructions in this chapter you need:

- A general understanding of the Select Access Policy Builder
- Access (local or remote) to a Select Access Server Version 6.1 or 6.2
- A Select Access Installation CD Version 6.1 or 6.2
- General Understanding of the SOA Manager

Overview

Select Access provides an identity management solution for securing access to various services and resources. SOA Manager uses Select Access for securing access on both the application channel and the management channel.

For the management channel, Select Access can be used to replace the default security provider that controls access to the HP SOA Manager web interface and the Broker Configurator Web applications. This allows for single sign-on scenarios where policies for user authentication are pre-established in the enterprise. See the "Authenticating HP SOA Manager web interface and Broker Configurator Login" section below after completing the instructions for setting up the Select Access integration.

For the application channel, Select Access is used to provide authentication and authorization for consumers of Web services. Select Access integration for the application channel requires the HP SOA Manager Intermediary to mediate Web service communication. See the "Using the Broker's Security Features" chapter in this guide for detailed instructions on setting up application channel security when using the Intermediary.

Setting Up the Select Access Integration

The Select Access integration must be set up and configured before using Select Access with SOA Manager. This entails tasks that are performed on SOA Manager and intermediary, as well as tasks that are performed on the Select Access Administration Server using the Select Access Administration console.

Install the Select Access Servlet Enforcer

The Select Access Servlet Enforcer must be installed on the same machine as SOA Manager as well as any computers that are hosting the intermediary. If SOA Manager and the intermediary are located on the same computer, The Enforcer only has to be installed once. This may be typical during testing scenarios.

To install the Select Access Servlet Enforcer:

- 1 Place the Select Access 6.1 installation CD in the CD ROM drive and close the tray. The installation's start screen displays.
- 2 If this is the first time a Select Access component is being installed on this system:
 - Click **Next**.
 - Read and agree to the License Agreement.
 - Select the location where you would like to install Select Access Components. (i.e., C:\Select Access).
 - Click **Next**.

Or,

If you have previously installed Select Access components on this system:

- Click **Modify – Install new components on this host**.
 - Click **Next**.
 - Read and agree to the License Agreement .
 - Click **Next**.
- 3 From the Choose HP OpenView Select Access Components screen, select the **Servlet Enforcer Plugin**. Click **Next**.
 - 4 Click **Install**.
 - 5 When asked if you would like to configure the components now, select **YES** and click **Next**. This will launch the Select Access Setup Tool.
 - 6 From the welcome screen, click **Next**. Keep clicking **Next** until you are asked to configure the Generic Enforcer Plugin. Follow the instruction and choose the location of the enforcer file, and click **Configure**.



Generic Enforcer Plugin is the same as Servlet Enforcer Plugin.

- 7 Provide the requested information to contact the Select Access Administration Server and click **Next**. Make sure to use an IP address.

- 8 In the Setup Options section, select **Typical** and click **Next**.
- 9 Click **Finish**.
- 10 Click **Next** on all the remaining prompts and close the wizards.

Copy the Required Jars

Once the Select Access Generic Enforcer is installed, move the required Enforcer's JAR files into the SOA Manager /lib directory. This procedure must be completed on every system where the Select Access Enforcer was installed.

To copy the required jars:

- 1 Stop SOA Manager and intermediary if they are currently started.
- 2 From the SOA Manager CD, copy `Addons\selectaccess\version6.1\miplib-addons.jar` to `<install_dir>/lib/addons`.
- 3 Copy the following files from the directory in which SA Enforcer was installed (`<SA_Enforcer Install_Dir>\shared`), to `<install_dir>/lib/addons`.
 - `activation.jar`
 - `EnforcerAPI.jar`
 - `Jakarta-oro-2_0.jar`
- 4 Copy the following files from the directory in which SA Enforcer was installed (`<SA_Enforcer Install_Dir>\shared\jetty\policy_builder\protected\`), to `<install_dir>/lib/addons`.
 - `castor-0.9.3.19-xml.jar`
 - `jdom.jar`
 - `ldapjdk.jar`
 - `msgsresources.jar`
 - `protomatter.jar`
 - `shared.jar`
 - `xercesImpl.jar`
 - `xml.jar`
 - `xml-apis.jar`
- 5 From `<install_dir>/lib/addons` directory, rename `xml.jar` to `AAA_sa_xml.jar`.
- 6 Move `AAA_sa_xml.jar` to `<install_dir>/lib`.

Configure SOA Manager to Use Select Access

Two procedures are used to configure the SOA Manager to use Select Access:

- Modify Security Provider Settings for Select Access
- Modify the Select Access Enforcer Properties File

Modify Security Provider Settings for Select Access

The Select Access security provider replaces the default SOA Manager security provider to provide security functions for the SOA Manager web interface authentication.

To modify the security provider settings:

- 1 Stop SOA Manager if it is currently started.
- 2 Use a text editor to open `<install_dir>/conf/networkservices/mipServer.xml`.
- 3 Add security provider names to the `com.hp.mip.security.providers` entry, separated by a semicolon. For example, to add the Select Access security provider to the provider list, use the following entry:

```
<entry name="com.hp.mip.security.providers">default; SelectAccess</entry>
```


- 4 Change the name of the console's security provider to Select Access:


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

The above setting configures Select Access as the security provider for authentication when accessing the SOA Manager web interface.

- 5 Add the security provider configuration file path entry. For example, when using the Select Access security provider:

```
<entry name="com.hp.mip.security.provider.SelectAccess">
  <install_dir>\\conf\\networkservices\\selectaccess.properties
</entry>
```

 The default file provided with SOA Manager has the line listed in step 5 disabled by using the comment tags. You must enable the line by removing the comment tags and also provide the correct path. You must also make sure that you override the backslash character in the line.

 It is recommended that the full path to the Select Access properties file be used. For example,

```
<entry name="com.hp.mip.security.provider.SelectAccess">Z:\Program Files\HP SOA Manager 2.5\conf\broker\selectaccess.properties</entry>
```

- 6 Configure the personalization attribute used for Role Based Security. Refer to section *Configuring Identities for SOA Manager Web Interface Role-based Authentication* for additional information. The personalization attribute name must be configured in the following entry:

```
<entry name="com.hp.mip.security.provider.SelectAccess.personalisation.attribute">employeeType</entry>
```
- 7 Save and close `mipServer.xml`.

Modify the Select Access Enforcer Properties File

SOA Manager must be configured to use the Select Access Enforcer at runtime.

To modify the Select Access Enforcer properties file:

- 1 Use a text editor to open `<install_dir>/conf/networkservices/selectaccess.properties` and configure the following settings:


```
Select AccessLoggingName = MipEnforcer
EnforcerDebugLevel = 9

#SelectAccess service for basic authentication


HttpServiceProtocol = HTTP
HttpServiceHost = <network_services_host_name>
HttpServicePort = Port where SOA Manager is servicing requests. Default is 5002.

#SelectAccess service for certificate-based authentication

HttpsServiceProtocol = HTTPS
HttpsServiceHost = <network_services_host_name>
HttpsServicePort = Port where SOA Manager is servicing SSL requests. Default is 8443

#SelectAccess service authentication resource path
AuthenticationResource = /authentication

EnforcerConfigFile = Specify the full path to the enforcer.xml file (i.e.,
C:\Program Files\HP OpenView\Select Access\bin\enforcer.xml).
```

 The above settings will be needed when defining a Select Access service.
- 2 Save and close `selectaccess.properties`.
- 3 Restart SOA Manager and make sure that there are no errors when starting.

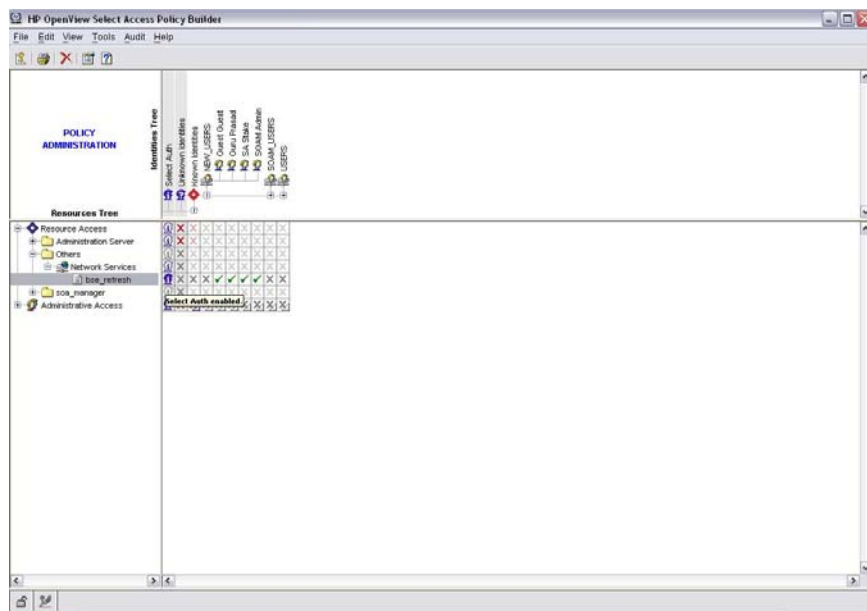
Authenticating SOA Manager Web Interface and Broker Configurator Login

The following instructions demonstrate how to use Select Access to authenticate users that log in to the SOA Manager web interface and the Broker Configurator. Before you complete this section, you must complete the “Setting Up the Select Access Integration” section above.

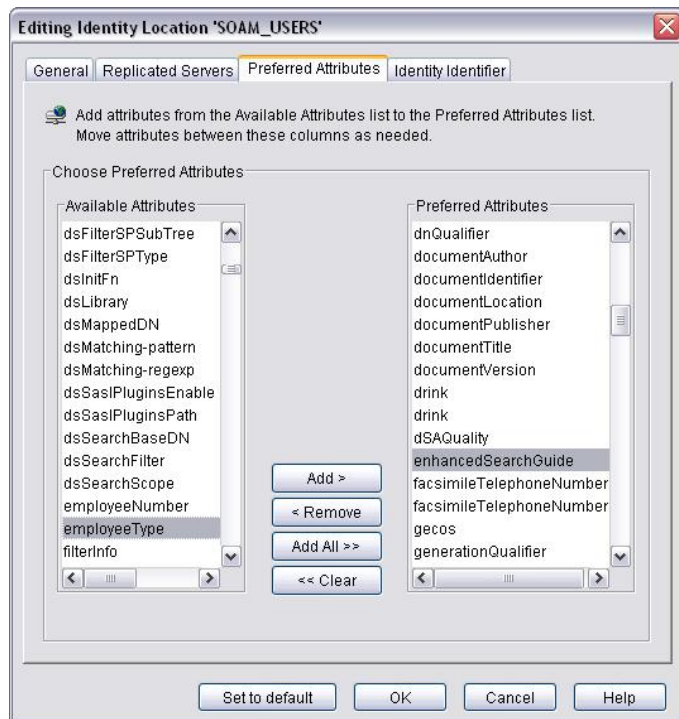
Configuring Identities for SOA Manager Web Interface Role-based Authentication

You must perform the following steps to configure SA as the security provider for SOA Manager:

- 1 Log in to HP Select Access Policy Builder. This displays the HP Select Access Policy Builder page as shown in the following figure.



- 2 Right-click **Known Identities** and select **New->Identity Location** from the shortcut menu. This displays the New Identity Location dialog box.
- 3 Type the required details in the General tabbed page.
- 4 Click the **Preferred Attributes** tab to display the corresponding tabbed page as shown in the following figure.



- 5 Select an attribute that supports plain text as a value from the **Available Attributes** list box. For example, in this case, the attribute `employeeType` is selected. This attribute type must be the attribute type configured in the `selectaccess.properties` file. If you are choosing another attribute type, make sure that you update the `selectaccess.properties` file with this attribute type.
- 6 Click **Add** to add the selected attribute to the **Preferred Attributes** list box.
- 7 Click **OK** to return to the HP Select Access Policy Builder page.
- 8 Right-click the new identity location you created from the HP Select Access Policy Builder page and select **New->Create Identity**. This displays the Editing Identity `<identity name>` dialog box as shown in the following figure. `<identity name>` signifies the identity that you created.

Editing Identity 'Guest Guest'

Identity Information | Group Membership | Profile Management

Enter information about the identity.

First Name: Guest

*Last Name: Guest

*Common Name: Guest Guest

User ID: guest

Phone:

Password: *****

Confirm Password:

E-Mail:

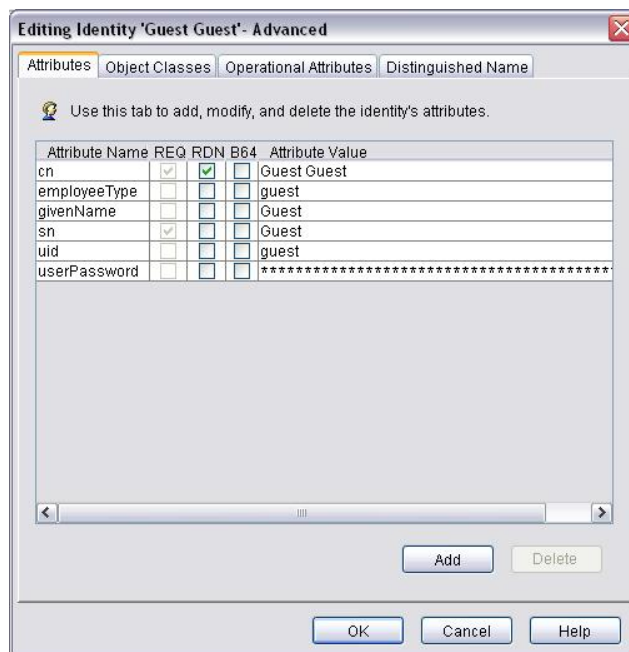
Fax:

Non-human identity

* You must enter information in these fields.

Advanced OK Cancel Help

- 9 Type the details in the respective boxes and click **Advanced**. This displays the Editing Identity `<identity name>` Advanced dialog box as shown in the following figure.



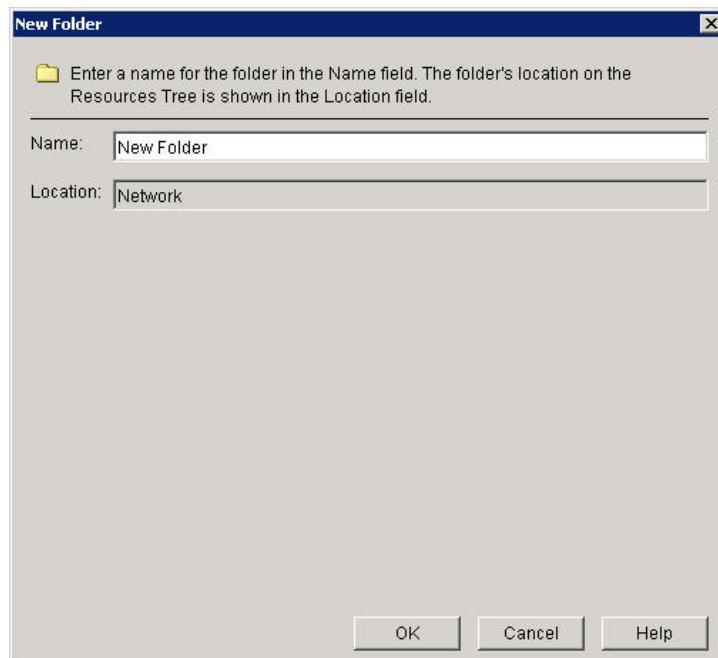
- 10 Click **Add** on the Editing Identity <identity name> Advanced dialog box and select the attribute that you used for role. (employeeType). Enter the value of the attribute as admin, stakeholder, or guest corresponding to the role you defined using SOA Manager web interface.
- 11 Click **OK** to go back to the HP Select Access Policy Builder page.

Define a Select Access Resource Server for the SOA Manager Web Interface

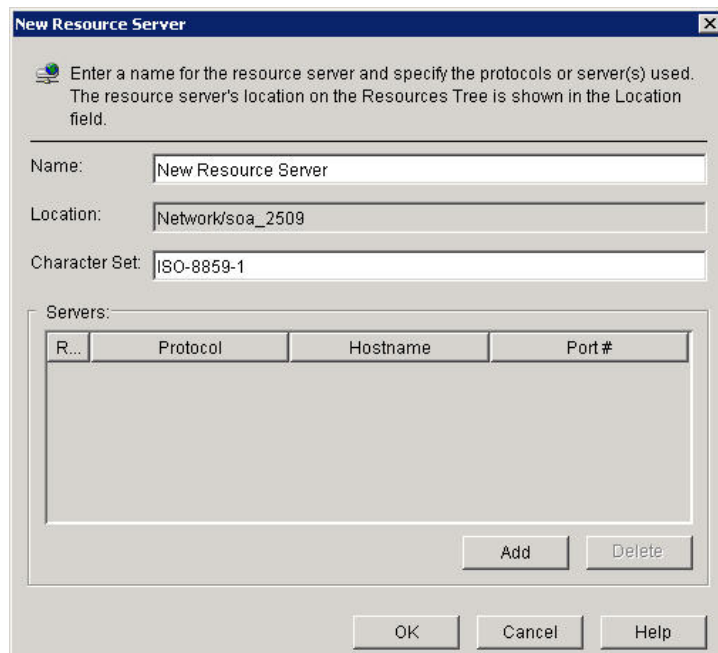
When the Select Access security provider is specified in the SOA Manager properties file, <install_dir>/conf/networkservices/mipServer.xml, the SOA Manager web interface automatically uses that security provider to authenticate a login. However, you must create a Select Access resource for the SOA Manager web interface using the HP Select Access Policy Builder.

To define a Select Access Service for the SOA Manager web interface:

- 1 Log in to HP Select Access Policy Builder. This displays the HP Select Access Policy Builder page.
- 2 From the **Select Access Policy Builder Resources Tree**, right-click **Resource Access** and select **New -> Folder**. The New Folder dialog box displays as shown below.

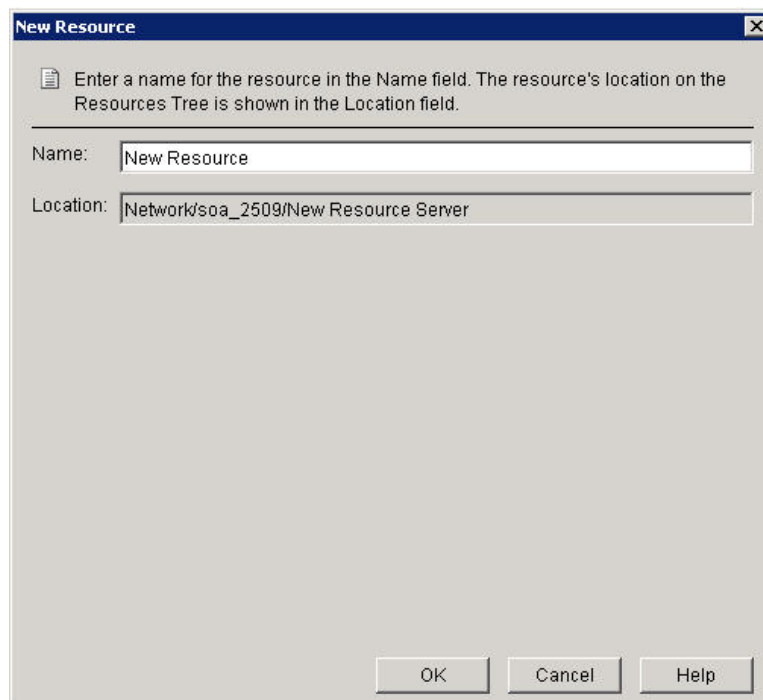


- 3 In the **Name** field, enter a name for the folder, for example, <SOA Manager_Host Name>
- 4 Click **OK**. When prompted to clear the Policy Validator cache, select **OK**. The folder is created and is added to the Policy Builder Resources Tree under Resource Access.
- 5 Right-click the newly created folder and select **New -> Resource Server**. The New Resource Server dialog box displays as shown below.

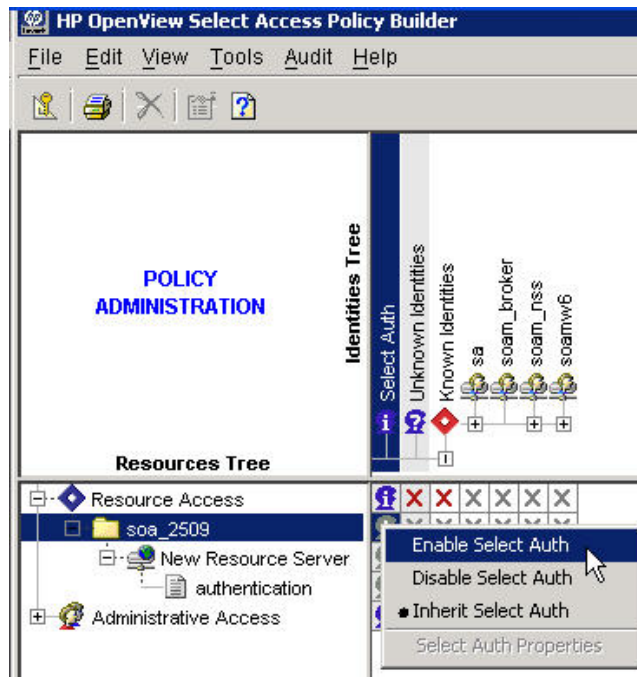


- 6 In the **Name** box, enter a name for this new resource server. Any name that clearly identifies the server can be used.
- 7 Click **Add**. A new entry displays under the Servers section.

- 8 Enter the following information for the server where SOA Manager web interface is located:
 - **Protocol:** The protocol used to access the SOA Manager web interface (HTTP or HTTPS).
 - **Hostname:** <SOA_Manager_host_name>. Make sure that this name is the same as the name you specified in the selectaccess.properties file for SOA Manager.
 - **Port #:** The port number (5002) of SOA Manager web interface. Make sure that this port number is the same as the port number you specified in the selectaccess.properties file for SOA Manager.
- 9 Click **OK** to close this dialog box. When asked to clear the validators cache, select **OK**. The resource server is listed in the Policy Builder Resources Tree.
- 10 From the **Select Access Policy Builder Resources Tree**, right-click on the newly created resource server and select **New -> Resource** from the menu. The New Resource dialog box displays.
- 11 In the **Name** field, enter the resource name authentication. This name must be the same as the name you specified for the authentication resource in the selectaccess.properties file for SOA Manager.

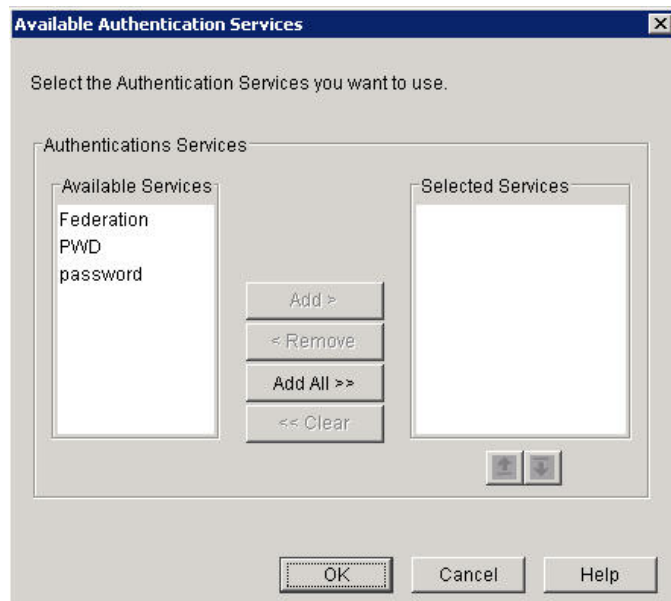


- 12 Click **OK** to save this new Select Access resource. When asked to clear the validators cache, select **OK**. The resource is listed under the service in the Policy Builder Resources Tree. You now have a new resource defined which is used to authenticate SOA Manager web interface users.
- 13 From the **Policy Builder Identities Tree**, right-click the first column on the same row as the resource server for SOA Manager and select **Enable Select Auth** from the shortcut menu as shown in the following figure.



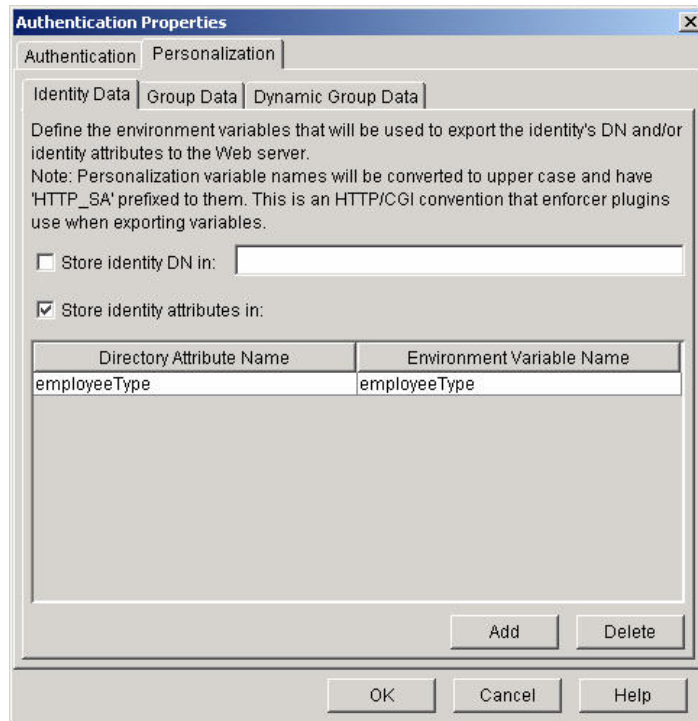
The Authentication Properties dialog box displays.

- 14 Click **Add** from the Authentication Properties dialog box. The Available Authentication Services dialog box displays.



- 15 Select the **password** authentication service and click **Add**. This service is listed in the **Selected Services** column.
- 16 Click **OK**. The Authentication service is added to the list of authentication services in the Authentication Properties dialog box.
- 17 Click the **Personalization** tab from the Authentication Properties dialog box. This displays the Personalization tab page.

- 18 Select the **Store identity attributes in:** check box from the Identity Data tab page and click **Add**. This allows you to add the attribute to the user role that you are using in SOA Manager.



- 19 Select a directory attribute name that supports plain text values from the **Directory Attribute Name** drop-down list. Type the corresponding environment variable in the **Environment Variable Name** box. You must make sure that this entry matches with the entry for the Select Access personalization attribute section in the mipServer.xml file for SOA Manager.
- 20 Click **OK**. This completes the process to define an SA resource server for SOA Manager web interface and configure the personalization attribute in SA for SOA Manager.

You can use the **Policy Builder Identities Tree** in SA to assign users configured in the SOA Manager web interface.

Configure the Intermediary to Use Select Access

Two procedures are used to configure the intermediary to use Select Access:

- Modify Security Provider Settings for Select Access
- Modify the Select Access Enforcer Properties File

Modify Security Provider Settings for Select Access

The Select Access security provider replaces the default intermediary security provider to provide security functions for intermediary console authentication, web service request authentication and authorization, and security auditing.

To modify the security provider settings:

- 1 Stop the intermediary if it is currently started.
- 2 Use a text editor to open `<install_dir>/conf/broker/mipServer.xml`.
- 3 Add security provider names to the `com.hp.mip.security.providers` entry, separated by a semicolon. For example, to add the Select Access security provider to the provider list, add the following entry:

```
<entry name="com.hp.mip.security.providers">default; SelectAccess
</entry>
```

- 4 Change the name of the security provider in the security service entries, for example:

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

```
<entry name="com.hp.mip.security.provider.authorization">
SelectAccess
</entry>
```

```
<entry name="com.hp.mip.security.provider.authentication">
SelectAccess
</entry>
```

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

The above settings configure Select Access as the security provider for authorization and authentication. The default security provider is used for the Broker Configurator and security auditing.

- 5 Add the security provider configuration file path entry. For example, when using the Select Access security provider:

```
<entry name="com.hp.mip.security.provider.SelectAccess">
  C:\\<install_dir>\\conf\\broker\\selectaccess.properties
</entry>
```



It is recommended that the full path to the Select Access properties file be used.

- 6 Save and close `mipServer.xml`.

Modify the Select Access Enforcer Properties File

The Broker must be configured to use the Select Access Enforcer at runtime.

To modify the Select Access Enforcer properties file:

- 1 Use a text editor to open `<install_dir>/conf/broker/selectaccess.properties` and configure the following settings:

```
Select AccessLoggingName = AxisEnforcer
EnforcerDebugLevel = 9
```

```
#SelectAccess service for basic authentication
```

```
HttpServiceProtocol = HTTP
```

```
HttpServiceHost = <broker_host_name>
```

```
HttpServicePort = Port intermediary is servicing requests on. Default is 9032.
```

```
#SelectAccess service for certificate-based authentication

HttpsServiceProtocol = HTTPS
HttpsServiceHost = <broker_host_name>
HttpsServicePort = Port intermediary is servicing SSL requests on. Default is
8443

#SelectAccess service authentication resource path
AuthenticationResource = /authentication

EnforcerConfigFile = Specify the full path to the enforcer.xml file (i.e.,
C:\\Program Files\\HP OpenView\\Select Access\\bin\\enforcer.xml).
```



The above settings will be needed when defining a Select Access service.

- 2 Save and close `selectaccess.properties`.
- 3 Restart the intermediary and ensure there are no errors when starting.

Modify Select Access Settings for Auto Configuration of Services

You must do as follows to modify SA settings for auto configuration of services:

- Add SA certificates to Broker keystore
- Enable auto configuration for SA

Adding SA Certificates to Broker Keystore.

- 1 Import the SA certificate on the broker Keystore as the trusted certificate.
- 2 Refer to the *Assign Keystores and Trust Stores* section of this guide for more information how to configure the key store for SOA Manager. The SA certificate should be added to the broker trust store location.

You can find the default certificate of SA at `<SA install folder>\Select Access\shared\jetty\etc\certs\mcacert.cer`. on SA policy Validator.

Enable Auto Configuration for SA

You must modify the `mipserver.xml` file present at `<install_dir>\conf\broker\` to enable auto configuration for Select Access. A sample of the lines that you must modify is provided below for your reference.

```
<entry name="com.hp.mip.security.provider.autoconfigrequired">true</entry>
<entry name="com.hp.mip.security.provider.sa.adminapi.id">sa_admin</entry>
<entry name="com.hp.mip.security.provider.sa.adminapi.pwd">sa_admin</entry>
<entryname="com.hp.mip.security.provider.sa.resourcepath">/network/SOA_Manager/In
termediary/Services</entry>
<!-- The following two properties indicate the authentication services configure
in SA. One for basic authentication and the other for certificate based
authentication. These properties define the authentication service type and the
authentication service name delimited by a '/'. This property has to be defined
in this format only. In The following example the "password" and "certificate"
```

indicates the authentication type and, "PWD" and "CERT" indicates the authentication service names-->

```
<entry
name="com.hp.mip.security.provider.sa.basicauthservicename">password/PWD</entry>
<entry
name="com.hp.mip.security.provider.sa.certauthservicename">certificate/CERT</entry>
<entry name="com.hp.mip.security.provider.sa.defaultuser">cn=Soam, ou=SOAM_USERS,
dc=asiapacific, dc=hpqcorp, dc=net</entry>
<entry
name="com.hp.mip.security.provider.sa.adminapi.wsd1">https://<sa_host_name>:9993/
axis/services/wsadmin</entry>
<entry
```

Line	Parameter Value	Description
<entry name="com.hp.mip.security.provider.autoconfig equired">true</entry>	true false	true- Enables auto configuration false- Disables auto configuration
<entry name="com.hp.mip.security.provider.sa.adminap i.id">sa_admin</entry>	User defined	Administrator ID to access Select Access Admin APIs
<entry name="com.hp.mip.security.provider.sa.adminap i.pwd">sa_admin</entry>	User defined	Password for the administrator ID
<entryname="com.hp.mip.security.provider.sa.re sourcepath">/network/SOA_Manager/Intermedia ry/Services</entry>	User defined	Path to the Select Access resource. You must specify /network at the start of the path only for specifying a network resource. All the other types of resources are present as separate folders in the /Services folder (SA resource server).
<entry name="com.hp.mip.security.provider.sa.basicaut hservicename">password/PWD</entry>	User defined	Basic authentication service name
<entry name="com.hp.mip.security.provider.sa.certauth servicename">certificate/CERT</entry>	User defined	Certificate authentication service name
<entry name="com.hp.mip.security.provider.sa.defaultu ser">cn=Soam, ou=SOAM_USERS, dc=asiapacific, dc=hpqcorp, dc=net</entry>	User defined	DN of the Select Access default user or group

<pre><entry name="com.hp.mip.security.provider.sa.adminapi.wsdli.wsdli">https://<sa_host_name>:9993/axis/services/wsadmin</entry></pre>	User Defined	Administration service URL to access the Select Access administration APIs. You must replace <sa_host_name> with the host name on which SA is installed.
---	--------------	--

Restart the intermediary after making the changes

Configure Select Access for SOA Manager Auto Configuration

The Select Access (SA) auto configuration feature allows you to automatically add a brokered web service URI into Select Access as a resource. You can assign permissions for a brokered web service either to a group of users or individual users by using auto configuration. This feature also helps you to define the policies to allow users access to the brokered web service. The default access permission is *Allow Access*.

Perform the following steps:

- 1 In the SA policy validator, create a new identity for SOA Manager admin user if the identity does not exist in the policy validator. The admin user name and password are configured in the intermediary (broker) configuration file.(mipserver.xml)
- 2 Create an authentication service in SA, one service each for basic authentication and certificate-based authentication if the authentication service is not created already. It is not compulsory to create both the type of services. You can create only the type of service that you are going to use.
- 3 Create a group or identity for which the service access must be enabled. The users or groups details must be configured in the brokers configuration file.(mipserver.xml)
- 4 Enable **SelectAuth** for the **Administration API Resource** server in the Administration Server folder. You must also associate the authentication services to the administration API resource server.
- 5 Enable **SelectAuth** for the **Delegated Administration Resource** server in the Administration Server folder. You must also associate the authentication services to the delegated administration resource server.
- 6 Enable **Delegated Administration** in the **SelectAuth** section for administrative Access and provide the authentication service.
- 7 Enable **Allow Access** to the SOA Manager admin user for **Administrative API** and **Delegated Administration**.
- 8 Enable **Delegate** for the SA admin user.

Note: When a service is provisioned in the intermediary, SOA Manager Intermediary configures the SA service for the service and provides access to the default user configured. You must provide access to other users when required. Refer the broker security feature for additional information.

Define a Select Access Service for the Broker Configurator

When the Select Access security provider is specified in the Broker's properties file, `<install_dir>/conf/broker/mipServer.xml`, the Broker Configurator automatically uses that security provider to authenticate a login. However, you must create a Select Access resource for the Broker Configurator using the Select Access Policy Builder.

To define a Select Access Service for the Broker:

- 1 From the Select Access Policy Builder Resources Tree, right-click Resource Access and select **New | Folder**. The New Folder dialog box displays.
- 2 In the Name field, enter a name for the folder.
- 3 Click **OK**. When asked to clear the Policy Validator cache, select **OK**. The folder is created and is added to the Policy Builder Resources Tree under Resource Access.
- 4 Right-click the newly created folder and select **New | Resource Server**. The New Resource Server dialog box displays.
- 5 In the Name box, enter a name for this new resource server (i.e., Broker). Any name that clearly identifies the server can be used.
- 6 On the bottom of the window click **Add**. A new entry displays under the Servers section.

New Resource Server

Enter a name for the resource server and specify the protocols or server(s) used. The resource server's location on the Resources Tree is shown in the Location field.

Name:

Location:

Character Set:

Servers:

R...	Protocol	Hostname	Port #
<input checked="" type="checkbox"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>

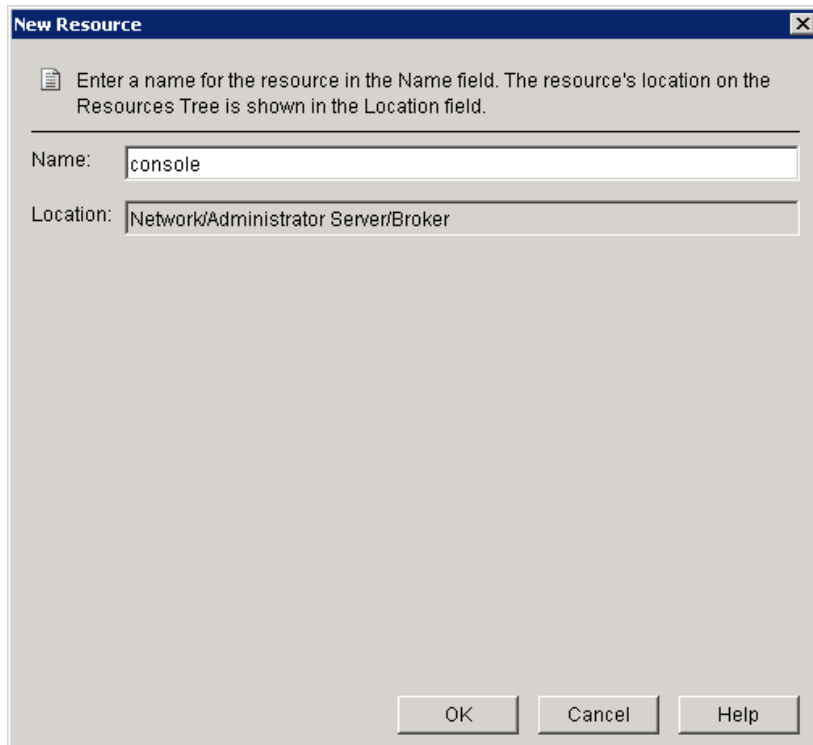
Add Delete

OK Cancel Help

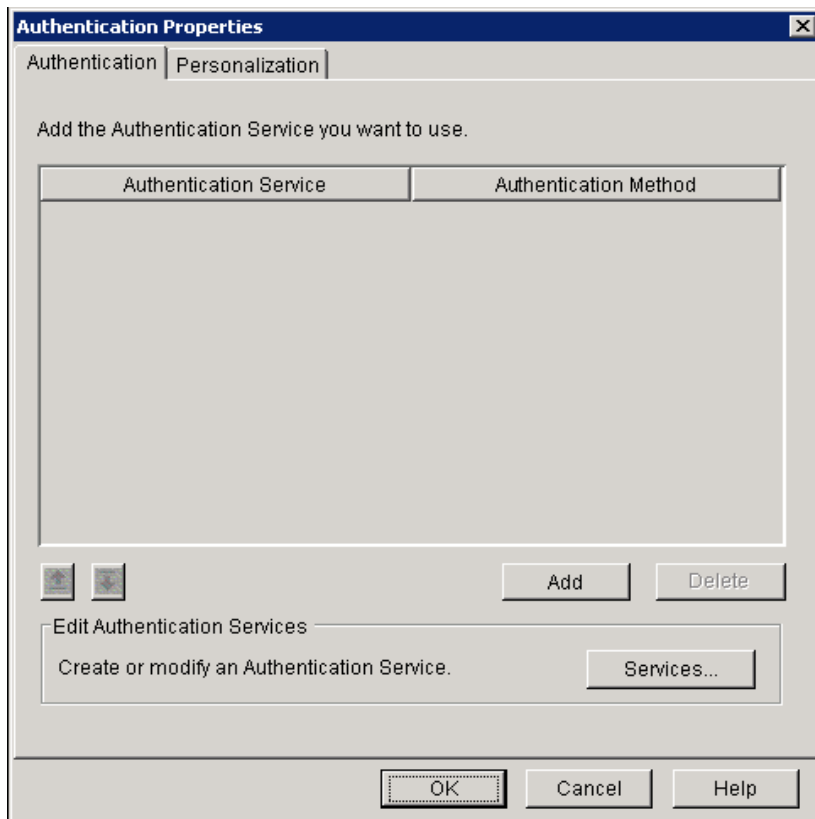
- 7 Enter the following server information for the server where the Broker is located:
 - **Protocol:** The protocol used to access the Broker Configurator (HTTP or HTTPS).
 - **Hostname:** `<broker_host_name>`.
 - **Port #:** The Broker Configurator's port number (9032).

- 8 Click **OK** to close this dialog box. When asked to clear the validator's cache, select **OK**. The resource server is listed in the Policy Builder Resources Tree.
- 9 From the Policy Builder Resources Tree, right-click on the newly created resource server and select **New | Resource** from the menu. The New Resource dialog box displays.
- 10 In the Name field, enter the resource name `console` as shown below. This name corresponds to the resource as it appears in the URL to access the Broker Configurator. For example:

`http://<broker_host_name>:9032/console`).



- 11 Click **OK** to save this new Select Access resource. When asked to clear the validator's cache, select **OK**. This resource is now listed under the service in the Policy Builder Resources Tree. You now have a new resource defined which is used to authenticate Broker Configurator users.
- 12 From the Policy Builder Identities Tree, right-click the first column on the same row as the resource server for the Broker, select **Enable Select Auth** from the pop-up menu. The Authentication Properties dialog box displays.



- 13 Click **Add**. The Available Authentication Services dialog box displays.
- 14 Select the password authentication service and click **Add**. The service is listed in the Selected Services column.
- 15 Click **OK**. The Authentication service is added to the list of authentication services in the Authentication Properties dialog box.
- 16 Click **OK**. When asked to clear the Policy Validator cache, select **OK**. The Select Auth icon shows that Select Auth for the selected resource server is enabled.
- 17 Use the Identities Tree to assign users of the SOA Manager web interface resource.

WSM Intermediary An Overview

The WSM Intermediary is responsible for collecting management data for Web services. The Intermediary runs in its own Java process and delegates service requests through a proxy (intermediary service) to Web services that are deployed in a Web Service Container. An intermediary service must be created for each Web service that you want to manage.

Prerequisites

Users must have fundamental knowledge of the Java programming language and Java platform technologies including security. Users should also have fundamental knowledge of Web services principles and be familiar with their application hosting environment.

Contextual Overview

Intermediary services utilize the Intermediary's handlers, which mediate the communication between a client and a Web service. The handler can be configured with sub-handlers (referred to as common handlers) that provide varying levels of manageability (Monitoring, Logging/Auditing, etc...). The Broker Configurator is used to create intermediary services and configure handlers for intermediary services.

The Smart Business Agent (SBA) provides a method of exposing data and metrics as Web services using WS-based management protocols. Managed objects collect data and metrics from the handlers. The data is represented in the SOA Manager and viewed using the SOA Manager web interface.

Broker Configurator

The Broker Configurator is a Web application that allows you to interact with the Intermediary. In particular, the Broker Configurator is used to configure the Intermediary, create intermediary services, and configure an intermediary service's handlers.

Common Handlers

As shown in **Error! Reference source not found.**, a handler can contain any number of sub-handlers known as common handlers. Together, the handlers are considered a handler chain. The common handlers for a simple intermediary service are described below. Some handlers are enabled by default when you create an intermediary service, while other handlers must be manually enabled. In addition, custom intermediary services provide an expanded list of handlers and the ability to add any custom handler.

Monitoring Handler

The Monitoring Handler collects performance data for a Web service. The data is reported over a period of time (the last 6 minutes, 1 hour, and 1 day). In particular, the handler reports:

- Average Response Time
- Maximum Response Time
- Minimum Response Time
- Security Violations
- Total Request Count
- Total Failure Count
- Total Success Count
- Availability %

- Uptime %

Logging Handler

The Logging Handler is used to collect and publish the Intermediary's log messages. The log messages can be used to troubleshoot any problems that occur with the Intermediary.

Auditing Handler

The Auditing Handler provides message tracing capabilities for a Web service. The handler can be configured to also include SOAP payload for the message.

Schema Validation Handler

The Schema Validation Handler is used to validate Doc Literal SOAP messages to ensure that they comply with the SOAP schema definitions.

Business Content Alerting Handler

The Business Content Handler generates alerts based on content that is found in SOAP requests, responses, or failure messages. The content is found in the message by applying an XPath expression.

Security Handlers

Security Handlers are used to provide both message-level and transport-level security for intermediary services. Authentication and Authorization is provided by HP OpenView Select Access integration.

Using Intermediary Services

This chapter explains how to manage the life cycle of intermediary services. It begins with an overview of intermediary services and then describes how to edit, deploy, view, and remove an intermediary service when using the WSM Intermediary.

Overview

An intermediary service is created for each Web service that you want to manage. The Broker Configurator creates and manages the life cycle of an intermediary service. Requests for managed Web services are sent to the intermediary service and then forwarded (dispatched) to the actual service's endpoints. Intermediary services can be created for both SOAP/HTTP and XML/HTTP Web services.



SOAP with attachments services is supported only if a WSDL is provided that describes the service.

Intermediary services are used to manage Web services when you want to do the following:

- Interpose manageability for Web services that are deployed in a Policy enforcement agent that does not offer native manageability.
- Separate the management of Web services from the services' implementation.
- Provide message-level and transport-level security when a Policy enforcement agent does not include native security features.

Viewing Intermediary Service Details

The Service Details screen lets you view the details of an intermediary service. The details include the intermediary service definition and endpoint, performance data, the Web service's endpoints, and features (handlers) configuration.

To view an intermediary service's details, follow these steps:

- 1 From the Intermediary Services screen, find the intermediary service that you want to view.
- 2 From the Name column, click the intermediary service's name. The Service Details screen opens. The intermediary service's details are listed in different sections. The Features section displays which handlers are enabled and their current configuration settings.

Performance Metrics

The Service Detail screen displays a subset of the performance metrics that are collected for an intermediary service. The metrics include the Average Response Time, Total Requests, Successes, and Failures. These metrics provide a general view of how an intermediary service is performing. The full set of performance metrics is displayed in the SOA Manager server when the intermediary service is managed as part of a business service.

Undeploying an Intermediary Service

An intermediary service that is undeployed is inactive, but is not removed from the Intermediary Service list. The intermediary service is not available for requests until it is deployed. You can configure an intermediary service that is undeployed, but you cannot view any of its management data.



Any Web service management data that has been collected is lost when an intermediary service is undeployed.

To undeploy an intermediary service, follow these steps:

- 1 From the Intermediary Service list, find the intermediary service that you want to undeploy.
- 2 From the Action column, click the **undeploy** link. The status of the service changes from **Operational** to **Inactive**.

Deploying an Intermediary Service

A deployed intermediary service can receive service requests and is considered operational. An intermediary service that is operational collects management data about the Web service that it is managing. An intermediary service is automatically deployed when the intermediary service is created.

To deploy an intermediary service, follow these steps:

- 1 From the Intermediary Service list, find the intermediary service you want to deploy.
- 2 From the Action column, click the **deploy** link. The Status field updates from *Inactive* to *Operational*.
- 3 Verify that the service is operational by clicking the intermediary service WSDL endpoint listed in the Service Interface (WSDL) column. The WSDL for the service is displayed.

Editing an Intermediary Service

You can edit an intermediary service at any time. Typically an intermediary service is edited to enable/disable different handlers depending on the type of manageability that is required for the Web service.

To edit an intermediary service, follow these steps:

- 1 From the Intermediary Service list, find the intermediary service that you want to edit.
- 2 From the Action column, click the **edit** link. The Edit Service screen opens.
- 3 From the Edit Service screen, edit the intermediary service using the fields provided. The handler configuration options are detailed in Chapter 11 “Configuring Handlers”. See Chapter 15 “Using the Intermediary’s Security Features” for detailed instructions if you want to secure communication with the intermediary service.
- 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*.

Changing an Intermediary Service’s Version

Each intermediary service has a description which includes a name that identifies the service in the Broker Configurator and a version number. An intermediary service name is automatically generated when the intermediary service is created. You cannot change the intermediary service’s name, but you can change the version number.

To change an intermediary service’s version, follow these steps:

- 1 From the Configurator's main toolbar, click **List Services**. The Broker Service screen opens.
- 2 From the Action column, click the **edit** link for the intermediary service. The Edit Service screen opens.
- 3 From the Service section, select the Version field and enter a version number for the intermediary service.
- 4 At the bottom of the screen, click **Save**. The Broker Service screen opens and the intermediary service is automatically deployed. The deployment is complete when the status changes to *Operational*.

Configuring an Intermediary Service's HTTP Path

An intermediary service's HTTP Path is the path that will be used by a client to invoke the managed Web service. For example, if the intermediary agent is installed on "MyHost.com" and the default intermediary port is used, the URL to the Web service would be:

```
http://MyHost.com:9032/<http_path_value>
```

A path value is automatically generated when the intermediary service is created. Changing the HTTP path of an intermediary service is useful when multiple intermediary services, with different configurations, are created for the same service or when a specific URL strategy is used by your organization.

To configure a service's HTTP Path, follow these steps:

- 1 From the Configurator's main toolbar, click **List Services**. The Broker Service screen opens.
- 2 From the Action column, click the **edit** link for the intermediary service. The Edit Service screen opens.
- 3 From the Inbound Transport section, select the HTTP Path field and enter a path. The path must consist of alpha-numeric characters and begin with a forward slash (/).
- 4 At the bottom of the screen, click **Save**. The Broker Service screen opens and the intermediary service is automatically deployed. The deployment is complete when the status changes to *operational*.

Removing an Intermediary Service

When an intermediary service is removed, it is deleted from the Intermediary Service list. In addition, the service definition (WSDL) for the intermediary service is deleted from the `<install_dir>\conf\broker` directory.

To remove an intermediary service, follow these steps:

- 1 From the Intermediary Service list, find the intermediary service that you want to remove.
- 2 From the Action column, click the **remove** link. A confirmation dialog box opens and asks you to confirm the removal of the intermediary service.
- 3 Click **OK** to remove the intermediary service.

Enabling Protocol Switching at the Intermediary

You can configure the intermediary to help in communication between service consumers and service providers that follow different protocols. The intermediary provides support to switch messages between JMS and HTTP/HTTPS protocols. The intermediary, before sending the request to an endpoint, transforms the request to a format supported by the protocol at the endpoint. The intermediary, after receiving a response from the endpoint, transforms the response back to the format supported by the protocol at the client that sent the request.

Prerequisites

To use JMS as the transport model, follow these steps to make sure that the prerequisites are satisfied:

- 1 Install the JMS server separately
- 2 Configure the destinations for both inbound and outbound service messages.
NOTE: The intermediary currently supports publish/subscribe and point to point messaging model. You can also enforce policies by using this feature in a JMS transport model.
- 3 Copy the JMS client jar files corresponding to the provider to the <install_dir>/lib/ext directory on the intermediary.
- 4 Make sure that the WSDL for the intermediary Web service contains the JMS binding information.

SOA Manager currently supports the following JMS service providers:

- WebLogic 8.1
- JBoss 4.0.4
- Tibco 4.4.0
- Sonic 7.0, 7.5

During protocol switching, the following sequence of events occurs at the intermediary:

- 1 The service consumer sends a SOAP or XML message over HTTP or JMS to the intermediary.
- 2 The intermediary receives the message and invokes the transport and XML handlers to transform the message based on the protocol supported at the endpoint.
- 3 For a two-way protocol switch interaction, the intermediary creates a temporary destination to receive a response from the endpoint.
- 4 After receiving the response from the endpoint, the intermediary invokes handlers to process the message back to the protocol supported by the service consumer.
- 5 After the message is processed it is converted to the protocol supported by the service consumer.

The intermediary supports two-way protocol switching for the following scenarios:

- Communication between different JMS service providers
- Communication from HTTP to JMS
- Communication from JMS to HTTP

Refer to the following scenarios to configure the intermediary to enable protocol switching.

Enabling JMS-to-JMS-Two-Way Protocol Switching

For a JMS-to-JMS two-way protocol communication, you must make sure that the following prerequisites are satisfied:

Configure three queues, one each for the following:

- Intermediary service
- Endpoint
- Client (to receive messages)

Make sure that the endpoint information present in the WSDL (that contains the queue information) is similar to the setup information of the queue.

To enable JMS-to-JMS protocol switching, follow these steps:

- 1 Start SOA Manager Intermediary and log in to the Broker Configurator.
- 2 Click **Add New Broker Web Service**. The Step1: Import WSDL screen of the Add New Broker Service page opens.
- 3 Type the modified WSDL that contains information about the JMS endpoint 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 Verify that the parameters are similar to what you specified in the WSDL.
- 6 Click **Next**. The Step 3: Configure Broker Service screen of the Add New Broker Service page opens.
- 7 Select **JMS Transport** from the Inbound Transport table. This displays the additional parameters that you must specify for JMS transport.
- 8 Specify the following details in the JMS Transport section. The examples shown in parenthesis for each of the options are specific to a WebLogic server used for inbound transport:
 - Destination Style: Specify either queue or topic for the destination type for the JMS transport model. You must make sure that the
 - Vendor URI: Specify the URL of the vendor that provides JNDI (<http://bea.com>).
 - Initial Context Factory: Specify the name of the JNDI context factory (`weblogic.jndi.WLInitialContextFactory`).
 - JNDI Provider Url: Specify the URL of the JNDI server (`t3://soamw2.ind.hp.com:7001`).
 - JNDI Connection Factory Name: Specify the JNDI lookup name for the connection factory (`weblogic.jms.ConnectionFactory`).
 - JNDI Destination Name: Specify the name of the JNDI destination name (`weblogic.wsee.WLSJbossInboundQueue`). You must make sure that the queue or topic name does not contain a white space character.
- 9 Click **Finish**. This enables the JMS-to-JMS two-way protocol switching at the intermediary.

Enabling HTTP-to-JMS-One-Way Protocol Switching

To enable HTTP-to-JMS protocol switching, follow these steps:

- 1 Start SOA Manager Intermediary and log in to the Broker Configurator.
- 2 Click **Add New Broker Web Service**. The Step1: Import WSDL screen of the Add New Broker Service page opens.
- 3 Type the modified WSDL that contains the JMS endpoint information 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 Verify that the parameters are similar to what you specified in the WSDL.
- 6 Click **Next**. The Step 3: Configure Broker Service screen of the Add New Broker Service page opens.
- 7 Select **Http Transport** from the Inbound Transport table.
- 8 Click **Finish**. This enables the HTTP-to-JMS one-way protocol switching at the intermediary.

Enabling JMS-to-HTTP-One-Way Protocol Switching

To enable JMS-to-HTTP protocol switching, follow these steps:

- 1 Start SOA Manager Intermediary and log in to the Broker Configurator.
- 2 Click **Add New Broker Web Service**. The Step1: Import WSDL screen of the Add New Broker Service page opens.
- 3 Type the modified WSDL that contains the HTTP endpoint information 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 Verify that the parameters are similar to what you specified in the WSDL.
- 6 Click **Next**. The Step 3: Configure Broker Service screen of the Add New Broker Service page opens.
- 7 Select **JMS Transport** from the Inbound Transport table.
- 8 Click **Finish**. This enables the JMS-to-HTTP one-way protocol switching at the intermediary.

Configuring Handlers

This chapter lists the management handlers available for intermediary services in alphabetical order. Each entry includes a description of each handler as well as the handler's fields. Where applicable, example entries for the fields are provided. You can refer to the entries in the chapter when you are editing or creating an intermediary service.



When using custom intermediary services, handler ordering is important, because handlers attach information to the executing operation for other handlers to find and use. Some handlers, like encryption/decryption handlers, also modify the message as it passes along the chain.

Audit Handler

The Audit Handler collects trace information on messages sent to Web services. The auditing feature can collect a message's SOAP payload. The information collected is sent to HP SOA Manager and is stored in a database. The HP SOA Manager web interface is used to query the database to retrieve audit information. Any management application can be extended to access the audit data. For more information on using the SOA Manager Auditing feature, see the “Using Auditing” chapter in the *SOA Manager Administrator Guide*.

Fields

- **Include detailed traces:** Captures profile data. The outcome of a Web service invocation as it passes through each handler in the handler chain for an intermediary service.
- **Payload Option:** Type of message payloads that should be logged.

- **Payload Filter:** Criteria to determine which message payloads should be logged.
- **Expression:** An XPath expression for determining which message payloads should be logged. This field is used if content-based payload logging is configured. This field is only available for custom intermediary services.
- **Namespaces:** Namespaces that are used in the expression field must be declared using the namespace prefix and the namespace URI. This field is only available for custom intermediary services.

Configuring the Audit Publisher

The Audit Publisher is an Intermediary component that publishes audit information that is collected by the audit handler.

There are two configuration options for the Audit publisher: `interval` and `threshold`. The Audit publisher sends trace messages using the value for whichever configuration option is reached first.

- `interval`– The entry sets the amount of time in milliseconds between publishing audit information.
- `threshold` – The threshold sets the number of messages that are published. When the number of messages reaches this threshold, the messages are published.

To configure the audit publisher, follow these steps:

- 1 Stop the Intermediary if it is currently started.
- 2 Use a text editor to open `<install_dir>\conf\broker\mipServer.xml`.
- 3 Edit the audit interval and threshold values. For example:

```
<entry name="com.hp.audit.publisher.interval">100000</entry>
<entry name="com.hp.audit.publisher.threshold">10</entry>
```
- 4 Save and close the properties file.
- 5 Restart the Intermediary.

Business Metric Alerts Handler

Business content alerting lets you define a business metric for specific content that is found in the SOAP request and response message for a service (for example, an order is placed with a total that is greater than \$25,000.00). When the business metric value is found, an alert is generated and sent to the Network Services server which notifies alert recipients (email, HP SOA Manager web interface console, and so on). For more information on SOA Manager business content alerting feature, see the “Using Alert Notification” chapter in the *SOA Manager Administrator Guide*.



Business content alerts are processed by the Network Services server and sent to any recipients configured to receive business content alerts. Recipients for business content alerts are viewed in the HP SOA Manager web interface.

Fields

- **Name:** Enter a user friendly name to identify the alert. (for example, HPQ Alert)
- **Operation:** Enter an operation in the service that contains the business content you want to monitor. The XPath expression is applied to the operation. (for example, getInfo)
- **Alert applies to:** Select when you want the intermediary to search for the operation. You can select to search during requests or responses.
- **Expression:** Enter an XPath expression which selects the business content from the operation. For example, `//tns1:InfoRequest/tns1:symbol/text()`. This expression traverses the SOAP message for the InfoRequest node and selects the text found for the symbol child node.
- **Message:** A user friendly message that is sent with the alert. (for example, A `${name}` alert has occurred)
- **Dynamic Properties:** A dynamic variable defined within the message. The Name field corresponds to the variable name. The XPath field corresponds to an XPath expression used to update the variable. For example, **Name:** name **Xpath:** `//s:Envelope/s:Body/t:InfoRequest/t:symbol/text()`. The XPath expression specified here is evaluated on the business content selected by the expression provided in the expression field.
- **Namespace Prefixes:** Any namespace prefixes that appear in the XPath expression (for example, **prefix:** tns1 **URI:** `http://wsm.hp.com/Finance/Request`).

Generic SOAP Contract Handler

The Generic SOAP Contract Handler detects the operation from a request. It can be used to replace the Soap Contract Handler. The handler is commonly used for SOAP services that do not have a WSDL. The handler generates a simple WSDL and does not perform any runtime checks. The handler must be used after decryption and before any handler that requires the operation. The handler only supports a single portType and binding. Operations are set as the runtime soap payload element. When using this handler, no WSDL is required in the Intermediary deployment unit.

Fields

- **namespace:** The target namespace for the generated WSDL
- **name:** The name of the WSDL
- **portType:** The name of the generated portType

- **binding:** The name of the generated binding

HTTP Pass-Through Transport Header Handler

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.

Invocation Handler

The Invocation Handler marshals XML to Java using JAXB and is used to invoke a Java class. The invocation handler is only available for custom intermediary services.

Fields

- **Classname:** The name of the Java class to be invoked
- **Packages:** The package of the Java class

Log Handler

An intermediary service's logging feature lets you indicate whether or not you want faults to be logged to the Intermediary's log file as well as the console. When enabled, log messages are included in the log file and the console. The intermediary's log file is named `broker.log` and is located at `<install_dir>/log`.

Fields

Category: This field is only available for custom intermediary services. This field lets you select a specific log category where log messages are sent. This field is optional.

Schema Validation Handler

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.

Security Auditing

The Security Audit Handler is used to collect security trace information (used for non-repudiation, and so on) and sends the payload to a security provider. For example, when using Select Access to control authorization, the traces can be viewed using the Select Access Audit Report Viewer.

Field

- **Payload Option:** Use this field if you want to constrain the type of message payloads that should be logged. Only payloads for the option selected are captured and sent to the security provider.

Configuring Security Auditing

When using the Security Audit Handler, you must configure the security provider where security trace information will be sent. For Select Access, the security provider will log audit messages to the Enforcer using `SOAP Messages.INFO`. See the Select Access documentation for information on how to configure Audit Policies and Servers.

To configure a security provider, add the following property in `<install_dir>\conf\broker\mipServer.xml`. The following example sets Select Access as the security provider:

```
<entry name="com.hp.mip.security.audit.provider">SelectAccess</entry>
```

Service Security Inbound Handler

The Service Security Inbound Handler performs authorization using the principal and credentials associated with an operation. The authorization is done using a configured security provider such as Select Access. This handler is used in conjunction with, and must come after the WS Security Message Processing Inbound handler. This handler must come before any handler that needs to be protected.

SOAP Contract Handler

The SOAP Contract Handler detects the operation from a request. It is a required handler that must be in every SOAP service. The handler must be used after decryption but before any handler that requires the operation. The handler can only be disabled, or the ordering changed, when using custom intermediary services.



The Generic SOAP Contract Handler can be used to replace the Soap Contract Handler for SOAP services that do not have a WSDL. For more information, see the “Generic SOAP Contract Handler” section above.

SOAP Dispatch Handler

The SOAP Dispatch Handler is used to dispatch a request to the Intermediary's Dispatcher component, which is responsible for forwarding a request to a Web service's endpoint. The handler must be last in the handler chain.

SOAP Monitoring Handler

The SOAP Monitoring Handler is used to decide if a service response is a success or failure. The handler must be used after any handler that requires the outcome and before any handler that might modify the outcome. Matches are based on SOAP fault codes.

Fields

- **Match:** Use the appropriate option to indicate whether the match means a success or a failure or all faults to be considered as failure.
- **Fault Codes:** Use this field to configure a list of codes to match. You must also include namespace for the code. For example, **Code:** `Server` **Namespace:** `http://schemas.xmlsoap.org/soap/envelope/`. When the option chosen for match is “All faults as failure”, fault codes need not be specified.

WS Security Outbound Handler

The Ws Security Outbound Handler provides support for WS-security on outbound messages (from the Intermediary to a Policy enforcement agent). This includes user name/password, signing, and encryption.

Fields

- **authMethod:** The outbound authentication method being used. Valid entries are `Cert`, `UsernameToken`, and `SSOToken`.
- **outMsgEncrypted:** Whether or not the message needs to be decrypted. This field is only relevant when using the `Cert` method. Valid entries are `true` and `false`.
- **outMsgUsername:** Username to be sent with the outgoing message
- **outMsgPassword:** Password to be sent with the outgoing message
- **outMsgSecurityProvider:** Not used
- **outMsgSignBeforeEncrypt:** Whether or not the message will be signed before it is encrypted. If set to `true`, the response is signed and then encrypted. If set to `false`, the response is encrypted and then signed. This field is only relevant when using the `Cert` method and when `outMsgEncrypted` and `outMsgSigned` are enabled.
- **outMsgSigned:** Whether or not the message has a signature that needs to be validated. This field is only relevant when using the `Cert` method. Valid entries are `true` and `false`.
- **relayRouter:** The alias to find the recipient certificate from the keystore.
- **securityProvider:** Not used

WS Security Message Processing Inbound Handler

The WS Security Message Processing Inbound handler provides support for WS-security signing and encryption. This handler is used in conjunction with the Service Security Inbound Handler and must come before any handler that reads the request body.

Both handlers are required because the authorization cannot be performed until the operation being invoked is known, but the handler that detects the operation requires the request to be decrypted first. Decryption and credential extraction is first completed using the WS Security Message Processing Inbound handler. The Soap Contract Handler detects the operation, and then the Service Security Inbound Handler uses the credentials and operation to perform authorization.

Fields

- **inMsgAuthMethod:** The inbound authentication method being used. Valid entries are `Cert`, `UsernameToken`, and `SSOToken`.
- **inMsgEncrypted:** Whether or not the message needs to be decrypted. This field is only relevant when using the `Cert` method. Valid entries are `true` and `false`.
- **inMsgResponseSecurity:** Whether or not the response is to be secured. If set to `false`, the response will not be signed or encrypted. If set to `true`, `inMsgEncrypted` and `inMsgSigned` will apply and the response will be signed and/or encrypted.
- **inMsgSSOEnable:** Not used

- **inMsgSignBeforeEncrypt:** Whether or not the message will be signed before it is encrypted. If set to `true`, the response is signed and then encrypted. If set to `false`, the response is encrypted and then signed. This field is only relevant when using the `Cert` method and when the `inMsgEncrypted` and `inMsgSigned` are enabled.
- **inMsgSigned:** Whether or not the message has a signature that needs to be validated. This field is only relevant when using the `Cert` method. Valid entries are `true` and `false`.

XML Contract Handler

The XML Contract Handler detects the operation from a request. It is a required handler that must be in every XML service. The handler must be used after decryption but before any handler that requires the operation. The handler can only be disabled, or the ordering changed, when using custom intermediary services for XML services.

XML Dispatch Handler

The XML Dispatch Handler is used to dispatch a request to the intermediary's Dispatcher component, which is responsible for forwarding a request to a Web service's endpoint. The handler is used for custom intermediary services for XML services. The handler must be last in the handler chain

XPath Monitoring

The XPath Monitoring handler is used to decide if a service response is a success or failure. The handler must be used after any handler that requires the outcome and before any handler that might modify the outcome. The handler is used for custom intermediary services for XML services. Matches are based on XPath expressions.

Fields

- **Match:** Use the options to indicate whether the match means a success or a failure.
- **XPath:** Use this field to enter an XPath expression used to match.
- **Namespace Mapping:** Enter any namespace prefixes that appear in the XPath expression (**prefix:** `tns1` **Namespace:** `http://wsm.hp.com`).

XSLT Handler

The XSLT Handler runs an XSLT template on the request or response messages. A different template can be assigned for the request and response. The templates must be included in the intermediary service JAR file in order to be loaded by the Intermediary's classloader.

Fields

- **requestTemplate:** The name of the XSLT template to be applied to a request message.
- **responseTemplate:** The name of the XSLT template to be applied to a response message.

Classifier Handler

The classifier handler forwards the requests to a specific endpoint configured to the handler.

Fields

- **Enter New Classifier:** The classifier name.
- **Expression:** An XPath expression for the endpoint of the classifier.
- **Context:** Specifies if the classifier must be used for message or transport.
- **Namespaces:** Any namespaces that you might want to specify.

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. The available handlers are detailed in the "Configuring Handlers" chapter.

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. See Chapter 11 "Configuring Handlers" for a detailed description of each handler.
- 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</TCP_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 Manager 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 Manager 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 not 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
 - 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**.
- 13 Click **Save**.
- 14 Click **Finish**.

Getting Started

This chapter provides detailed instructions for starting and configuring the WSM Broker. The WSM Broker is installed as part of the SOA Manager installation. Before beginning the instructions in this chapter, make sure you have installed SOA Manager following all the instructions in the *SOA Manager Installation Guide*. The directory where you installed SOA Manager is referred to as *<install_dir>* throughout these instructions.

This chapter also provides instructions for configuring the Broker using the Broker's configuration files. The chapter covers common configuration changes and does not include every configuration option. The Broker's configuration files are located in the *<install_dir>\conf\broker* directory of the distribution. The configuration files can be edited with a text editor. In addition, several of the configuration options discussed here can be set using the Broker Configurator.

Starting the WSM Broker

A script for both Windows and UNIX is provided to start the Broker. The script is located in *<install_dir>/bin/win32* and *<install_dir>/bin/unix*, respectively. Windows users can choose to create product icons during installation. If you accepted the default program group during installation, you can start the Broker by clicking **Start | Program Files | HP OpenView | SOA Manager 2.1 | Broker**.



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

To start the WSM Broker:

- 1 Open a command prompt.

- 2 Depending on your platform, change directories to `<install_dir>\bin\win32` or `<install_dir>\bin\unix`.
- 3 Run the “broker” startup script. The console outputs log messages as the broker starts. The broker has started when you see the message:
MIP Server startup completed in # seconds.



If you selected to install the WSM Broker as a Windows service, the Broker may already be running. If you attempt to start the Broker again, an error message is displayed.

Stopping the WSM Broker

The WSM Broker 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 the WSM Broker 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 WSM Broker 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 WSM Broker process and issue the `kill` command to stop the process. For example:

```
kill <process number>
```

Starting the Broker Configurator Console

Typical interaction with the Broker is through its console. The console is a Web application that runs on port 9032. To change the default port, see the “HTTP Settings” section below.

To start the Broker Configurator:

- 1 Start the Broker as described above.
- 2 Open a Browser.

- 3 Enter the following URL and substitute *<host>* with the host name where the Broker Agent is running:

```
http://<host>:9032/console
```
- 4 The login screen already contains default credentials: `admin` is the username and `password` is the password.
- 5 Click **Login**. The Brokered Services screen displays.



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

Installing the Broker as a Windows Service

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

To install the Broker to run 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 broker <install_dir>
```

The service has been successfully installed when the following message is outputted to the console:

```
Service "HP OpenView SOA Manager v2.1 Broker" installed.
```



The script configures the HP SOA Manager 2.1 broker service 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 broker
```

Configuring HTTP Settings

The WSM Broker contains both an HTTP server and an HTTP client. The server is used to accept HTTP requests for Web services and is also used to interact with the Broker Configurator. The HTTP client is used to communicate with HTTP-based servers that are hosting Web services in your environment (i.e., WebLogic server). The HTTP settings allow you to change the behavior of HTTP communication and in some circumstances may help improve the performance of HTTP communication.

This section covers:

- Configuring the HTTP Server Port Number
- Changing the Broker's Management Channel Port
- Configuring the HTTP Server Thread Settings
- Configuring the HTTP Client Settings
- Configuring the HTTP Proxy Settings

Configuring the HTTP Server Port Number

The default port used by the HTTP Server and the Broker Configurator is 9032. If port 9032 is currently being used, the Broker will not start.

To change the port number:

- 1 Stop the Broker if it is currently started.
- 2 Using a text editor, open `<install_dir>\conf\broker\mipServer.xml`.
- 3 Change the port number the `com.hp.http.server.port` entry. For example:

```
<entry name="com.hp.http.server.port">9035</entry>
```
- 4 Save and close the file.
- 5 Restart the Broker.

Configuring the Broker's Management Channel Port

The Broker's management channel port is used to publish the management WSDLs for brokered Web services (i.e., `http://host:9032/wsmf/services`). The management WSDLs are used by the Network Services server to get management data about brokered Web services.

By default, the management channel port is set to port 9032 which is also the application channel port that receives Web service requests. To separate management channel and application channel traffic, change the management channel port.



The management channel port is required when registering a WSM Broker with the Network Service server. If the default port number is changed, make sure that the new port number is known when the WSM Broker is being registered with the Network Service server.

For more instructions on securing the management channel, see the *SOA Manager Administrator Guide*.

To define a different server port for the management channel:

- 1 Stop the Broker if it is currently started.
- 2 Use a text editor to open `<install_dir>\conf\broker\mipServer.xml`.
- 3 Specify a port value for the `com.hp.http.server.managementPort` element. Make sure the port is not being used by any other application on your system. For example:

```
<entry name="com.hp.http.server.managementPort">9033</entry>
```

- 4 Save and close `mipServer.xml`.
- 5 Start the Broker server.

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 change HTTP server thread settings:

- 1 Stop the Broker if it is currently started.
- 2 Use a text editor to open `<install_dir>\conf\broker\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 the Broker.

Configuring HTTP Client Settings

The WSM Broker contains an HTTP client used to communicate to an HTTP server. In particular, the client is used to send requests to and receive responses from the containers that are hosting Web services. The client settings can improve performance between the HTTP client and an HTTP server.

- `<entry name="com.hp.http.client.keepAlive">` – Indicates the HTTP client will reuse a network connection to the server. This usually has performance benefits because the client does not need to keep opening and closing sockets. Typically, the value is set to `true`. Valid values are either `true` or `false`.
- `<entry name="com.hp.http.client.chunking">` – Allows the HTTP client to send data by breaking it into smaller chunks. Chunking information allows the client and server to process large amounts of data without using as much memory. Typically the value is set to `true`. However, some HTTP servers may not support this feature, in which case the value should be set to `false`.

To configure HTTP client settings:

- 1 Stop the Broker if it is currently started.
- 2 Use a text editor to open `<install_dir>\conf\broker\mipServer.xml`.

- 3 Configure the HTTP Server Thread settings. For example:

```
<entry name="com.hp.http.client.chunking">true</entry>  
<entry name="com.hp.http.client.keepAlive">true</entry>
```

- 4 Save and close the file.
- 5 Restart the Broker.

Configuring HTTP Proxy Settings

Many networks use a proxy server that enables access to resources that are external to a network. This is also true for external Web services that are being managed by a Broker. The `Proxy Host` and `Proxy Port` settings allow you to define a proxy server. If set, all requests sent to a brokered service are dispatched to the final endpoint through the proxy server.

However, a proxy server is not required to access addresses that are internal to the network. Therefore, if you are managing Web services that are both internal and external to the network, the `Non-proxy Hosts` setting allows you to define a set of hosts that never require the use of a proxy server.



You do not need to set the `Non-proxy Hosts` setting if you do not define a proxy server.

To configure the HTTP proxy settings:

- 1 From the Broker Configurator's main toolbar, click **HTTP Settings**. The **HTTP Settings** screen displays.
- 2 Use the `Proxy Host` and `Proxy Port` text boxes to enter a proxy server's host and port. The host value must be an IP address or the full DNS name of the server.
- 3 Use the `Non-proxy Hosts` text box to enter a list of hosts that do not require the use of a proxy server. Use the pipe character (|) to separate entries. For example:

```
localhost | 15.* | 16.* | 127.*
```

The local host and any hosts in the 15, 16, and 127 domain space do not require the proxy server.

- 4 Click **Save** to save your changes.

Assigning Access to the Console

The `<install_dir>\conf\broker\mipServer.xml` file allows you to define user credentials for accessing the Broker's console. In particular, you can define usernames and passwords for accessing the console. A single role, `admins`, has been implemented. All users must be associated with this role.



The SOA Manager also integrates with **Select Access**, which can be used to secure access to the Broker's console. See Chapter 9 "Integrating with Select Access" in the *SOA Manager Administration Guide* for more information.

To add console access rights for a user:

- 1 Stop the Broker if it is currently started.
- 2 Using a text editor, open `<install_dir>\conf\broker\mipServer.xml`.
- 3 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>
```
- 4 Save and close the file.
- 5 Restart the Broker.



You can use the Broker Configurator to change a user's password. You must be logged into the Configurator as the user in order to change the password. See the *Broker Configurator Online Help* for detailed instructions.

Using XPL Logging

SOA Manager uses HP OpenView Cross Platform (XPL) logging. Installation, configuration, and usage are described below.

Installing XPL Logging

During the SOA Manager installation, you may be prompted to select the HP OpenView installation and data directories. You will only be prompted for this information if this is the first time you have installed an HP OpenView product.

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

XPL Tools

The HP OpenView 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 Manager CD.

Configuring XPL

The Broker automatically creates log files in the log subdirectory of the HP OpenView data directory. The Broker log file name has the format:

broker[unique].sequence.locale

For example:


```
broker0.0.en_US
```

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

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

The Broker creates up to 10 log files, each file containing up to 1 megabyte of data. The log files will 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 OpenView Cross Platform tool, `ovconfchg`. After installing the The HP OpenView Cross Platform Component, this program is in the bin directory of the HP OpenView installation directory. An example of using this tool is shown below.

```
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 the Broker for the new configuration to take effect.

You can see the current configuration using this command:

```
ovconfget
```

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

Configuring Log Levels

You can change the Broker log levels using the BSE. Alternatively, you can change the log levels by editing the `logging.properties` file in the JDK lib directory or the `xpllogging.properties` in the `<install_dir>/conf/broker` directory. The log levels are: SEVERE, WARNING, INFO, FINE, FINER, and FINEST. By default the log level is set to INFO.

Using the BSE

To change a Broker's log levels from the BSE:

- 1 From the BSE main tool bar, click **IT Services**. The IT Services screen displays and the Summary tab is selected.
- 2 From the list of **WS Intermediary Services**, click the IT service you want to view. The WS Intermediary Service view screen displays.
- 3 From the Contained Resources section, click the resource you want to view. The Resource View screen displays.
- 4 Click **Edit/Query Log Levels**.
- 5 Specify MIP for the logger. You can also set the log level for individual packages. The Broker packages begin with `com.hp.ov.mip`.

Using JRE Properties File

You can change the log level for the Broker by editing the `logging.properties` file in the JRE lib directory. You must restart Network Services and the Broker to make the changes take effect. For example, you can add the following line to the end of

`logging.properties`:

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

This sets the log level for the Broker to `FINE`.

Using the XPL Properties File

You can change the log level for the Broker by editing the `xpllogging.properties` in the `<install_dir>/conf/broker` directory. You must restart the Broker 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 Broker to `FINE`.

Viewing Logs

You can use an editor or the BSE to view the Broker log files. In the BSE, go to a Broker's Resource View screen and click the **View Log** link. Alternatively, use an editor to view the Broker log files in the HP OpenView data log directory.

Using XPL Tracing

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

Installation

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

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

Windows

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

HP-UX

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

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

Linux

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

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

Example Configuration Entries

The following SOA Manager 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
```

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

Load balancing and failover share the same common architecture shown in **Error! Reference source not found.** 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 the HTTP service that is provided by your Policy enforcement agent supports the `Retry-After` HTTP header property, the Intermediary uses the interval specified by this property instead of the 30 second retry interval.



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 HP SOA Manager web 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.

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. If you are using a WSM deployment scenario that uses the WSM Agents (J2EE Agent or .NET Agent), then you can implement the security features natively provided by the Policy enforcement agent (WLS or IIS). 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 HTTPS: X.509 certificates HTTP/S: SSO tokens Select Access	WS-Security: User password WS-Security: X.509 certificates WS-Security: SSO tokens Select Access
Authorization	Select Access	Select Access
Confidentiality	SSL	WS-Security: XML-Encryption
Integrity	SSL	WS-Security: D-Sig
Auditing	SOA Manager, Select Access	SOA Manager, Select Access
Non-Repudiation	SOA Manager Audit Service (using D-Sig), Select Access Audit Server	SOA Manager Audit Service (using D-Sig), Select Access Audit Server
Administration	Select Access	Select Access

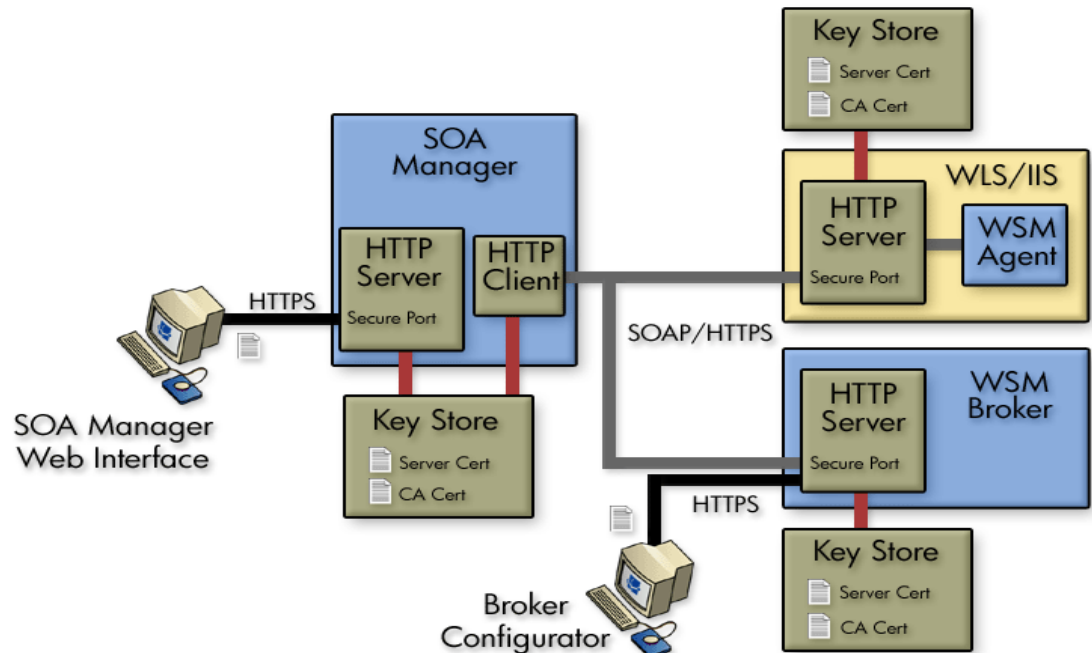
- All User Identity Management – authentication, authorization, and administration is deferred to enterprise security products. SOA Manager currently integrates with HP OpenView Select Access.
- WS-Security implementation in the Intermediary (D-Sig, Encryption) is done using Verisign TSIK toolkit.
- 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.

Error! Reference source not found. below shows a high level view of the Intermediary security implementation and includes all three scenarios.



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 such as Select Access.

Authentication/Authorization failures are tracked and sent to the SOA Manager 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.

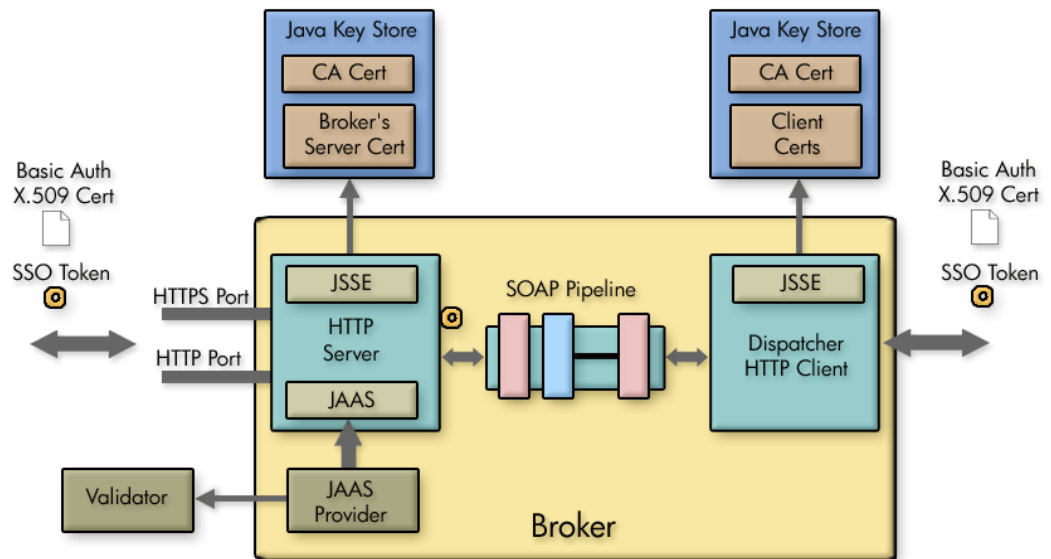
Scenario 3: Intermediary is the Exit Point for External Providers

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 Manager 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. **Error! Reference source not found.** shows a common view of transport level security.



Message Level Security

Message level security is offered using SOAP handlers. Figure 17-1 shows a common view of message level security.

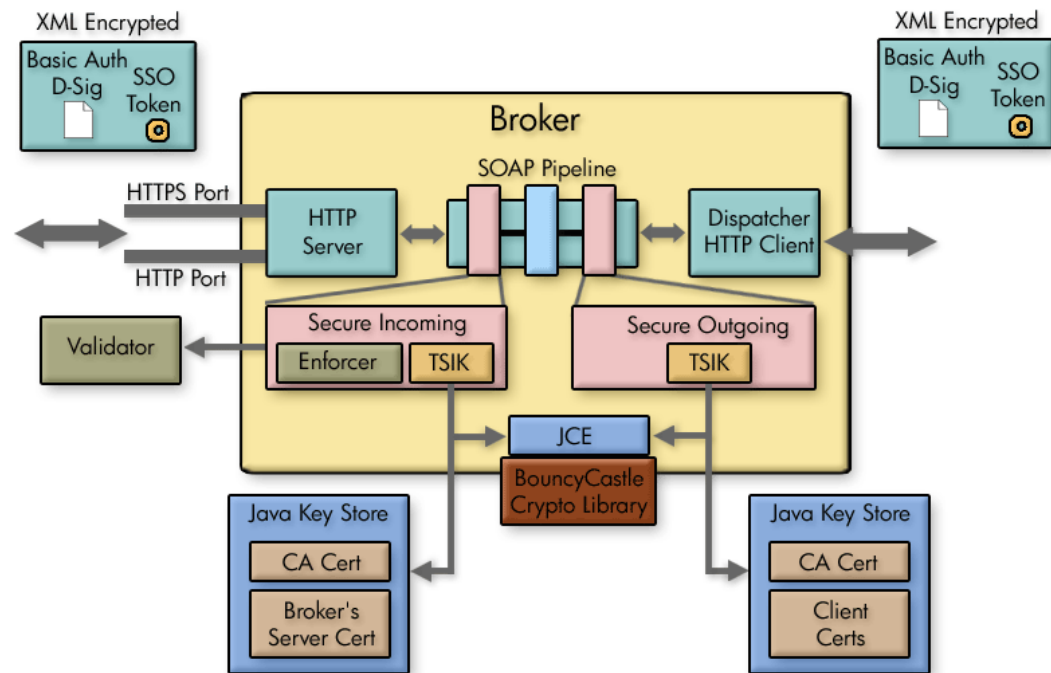


Figure 17-1: Message Level Security

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 Manager 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 Select Access 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. At the present time integrations are provided with Hewlett-Packard's Select Access

Using Select Access

The Intermediary security solution supports integration with Select Access. Select Access provides an identity management solution for securing access to Policy enforcement points and resources. To complete the Select Access instructions in this section, you must have:

- Completed the Select Access integration instructions in Chapter 8 of the *SOA Manager Administrator Guide*
- A general understanding of the Select Access Policy Builder
- General understanding of the WSM solution
- A WSM Intermediary deployed and running
- Access to a Web service endpoint deployed and running

Setting Up Basic Authentication Only

The instructions in this section are used to set up Select Access to provide Basic Authentication Only for users that are accessing resources on the Intermediary's application channel. The instructions use the Select Access Administration console. The Select Access Administration console is used to define resources that are to be secured as well as create policies and permissions for those resources. If you are not familiar with Select Access, you may need to consult the Select Access documentation while completing some of the instructions in this section.

Define a Select Access Resource Server for the Intermediary

You must create a Select Access resource server that is mapped to the Intermediary server. This Select Access resource server is used to control access to the server using basic authentication only. The resource server contains the protocol, host name, and port number of the Intermediary.

To define a Select Access resource server for the Intermediary,

- 1 From the Select Access Policy Builder Resources Tree, right-click Resource Access and select **New | Folder**. The New Folder dialog box displays.
- 2 In the Name field, enter a name for the folder.
- 3 Click **OK**. When asked to clear the Policy Validator cache, select **OK**. The folder is created and is added to the Policy Builder Resources Tree under Resource Access.
- 4 Right-click the newly created folder and select **New | Resource Server**. The New Resource Server dialog box displays.
- 5 In the Name box, enter a name for this new resource server (i.e., Intermediary BA_Only). Any name that clearly identifies the server can be used.
- 6 On the bottom of the window click **Add**. A new entry displays under the Servers section.

New Resource Server

Enter a name for the resource server and specify the protocols or server(s) used. The resource server's location on the Resources Tree is shown in the Location field.

Name:

Location:


Character Set:

Servers:

R...	Protocol	Hostname	Port #
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

- 7 Enter the following server information for the server where the Intermediary is located:

- **Protocol:** The protocol used to access the Intermediary (HTTP)
- **Hostname:** *<intermediary_host_name>*
- **Port #:** The Intermediary 's port number used for HTTP connections (9032)

 Only one server is added to this resource server and must match the settings in the `selectaccess.properties` file.

- 8 Click **OK** to save this resource server. When asked to clear the Policy Validator cache, select **OK**. The resource server is listed in the Policy Builder Resources Tree.

Mapping a Select Access Resource for Basic Authentication Only

You must create a Select Access resource that is mapped to the Intermediary server's Select Access security settings. This Select Access resource is used to control access to the server using Basic Authentication only for HTTP transport level security.

A property entry in `selectaccess.properties` sets the authentication resource path in Select Access:

```
#SelectAccess service authentication resource path
AuthenticationResource = /authentication
```

To define a Select Access Resource for Basic Authentication Only:

- 1 From the Policy Builder Resources Tree, select the resource server that contains the Intermediary Server that you want to secure using Basic Authentication Only.
- 2 Right-click on the resource server, and select **New | Resource** from the menu. The New Resources dialog box displays.
- 3 In the Name field, enter the authentication-only resource path as pre-defined in the AuthenticationResource property in the selectaccess.properties file. For example, authentication.
- 4 Click **OK**. When asked to clear the Policy Validator cache, select **OK**. The resource is listed in the Policy Builder Resources Tree under the appropriate intermediary resource server. Any requests to the Intermediary are authenticated but no authorization is performed.

Setting up Basic Authorization

The instructions in this section are used to set up Select Access to provide basic authorization for users that are accessing resources on the Intermediary's application channel. The instructions use the Select Access Administration console. The Select Access Administration console is used to define resources that are to be secured as well as create polices and permissions for those resources. If you are not familiar with Select Access, you may need to consult the Select Access documentation while completing some of the instructions in this section.

Define a Select Access Resource Server for the Intermediary

You must create a Select Access resource server that is mapped to the Intermediary server. This Select Access resource server is used to control access to the server using basic authorization. The resource server contains the protocol, host name, and port number of the Intermediary.

To define a Select Access resource server for the Intermediary:

- 1 From the Select Access Policy Builder Resources Tree, right-click Resource Access and select **New | Folder**. The New Folder dialog box displays.
- 2 In the Name field, enter a name for the folder.
- 3 Click **OK**. When asked to clear the Policy Validator cache, select **OK**. The folder is created and is added to the Policy Builder Resources Tree under Resource Access.
- 4 Right-click the newly created folder and select **New | Resource Server**. The New Resource Server dialog box displays.
- 5 In the Name box, enter a name for this new resource server (Intermediary_BA). Any name that clearly identifies the server can be used.
- 6 On the bottom of the window click **Add**. A new entry displays under the Servers section.

- 7 Enter the following server information for the server where the Intermediary is located:

- **Protocol:** The protocol used to access the Intermediary (HTTP)
- **Hostname:** < *intermediary_host_name* >
- **Port #:** The Intermediary 's port number used for HTTP connections (9032)

▶ Only one server is added to this resource server and must match the settings in the basic authentication section in the `selectaccess.properties` file.

- 8 Click **OK** to save this resource server. When asked to clear the Policy Validator cache, select **OK**. The resource server is listed in the Policy Builder Resources Tree.
- 9 Use the Identities Tree to apply the basic authorization rules for users of this resource server.

Setting Up X.509 Authorization

The instructions in this section are used to set up Select Access to provide x.509 authentication for users that are accessing resources on the Intermediary's application channel. The instructions use the Select Access Administration console. The Select Access Administration console is used to define resources that are to be secured as well as create polices and permissions for those resources. If you are not familiar with Select Access, you may need to consult the Select Access documentation while completing some of the instructions in this section.

Define a Select Access Resource Server for the Intermediary

You must create a Select Access resource server that is mapped to the Intermediary server. This Select Access resource server is used to control access to the server using certificate-based authorization. The resource server contains the protocol, host name, and port number of the Intermediary.

To define a Select Access resource server for the Intermediary:

- 1 From the Select Access Policy Builder Resources Tree, right-click Resource Access and select **New | Folder**. The New Folder dialog box displays.
- 2 In the Name field, enter a name for the folder.
- 3 Click **OK**. When asked to clear the Policy Validator cache, select **OK**. The folder is created and is added to the Policy Builder Resources Tree under Resource Access.
- 4 Right-click the newly created folder and select **New | Resource Server**. The New Resource Server dialog box displays.
- 5 In the Name box, enter a name for this new resource server (Intermediary_Cert). Any name that clearly identifies the server can be used.
- 6 On the bottom of the window click **Add**. A new entry displays under the Servers section.

Enter a name for the resource server and specify the protocols or server(s) used. The resource server's location on the Resources Tree is shown in the Location field.

Name:

Location:

Character Set:

Servers:

R...	Protocol	Hostname	Port #
<input checked="" type="checkbox"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>

Add Delete

OK Cancel Help

- 7 Enter the following server information for the server where the Intermediary is located:
 - **Protocol:** The secure protocol used to access the Intermediary (HTTPS)
 - **Hostname:** *<intermediary_host_name>*
 - **Port #:** The port number used for SSL connections (i.e., 443)

▶ Only one server is added to this resource server and must match the settings in the certificate-based authentication section in the `selectaccess.properties` file.

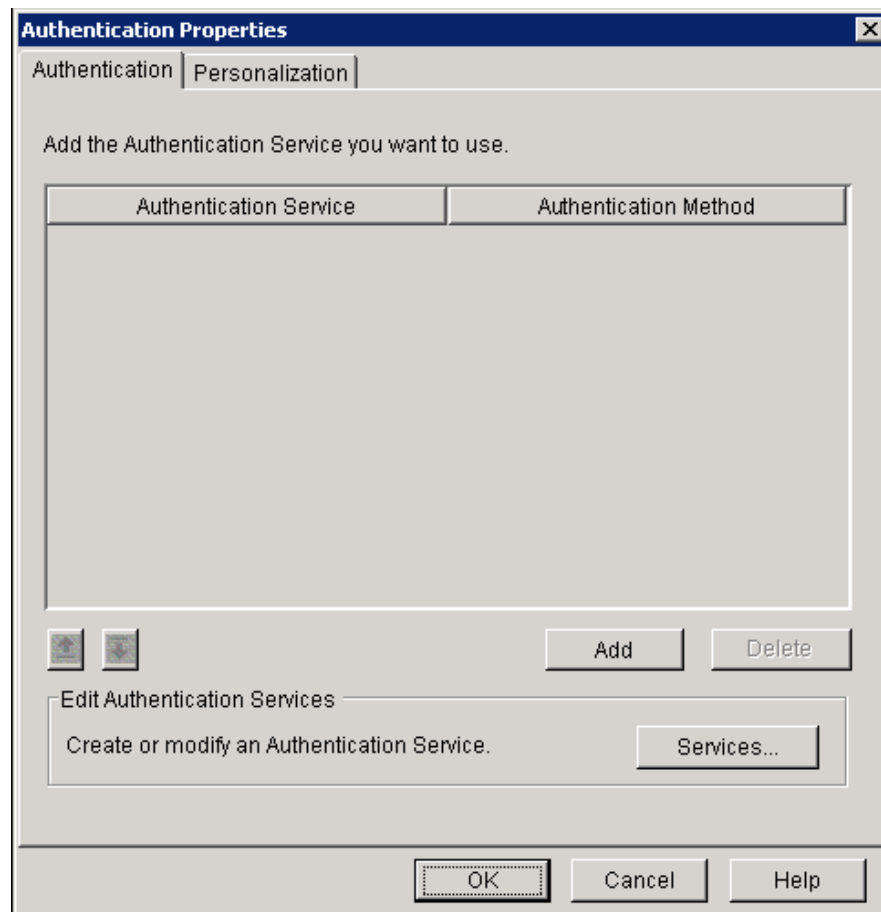
- 8 Click **OK** to save this resource server. When asked to clear the Policy Validator cache, select **OK**. The service is listed in the Policy Builder Resources Tree.
- 9 Use the Identities Tree to apply the x.509 rule for users of this resource server.

Enable Select Auth for Basic Authorization and X.509 Certificate Authorization

The Policy Builder Identities Tree contains a Select Auth column that must be enabled for each intermediary resource server for which you want to enable authorization.

To enable Select Auth for Basic and X.509 authorization:

- 1 From the Policy Builder Identities Tree, right-click the first column on the same row as the resource server for Basic Authorization, select **Enable Select Auth** from the pop-up menu. The Authentication Properties dialog box displays.



- 2 Click **Add**. The Available Authentication Services dialog box displays.
- 3 Select the **password** authentication service and click **Add**. The service is listed in the Selected Services column.

- 4 Click **OK**. The Authentication service is added to the list of authentication services in the Authentication Properties dialog box.
- 5 Click **OK**. When asked to clear the Policy Validator cache, select **OK**. The Select Auth icon shows that Select Auth for the selected resource server is enabled.
- 6 Repeat this procedure for a resource server for X.509 Certificate Authorization.

Mapping Resources in Select Access

A Select Access resource server provides authorization on a global basis. In other words, for every intermediary service configured within the Intermediary to use authorization, the resource server is used to authorize users. These services have the same group of consumers authorized to use them.

There may be circumstances where you want to identify a specific consumer or group of consumers to be authorized to use specific intermediary Web services or specific intermediary Web service operations. To do this, you must create resources in Select Access that maps to the intermediary Web services or intermediary Web service operations in the Intermediary and then apply user permissions to the resources.

To map resources in Select Access:

- 1 Login to the Select Access Policy Builder.
- 2 From the Policy Builder Resources Tree, select the resource server that contains the resources that you want to secure. Make sure the server you select represents the authorization type you are using (basic authorization or certificate authorization).
- 3 Right-click on the resource server, and select **New | Resource** from the menu. The New Resources dialog box displays.
- 4 In the Name field, provide a name for this resource which maps to a name of a resource on the Intermediary. The name is different depending on whether you are implementing transport-level or message-level security. In addition, if you are securing a specific operation, the name must include the operation name.
 - If you are authorizing at the transport level, the name must match the intermediary service's URL path as defined in the HTTP header. For example, if an intermediary service has the URL:
`HTTP://<host_name>:<port>/FinanceServiceProxy`, then the resource name would be `FinanceServiceProxy`. If you do not know the URL used to access to the intermediary service, you can use the Broker Configurator's Service detail screen to get an intermediary service's URL value.

Web services are typically associated with a port binding that is used as part of the URL which is passed in the message header. For example:

```
HTTP://<host_name>:<port>/FinanceServiceProxy/  
FinanceServiceSoapBinding
```



If your client invokes an intermediary service using the port binding in the address, you must include the binding in the name for the resource. For the above example, the resource name would be `FinanceServiceSoapBinding`. This URL is defined in the WSDL for an intermediary service. You can use the Broker Configurator's Brokered Services screen to view a brokered service's WSDL. The URL is typically located near the bottom of the WSDL.

- If you are authorizing at the message level, the resource name must match the Web service name that is defined in the WSDL for the intermediary service. If you are not sure of the Web service name, you can use the Broker Configurator's Brokered Services screen to view the WSDL for an intermediary service. The service name is typically located towards the bottom of the screen in the `<wsdl:service>` element.

- 5 Click **OK**. The resource is saved and is listed on the Resources Tree.
- 6 Use the Users Tree policy matrix to apply access permissions to this resource.
- 7 Repeat this procedure to create and map additional resources.

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 Policy enforcement agent.



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 such as Select Access. 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. Refer to **Error! Reference source not found.** for a conceptual architecture of transport-level security.

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 such as Select Access (See “Using Select Access” above). 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. Select Access is used to verify the credentials and which resources can be accessed.

- **X.509 Client Certs:** All requests to the Intermediary need to be authenticated using an X.509 certificate. Select Access is used to verify the credentials and which resources can be accessed.
- 3 At the bottom of the screen, click **Save Changes**. Once this service is deployed, the Intermediary will communicate with Select Access 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. Refer to **Error! Reference source not found.** for a conceptual architecture of transport-level security.

 When using outbound SSL Security, a Web Service deployed in a Policy enforcement agent must be configured to use SSL from within that Policy enforcement agent. See your Policy Enforcement Agent documentation for more instructions on setting up SSL communications.

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 17-1 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 such as Select Access (See “Using Select Access” above) 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. Select Access is used to verify the credentials and which resources can be accessed.
 - **Digital Signature Authentication:** All messages to the Intermediary need to be authenticated using a digital signature. Select Access is used to verify the credentials and which resources can be accessed.
 - **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. Select Access is used to verify the credentials and which resources can be accessed while the Intermediary's Key Store is used to manage the private key used for decryption.
- 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 Select Access 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

In this scenario, The Intermediary must authenticate itself with a Policy enforcement agent before messages are processed at the Policy enforcement agent. The Policy enforcement agent and the Intermediary can share the same security provider or a Policy enforcement agent's security provider is used to complete the authentication. In addition, the Intermediary can encrypt messages before sending them to the Policy enforcement agent; in which case, the Policy enforcement agent must be able to decrypt the messages. Refer to Figure 17-1 for a conceptual architecture of message-level security.

The Broker Configurator is used to configure an outbound message security handler for a brokered service. Requests are authorized using a Policy enforcement agent's security provider and encryption 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 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 Click the security option you want to enable:
 - **Username-Password Authentication:** All messages dispatched to a Policy enforcement agent need to be authenticated using a user name and password. The Policy enforcement agent's security provider is used to verify the credentials and which resources can be accessed. Enter a valid Username and Password for your Policy enforcement agent in the fields provided.
 - **Sign:** All messages dispatched to a Policy enforcement agent will include a digital signature. The Intermediary's Key Store is used to sign the outbound message.
 - **Sign and Encrypt:** All messages dispatched to a Policy enforcement agent will include a digital signature and will be encrypted. The Intermediary's Key Store is used to sign the outbound message. In addition, the Intermediary's private key must be located at the Policy enforcement agent to decrypt the message.
- 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.

For example:

A Intermediary is running on `Myhost` and its management channel is running on non-secure port 9035. The security provider, `SelectAccess`, sets up web access control for any resources under `http://Myhost:9035/wsmf/`. User `jsmith`, with password, `johnspassword`, is authorized to access these Web resources. The values of the three entries are set to:

```
<entry name="com.hp.mip.security.provider.management">
    SelectAccess</entry>
<entry name="com.hp.mip.security.sba.user">jsmith</entry>
<entry name="com.hp.mip.security.sba.password">
    johnspassword</entry>
```

- 4 Save and close `mipserver.xml`.
- 5 Start the Intermediary server.

Introduction to WSM .NET Agent

The WSM .NET Agent is a software technology for managing Microsoft .NET Web services on multiple machines and integrating with the enterprise management platform.

Prerequisites

Users of this guide should be familiar with Web services management in general and with the .NET platform specifically.

Contextual Overview

The major components of the .NET Agent are:

- Control point – wsmflibs.dll
- Web services that implement WS-based management protocols – wsmf.asmx
- Set of sample Web services

Figure 18-1 provides a high-level overview of the architecture and context of the .NET Agent.

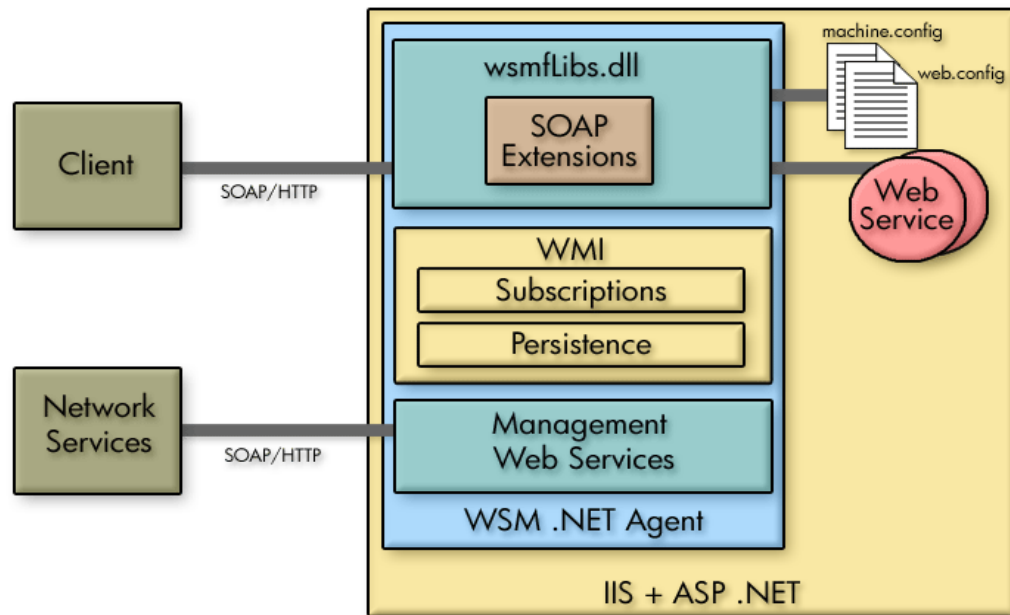


Figure 18-1: Contextual Overview

Control Point

The Control Point for the .NET Agent is wsmfLibs.dll. The DLL lives in the Global Assembly Cache (GAC) and has the SoapExtension component. The DLL integrates with IIS and monitors incoming and outgoing Web service messages at a managed machine and collects runtime performance metrics.

WS Management Web Services

The management Web services provide management information and performance metrics through Web service interfaces. The interfaces allow clients to subscribe/unsubscribe to management events and notifications as well as discover all the Web services on a managed machine.



The SOA Manager currently implements the Web Service Management Framework (WSMF), which is an HP-authored precursor version of standard WS-based management protocols.

Runtime Description

The following table provides a description of what happens when a web service request enters a .NET Agent managed machine.

Phase	Step	Description
Monitoring	1	Web service message enters a machine that has a .NET Agent installed.
	2	Message passes through Control Point, which records its arrival time.
	3	Message is processed by its target Web service and then sent back to the original sender.
	4	On its way out, it again passes through Control Point which records its departure time and collects other management information, like request and response sizes, receiver URL, transport protocol etc. The Control Point also calculates the response time for the Web service.
	5	The Control Point is able to recognize failed request and send auditing information including the payload if a SOA Manager business service subscribes for those features.
	6	The Control Point keeps tracking outgoing Web services calls to another Web service to maintain relationships and dependencies in the WMI.
Management	1	On the management side, when an operator runs the Business Service Explorer, it communicates with the .NET Agent through the management Web services. Using the Business Service Explorer, a user can get data about the Web services being managed on the machine, such as their response time and failure statistics.
	2	The operator can also use the Business Service Explorer to subscribe and receive events.

Installing the WSM .NET Agent

The WSM .NET Agent must be installed on every machine that is hosting Web services that you want to manage. The .NET Agent software is located on the HP Service Oriented Architecture (SOA) Manager installation CD. Make sure you have the CD before beginning the instructions in this chapter.

- ▶ The .NET Agent has been tested on Windows 2003 Server.

Software Requirements

The following software is required on a system before the WSM .NET Agent can be used.

- Microsoft .Net Framework version 1.1.4322. You can download this from: <http://msdn.microsoft.com/netframework/downloads/>
- Microsoft Internet Information Services (IIS) version 5+. You can download this from: <http://www.microsoft.com/iis/>

- ▶ If you install the Microsoft .NET framework before installing IIS, you must run `<winnt>\Microsoft .NET\Framework\v1.1.4322\aspnet_regiis.exe` to register version 1.1 of the .NET framework before you install the WSM .NET agent. (By default, IIS is registered to run .NET 1.0.)

Pre-Installation Steps

The .NET framework must be configured with the correct proxy settings if you intend to use a Web proxy inside your corporate firewall.

To configure a Web proxy:

- 1 Using a text editor, open `machine.config` located in `<winnt>\Microsoft .NET\Framework\v1.1.4322\CONFIG`
- 2 Set the `usesystemdefault` attribute for the `<proxy>` element to `false`.
- 3 Set the `proxyaddress` attribute to the name of your proxy server including the port number. For example:

```
<proxy usesystemdefault="false"  
      proxyaddress="http://MyProxyServer:1234"  
      bypassonlocal="true" />
```

Running the Installer

Before installing, make sure all applications are closed and ensure that all the prerequisite software has been installed.



The WSM .NET Agent is licensed. During the installation, an evaluation license is installed, and these components may be evaluated for 60 days. Thereafter, a license must be purchased. For information on licensing, see the “Entering a SOA Manager License” section in the *SOA Manager Installation Guide* after completing the installation instructions in this section.

To run the installer:

- 1 Log in to the system as a root/admin user.
- 2 Place the SOA Manager CD into the CD ROM drive.
- 3 Exit the Auto Run screen.
- 4 From a command prompt, change directories on the CD to `\Autopass\win`.
- 5 Run the following command:

```
msiexec.exe /i HPOvLic-WinNT4.0-release.msi /q /Lv  
<path>/AutoPass_install.log INSTALLSTANDALONE=1
```

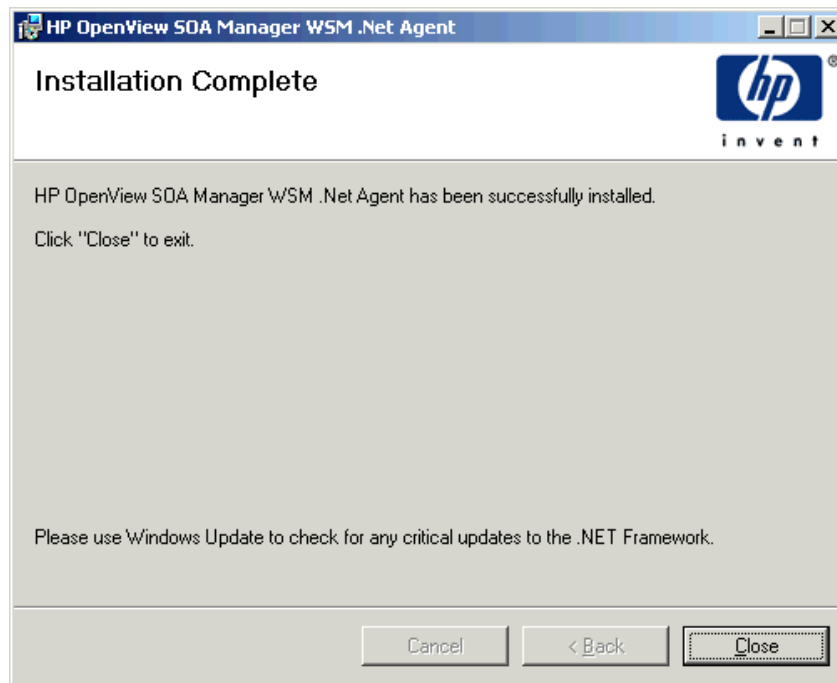
AutoPass is installed and the `AutoPass_install.log` is written to the directory indicated for `<path>`.

- 6 From the command line, change directories on the CD to `\WSM_Agents\DotNet`.
- 7 Double click on `HPWsmAgentInstall.msi`.

The installation wizard start screen displays.



- 8 Click **Next** to start the installation process. Follow the on-screen instructions to complete the installation.
- 9 After the installation has completed, the Installation Complete screen displays. Click **Close** to complete the installation.



Verifying the Installation

There is a quick method of verifying that .NET Agent was successfully installed and is managing the Web services environment.

To verify the installation:

- 1 Open an Internet browser.
- 2 Enter the following address:

`http://<fully-qualified machine name>/hpwsm/wsmf.asmx?http://<non-qualified machine name>/wsee`

Replace *<fully-qualified machine name>* with the name of the host that is running the IIS instance where the WSM .NET Agent was installed.

- 3 A management WSDL for your Web services environment displays:

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="WLS" targetNamespace="http://www.hp.com/mso/mip/2004/02/Wsee"
  xmlns:Events="http://schemas.hp.com/wsmf/2003/03/Events#"
  xmlns:Foundation="http://schemas.hp.com/wsmf/2003/03/Foundation"
  xmlns:WsExecutionEnvironment="http://schemas.hp.com/wsmf/2003/03/WsExecutionEnvironment"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.hp.com/mso/mip/2004/02/Wsee"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:import location="http://15.40.235.105:7001/wsmf/wsdl/WS-Events.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/Events#" />
  <wsdl:import location="http://15.40.235.105:7001/wsmf/wsdl/WSMF-Foundation.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/Foundation" />
  <wsdl:import
    location="http://15.40.235.105:7001/wsmf/wsdl/WsExecutionEnvironment.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/WsExecutionEnvironment" />
  <wsdl:types>
  - <xsd:schema elementFormDefault="qualified"
    targetNamespace="http://www.hp.com/mso/mip/2004/02/Wsee"
    xmlns:ManagementNotification="http://schemas.hp.com/mip/Events/ManagementNotification"
    xmlns:tns="http://www.hp.com/mso/mip/2004/02/Wsee"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:import
      namespace="http://schemas.hp.com/mip/Events/ManagementNotification"
      schemaLocation="http://15.40.235.105:7001/wsmf/schema/MIP-Events.xsd" />
    - <xsd:element name="DeployService">
      - <xsd:complexType>
        - <xsd:sequence>
          <xsd:element name="param0" type="xsd:string" />

```

Uninstalling the WSM .NET Agent

To uninstall the WSM .NET Agent


- 1 Click **Start** | **Settings** | **Control Panel**.
- 2 Click **Add/Remove Programs**.
- 3 Select **HP SOA Manager WSM .NET Agent**.
- 4 Click **Remove**.

Configuring SOAP Extensions

This chapter provides instructions for configuring audit, business metric alerts, and logging. These management features are implemented using SOAP extensions. The chapter includes general instructions for creating service configurations and then provides a reference for each of the SOAP extensions.

Overview

The .NET Agent uses SOAP extensions to add manageability to a Web service. Default SOAP extensions are applied to a managed Web service when the service is discovered by the Agent. However, the SOAP extensions for auditing, business metric alerts and logging are not used by default and must be configured in order to take advantage of their management features. These SOAP extensions are configured in either a .NET application's `Web.config` file or in the .NET server's `Machine.config` file. These files are both XML-based files and can be edited using an XML editor or a text editor.

 Always back up the `web.config` or `machine.config` files before making any changes to the files. If you are not familiar with using .NET configuration files, see the .NET documentation included with the .NET SDK.


Creating a Service Configuration

Service configurations are used to configure SOAP extensions. Service configurations are added to either the `Web.config` file or the `Machine.config` file. There are two types of service configurations that are discussed in this section:

- **Service Specific Configurations:** A method for configuring extensions for specific Web services.
- **Default Service Configurations:** A method for configuring all Web services in an application or all Web services in the .NET Server.

Registering Configuration Handlers

For each type of configuration (either specific or default), you must register the .NET Agent's configuration handlers: `ServiceConfigurationHandler` and `DefaultServiceConfigurationHandler`. The handlers are contained in the Agent's `WsmfLibs` .NET Assembly and are used to gather management data from the SOAP Extensions.

 Whether you are using `Machine.config` or `Web.config`, the Agent's configuration handler definition must precede any SOAP Extension definitions.


To register the .NET Agent's Configuration handlers:

- 1 Using a text editor, open `Machine.config` or `Web.config`.
- 2 Create a `<configSections>` node within the `<configuration>` node. The `<configSections>` must be the first child element under `<configuration>`. For example:

```
<configuration>
  <configSections>
```

- 3 Add a `<section>` node for `ServiceConfigurationHandler` and `DefaultServiceConfigurationHandler` as demonstrated below:

```
<configuration>
  <configSections>
    <section name="defaultServiceConfig"
      type="WsmfLibs.Config.DefaultServiceConfigurationHandler,
      WsmfLibs, Version=1.0.334.40457, Culture=neutral,
      PublicKeyToken=ca89dbe4f7ff13b8" />
    <section name="services"
      type="WsmfLibs.Config.ServiceConfigurationHandler,
      WsmfLibs, Version=1.0.334.40457, Culture=neutral,
      PublicKeyToken=ca89dbe4f7ff13b8" />
  </configSections>
</Configuration>
```

 The `Version` and `PublicKeyToken` attributes for the `WsmfLibs` Assembly is different for each release of the .NET Agent. To find these values, use the .NET configuration tool (**Control Panel | Administrative Tools | Microsoft .NET Framework 1.1 Configuration**) and select **Assembly Cache**. From the Assembly list, double-click **WsmfLibs** to view its properties. The Properties window contains the above attribute values.

- 4 Save the file.

Using Service Specific Configurations

A service specific configuration allows you to configure SOAP extensions for specific Web services. When using this configuration method, a `<services>` node is added to the `Web.config` or the `Machine.config` file. The `<services>` node must be contained within the `<configuration>` node. Within the `<services>` node, a `<service>` node is added that contains a name attribute that identifies the Web service's name. Multiple `<service>` nodes can be used for each service. SOAP extensions are configured within the `<service>` node. For example:

```
<configuration>
  <configSections>
    ...
  </configSections>
  <services>
    <service name="FinanceService.asmx">
      <audit payload-option="REQUEST-RESPONSE"
        payload-filter="ALL" />
    </service>
    <service name="FinanceService1.asmx">
      <audit payload-option="REQUEST-RESPONSE"
        payload-filter="ALL" />
      <logging>
        <category>INFO</category>
      </logging>
    </service>
  </services>
</configuration>
```

Using a Default Service Configuration

A default service configuration allows you to configure SOAP extensions for all Web services in an application or all Web services that are hosted in the .NET server. When a default service configuration is added to an application's `Web.config` file, SOAP extensions are configured for all Web services in the application. When a default configuration is added to the .NET server's `Machine.config` file, SOAP extensions are configured for all Web service deployed in the server. Default service configurations are easier to maintain since they only have to be defined once.



Service specific configurations override default service configurations.

The structure for the default service configuration is exactly the same as that for a specific service except the `<defaultServiceConfig>` element replaces the `<services>` element. For example:

```
<configuration>
  <configSections>
    ...
  </configSections>
  <defaultServiceConfig>
    <service>
      <audit payload-option="REQUEST-RESPONSE"
        payload-filter="ALL" />
      <logging>
        <category>INFO</category>
      </logging>
    </service>
  </defaultServiceConfig>
</configuration>
```

```

    </service>
  </defaultServiceConfig>
</configuration>

```

Verifying Extension Types

SOAP extensions have a corresponding type definition that must be located in the `Web.config` or the `Machine.config` file. The type definitions are automatically added to the `Machine.config` file during the .NET Agent installation. However, it is good practice to verify that the type definitions are included in the file. The SOA Manager server will not be able to retrieve management information from the .NET Agent unless the extension types are defined.

```

<soapExtensionTypes>
  <add type="WsmfLibs.MonitoringModule.WsmfSoapExtension,
    WsmfLibs, Version=2.10.3.4113, Culture=neutral,
    PublicKeyToken=ca89dbe4f7ff13b8" priority="1" group="0" />
  <add type="WsmfLibs.Handlers.AuditHandler, WsmfLibs,
    Version=2.10.3.4113, Culture=neutral,
    PublicKeyToken=ca89dbe4f7ff13b8" priority="1" group="0" />
  <add type="WsmfLibs.Handlers.LoggingHandler, WsmfLibs,
    Version=2.10.3.4113, Culture=neutral,
    PublicKeyToken=ca89dbe4f7ff13b8" priority="1" group="0" />
  <add type="WsmfLibs.Handlers.BizMetricHandler, WsmfLibs,
    Version=2.10.3.4113, Culture=neutral,
    PublicKeyToken=ca89dbe4f7ff13b8" priority="1" group="0" />
</soapExtensionTypes>

```



The `Version` and `PublicKeyToken` attribute for the `WsmfLibs` Assembly is different for each release of the .NET Agent. To find these values, use the .NET configuration tool (**Control Panel | Administrative Tools | Microsoft .NET Framework 1.1 Configuration**) and select **Assembly Cache**. From the Assembly list, double-click **WsmfLibs** to view its properties. The Properties window contains the above attribute values.

Logging Extension

The Logging extension allows you to capture logged events for a Web service. Log data is viewable in the `Agent.log` file located in the `Intepub\wwwroot\hpwsm\log` directory.

To configure the Logging extension:

- 1 Create a service configuration in either `Web.config` or `Machine.config`.
- 2 Under the `<service>` element add the following XML content:

```

<logging>
  <category>INFO</category>
</logging>

```

— **category**: Defines the log level that is captured. Valid entries include `INFO`, `DEBUG`, `WARN`, and `ERROR`.

- 3 Save and close the file. Check the agent log file for any logged messages.

Example

```
<defaultServiceConfig>
  <service>
    <logging>
      <category>INFO</category>
    </logging>
  </service>
</defaultServiceConfig>
```

Audit Extension

The Audit extension collects trace information on messages sent to Web services. In addition, the auditing feature can collect a message's SOAP payload. The information collected is sent to the SOA Manager server and is stored in a database. The HP SOA Manager web interface is used to query the database to retrieve audit information. Moreover, any management application can be extended to access the audit data. For more information on using the SOA Manager Auditing feature, see the “Using Auditing” chapter in the *SOA Manager Administrator Guide*.

To configure the Logging extension:

- 1 Create a service configuration in either `Web.config` or `Machine.config`.
- 2 Under the `<service>` element add the following XML content:

```
<audit payload-option="REQUEST-RESPONSE"
      payload-filter="ALL" />
```

- **payload-option:** Defines what payloads to audit. Valid entries are `REQUEST`, `RESPONSE`, `REQUEST-RESPONSE`, or `NONE`.
 - **payload-filter:** Defines when to capture the payload. Valid entries are `ALL` or `ERROR`. Setting this attribute to `ALL` captures payloads that are successful and payloads that encountered errors.
- 3 Save and close the file. Audit data is sent to the SOA Manager Server and viewed using the HP SOA Manager web interface.

Example

```
<defaultServiceConfig>
  <service>
    <audit payload-option="REQUEST-RESPONSE"
          payload-filter="ALL" />
  </service>
</defaultServiceConfig>
```

Business Metric Extension

The Business Metric extension allows you to find specific content within a SOAP request, response, and failure. When the content is found, an alert is sent to the SOA Manager server and viewed in the HP SOA Manager web interface. For example, an OrderEntry Web service may have an OrderTotalAmount field. Using business content monitoring, you can be alerted whenever the value of OrderTotalAmount is greater than \$25,000.00. For more information on using the SOA Manager Alert feature, see the “Using Alert Notifications” chapter in the *SOA Manager Administrator Guide*.



Configuring the Business Metric extension requires knowledge of the W3C XPath expression language. It is beyond the scope of this documentation to cover the details of XPath. Several books on XPath are available and you can also refer to the W3C website for details. If you are not familiar with XPath, you should consult a developer before defining a business content alert.

In addition, the business metric extension is only applicable in a service specific configuration because it is always service specific.

To configure a Business Metric extension:

- 1 Create a service specific configuration in either `Web.config` or `Machine.config`.
- 2 Under the `<service>` element add a `<bizmetric>` node using the following example as a guide. Element definitions are provided after the example.

```
<bizmetric>
  <name>HPQ Info</name>
  <expression>
    //s:Envelope/s:Body/t:InfoRequest/t:symbol[text() = 'HPQ']
  </expression>
  <message>InfoRequest = ${company}</message>
  <operation>getInfo</operation>
  <direction>REQUEST</direction>
  <properties>
    <property>
      <name>company</name>
      <value>text()</value>
    </property>
  </properties>
  <namespaces>
    <property>
      <name>s</name>
      <value>http://schemas.xmlsoap.org/soap/envelope/</value>
    </property>
    <property>
      <name>t</name>
      <value>http://wsm.hp.com/Finance/Request</value>
    </property>
  </namespaces>
</bizmetric>
```

- **name:** A user friendly name to identify the alert.
- **expression:** An XPath expression which selects the business content from the operation.

- **message:** A user friendly message that is sent with the alert. Any alert service variables can also be used in the message. Alert variables are described in the HP SOA Manager web interface.
 - **operation:** The operation in the service that contains the business content you want to monitor.
 - **direction:** When to search for the operation. Valid entries are REQUEST, RESPONSE, and BOTH.
 - **properties:** A dynamic variable defined within the message. The name attribute corresponds to the variable name. The value attribute corresponds to an XPath expression used to update the variable.
 - **namespace:** Any namespace prefixes that appears in the XPath expression. The name attribute refers to namespace prefix. The value attribute refers to the namespace URI.
- 3 Save and close the file. Business Metric Alerts are viewed in the HP SOA Manager web interface.

Example

The following example can be used to create a business metric alert for the FinanceService sample Web service.

```
<services><Service name="FinanceService.asmx">
  <bizmetric>
    <name>HPQ Info</name>
    <expression>
      //s:Envelope/s:Body/t:InfoRequest/t:symbol[text() = 'HPQ']
    </expression>
    <message>InfoRequest = ${company}</message>
    <operation>getInfo</operation>
    <direction>REQUEST</direction>
    <properties>
      <property>
        <name>company</name>
        <value>text()</value>
      </property>
    </properties>
    <namespaces>
      <property>
        <name>s</name>
        <value>http://schemas.xmlsoap.org/soap/envelope/</value>
      </property>
      <property>
        <name>t</name>
        <value>http://wsm.hp.com/Finance/Request</value>
      </property>
    </namespaces>
  </bizmetric>
</service>
```


Introduction- WSM J2EE Agent

The WSM J2EE Agent is responsible for collecting and exposing management data for Web services that are deployed on Java-based Web services platforms.

Prerequisites

Users must have fundamental knowledge of the Java programming language and Java platform technologies. Users should also have fundamental knowledge of Web services principals and be familiar with the WebLogic Server platform.

Contextual Overview

The Agent runs in the same Java Virtual Machine (JVM) process as the Web services platform. In the case of the WSM J2EE Agent for WLS, the Web service platform is WebLogic's Web services platform.



If you want to use your current Java-based Web services platform (not WebLogic), you can use a WSM Intermediary to manage Web services instead of the WSM J2EE Agent.

The J2EE Agent provides a JAX-RPC handler that mediates the SOAP communication between a client and a Web service. The JAX-RPC handler is configured with sub-handlers (referred to as common handlers) that provide varying levels of manageability (Monitoring, Logging/Auditing, and so on). A JAX-RPC handler chain is automatically created for each Web service that the J2EE Agent discovers in the Web services platform.

Another core component of the Agent is the SBA. The SBA contains Managed Objects (MOs) that represent the data and metrics collected by the handlers. The SBA is also responsible for exposing MOs as Web services that comply with the Web Services Management Framework (WSMF) specification. WSMF is an HP-authored precursor version of standard WS-based management protocols. Lastly, SOA Manager uses the management Web services to retrieve management data and metrics and display them in the HP SOA Manager web interface.

Common Handlers

As shown in **Error! Reference source not found.**, a JAX-RPC handler can contain any number of sub-handlers known as common handlers. Together, the handlers are considered a handler chain. Three handlers are included with the WSM J2EE Agent for WLS and are described below. For more information on configuring the common handlers, see Chapter 3.

Monitoring Handler

The Monitoring Handler collects performance data for a Web service. In particular, the handler reports:

- Average Response Time
- Maximum Response Time
- Minimum Response Time
- Total Message Count
- Failed Message Count
- Success Message Count

Audit Handler

The Audit Handler provides message trace capabilities for a Web service. In particular, the handler captures all SOAP requests, responses, and failures. Trace messages are sent to the SOA Manager server database and viewed using the HP SOA Manager web interface.

Business Metric Handler

The Business Metric Handler captures specific data that is found in a SOAP request, response, and failure message. The data is defined using an XPath expression and is applied to individual Web services. When the data is found in a SOAP message, the data is sent to the SOA Manager server and a business content alert is displayed in the HP SOA Manager web interface.

Deploy the WSM J2EE Agent for WebLogic

This chapter provides detailed instructions for installing/deploying the WSM J2EE Agent for WebLogic Server (WLS). The Agent supports WLS scenarios that use standalone managed servers as well as managed servers that are part of a server cluster. The WebLogic instructions in this chapter are specific to deploying the Agent and do not include detailed WLS instructions. For detailed WLS instructions, refer to the WLS documentation.

The WSM J2EE Agent software for WLS is located on the SOA Manager installation CD. Make sure you have the CD before beginning the instructions in this chapter.

- ▶ The WSM J2EE Agent is licensed. During the installation, an evaluation license is installed, and the Agent may be evaluated for 60 days. Thereafter, a license must be purchased. Unlicensed components will fail to start and a license expiration message is written to the log file. For information on licensing, see the “Entering a SOA Manager License” section in the *SOA Manager Installation Guide* after completing the installation.

Software Requirements

The following software is needed when using the WSM J2EE Agent for WebLogic:

- WebLogic Server 8.1.x – A free developer edition of the WebLogic application server can be downloaded from the BEA Web site, <http://www.bea.com>. The Agent supports Windows, UNIX, and Linux installations.
- ▶ The WSM Agent has been tested against WebLogic Server 8.1 service pack 2 (8.1.2.0).

- Ant 1.5.x – Ant is a Java-based build tool available from the Apache open source organization. You can download Ant from <http://ant.apache.org/>.

Pre-Deployment Steps

You must install ANT and the WebLogic Server software before you can deploy the Agent. General setup requirements are provided in this section.

ANT 1.5.x Setup

ANT is used by the agent's setup script and does not require that you have any practical knowledge of ANT. ANT should be installed to a directory with no spaces in the path.

Create an environment variable called ANT_HOME and set it to the directory where you installed ANT.

WebLogic Server 8.1.x Setup

WLS should be installed to a directory with no spaces in the path. Create an environment variable called BEA_HOME and set it to the directory where you installed BEA.

In addition, the WSM J2EE Agent must be deployed to a WLS domain. If you do not currently have a WLS domain set up, use the Configuration Wizard that comes with WLS to create a basic WebLogic Server domain. See the WLS documentation for instructions on using the Configuration Wizard to create a WLS Domain.



It is recommended that a WLS Domain be created specifically for testing the WSM J2EE Agent.

Deploy to the Administration Server

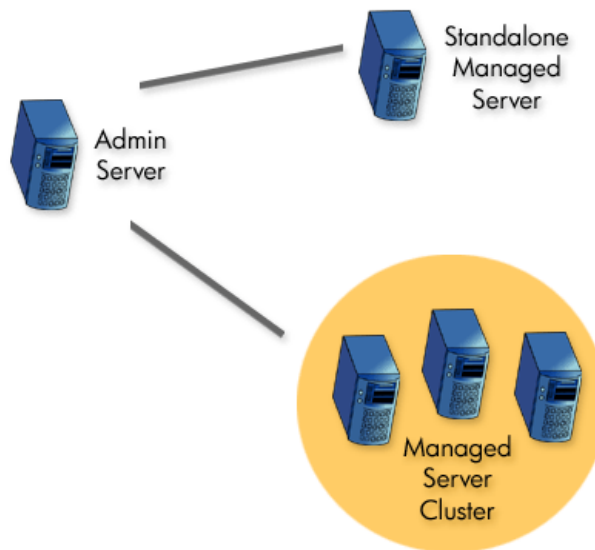
The WSM J2EE Agent must be deployed to the WLS Administration Server. Each WLS domain must have one server that acts as the Administration Server. In a typical production environment, the Administration Server is where the Administration Console is run and used to perform administrative tasks. By default, the Administration Server is called `myserver`.



Make sure you are logged in to the system as a `root/admin` user.

A typical production environment may also have any number of managed servers, which are instances of WebLogic servers used to host enterprise applications. These managed servers can be standalone or part of a cluster. The WSM J2EE Agent must be deployed to the Administration Server before you can deploy additional agents to standalone managed servers or a cluster of managed servers.

Error! Reference source not found. shows an example WLS environment that includes an administrator server, a standalone managed server, and managed servers in a cluster.



If your current environment contains standalone managed servers or a cluster of managed servers, see the “Deploying to Managed Servers” section after completing the instructions in this section.

Extract the WSM J2EE Agent for WLS

Log in to the Administration Server computer as a root/admin user. From the SOA Manager CD, extract `WSM_Agents/Java/WebLogic/wsm-agent-wls8.zip` to a location on the system (i.e., `C:\wsm-agent-wls8` or `/opt/wsm-agent-wls8`). This location will be referred to as `<install_dir>`.

Modify the Environment Setup Script

The WSM J2EE Agent for WLS includes a script that is used to setup its environment. A script is provided for both the Windows and UNIX platforms. As part of the script, a variable called `WSM_AGENT_HOME` is set. You must modify this variable to point to `<install_dir>`.

To edit the environment setup script:

- 1 From the `<install_dir>\bin` directory, open the environment setup script for your platform (`setLocalAgentEnv.bat` or `setLocalAgentEnv.sh`).
- 2 Change the value of the `WSM_AGENT_HOME` variable to point to `<install_dir>`. For example:

```
WSM_AGENT_HOME=c:\wsm-agent-wls8
```

Or,

```
WSM_AGENT_HOME=/opt/wsm-agent-wls8
```

- 3 Save and close the file.

Modify the Agent-Setup Script

The agent-setup script is used to configure the Agent based on your WLS environment. A script is provided for both the Windows and UNIX platforms.

To edit the agent-setup script:

- 1 From the `<install_dir>\bin` directory, open the agent-setup script for your platform (`agent-setup.bat` or `agent-setup.sh`).
- 2 Modify the following variables:
 - **WL_DOMAIN** – The name of the WebLogic domain where the Agent will be deployed.
 - **WL_USER** – The WebLogic administrator user name.
 - **WL_ADMIN_SERVER_NAME** – The name of the WebLogic Administration Server.
 - **AGENT_HOST** – The fully qualified DNS address of the host on which the Agent will run. This value is the same as the host where your WebLogic instance is installed.
 - **AGENT_PORT** – The port number where the Agent will listen for SOAP requests. This value is the same as the port number being used by the Administration Server.
 - **AGENT_PROTOCOL** – The protocol used by the Agent within the WLS environment. This value can be either HTTP or HTTPS. If you select HTTPS, see the “Configuring a Trust Keystore” in Chapter 3 before completing the deployment.
 - **OV_LOG_DIR** – The name of the OV log directory where the agent log file will be stored.
- 3 Save and close the agent-setup script.

Run the Agent-Setup Script

The agent-setup configures the Agent based on your WLS environment and automatically deploys two Web applications called `wsm` and `wsmf` to the Administration Server instance. The applications are automatically deployed the next time you start the Administration Server.

To run the agent-setup script:

- 1 Stop WLS if it is currently started.
- 2 From a command prompt, change directories to `<install_dir>/bin`.
- 3 Run the agent-setup script that is appropriate for your platform (either `agent-setup.bat` or `agent-setup.sh`).

Modify the WebLogic Startup Script

The WSM J2EE Agent for WLS runs in the same process as WebLogic. Therefore, you must modify your domain's startWebLogic startup script to include the Agent's dependencies.

To modify the WebLogic startup script:

- 1 From your WLS domain (that is, `<bea_home>\user_projects\domains\mydomain`), open either `startWebLogic.cmd` or `startWebLogic.sh` depending on your platform.
- 2 Just before the comment near the end of the startWebLogic script which reads "Call WebLogic Server", add a call to the `<install_dir>/bin/setLocalAgentEnv` script that is appropriate for your platform. For example:

```
call c:\wsm-agent-wls8\bin\setLocalAgentEnv.bat
```

Or,

```
. /opt/wsm-agent-wls8/bin/setLocalAgentEnv.sh
```

- 3 In the line `%JAVA_HOME%\bin\java %JAVA_VM%... weblogic.Server`, add `-Djava.util.logging.config.class=com.hp.ov.xpl.log.OvLogConfig` right before `weblogic.Server`. For example:

```
%JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%
-Dweblogic.Name=%SERVER_NAME%
-Dweblogic.ProductionModeEnabled=%PRODUCTION_MODE%
-Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy"
-Djava.util.logging.config.class=com.hp.ov.xpl.log.OvLogConfig
weblogic.Server
```

- 4 Save and close the startup script.

Modify the WebLogic Policy File

The WSM J2EE Agent for WLS requires specific permissions in order to run within WLS. The permissions are set in the WLS `weblogic.policy` file.

To modify the WLS policy file:

- 1 Open the `<bea_home>\weblogic81\server\lib\weblogic.policy` file.
- 2 Open `<install_dir>\config\weblogic_policy_include.txt`.
- 3 Copy the content of `weblogic_policy_include.txt` and append it to the end of `weblogic.policy`.
- 4 Save `weblogic.policy` and close both files.

Start the WebLogic Administration Server

To start the WLS Administration Server for a domain:

- 1 From the command line, change directories to your WLS domain (i.e., `bea\user_projects\domains\mydomain`).

- 2 Depending on your platform, run either `startWebLogic.cmd` or `startWebLogic.sh`. The startup is complete when the following log message is received in the console window:

```
Smart Business Agent Controller has completed initial discovery;  
it is now fully initialized and started!
```

Verify the Deployment

There is a quick method to verify that the WSM J2EE Agent for WLS is successfully deployed to the Administration Server and that it is working properly.

To verify the deployment:


- 1 Open a browser.
- 2 Enter the address where the `wsmf` Web application is deployed. For example:

```
http://myhost.com:7001/wsmf
```

(Change the host and port to match you deployment.)

- 3 From the WSMF SBA WebApp screen, click the **List Deployed Services** link.

You should have at least two management Web services on the list: `WLS` and `WLS.<administration_server_name>`. Additional management Web services may be listed if the domain contains: standalone managed servers, managed servers in a cluster, and Web services that are deployed in the Administration Server.

 If your domain contained standalone managed servers and/or a cluster of managed servers. You must deploy a WSM J2EE Agent to each managed server before the management Web services become operational and Web services in those servers are managed. See the “Deploy to Managed Servers” section below.

- 4 Click the WSDL link for the `WLS` managed service. The WSDL displays as shown below:


```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="WLS" targetNamespace="http://www.hp.com/mso/mip/2004/02/Wsee"
  xmlns:Events="http://schemas.hp.com/wsmf/2003/03/Events#"
  xmlns:Foundation="http://schemas.hp.com/wsmf/2003/03/Foundation"
  xmlns:WsExecutionEnvironment="http://schemas.hp.com/wsmf/2003/03/WsExecutionEnvironment"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.hp.com/mso/mip/2004/02/Wsee"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:import location="http://15.40.235.105:7001/wsmf/wsdl/WS-Events.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/Events#" />
  <wsdl:import location="http://15.40.235.105:7001/wsmf/wsdl/WSMF-Foundation.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/Foundation" />
  <wsdl:import
    location="http://15.40.235.105:7001/wsmf/wsdl/WsExecutionEnvironment.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/WsExecutionEnvironment" />
  - <wsdl:types>
  - <xsd:schema elementFormDefault="qualified"
    targetNamespace="http://www.hp.com/mso/mip/2004/02/Wsee"
    xmlns:ManagementNotification="http://schemas.hp.com/mip/Events/ManagementNotification"
    xmlns:tns="http://www.hp.com/mso/mip/2004/02/Wsee"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:import
      namespace="http://schemas.hp.com/mip/Events/ManagementNotification"
      schemaLocation="http://15.40.235.105:7001/wsmf/schema/MIP-Events.xsd" />
    - <xsd:element name="DeployService">
      - <xsd:complexType>
        - <xsd:sequence>
          <xsd:element name="param0" type="xsd:string" />

```



The WLS and WLS.<administration_server_name> managed services represent a set of basic operations and attributes for the WLS environment. The Agent created these services based on management information being exposed through WebLogic's JMX MBeans.

Verify the Automatic Discovery of Web Services

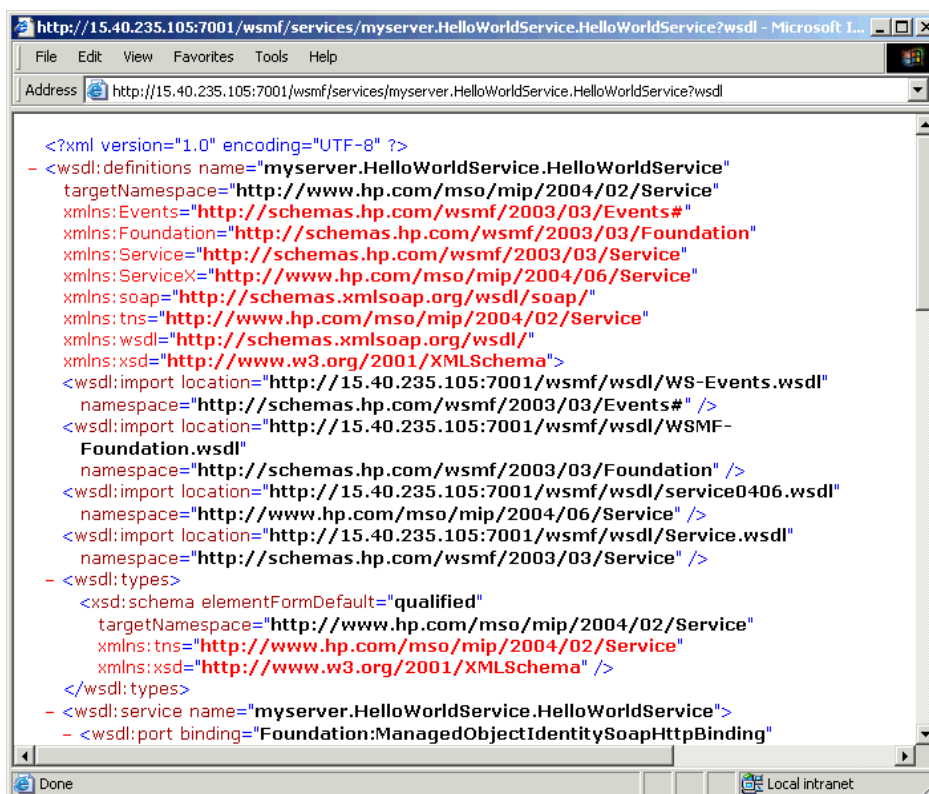
You can verify that the J2EE Agent is automatically discovering Web services by deploying a Web Service and verifying that a management service has been created for it. If you do not have a Web service to deploy, a simple Web service (HelloWorldService) is included in the <install_dir>\sample directory. The service is packaged as a Web application.

To verify the automatic discovery of Web Services:

- 1 Deploy <install_dir>\sample\HelloWorldService to the Administration Server using the WLS Administration Console.
- 2 Open a browser.
- 3 Enter the address where the wsmf Web application is deployed. For example:
http://myhost.com:7001/wsmf
(Change the host and port to match you deployment.)
- 4 Click the **List Deployed Services** link. The management service for the HelloWorldService is listed.

- ▶ The Agent polls The WLS environment every 2 minutes to discover any changes that have occurred in the WLS servers. It may take up to 2 minutes for the management service for the HelloWorldService to be listed. You can change the polling interval using the Agent's SBA configuration file. See "Changing the SBA Polling Frequency" in chapter 3.

- 5 Click on the WSDL link for the `<admin_server_name>.HelloWorldService.HelloWorldService` service. The management WSDL for this service displays as shown below:



```

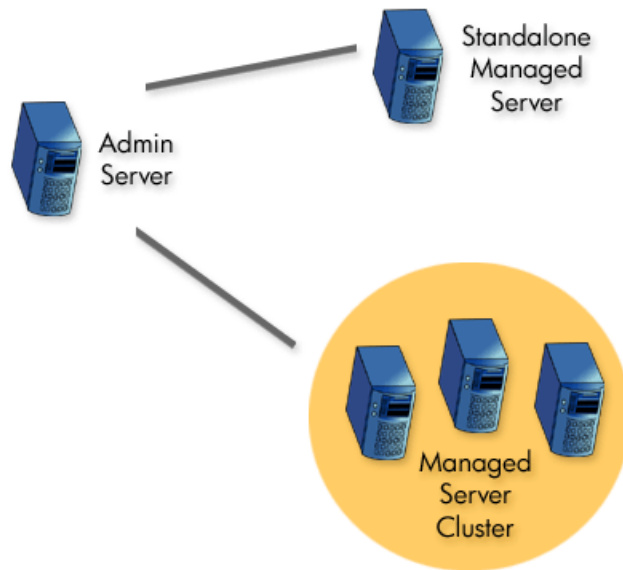
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="myserver.HelloWorldService.HelloWorldService"
  targetNamespace="http://www.hp.com/mso/mip/2004/02/Service"
  xmlns:Events="http://schemas.hp.com/wsmf/2003/03/Events#"
  xmlns:Foundation="http://schemas.hp.com/wsmf/2003/03/Foundation"
  xmlns:Service="http://schemas.hp.com/wsmf/2003/03/Service"
  xmlns:ServiceX="http://www.hp.com/mso/mip/2004/06/Service"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.hp.com/mso/mip/2004/02/Service"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:import location="http://15.40.235.105:7001/wsmf/wsd/WS-Events.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/Events#" />
  <wsdl:import location="http://15.40.235.105:7001/wsmf/wsd/WSMF-
    Foundation.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/Foundation" />
  <wsdl:import location="http://15.40.235.105:7001/wsmf/wsd/service0406.wsdl"
    namespace="http://www.hp.com/mso/mip/2004/06/Service" />
  <wsdl:import location="http://15.40.235.105:7001/wsmf/wsd/Service.wsdl"
    namespace="http://schemas.hp.com/wsmf/2003/03/Service" />
  - <wsdl:types>
    <xsd:schema elementFormDefault="qualified"
      targetNamespace="http://www.hp.com/mso/mip/2004/02/Service"
      xmlns:tns="http://www.hp.com/mso/mip/2004/02/Service"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
    </wsdl:types>
  - <wsdl:service name="myserver.HelloWorldService.HelloWorldService">
  - <wsdl:port binding="Foundation:ManagedObjectIdentitySoapHttpBinding"
  
```

- ▶ To actually see the runtime data and metrics being collected for the HelloWorldService by the HelloWorldService management service, you must use the SOA Manager's Business Service Explorer. See the *SOA Manager User Guide*.

Deploy to Managed Servers

Beyond the Administration Server, a typical WLS production environment may have any number of managed servers, which are instances of WLS used to host enterprise applications. These managed servers can be standalone or part of a cluster. In addition, managed servers can be local to the Administration Server process or located on remote computers. To support such scenarios, the WSM J2EE Agent for WLS can be deployed to any managed server.

Error! Reference source not found. shows an example WLS environment that includes an administrator server, a standalone managed server, and managed servers in a cluster.



The WSM J2EE Agent for WLS must be deployed to an Administration Server before it can be deployed to a managed server. See the “Deploy to Administration Server” section before completing the instructions in this section. In addition, when deploying the Agent to a remote computer, access to the computer is required in order to complete the instructions in this section.

This section assumes knowledge of setting up and configuring managed servers and clusters. To complete the instructions in this section, you must have at least one managed server in addition to the Administration Server. The managed server can be standalone and /or part of a cluster. To simplify testing, setup the managed server on the same computer as the Administration Server. See the BEA WLS documentation if you do not know how to create and configure servers and/or clusters.

Extract the WSM J2EE Agent for WLS

From the SOA Manager CD, extract `WSM_Agents/Java/WebLogic/wls8-wsme-managed-server.zip` to a location on the managed server computer (i.e., `C:\wsm-agent-wls8` or `/opt/wsm-agent-wls8`). This location will be referred to as `<managed_install_dir>`. The Agent installed on a managed server is a “lite” version of the Agent.

Modify the Environment Setup Script

The WSM J2EE Agent for WLS includes a script that is used to setup its environment. A script is provided for both the Windows and UNIX platforms. As part of the script, a variable called `WSM_AGENT_HOME` is set. You must modify this variable to point to `<managed_install_dir>`.

To edit the environment setup script:

- 1 From the `<managed_install_dir>\bin` directory, open the environment setup script for your platform (`setLocalAgentEnv.bat` or `setLocalAgentEnv.sh`).
- 2 Change the value of the `WSM_AGENT_HOME` variable to point to `<managed_install_dir>`. For example:

```
WSM_AGENT_HOME=c:\wsm-agent-wls8
```

Or,

```
WSM_AGENT_HOME=/opt/wsm-agent-wls8
```
- 3 In each line with `%JAVA_HOME%\bin\java %JAVA_VM%...`
`weblogic.NodeManager`, add `-Djava.util.logging.config.class=com.hp.ov.xpl.log.OvLogConfig` right before `weblogic.NodeManager`.
- 4 Save and close the file.

Modify the BEA Node Manager

The BEA Node Manager is a Java utility that runs as a separate process from WLS on each managed server. An Administration Server uses the Node Manager to perform common operations and tasks on managed servers (i.e., starting and stopping). The WSM J2EE Agent for WLS leverages the Node Manager's capabilities. To achieve this integration, the Node Manager's classpath must be modified on each managed server host to include the Agent's dependencies.



The use of the Node Manager is required to use the Agent on a managed server.

The Node Manager's classpath can be modified two different ways depending on how the Node Manager is running (either a standalone process or as a Windows service). Instructions for both scenarios are provided.

Modify the Node Manager Windows Service

To modify the Node Manager when using it as a Windows service:

- 1 Stop the managed server if it is currently started.
- 2 Stop and uninstall the Node Manager Windows service if it is currently started.
- 3 Open `<bea_home>\weblogic81\server\bin\installNodeMgrSvc.cmd`.

- 4 Just after the comment near the end of the `installNodeMgrSvc.cmd` script which reads “rem *** Set Command Line for service to execute within created JVM” add a call to the `<managed_install_dir>/bin/setLocalAgentEnv.bat`. For example:


```
call c:\wsm-agent-wls8\bin\setLocalAgentEnv.bat
```
- 5 Save and close the `installNodeMgrSvc.cmd` script.
- 6 Run the `installNodeMgrSvc.cmd` script.
- 7 Start the service using the Windows Services Administrative tool.

Modify the Node Manager Standalone Process

To modify the Node Manager when using it as a standalone process:

- 1 Shut down the Node Manager process if it is currently started.
- 2 Open the `<bea_home>\weblogic81\server\bin\StartNodeManager` script that is appropriate for your platform (`startNodeManager.cmd` or `startNodeManager.sh`).
- 3 On Windows, just after the `set JAVA_VM=` entry, add a call to set the Java vendor to Sun. For example:

```
set JAVA_VENDOR=Sun
```

- 4 Just after the `set CLASSPATH` entry, add a call to the `<managed_install_dir>/bin/setLocalAgentEnv` script that is appropriate for your platform. For example:

```
call c:\wsm-agent-wls8\bin\setLocalAgentEnv.bat
```

Or,

```
./opt/wsm-agent-wls8/bin/setLocalAgentEnv.sh
```

- 5 Save and close the script.
- 6 Run the `startNodeManager` script to start the Node Manager.

Start the Managed Server

Managed servers are started through the Administration Server console. When a managed server is started, the Agent is initialized and a managed service is created for each Web service that is running in the managed server.

To start a managed server:

- 1 Log in to the WLS Administration Console.
- 2 From the Left frame, select the managed server that you want to start.
- 3 From the Right frame, click the **Control** tab.
- 4 From the Start/Stop tab, select **Start this server...** The server is started. It may take several seconds for the managed server to become operational.
- 5 Repeat this procedure to start additional managed servers.

Verify the Agent Deployment to a Managed Server

There is a quick method to verify that the WSM J2EE Agent for WLS is successfully deployed to a managed server and that the Agent created a managed service for it. If a cluster is configured, a managed service is also created for the cluster and the managed servers in the cluster.

To verify the Agent deployment to a managed server:

- 1 Open a browser.
- 2 Enter the address where the `wsmf` Web application is deployed on the Administration Server. For example:

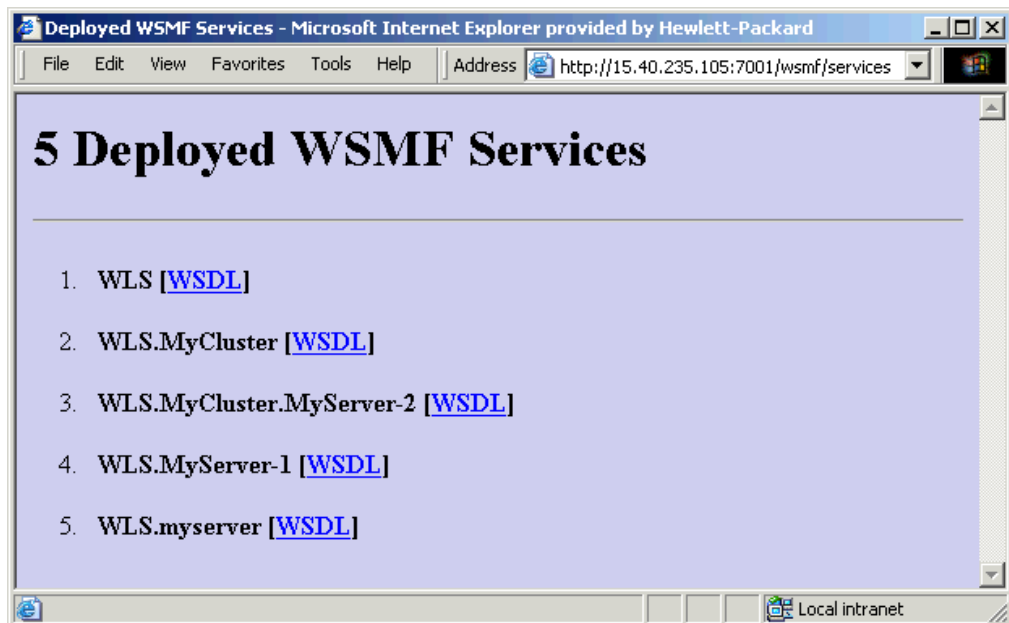
`http://myhost.com:7001/wsmf`

(Change the host and port to match you deployment.)

- 3 From the WSMF SBA WebApp screen, click the **List Deployed Services** link.

The services list displays the management services that were created based on the managed servers and/or clusters that were discovered in your environment. The following screen shot displays the management services created for an environment that contains:

- 1 cluster: `WLS.MyCluster`
- 1 managed server in the cluster: `WLS.MyCluster.MyServer-2`
- 1 standalone managed server: `WLS.MyServer-1`
- 1 Administration Server: `WLS.myserver`



The management services are organized in a `contains` relationship. That is, WLS is the root management service. All management services are contained by the root. If a cluster is used, the managed server is contained in the cluster which is contained in the root. This hierarchy mirrors the WLS hierarchy. Any Web services that are being discovered are also organized according to this convention. See “Verify the Automatic Discovery of Web Services” section below.

- 4 Click on a management service’s WSDL link to see its WSDL.

Verify the Automatic Discovery of Web Services

You can verify that the J2EE Agent is automatically discovering Web services on a managed server by deploying a Web Service to each managed server and verifying that a management service has been created for it. If you do not have a Web service to deploy, a simple Web service (HelloWorldService) is included in the `<install_dir>\sample` directory. The service is packaged as a Web application.

To verify the automatic discovery of Web Services:

- 1 Start the WLS Administration Console on the Administration Server.
- 2 Deploy `<install_dir>\sample\HelloWorldService` to each managed server.
- 3 Open a browser.
- 4 Enter the address where the `wsmf` Web application is deployed. For example:
`http://myhost.com:7001/wsmf`
(Change the host and port to match you deployment.)
- 5 Click the **List Deployed Services** link. The management services for the HelloWorldService are listed based on your environment. Notice in the following example that each management server where the application is deployed has its own HelloWorldService management service.



▶ The Agent polls The WLS environment every 2 minutes to discover any changes that have occurred in the WLS servers. It may take up to 2 minutes for the management service for the HelloWorldService to be listed. You can change the polling interval using the Agent's SBA configuration file. See "Changing the SBA Polling Frequency" in chapter 3.

- 6 Click on a HelloWorldService management service WSDL link to see its WSDL.

Un-Deploying on the Administration Server

The following steps are used to un-deploy the WSM J2EE Agent for WLS from the Administration Server.

To un-deploy the Agent from the WLS Administration Server:

- 1 Open a browser.
- 2 Go to the Agent's `wsm/handlerRemoval.jsp`. For example:
`http://myhost.com:7001/wsm/handlerRemoval.jsp`
(Change the host and port to match you deployment.)
- 3 On the Handler Removal Form, select the remove check box to remove the WSM Agent handlers from the Web service.
- 4 Click the **Remove Handlers** button. The Processing screen displays and verifies the removal of the WSM handlers for the selected Web services.

- 5 Close the browser.
- 6 Open a command prompt and change directories to `<install_dir>\bin`.
- 7 Run the appropriate agent-setup script (.bat, .sh) for your platform and specify the “undeploy” parameter. For example,


```
agent-setup undeploy-from-wls
```
- 8 Close the command window.
- 9 Remove the `JLA-removed.properties` file from your home directory (i.e., `C:\Documents and Settings\Administrator` or `/usr/admin`).
- 10 Open the `<bea_home>\weblogic81\server\lib\weblogic.policy` file and remove the policies for the WSM Agent.
- 11 Save and close the file.
- 12 Remove the call to configure the local agent environment that was added to the `startWebLogic` script. For example remove:


```
call c:\wsm-agent-wls8\bin\setLocalAgentEnv.bat
```

 Or,


```
./opt/wsm-agent-wls8/bin/setLocalAgentEnv.sh
```
- 13 Save and close the startup script.
- 14 Stop and restart WLS.
- 15 Delete `<install_dir>`.

Un-Deploying on Managed Servers

The following steps are used to un-deploy the WSM J2EE Agent for WLS on any managed servers that are standalone or part of a cluster.

To un-deploy the Agent from a managed server:

- 1 On the managed server, stop the BEA Node Manager if it is currently started.
- 2 Remove the call to `setLocalAgentEnv` from the `<bea_home>\weblogic81\server\bin\StartNodeManager` script that is appropriate for your platform (`startNodeManager.cmd` or `startNodeManager.sh`).

Or if using the Node as a Windows service,

Remove the call to `setLocalAgentEnv` from `<bea_home>\weblogic81\server\bin\installNodeMgrSvc.cmd`. You must then uninstall and reinstall the node manager.
- 3 Delete `<managed_install_dir>`.
- 4 Repeat this procedure for each managed server where the Agent is installed.

Deploying the MOM Agent for WLS

The SOA Manager's MOM Agent for WLS is an optional management agent that provides monitoring capabilities for Web services that utilize SOAP/JMS. In particular, the agent provides monitoring capabilities to detect when messages get stuck in the messaging infrastructure and are not delivered to the end destination.



The MOM Agent is not required when running the WSM J2EE Agent for WebLogic. The MOM Agent is an optional standalone application for JMS support. In addition, a tutorial demonstrating JMS management is provided in the `/Documentation` directory of the distribution. Refer to the instructions included with the tutorial.

The MOM Agent communicates management information about WLS JMS servers, topics, and queues to the SOA Manager server. The steps in this section assume that a WLS JMS server has been created and that the server contains topics and/or queues. If you are unfamiliar with JMS, refer to the WLS documentation for detailed instructions.

To deploy the MOM Agent:

- 1 From the SOA Manager CD, copy `/MOM_Agents/Java/Weblogic/MOMAgent.ear` to a location on the WLS computer. This location is referred to as `<install_dir>`.
- 2 Connect to the BEA WLS Administrator Console.

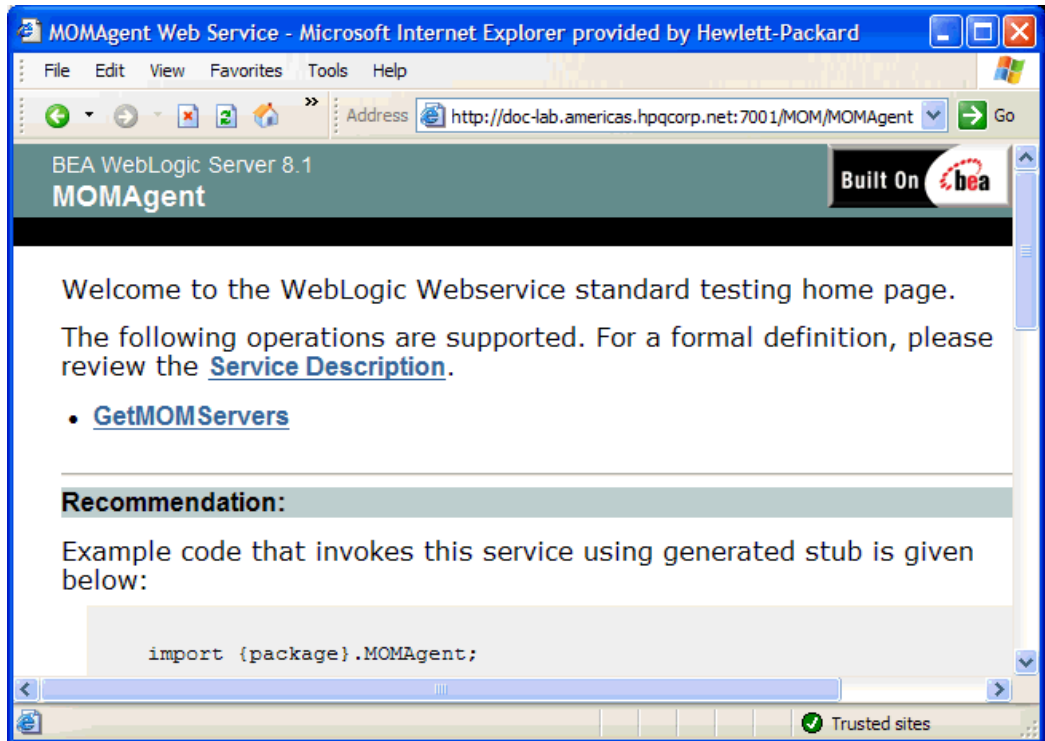


The WLS user name and password must be `weblogic/weblogic`.

- 3 In the left frame, click **Applications**. The Applications screen displays in the right frame.
- 4 Click **Deploy a new Application....** The Deploy an Application screen displays.
- 5 In the right frame, use the Location: URL to navigate to the `<install_dir>/MOMAgent.ear` file.
- 6 Select **MOMAgent.ear**.
- 7 Click **Continue**. The Deploy an Application screen redisplay and allows you to review your deployment selection.
- 8 Click **Deploy**. The MOM Agent (`MOMAgent`) is deployed and is listed under Applications in the left frame. The deployment may take a few seconds to complete. Once deployed, the MOM Agent automatically discovers all JMS servers including their contained topics and queues.
- 9 Open a browser and go to the following address:

`http://<wls_system>:7001/MOM/MOMAgent`

The WLS standard Web service testing page displays for the MOM Agent's management Web service as shown below.



- 10 Click **Service Description** to see the MOM Agent's WSDL.
- 11 Refer to the *SOA Manager User Guide* for instructions on creating a MOM Policy enforcement point that contains a MOM Agent so that the JMS servers, topics, and queues can be monitored from the HP SOA Manager web interface.



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 now 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.

Step 4: Import Signed Server Certificate to Key Store


```
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.

Appendix B Troubleshooting SOA Manager

This chapter provides common troubleshooting tasks when using the SOA Manager.

Troubleshooting Tips/FAQ

- If I send an invalid JMS message, SOA Manager Intermediary continuously processes the same message repeatedly. What do I do?
 - When an invalid message is sent to SOA Manager Intermediary, SOA Manager 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 Manager 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 Manager 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 Manager updates the routing table only after successful deployment. Using Lifecycle Management link, check the status of the deployment.

Installation and Configuration Problems

Errors occurred during installation

Receive an error message at the end of the installation:

The installation of HP SOA Manager is finished, but some errors occurred during the install. Please see the installation log for details.

Solution:

- 1 Check the `<SOAM dir>/HP_OpenView_SOA_Manager_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 `HPSOAManagerInstaller.bin` file from the SOA Manager 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 Manager.

AutoPass fails to install

Receive an error dialog during installation:

AutoPass, the OpenView 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 Manager.
 - 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 Manager.
- e Change the value of the TMP environment variable back to its original value.

Unable to add Intermediary to Policy Enforcement Agent Group

Receive the message:

```
http://<intermediary system>:<intermediary port>/wsmf/services/
Runtime$service=Wsee?wsdl does not seem to be a valid WSEE or is
offline.
```

Solution:

- 1 Check to see if the WSM Intermediary is available. Access the URL from a browser.
- 2 If the URL is accessible, make sure you can access each imported WSDL and schema. For example, the following is a portion of a WSDL displayed in the browser.

```
<wsdl:import location=
  "http://ovw017.cup.hp.com:9032/wsmf_generated/WS-Events.wsdl"
  namespace="http://schemas.hp.com/wsmf/2003/03/Events#" />
...
<xsd:import namespace=
  "http://openview.hp.com/xmlns/mip/2005/03
  /mip-manageability.xsd"
  schemaLocation="http://ovw017.cup.hp.com:9032/wsmf_generated
  /mip-manageability.xsd" />
...
```

In the browser, access:

```
http://<intermediary system>:<intermediary port>/wsmf_generated/WS-
Events.wsdl http://<intermediary system>:<intermediary
port>/wsmf_generated/
mip-manageability.xsd.
```

- 3 If the URL is not accessible, make sure the Intermediary is running. If the intermediary is not running, start it up. If the intermediary is running, look in the log file to see if there are any errors.

Runtime Problems

Could not start monarch-sba

When trying to start SOA Manager, 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 Manager, 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 Manager.
- 3 Look for errors in the `C:\temp\soam-ns-sba.log` file.

Failed to initialize listener

When trying to start SOA Manager, receive a message:

```
...;SEVERE;An error occurred while initializing the MIP Server: ... : failed to initialize listener
```

Solution:

- 1 Check to see if the SOA Manager is already running. If you are running on Windows and selected to install SOA Manager as a service during the installation process, SOA Manager is automatically started when you reboot the system.
- 2 If SOA Manager is not running, then another application must be using the port. By default, the SOA Manager uses port 5002. Change the SOA Manager to use a different port.
 - a Change directories to `<soam_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 Manager

Timezone error when using Oracle 9i

Receive the message when starting the SOA Manager:

```
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 Manager 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 (timezlg.dat), then you must set the ORA_TZFILE environment variable to point to the larger file.”

Performance data not showing up in Business Service

Requests are sent to the service but the performance data is not updated in the Business Service.

Solution:

Verify service is a resource in the Business Service:

- 1 Click the **Business Services** tab to view the Business Services List screen.
- 2 From the Business Services List screen, expand a business service to view its contained configurations.
- 3 Click the resource configuration you want to view. The appropriate view screen displays. The service should be listed in Discovered Resources at the bottom of the window. If the service is not listed, verify that the configuration is bound to the IT Service that contains the container/intermediary the service is in. The **Bind to IT Service** field should contain a value.
- 4 If there is no value for the **Bind to IT Service** field, click on the **Edit** link next to Configuration: <container name>. Select the correct IT Service for the **Bind to IT Service** field in the configuration details window.
- 5 If there is a value for the Bind to IT Service field:
 - a Click on the value to get the IT Service View screen. Scroll down to the bottom of the window and verify that the service's container/intermediary is in the Contained Policy Enforcement Agent or Intermediary section. If the service's Contained Policy Enforcement Agent or Intermediary is not in the Container Resources section, click on the **Add** link and add it.
 - b Check that the Resource Discovery value is correct. The format for the resource discovery is {namespace}localname where namespace is the service's target namespace and localname is the service's name. For example, if the service is the FinanceService with the following WSDL, the resource discovery is:


```
{http://wsm.hp.com/finance}FinanceService.
```

```

<definitions targetNamespace="http://wsm.hp.com/finance">
  <types>
    ...
  </types>
  <message name="getQuote">
    <part name="parameters" element="partns:QuoteRequest">
    </part>
  </message>
  ...
  <portType name="FinanceServiceSoap">
    <operation name="getQuote">
      <input message="tns:getQuote">
      </input>
      <output message="tns:getQuoteResponse">
      </output>
    </operation>
  </portType>
  <binding type="tns:FinanceServiceSoap"
    name="FinanceServiceSoap">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http">
    </soap:binding>
    <operation name="getQuote">
      <soap:operation style="document" soapAction="">
      </soap:operation>
      <input>
        <soap:body namespace=http://wsm.hp.com/finance
          use="literal">
        </soap:body>
      </input>
      <output>
        <soap:body namespace=http://wsm.hp.com/finance
          use="literal">
        </soap:body>
      </output>
    </operation>
  </binding>
  <service name="FinanceService">
    <port name="FinanceServiceSoap"
      binding="tns:FinanceServiceSoap">
      <soap:address location=
        "http://ovw017.cup.hp.com:7001
          /FinanceService/FinanceService">
      </soap:address>
    </port>
  </service>
</definitions>

```

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 Manager 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 Manager 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 Manager Server process.

Intermediary audit traces not showing up in BSE

When querying for audit messages in the BSE, 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 Manager, 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 Manager:

- 1 In the `<install_dir>\conf\networkservices\xpllogging.properties` file, set the logging level for SOA Manager to fine:

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

- 2 Restart the SOA Manager.

- 3 Confirm that SOA Manager 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.CollectionService;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 Manager. Look for the following message in the trace file:

```
Jul 5, 2005 9:58:04
AM;581;19;com.hp.wsm.sn.monitoring.collectionservice.CollectionService;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/MessageTraceNotification
```

If the message is there, then you know the SOA Manager 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 Manager 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 Manager to make sure that the SOA Manager subscribes to the Intermediary. Wait until you see the following message in the SOA Manager log file:

```
Jul 5, 2005 9:56:35
AM;157;13;com.hp.wsm.sn.monitoring.collectionservice.CollectionService;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$QueueThread;run;com.hp.ov.mip.wsm.sn.router.server.audit.MessageTraceBuffer;FINE;>!  
<Dispatched 1 traces.
```

- 7 Verify that the SOA Manager 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 Manager 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;>!  
<Current services subscribed for audit traces:
```



```

<SubscriptionTableList>
  <SubscriptionTable>
    <ManagedObject>Endpoint:id=e4f85099f4ab9246c0595be76856c2d3
    </ManagedObject>
    <SubscriptionList>
      <PushSubscriptions />
      <PullSubscriptions />
    </SubscriptionList>
  </SubscriptionTable>
</SubscriptionTableList>

  <ManagedObject>SoapDispatcher:serviceId=financeServiceProxy
  </ManagedObject>
  <SubscriptionList>
    <PushSubscriptions />
    <PullSubscriptions />
  </SubscriptionList>
</SubscriptionTable>
<SubscriptionTable>

  <ManagedObject>SmartBusinessAgent:service=WebServiceDirectory
  </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 Manager sets the expiration time to the current time + 1 hour. If there is not an entry for the SOA Manager and you are running the SOA Manager 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 Manager 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 Manager.

Out of Memory

Receive an error that ran out of memory when running SOA Manager 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 Manager service (`service-manager.bat -remove networkservices`).
- 3 Run the bat file again to add SOA Manager 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 Manager 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 Manager.

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.

WebLogic Server WSDL with JMS Topics Fail

This is a known limitation of the product. For topic type of a service, you must modify the JMS service as follows for the parsing to be successful:

For example: If the WSDL has the following service:

```
<service name="TopicBounceService">
  <port name="TopicBounceServicePortJMS"
```

```

        binding="tns:TopicBounceServicePortJMS">
      <soap:address
location="jms://soamw2:7001/weblogic.jms.ConnectionFactory/weblogic.j
ms.intopic?
URI=/TopicBounceBean/TopicBounceService">
      </soap:address>
    </port>
  </service>

```

You must change it as follows:

```

<service name="TopicBounceService">
  <port name="TopicBounceServicePortJMS"
    binding="tns:TopicBounceServicePortJMS">
    <jms:address destinationStyle="topic"
initialContextFactory="weblogic.jndi.WLInitialContextFactory"
jmsVendorURI="http://bea.com"
jndiConnectionFactoryName="weblogic.jms.ConnectionFactory"
jndiDestinationName="weblogic.jms.intopic"
jndiProviderURL="t3://soamw2.ind.hp.com:7001"/>
  </port>
</service>

```

For WebLogic 8.1, SOA Manager parses WSDLs automatically for queue type service.

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 Protocol Switching 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.



Appendix C Troubleshooting Intermediary

This chapter provides common troubleshooting tasks when using the WSM Intermediary.

Installation and Configuration Problems

Errors occurred during installation

Receive an error message at the end of the installation that:

The installation of HP SOA Manager is finished, but some errors occurred during the install. Please see the installation log for details.

Solution:

- 1 Check the `<SOAM dir>/HP_OpenView_SOA_Manager_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 `HPSOAManagerInstaller.bin` file from the SOA Manager 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 the intermediary.

AutoPass fails to install

Receive an error dialog during installation that:

AutoPass, the OpenView 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 Network Services.
 - b Save the value of the TMP environment variable.
 - c Change the TMP environment variable to a directory with all English characters.
 - d Install Network Services.
 - 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 the intermediary, 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:

- 1 Verify that the environment variable MIP_JAVA_HOME is assigned to the Java 1.4 SDK and not the JRE.

When trying to start the intermediary, 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.

f Change directories to <install_dir>/conf/broker.

g Edit the logging.properties file.

1. Change log4j.category.com.hp.wsm.impact=OFF to log4j.category.com.hp.wsm.impact=INFO, ROLL_FILE

2. 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-broker-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 the intermediary.
- 3 Look for errors in the C:\temp\soam-broker-sba.log file.

Failed to initialize listener

When trying to start the intermediary, receive a message:

```
...;SEVERE;An error occurred while initializing the MIP Server: ... :
failed to initialize listener
```

Solution:

- 1 Check to see if the Intermediary is already running. If you are running on Windows and selected to install the Intermediary as a service during the installation process, the Intermediary is automatically started when you reboot the system.
- 2 If the Intermediary is not running, then another application may be using the port. By default, the Intermediary uses port 9032. Change the Intermediary to use a different port.
 - a Change directories to <install_dir>/conf/broker.
 - b Edit the mipServer.xml file. Change the <entry name="com.hp.http.server.port">9032</entry> property.
 - c Start the Intermediary.

Unable to determine binding from message element

Receive the message when a request is sent to a custom intermediary service:

```
Unable to determine binding from message element: {xxx}yyy
```

Solution:

- 1 Verify that the request matches the binding specified in the WSDL.
- 2 Verify that the namespace in the request matches the namespace in the WSDL.

Authentication header not progressed to backend

The authentication header is not progressed to the backend service when a request is sent to a custom XML intermediary service.

Solution:

- 1 Verify that the `com.hp.wsm.sn.router.xml.handlers.outbound.SoopPassThroughTransportHeaderHandler` handler is configured for your custom XML intermediary service. This handler works for XML services even though it's called a SOAP handler.

Select Access enforcer cannot connect to validator

Receive the message

```
Error for /console/auth/j_security_check
java.lang.NoClassDefFoundError:
com/hp/wsm/sn/common/security/auth/selectaccess/RouterTransaction
```

Solution:

- 1 The `mip-addons.jar` file is missing.
- 2 On the Intermediary system, create an `addons` directory under `<install_dir>/lib`.
- 3 From the SOA Manager CD, copy `/Addons/mip-addons.jar` to `<install_dir>/lib/addons`.
- 4 Refer to the “Integrating with Select Access” section in the *User Guide* for more details about the Select Access integration.

XML message not being passed to Select Access

The XML message is not being passed to Select Access in a custom XML service.

Solution:

- 1 Verify that an XML introspection handler is configured in the custom XML service's handler list. This handler is needed since the intermediary does not pass XML requests to Select Access automatically.
- 2 Refer to the Creating an “XML Introspection Service” section of the *Integrator Guide* for details.

Out of Memory

Receive an error that ran out of memory when running the Intermediary as a service.

Solution:

Increase the stack and heap sizes.

- 1 Modify the `<soam_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 Intermediary service (`service-manager.bat -remove broker`).
- 3 Run the bat file again to add the Intermediary as a service with the new parameters (`service-manager.bat -install broker`).

- 4 Check that the new parameters are configured by looking in the registry under HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/broker<version num>.

Receive an error that ran out of memory when running the Intermediary 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 the Intermediary.



Appendix D Troubleshooting J2EE Agent

This chapter provides common troubleshooting tasks when using the WSM J2EE Agent. In addition, refer to the *SOA Manager Release Notes* for the latest information on the WSM J2EE Agent.

Runtime Problems

Remote Services is not Discovered

The WSM J2EE Agent does not discover services deployed on a remote managed server.

Solution:

Check for errors in the remote managed server's log file

(<bea_home>/weblogic8/common/nodemanager/<managed_server>/<managed_server>.log file on the remote system. If there's an error similar to:

```
####<Sep 1, 2004 7:36:31 AM PDT> <Error> <HTTP> <ovw025> <remote2>
<ExecuteThread: '2' for queue: 'weblogic.kernel.System'> <<WLS
Kernel>> <> <BEA-101216> <Servlet: "WebServiceServlet" failed to
preload on startup in Web application: "wsm".
javax.servlet.ServletException: ERROR: The handler class:
com.hp.wsm.agent.handler.WsmfHandler from handler-chain:
WsmfEventHandlerChain could not be loaded. Please ensure that this
class name is correct and is included in the web service.
    at
weblogic.webservice.server.servlet.WebServiceServlet.initLocal()V(Web
ServiceServlet.java:132)
    at
weblogic.webservice.server.servlet.WebServiceServlet.init()V(WebServi
ceServlet.java:86)
    at
javax.servlet.GenericServlet.init(Ljavax.servlet.ServletConfig;)V(Gen
ericServlet.java:258)
    at
weblogic.servlet.internal.ServletStubImpl$ServletInitAction.run()Ljav
a.lang.Object; (ServletStubImpl.java:1018)
```

- 1 Verify that the `WSM_AGENT_HOME` variable is set correctly. In the `<managed_install_dir>/bin/setLocalAgentEnv.<bat | sh>` file, `WSM_AGENT_HOME` must be set to the `<managed_install_dir>`. The `<managed_install_dir>` is the directory where you unzipped the `wls8-wsme-managed-server.zip` file.
- 2 Verify that the WebLogic Node Manager startup script (`<bea_home>/weblogic81/server/bin/startNodeManager.<cmd | sh>`) includes a call to `<managed_install_dir>/bin/setLocalAgentEnv.bat`.
- 3 Restart the Node Manager if you modified either of the files in the above steps.

Performance Data not Available from Cluster

Requests are sent to servers in a cluster but the performance data is not updated.

Solution:

- 1 Check the CLASSPATH, `wsm-commons-qname.jar` must be at the beginning. If `weblogic.jar` appears earlier, the call to `setLocalAgentEnv.<bat | sh>` is most likely misplaced. In this case, you may find an exception in the Weblogic log files.


```
java.lang.NoSuchMethodError:
javax.xml.namespace.QName.<init>(Ljava/lang/String;Ljava/lang/Stri
ng;Ljava/lang/String;)V.>
```
- 2 Restart all of the servers in the cluster.

Non-Existent Services in deployed services list

When you run <http://myhost.com:7001/wsmf> (change host and port to match your deployment), there are services listed that no longer exist.

Solution:

This situation should only occur if you modified the `config.xml` file by hand. You'll need to manually cleanup the `<install_dir>/webapps/wsmf/WEB-INF/server-config.wsdd` file. Remove all the services that no longer exist and restart the WebLogic admin server.

WsmfEventHandlerChain could not be loaded

Received a message when starting up the WebLogic Server:

```
ERROR: The handler class: com.hp.wsm.agent.handler.WsmfHandler from
handler-chain: WsmfEventHandlerChain could not be loaded. Please
ensure that this class name is correct and is included in the web
service.
```

Solution:

- 1 Verify that the `WSM_AGENT_HOME` is set to the correct location. Refer to the "Modify the WebLogic Startup Script" section in Chapter 2 for details.
- 2 Verify that the `setLocalAgentEnv` script is called from the WebLogic startup script. Refer to the "Modify the WebLogic Startup Script" section in Chapter 2 for details.



Appendix E Technical Policies

Technical Policies- Policy Mapping

Refer to the information below for information about the number of policies that you can associate with a web service:

Route Policy

All the route policies are used while mapping the policies to the Intermediary. If there is a route policy with a JMS endpoint then only that route policy is used for a service, all the HTTP end point route policies are ignored while mapping.

This is due to the limitation in the Intermediary which cannot handle multiple types of end points for the same service. Provisioning using Broker GUI is used as the first endpoint in the list.

Audit Policy

You can map two audit polices for a web service, one with security enabled and one with secure disabled. If more than one audit policy with security enabled intermediary mapping feature is enabled, the SOA Manager uses the first policy in the policy list. SOA Manager uses a similar method for one or more audit policies with security disabled feature enabled.

JMS Mediation Policy

You can map only one JMS mediation policy for a web service. If there is more than one JMS mediation policy, SOA Manager uses the first policy in the list for mapping.

Message Security Policy

You can map only two message security policies to a web service, one for inbound message and other for outbound message. If there are multiple inbound or outbound message policies, SOA Manager uses the first policy in the list.

Transport Security Policy

You can map only two transport security policies to a web service, one for the inbound transport and the other for outbound transport. If there are multiple inbound or outbound policies, SOA Manager uses the first policy in the list.

Log Policy

You can map only one log policy for a web service. SOA Manager uses the first policy in the list if more than one policy is defined.

Schema Validation Policy

You can map only one schema validation policy for a web service. SOA Manager uses the first policy in the list if more than one policy is defined.

Event Policy

You can map only one event policy for a web service. SOA Manager uses the first policy in the list if more than one policy is defined.

Transform Policy

You can map only one transform policy with directions Both, Request, and Response. If an individual policy is defined (request or response) then this policy takes priority over other transform policies defined. If multiple policies with the same direction are specified, SOA Manager uses the first policy in the list.

Load Balancing Policy

You can map only one load balancing policy for a web service. SOA Manager uses the first policy in the list if more than one policy is defined.

Notes

If the web service is an XML service, then SOA Manager ignores the message security policy and schema validation policy. If a policy is invalid, the Intermediary might not create a handler for the policy. If there are multiple policies, one an invalid policy and the other, a valid policy, SOA Manager selects the first policy in the list and tries to process it irrespective of whether the policy is valid or not valid.

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.

Broker Configurator

The Broker Configurator is the WSM Broker's administration console. It is used to create and configure brokered services as well as configure the Broker's server properties.

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.

Business Service Configuration

A business service configuration is a part of the service model that contains an IT service and provides varying levels of automation.

Content Monitoring

Content monitoring is a management feature that searches Web service request and/or response messages for specific content.

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 Manager 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 Manager or add additional management data to the SOA Manager.

Interposed Manageability

Interposed manageability means inserting management policies in the request/response path of Web services.

Logging

Logging in the SOA Manager 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 Manager, 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 Manager.

Management Agents

Management agents are software components that get installed on a computer and are responsible for performing management tasks.

Management Channel

The Management channel refers to the communication between the SOA Manager and one or more management agents. In the SOA manager, the management channel can be different than the application channel.

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 WSM Agents or 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 Manager 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 Agents and the WSM Broker expose their management information as management Web services.

Northbound Interfaces

Northbound interfaces are Web services-based integration interfaces that are used to extract the information contained in the SOA Manager's management model.

Performance Monitoring

Performance monitoring is a management feature that captures a set of real-time performance metrics that clearly indicate the health, availability, and performance of Web services.

Policy Handlers

Policy handlers are the actual implementation of the management policies in the WSM Agents and WSM Broker. Policy handlers are often referred to as simply handlers.

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 Level Objective (SLO)

An SLO is a set of preferred operating limits for a Web 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.

SLO Monitoring

SLO Monitoring is a management feature that evaluates a Web service's performance against an SLO to insure it is within acceptable operating limits.

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.

Southbound Interfaces

The Southbound interfaces are Web services-based integration interfaces that are used to interact with a managed WS Container/WSM Broker.

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 Manager 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 Management (WSM) Agent

A WSM Agent is an enablement component that is installed in a WS Container in order to manage the Web services in the container as well as the container itself. There is a WSM Agent for both the Microsoft Internet Information Server (IIS) and for the BEA WebLogic Server (WLS).

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.

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.

XML (Extensible Markup Language)

XML is a markup language used to describe data and does not include any presentation logic for the data.

A

- acknowledge alerts, 7-11
- administration server, 22-2, 22-8
- agent
 - MOM, 22-16
- agent for .NET, 2-7
- agent for WLS, 2-7
- agent handlers, 3-5
- agent setup script, agent-setup, 22-4
- agent/intermediary
 - log traces, 5-5
 - managing, 5-4
- AGENT_HOST, 22-4
- AGENT_PORT, 22-4
- AGENT_PROTOCOL, 22-4
- alert category, 7-2
- alert recipients
 - email, 7-14
 - log, 7-15
 - setup, 7-12
 - SNMP, 7-15
- alerts
 - acknowledge, 7-11
 - business content, 7-3
 - customize message, 7-11
 - overview, 7-1
 - Policy enforcement agent/intermediary
 - availability notifications, 5-6
 - query, 7-12
 - SLO, 7-1
- ANT
 - general setup, 22-2
 - requirements, 22-1
- application channel, 2-6, G-1
- architecture
 - multiple brokers, 16-4
 - WSM agents, 3-5

- WSM broker, 3-5
- audit handler
 - enable, 6-10
- audit publisher, 3-6, 13-2
 - configure, 6-12
- auditing, 2-3, 13-1, G-1
 - audit publisher, 13-2
 - business service reports, 6-16
 - viewing message trace, 6-15
- auditing handler, 11-3, 21-2
- authentication, 17-9
 - broker configurator, 10-5
 - enabling, 17-18
- authorization, 17-9
- availability % metric, 6-29
- availability monitoring, 2-2, G-1
- availability notifications
 - contained resource, 5-6
 - PEPs, 5-9
- average response time, 21-2
- average response time metric, 6-29
- axis enforcer, 10-2

B

- backup endpoint, 16-2
- BEA node manager
 - modify standalone process, 22-11
 - modify Windows service, 22-10
 - overview, 22-10
- breach alert, 7-1
- broker, 2-8
 - contextual overview, 11-1
 - install as Windows service, 15-3
 - management channel port, 15-4
 - SSL port, 17-9
 - starting, 15-1
 - stopping, 15-2

- using multiple, 16-4
 - broker configurator, 2-8, 11-2, G-1
 - assign access, 15-6
 - secure access, 9-6
 - starting, 15-2
 - broker configurator authentication, 10-5
 - broker-based deployment scenario, 3-11
 - brokered service
 - auditing, 13-1
 - business content, 13-2
 - convert simple, 14-2
 - custom, 14-1
 - deploy, 12-2
 - edit, 12-3
 - fault logging, 13-1, 13-4
 - HTTP path, 12-4
 - overview, 12-1
 - performance metrics, 12-2
 - remove, 12-4
 - schema validation, 13-5
 - undeploy, 12-2
 - version, 12-3
 - view details, 12-2
 - brokered services, 2-8, G-1
 - BSE, 2-11, 20-6, 21-2, 22-8
 - business and IT alignment, 3-8
 - business content alerting, 13-2
 - business content alerts
 - define, 7-3, 7-8
 - business content handler, 11-3
 - business metric handler, 21-3
 - configure, 20-6
 - business seervice
 - assign roles, 8-9
 - business service, 1-3, 8-1, G-1
 - add configuration, 8-3
 - add resource, 8-4
 - create, 8-2
 - delete, 8-12
 - export, 8-11
 - import, 8-11
 - overview, 8-1
 - publish to UDDI, 8-10
 - relationships, 8-7
 - business service configuration, G-1
 - business service explorer, 2-11
 - business service model, 8-1
- ## C
- certificate authority, 9-2
 - cluster, 22-2, 22-8, 22-12
 - common handlers, 11-2, 21-2
 - auditing, 21-2
 - business metric, 21-3
 - configure business metric, 20-6
 - monitoring, 21-2
 - components
 - WSM overview, 3-5
 - conceptual architecture
 - failover and load balancing, 16-2
 - multiple brokers, 16-4
 - configuration
 - add to business service, 8-3
 - business metric handler, 20-6
 - delete, 8-12
 - WS intermediary, 8-3
 - configuration (model), 1-5
 - configure
 - alert recipients, 7-12
 - audit publisher, 6-12, 13-2
 - auditing, 13-1
 - brokered service HTTP path, 12-4
 - brokered service version, 12-3
 - business content alert, 13-2
 - business content alerts, 7-3
 - database, 4-7, 6-14
 - email recipients, 7-14
 - failover and load balancing, 16-3
 - fault logging, 13-1, 13-4
 - HTTP, 4-5, 15-3
 - HTTPS, 9-4
 - key store, 17-8
 - key store and trust store, 9-2
 - log recipients, 7-15
 - refresh, 4-6

- schema validation, 13-5
 - SLO alerts, 7-1
 - SNMP recipients, 7-15
 - SNMP TRAP, 7-15
 - SSL, 9-4
 - SSL port, 17-9
 - trust store, 17-8
 - UDDI, 4-9
 - content monitoring, 2-4, G-1
 - contextual overview, 11-1, 21-1
 - counfigure
 - inbound message security, 17-20
 - inbound transport security, 17-18
 - outbound message security, 17-21
 - outbound transport security, 17-19
 - custom alert message, 7-11
 - custom handlers, 14-3
- D**
- database, 4-7
 - configure auditing, 6-14
 - database properties, 4-7, 6-14
 - delete
 - business service, 8-12
 - configuration, 8-12
 - deploy
 - WSM J2EE Agent to administrator server, 22-2
 - WSM J2EE Agent to managed server, 22-8
 - deployment, 2-4
 - deployment scenarios
 - broker-based, 3-11
 - WSM-based, 3-11
 - deployment service, 3-6
 - dispatcher, 3-6
 - distributed management, G-1
- E**
- email alert recipients, 7-14
 - endpoint
 - backup, 16-2
 - multiple in WSDL, 16-3
 - primary, 16-2
 - environment setup script setLocalAgentEnv, 22-3
 - environment setup script, setLocalAgentEnv, 22-10
 - environment variable, 4-2, 15-1
 - ANT_HOME, 22-2
 - BEA_HOME, 22-2
 - WSM_AGENT_HOME, 22-3, 22-10
 - export business service, 8-11
- F**
- failed message count, 21-2
 - failover and load balancing
 - conceptual architecture, 16-2
 - multiple brokers, 16-4
 - overview, 16-1, 16-2
 - scenarios, 16-2
 - setup, 16-3
 - failure metric, 6-29
 - fault logging, 13-1, 13-4
 - finance sample application, 4-1
- G**
- generic soap contract handler, 13-3
- H**
- handlers. *See* common handlers, *See* common handlers, *See* policy handler
 - add to custom, 14-2
 - configuring, 13-1
 - custom, 14-3
 - overview, 11-2
 - HSQL database, 6-14
 - HTTP, 22-4
 - brokered service URL, 12-4
 - client settings, 15-5
 - proxy settings, 15-6
 - secure port, 9-4, 17-9
 - server port, 15-4
 - server settings, 15-3
 - threads, 15-5
 - HTTP server port number, 4-5
 - HTTP server thread settings, 4-5
 - HTTP settings, 4-5
 - HTTPS, 9-4, 17-3, 22-4

I

- impact analysis, 2-3, 8-7, G-2
- inbound message security, 17-20
- inbound transport security, 17-18
- installation problems, B-1, C-1
- integration points, G-2
- intermediary agent, 2-8
- interposed manageability, 2-5, G-2
- Invocation handler, 13-4

J

- J2EE agent, 2-7
- J2EE-based deployment scenarios, 3-11
- Java Keytool, A-1
- JAX-RPC, 21-2
- JMS, 22-16
- JMX, 2-7

K

- key store, 9-2, 17-8
- Key Store
 - create, A-1
 - generate CSR, A-2
 - import certificate, A-3
 - obtain certificate, A-2
- keystore, 22-4

L

- LCM4WS
 - alerts, 7-1
 - security, 9-1
- life cycle
 - deployment and configuration, 3-10
 - model definition, 3-9
 - problem resolution, 3-10
 - resource discovery, 3-10
 - SLO monitoring, 3-10
- log alert recipients, 7-15
- log traces for agent/intermediary, 5-5
- log4j, 7-15
- logging, 2-4, 4-11
 - agent/intermediary, 5-5
 - brokered service fault, 13-1, 13-4

- edit/query levels, 5-5
- levels, 4-12
- logging handler, 11-3

M

- managed object, G-2
- managed server, 22-2, 22-8, 22-12
- managed web services, G-2
- management agents, 2-7, G-2
- management channel, 2-6, 15-4, G-2
- management channel SSL, 9-4
- management client, 2-10
- management enablement
 - WSM agents, 2-7
 - WSM broker, 2-8
- management information model, G-2
- management integration, G-2
- management policies, 2-5, G-3
- management proxies, G-3
- management server, G-3
- management service
 - WLS, 22-7
- management service relationships, 22-13
- management service WSDL, 22-6, 22-14
- management web service, G-3
- management web services, 2-6
- maximum idle threads, 4-6
- maximum response time, 21-2
- maximum threads, 4-5
- maximum time metric, 6-29
- message level security, 17-6
 - inbound processing, 17-7, 17-20
 - outbound processing, 17-7, 17-21
- message trace, 13-1
- metrics
 - availability %, 6-29
 - average response time, 6-29
 - failure, 6-29
 - maximum time, 6-29
 - minimum time, 6-29
 - security violation, 6-29
 - success, 6-29
 - total request, 6-29

- uptime %, 6-29
 - Microsoft .NET-based deployment scenarios, 3-11
 - minimum response time, 21-2
 - minimum threads, 4-5
 - minimum time metric, 6-29
 - MIP_JAVA_HOME, 15-1
 - MIP_JAVA_HOME variable, 4-2
 - MOM agent, 22-16
 - monitoring handler, 11-2, 21-2
- N**
- network services
 - key store and trust store, 9-2
 - starting, 4-2
 - network services server, 2-10, 20-6, 21-2
 - BSE console, 2-11
 - node manager
 - modify standalone process, 22-11
 - modify Windows service, 22-10
 - overview, 22-10
 - normal alert, 7-1
 - north bound interfaces, G-3
 - notifications. *See* alerts
- O**
- operations
 - add to business service, 8-6
 - Oracle database, 4-7, 6-15
 - upgrading, 4-8
 - outbound message security, 17-21
 - outbound transport, 17-19
 - overview
 - WSM J2EE Agent, 21-1
 - overview
 - architecture, 11-1
 - brokered service, 12-1
 - failover and load balancing, 16-1
 - security, 17-1
 - overview, 21-1
 - owner roles, 8-9
- P**
- payload, 13-1
- PEP**
- availability notification, 5-9
 - remove resources from, 5-8
- PEPs**
- add resources to, 5-8
 - deleting, 5-10
- performance graph, 6-30
- performance metrics, 12-2
- polling interval, 6-30
 - web services, 6-29
- performance monitoring, 2-2, G-3
- PKI, 17-3, G-3
- Policy enforcement agent/intermediary
- availability notifications, 5-6
 - delete, 5-7
- policy enforcement intermediary
- configure auditing publisher, 6-12
 - configure management channel SSL, 9-4
- policy handler, 2-5, 2-7
- chain, 2-5
 - custom, 2-10
 - types of, 2-9
- policy handlers, G-3
- polling interval
- metrics, 6-30
 - SLO, 7-2
- port, 15-4
- management channel, 15-4
- port number, 4-5
- prerequisites, 11-1
- primary endpoint, 16-2
- proxy settings, 15-6
- Q**
- query
- alerts, 7-12
 - audit message trace, 6-15
- queues, 22-16
- R**
- recipient category
- add recipient, 7-13
 - create, 7-13

- modify, 7-13
- refresh settings, 4-6
- relationships
 - uses, 8-7
- relationships among business services, 8-7
- reports
 - business service, 6-16
- requirements
 - software, 22-1
- requirements
 - WLS domain, 22-2
- resource
 - add to business service, 8-4
 - WS container, 8-4
- resource management, 2-1
- roles
 - business service, 8-9
 - owner support, 8-9
- root cause analysis, 2-3, 8-7, G-4
- runtime problems, B-4, C-2, C-3, D-1

S

- sample application, 4-1
- SBA, 11-1
- schema validation, 13-5
- schema validation handler, 11-3
- secure port, 9-4
- security, 2-5
 - basic authentication, 17-3
 - feature matrix, 17-2
 - implement scenario, 17-17
 - inbound message, 17-20
 - inbound transport, 17-18
 - key stores and trust stores, 9-2
 - message level, 17-6
 - outbound message, 17-21
 - outbound transport, 17-19
 - overview, 9-1, 17-1
 - scenarios, 17-3
 - setup components, 17-7
 - SSL, 9-2, 17-3
 - transport level, 17-5
 - XML encryption, 17-3

- security handler, 11-3
- security violation metric, 6-29
- Select Access, 17-4, 17-9
 - axis enforcer, 10-2
 - define resource, 10-8
 - define service, 10-8
 - setup, 10-2, 17-10
- Select Access resource, 10-17, 17-12, 17-14
- Select Access:, 10-1
- server cluster, 22-2, 22-8, 22-12
- server port, 15-4
- service, G-4
- service consumer, G-4
- service producer, G-4
- service security inbound handler, 13-5
- service version, 12-3
- service-manager.bat, 15-3
- services management
 - enablement architecture, 2-5
 - features, 2-2
- services model, G-4
 - business service, 1-3
 - configuration, 1-5
 - define, 1-6
 - overview, 1-1
- settings
 - alert recipients, 7-12
 - audit publisher, 6-12, 13-2
 - business content alerts, 7-3
 - database, 4-7, 6-14
 - email recipients, 7-14
 - HTTP, 4-5, 15-3
 - HTTPS, 9-4
 - key store, 17-8
 - key store and trust store, 9-2
 - log recipients, 7-15
 - SLO alerts, 7-1
 - SNMP recipients, 7-15
 - SNMP TRAP, 7-15
 - SSL, 9-4
 - SSL port, 17-9
 - trust store, 17-8

- UDDI, 4-9
- SLA, 2-3, G-4
- SLO, G-4
 - alerts, 7-1
 - breach, 7-1
 - normal, 7-1
 - warning, 7-1
- SLO alert
 - assign alert category, 7-2
 - polling interval, 7-2
- SLO engine, 7-2
- SLO monitoring, 2-3, G-4
- SNMP alert recipients, 7-15
- SNMP TRAP, 7-15
- SOA, G-4
- SOA Manager
 - configure SSL, 9-4
 - deployment scenarios, 3-10
- SOA Manager life cycle, 3-9
- SOA Manager prerequisites, 3-1
- SOA Manager roles
 - development, 3-8
 - IT and support, 3-8
 - line of business, 3-7
- SOA manager web interface
 - secure access, 9-6
- SOAP, 2-10, G-4
 - endpoint, 16-3
- SOAP client, 2-10
- SOAP contract handler, 13-6
- SOAP dispatch handler, 13-6
- SOAP extensions, 3-5
- SOAP handlers, 3-5
- SOAP message, 21-2
- SOAP monitoring handler, 13-6
- SOAP pass-through transport header handler, 13-4
- SOAP payload, 13-1
- SOAP pipeline, 2-5
- solution, G-5
- southbound interfaces, G-5
- SSL, 9-2, 17-3, 17-4, 17-5, 17-18, 17-19, G-4
 - configure, 9-4

- enabling, 17-18
 - port, 17-9
- SSO token, 17-3
- standalone managed server, 22-2, 22-8, 22-12
- start
 - network services, 4-2
- stop broker, 15-2
- success message count, 21-2
- success metric, 6-29
- support roles, 8-9

T

- topics, 22-16
- total message count, 21-2
- total request metric, 6-29
- trace bucket size, 6-12
- trace interval, 6-12
- trace message, 13-1
- trace messages, 21-2
- transport level security, 17-5, 17-18, 17-19
- trend analysis, 2-2, G-5
- troubleshooting, D-1
 - business content alerts, 7-8
 - installation problems, B-1, C-1
 - runtime problems, B-4, C-2, C-3
- trust keystore, 22-4
- trust store, 17-8
- trust Store, 9-2

U

- UDDI, G-5
- UDDI registry, 8-10
- un-deploy
 - WSM J2EE Agent from administration server, 22-14
 - WSM J2EE Agent from managed server, 22-15
- uptime % metric, 6-29

URL

- changing, 12-4

W

- warning alert, 7-1
- Web Interface

- refresh settings, 4-6
- web service, G-5
 - operation, 8-6
 - performance metrics, 6-29
- Web services
 - autodiscovery, 22-7, 22-13
- web services intermediary, G-5
- web services management, G-5
- web-services.xml, 20-6
- Win32 service, 15-3
- Windows service, 15-3
- WL_DOMAIN, 22-4
- WL_USER, 22-4
- WLS
 - administration server, 22-2, 22-8
 - administrator console, 22-2, 22-7
 - cluster, 22-2, 22-8
 - general setup, 22-2
 - JMS server, 22-16
 - managed server, 22-2, 22-8
 - modify policy file, 22-5
 - modify startup script, 22-5
 - policy file, weblogic.policy, 22-5
 - production environment, 22-2, 22-8
 - requirements, 22-1
 - start administration server, 22-5
 - start managed servers, 22-11
 - startup script, startWebLogic, 22-5
- WMI, 2-7
- WS container, G-5
- WS container/intermediary
 - registering secure, 9-5
- WS container/intermediary
 - register, 5-3
- WS management
 - enablement, 3-6
- ws security message processing inbound handler, 13-7
- ws security outbound handler, 13-6
- WSDL, G-5
 - import, 8-6
 - management service, 22-6, 22-14
 - multiple endpoints, 16-3
- WSDM, G-5
- WSM .NET agent
 - configure auditing publisher, 6-13
 - define business content alerts, 7-6
 - enable auditing, 6-12
- WSM agent, G-5
- WSM agent-based deployment scenarios, 3-11
- WSM agents, 2-7
 - configure SSL, 9-5
 - key store and trust store, 9-4
- WSM broker, 2-8, G-6
 - configure SSL, 9-5
 - key store and trust store, 9-3
 - profile data, 6-11
- WSM components
 - overview, 3-5
- WSM intermediary
 - enable auditing, 6-10
- WSM J2EE agent
 - configure auditing publisher, 6-13
 - define business content alerts, 7-5
 - enable auditing, 6-11
- WSM J2EE Agent
 - administration server install/deploy, 22-2
 - agent setup script, 22-4
 - environment setup script, 22-3
 - environment setup script, 22-10
 - host, 22-4
 - managed server install/deploy, 22-8
 - overview, 21-1
 - port, 22-4
 - run agent setup script, 22-4
 - software requirements, 22-1
 - starting, 21-1, 22-5, 22-11
 - un-deploy from administration server, 22-14
 - un-deploy from managed server, 22-15
 - verify autodiscovery, 22-7, 22-13
 - verify deployment, 22-6, 22-12
- wsm web application, 22-4
- WSMF, 3-6
- wsmf web application, 22-4, 22-6
- WS-Security, 17-3

X

XML, G-6

xml contract handler, 13-8

xml dispatch handler, 13-8

XML encryption, 17-3

XPath, 20-6

XPath monitoring handler, 13-8

XPL, 4-11

configure, 4-11

tools, 4-11

tracing, 4-13

XSLT handler, 13-9

