

HP OpenView Service Oriented Architecture Manager

Integrator Guide

Version: 2.1

Windows, HP-UX, Linux



June 2006

© Copyright 2004-2006 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 2004-2006 Hewlett-Packard Development Company, L.P.

Trademark Notices

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Linux is a U.S. registered trademark of Linus Torvalds

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 OpenView web site at:

<http://www.hp.com/managementsoftware/support>

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

HP OpenView 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 support site to:

- Search for knowledge documents of interest
- Submit enhancement requests online
- Download software patches
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter 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 log in. Many also require a support contract.

To find more information about access levels, go to:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

<http://www.managementsoftware.hp.com/passport-registration.html>

Table of Contents

1 Introduction	1-1
Document Overview	1-1
Audience	1-1
Prerequisites	1-2
Integration Points	1-2
General Integrations.....	1-2
Startup Tasks.....	1-3
Brokered Services.....	1-3
Java API Integrations	1-3
Security Customization	1-4
Database Integrations	1-4
Web Services Integrations	1-4
Interfaces Overview	1-5
Northbound (i.e. Offered) Interfaces	1-5
Southbound (i.e. Consumed) Interfaces	1-5
SOA Services Model Mapping to UDDI.....	1-6
Role of UDDI in the SOA	1-6
2 General Integration	2-1
Startup Tasks for Network Services and Broker.....	2-1
StartupTask Interface.....	2-1
Configuration File (server.xml)	2-2
Manually Implementing Brokered Services	2-2
Creating a Brokered Service JAR	2-3
Writing the Brokered Service Definition	2-3
Creating a Custom Handler	2-5

Adding Custom Handlers to a Brokered Service Definition.....	2-7
Deploying a Brokered Service Jar.....	2-7
3 Producing Custom SLA Reports	3-1
Overview	3-1
Database Schema Reference	3-2
MESSAGE_TRACE Table	3-2
MESSAGE Table	3-4
Example SLA Report.....	3-4
Sample SLA Reports for Month of Nov 2004	3-5
4 Security Customizations	4-1
Overview	4-1
Implementing Custom Authentication based on User Profile	4-2
Customizing the RealmAuthorizationHandler	4-2
Create an Implementation Class.....	4-2
Add the Implementation Class	4-4
Configuring the Broker Agent.....	4-4
Verify the Integration.....	4-4
Creating Custom Security Handlers.....	4-5
Create the Handlers.....	4-5
Add the Handlers to a Custom Brokered Service	4-6
Creating an XML Introspection Service	4-6
Create an XML Introspection Handler Class.....	4-7
Create an XML Introspection Brokered Service	4-8
5 Web Services API Reference	5-1
Retrieving SLO Data	5-2
ServicePerformanceSnapshotPT	5-2
queryPerformanceData.....	5-2
queryPerformanceDataResponse.....	5-3
Schema Types	5-6
PerformanceDataType.....	5-6
PerformanceWindowType	5-8

Modifying Business Service SLO Policy.....	5-9
ServiceMonitoringPolicyPT	5-9
setPolicy	5-9
getPolicy	5-10
Schema Types.....	5-13
SloPolicy.....	5-13
AlertPolicyType.....	5-14
Querying Audit Message Traces	5-16
AuditQueryPT	5-16
queryAuditTraces.....	5-16
retrieveMessageTrace	5-18
retrieveProfile.....	5-21
queryServiceReport	5-23
Schema Types.....	5-26
AuditQueryType.....	5-26
MessageTraceResultSet	5-27
MessageTraceType	5-28
ProfileTraceType	5-31
ServiceReportQueryType	5-32
ReportEntryType.....	5-33
ReportEntryListType	5-33
ServiceReportEntryType.....	5-34
ServiceReportListType	5-34
Importing/Exporting a Business Service.....	5-35
ImportExportPT	5-35
exportBusinessService	5-35
importBusinessService	5-36
Deleting a Business Service.....	5-38
LifecyclePT	5-38
delete	5-38
Sample Request	5-38
Sample Response	5-38
Managing Alerts	5-40
AlertService	5-40
Methods	5-40

Data Structures.....	5-40
sendAlert(AlertType Alert).....	5-41
Recovering From an Alert.....	5-43
queryAlerts(AlertQueryType query)	5-44
queryAlertProperties(int alertId)	5-46
deleteAlerts(AlertQueryType query)	5-47
deleteAlertsByIds(AlertIdListType AlertIdList).....	5-48

6 UDDI Data Mappings Reference 6-1

Overview	6-1
Business Service Mappings	6-2
Configuration Mappings	6-4
WS Intermediary/Container IT Service Mappings	6-6
IT Service Mappings	6-8
Dependency Relationships Mappings	6-9
tModel Definition	6-10
tModel Structure.....	6-10
Valid Values	6-10
Example	6-10
Managed Endpoint Relationships Mappings	6-11
tModel Definition	6-11
tModel Structure.....	6-12
Valid Values	6-12
Example	6-12
Organizational Contact Information Mappings	6-13
Contact Name	6-14
tModel Definition	6-14
tModel Structure	6-14
Valid Values.....	6-14
Example.....	6-14
Contact Distinguished Name	6-15
tModel Definition	6-15
tModel Structure	6-16
Valid Values.....	6-16

Example.....	6-16
Contact Email	6-17
tModel Definition	6-17
tModel Structure	6-17
Valid Values.....	6-17
Example.....	6-18
Contact Relationship.....	6-18
tModel Definition	6-19
tModel Structure	6-19
Valid Values.....	6-19
Example.....	6-19

Index

Introduction

The *Integrator Guide* is intended for users who want to take advantage of the HP OpenView Service Oriented Architecture (SOA) Manager's integration possibilities. SOA Manager Integration is often performed to either customize how the SOA Manager works or to reuse the SOA Manager's assets within other enterprise applications.

Much attention has been given to ensure that the SOA Manager's architecture is flexible and easy to integrate with. The architecture is pluggable and uses popular industry standards such as Java and XML. More important, the SOA Manager is among the first distributed management software products to use a services-based architecture. For example Web services are used to expose many parts of the SOA Manager's management model. This practice not only ensures standard and open integration possibilities, it clearly sets the SOA Manager within current and future SOA-based environments.

Document Overview

The SOA Manager Integrator Guide provides instructions for performing integrations with the SOA Manager. The guide provides a brief contextual overview of the customization and integration possibilities. Generally, each chapter is dedicated to an integration possibility and is self contained. Therefore, the book need not be read in sequence.

Every effort has been made to explain general concepts. However, much of the content in this guide assumes that the user is already familiar with the SOA Manager and has a basic level of experience. If you are new to the SOA Manager product, it is recommended that you first become familiar with the product. A good starting point is the SOA Manager Concept guide and the SOA Manager Tutorials. These and other documents are located in the distribution in the `/Documentation` directory.

Audience

The Integrator Guide is primarily intended for Solution Architects, Systems Integrators, and Application Developers who are responsible for integrating and enabling management components in their environments.

Prerequisites

Users should be familiar with the SOA Manager and Web Services Management (WSM) before completing the instructions provided in this guide. Users will require fundamental knowledge of the Java programming language and Java platform technologies including security. Users should have experience creating and using Web services, including fundamental knowledge of Web services standards such as XML, WSDL, SOAP, and UDDI.

Integration Points

The SOA Manager provides several integration points. The integration points can facilitate the deployment of Web services management solution as well as provide an opportunity to repurpose some of the management data that is collected for Web services. In some cases, there is also the ability to customize the solution with management features that are not supported with the solution, but important to fulfill the management needs of a Web service deployment.

In general, the integration points have been organized into the categories listed below. This is purely an organizational scheme and does not suggest any functional boundaries within the product.

- **General** – These integration points are used to manually create SOA Manager’s assets as well as customize the Network Services server and the WSM Broker Agent startup.
- **Database** – This integration point is used to create custom audit reports or to include audit information within other management applications.
- **Java API** – This integration point is used to create custom management handlers that address an organization’s specific Web service management needs.
- **Web Services Interfaces** – This integration point is used to create new (or augment existing) enterprise management applications by reusing the SOA Manager’s management data and management model. The data and model are exposed as management Web services.

General Integrations

General integrations make use of the SOA Manager’s flexible architecture in order to automate many tasks that are associated with using the software. General integrations provide the following benefits:

- Customize startup behaviors.
- Automate repetitive tasks.
- Save time when updating SOA Manager assets over multiple installations.
- Reuse current tool sets (i.e., IDE, Content Versioning System, etc...).
- Maintain current development processes.

Startup Tasks

The Network Services server and WSM Broker Agent are server processes that have the ability to execute user-defined Java code at startup. This is useful for initializing any required services when the processes are started or executing any type of process initialization code.

Integrators are responsible for creating the user-defined Java code as well as configuring each server to use the custom code. Configuration of startup classes is done in the servers' XML configuration file (`server.xml`).

Brokered Services

The WSM Broker's Broker Configurator tool is typically used to create brokered services, configure a brokered service's management handlers, and deploy brokered services. These steps can become repetitive and time consuming depending on the number of brokered services that are being deployed. However, these tasks can be completed without using the Broker Configurator.

Integrators are responsible for creating the brokered service definition file (a proprietary file written using XML), packaging the brokered service as a Java Archive (jar file), and copying it to the appropriate directory on the WSM Broker Agent host. Depending on the requirements, some or all of these tasks can be automated.

Java API Integrations

Policy handlers implement management logic that is used to interpose visibility and controls on Web services. These handlers are inserted in the HTTP or SOAP pipeline that is responsible for processing request and response messages. Brokered services use a set of standard handlers and can also be configured to use a set of simple or advanced handlers. However, an organization may have special management requirements that are not covered by any of the provided handlers.

In such cases, the SOA Manager's Java API can be used to create custom handlers that can implement any management or processing logic that is required.

Integrators are responsible for:

- Creating the custom handler Java logic by extending a base class
- Compiling the handler
- Configuring the handler in the brokered service definition file
- Packaging the handler in the broker service jar file

Security Customization

The SOA Manager's Java API integration point is used to create custom security implementations that allow an organization to enforce security policies that are not covered by the WSM Broker's default security features. This includes custom authentication based on user profile, custom security handlers, and an XML Introspection service.

Database Integrations

Web services audit trace messages are persisted to a central database. This information in conjunction with the model relationships which are also populated into the database (such as which Business Service contains which Web services) can be used to enable reporting and analytical applications such as SLA reporting, billing, non-repudiation, forecasting, etc. Any application can potentially connect to the SOA Manager database and make use of the data. For example, the database can be used to create audit reports using packages like Crystal Reports.

Integrators are responsible for creating and maintaining database connections from within their applications. In addition, integrators are responsible for upgrading their applications as the SOA Manager's database schema changes.

Web Services Integrations

The most common and robust method for integrating with the SOA Manager is by using the SOA Manager's published Web services interfaces. The Web services are SOAP based Web services that are defined using WSDL. The Web services follow standard Web services management protocols.



Two Web services management specifications are: Web Services Distributed Management (WSDM) and Web Services for Management (WS-Management). At present, the SOA Manager software utilizes the Web Services Management Framework (WSMF), which is an HP authored precursor to the standard WS management protocols. The SOA Manager software will support these specifications as they mature and stabilize.

Web services integrations allow system architects to leverage current management investments and provide a broader and more thorough view of the enterprise. In general, Web Services integrations are used to:

- Link the SOA Manager with other enterprise management products to create composite and custom applications
- Create and/or reuse current management consoles to display the SOA Manager's management data
- Create custom management consoles

Interfaces Overview

The SOA Manager's Web services interfaces can be categorized into two groups:

- **North Bound Interfaces** – Interfaces used to interact with the SOA Manager's management model.
- **South Bound Interfaces** – Interfaces used to interact with the SOA Manager's WSM Broker and WSM Agents (including the WS Container).

Northbound (i.e. Offered) Interfaces

The northbound interfaces include the entire Management Information Model. The interfaces include operations to:

- Create business Service
- List Business Services
- List IT Services
- Get Metrics, Status, and Alerts
- Query Audit Traces
- Dismiss Alerts
- Delete Business Services
- Status Change Events
- SLO Violation Alerts

The management information model is a set of Web services that are discoverable through meta-data populated in a UDDI registry.

Southbound (i.e. Consumed) Interfaces

The Southbound interfaces are used to interact with a managed WS Container/WSM Broker. The interfaces include operations to:

- Deploy Web services including configuration
- List Web Services
- Get Metrics
- Get Log Messages
- Reset Metrics
- Change Log Levels
- Get Business Content Alerts
- Get Audit Traces

The southbound interfaces include interfaces that are required to be exposed by Web services as defined by the WSMF Specification. The interfaces also include extension interfaces created by HP SOA Manager that implement metrics required for basic features that are not covered by the WSMF specification. Lastly, the interfaces include interfaces that are additional extensions defined only for the SOA Manager.



Contact the developer alliance team to get detailed documentation about the southbound interfaces.

SOA Services Model Mapping to UDDI

As described above, management interfaces are based on Web services. Like any Web services, the SOA Manager's Web services can be published to UDDI based on mapping recommendations.

The detailed mapping of the management information model to UDDI is described in Chapter 6 "UDDI Data Mappings Reference".

Role of UDDI in the SOA

SOA implementations use UDDI as a system of record. The positioning of the role of registry technology (such as UDDI) in an SOA has evolved over the past few years since its inception. Originally, the registry was conceived as a central discovery point for design-time and run-time reuse.

However, most recent thinking in this direction is that if the registry is used as the only way to offer and discover Web services in the SOA, it provides an excellent control point to achieve Governance during various stages of development, deployment, and runtime management. Business stakeholders and enterprise architects define various policies that must be adhered to in the enterprise SOA. These policies are captured and attached to various entities in UDDI.

The UDDI Registry is used to achieve the following:

- **Reuse** – capture meta-data about Web services as well as other technology assets so that effective search capabilities from various environments may be written against the registry.
- **Policy Definition** – capture various policy definitions that provide the ability for a business person or enterprise architect to mandate policies on various entities.
- **Capture SOA Environment Model** – capture various entities participating in an SOA; their relationships; and some meaningful subset of state information about these entities. This information is used to create a standardized/normalized information store that is used to support various IT governance processes.
- **Integration** – the agreement of various participants in the SOA to settle upon ontologies that are linked together in the UDDI Registry creates very good potential for different participants to consume and populate information out of UDDI for their own narrow domains. The registry then provides a central store to create the ultimate spider that links together all this information.

General Integration

General integrations make use of the SOA Manager's flexible architecture in order to automate many tasks that are associated with using the software. General integrations provide the following benefits:

- Customize startup behaviors
- Automate repetitive tasks
- Save time when updating SOA Manager's assets over multiple installations
- Reuse current tool sets (i.e., IDE, Content Versioning System, etc...)
- Maintain current development processes

Startup Tasks for Network Services and Broker

The Network Services and Broker have the ability to execute user-defined Java code at startup. This is useful for initializing any required services when the process is started, or for executing any other type of process initialization code. To have either Network Services or Broker execute startup code, a class must be created that implements the `com.hp.wsm.sn.server.StartupTask` interface. This startup class should be registered in the `service.xml` file in either the Network Service or Broker `/conf` directory.

StartupTask Interface

A startup class must implement the `com.hp.wsm.sn.server.StartupTask` interface in order to be executed. The `StartupTask` interface is as follows:

```
public interface StartupTask {
    public void startup(MipServer server) throws
        StartupTaskFailureException,
        LicenseException.LicenseRunTimeError,
        LicenseException.LicenseNotFound;
}
```

The class must be available on the classpath. When the server executes the startup task, it creates a new instance of the class and executes the startup method, passing in the current `com.hp.wsm.sn.server.MipServer` instance. The `MipServer` class is used for internal processing and is generally not used by user-defined startup tasks. The `MipServer` interface is not documented and is subject to change.

The startup class should throw a subclass of the `com.hp.wsm.sn.server.StartupTaskFailureException` class if a critical problem occurred and the server should shutdown.

Configuration File (server.xml)

The `server.xml` file is located in the `<install_dir>/conf/broker` and `<install_dir>/conf/networkservices` directories, respectively. Underneath the root `<server>` element, it has a `<startup>` element. This is the element in which the list of startup classes is contained. Each startup class should be specified within `<classname>` tags. For example:

```
<server>
  <startup>
    <classname>
      com.hp.wsm.sn.networkservices.NetworkServicesStartupTask
    </classname>
  </startup>
</server>
```

The startup tasks are executed in the order in which they appear in the startup list. User-defined startup tasks should follow the existing `com.hp.wsm.sn` startup tasks. While it is possible to put a startup task before the existing `NetworkServicesStartupTask` or `BrokerStartupTask`, it is not recommended.

Manually Implementing Brokered Services

When using a Broker-based deployment scenario, the Broker Configurator is typically used to create brokered services. This includes configuring a brokered service's management handlers. As an option, users can manually create brokered services and configure their management handlers. Brokered services are manually created and deployed when:

- You do not want to use the Broker Configurator.
- You want to develop and manage brokered services using your own development environment (i.e., IDE, Content Versioning System, development processes).
- You want to create custom brokered services.

The instructions in this section demonstrate how to manually create a brokered service and deploy it to the WSM Broker.

Creating a Brokered Service JAR

The artifacts of a brokered service are packaged as a JAR file. A good method for learning about brokered services is to create a broker service using the Broker Configurator and then inspecting the brokered service JAR file or using the files in the JAR as a template for creating your own brokered services. When using the Broker Configurator, the JAR file is written to the `<install_dir>/conf/broker`.

The archive contains two files which can be manually created:

- `service.wsdl` – This file contains a service’s definition without the address (endpoints) for the service. The service can be defined using either SOAP semantics, or can be created using straight XML.
- `service.xml` – This file contains a service’s endpoints and also the management capabilities that will be interposed for the service.

Writing the Brokered Service Definition

The brokered service definition (`service.xml`) is an XML-based file and can be created using a text editor or an XML editor. Among other things, the definition contains a service’s endpoints and references to any management handlers. The file can contain handlers that are included with the WSM solution, or can contain any custom handlers you create. The `service.xml` contains the following elements:

<service>

This is the root element of the brokered service definition. The following attributes are included as part of the `<service>` element.

- `class`: Defines the class used to create the brokered service. There are four possible classes that you can use:
 - `com.hp.wsm.sn.router.xml.SimpleSoapServiceFactory`: This class is used to create a brokered service for SOAP-based services as defined in the `service.wsdl` file. The brokered service can contain a predefined set of handlers often referred to as simple handlers.
 - `com.hp.wsm.sn.router.xml.SimpleXmlServiceFactory`: This class is used to create a brokered service for XML-based services as defined in the `service.wsdl` file. The brokered service can contain a predefined set of handlers often referred to as simple handlers.
 - `com.hp.wsm.sn.router.xml.CustomSoapServiceFactory`: This class is used to create a brokered service for SOAP-based services as defined in the `service.wsdl` file. The brokered service can contain a broad set of handlers and can also contain any custom handlers you create.
 - `com.hp.wsm.sn.router.xml.CustomXmlServiceFactory`: This class is used to create a brokered service for XML-based services as defined in the `service.wsdl` file. The brokered service can contain a broad set of handlers and can also contain any custom handlers you create.
- `name`: The brokered service's name
- `version`: A version for the brokered service

- namespace: A list of namespaces entered as *prefix=url*

The following definition creates a simple brokered service for a SOAP-based Web service:

```
<?xml version="1.0" encoding="UTF-8"?>
<service class="com.hp.wsm.sn.router.xml.SimpleSoapServiceFactory"
  name="financeServiceProxy" version="1.0"
  xmlns="com.hp.wsm.sn.router.persistence.servicedesc.v1"
  xmlns:ns1="urn:com.hp.wsm.sn.common.misc.utils"
  xmlns:ns2="http://wsm.hp.com/finance"
  xmlns:ns3="com.hp.wsm.sn.common.soap.dispatch.v1"
  ...
```

<transport>

The <transport> element is contained within the <service> element and is used to define the transport layer used to access the brokered service. It contains a reference to the transport provider as well as the http context that is used to access the brokered service and whether the transport layer is secured.



Only HTTP is supported.

The following example demonstrates the transport definition:

```
<transport
  provider="com.hp.wsm.sn.router.http.HttpTransportProvider">
  <ns1:property name="http.context" value="/financeServiceProxy"/>
  <ns1:property name="http.secure" value="false"/>
</transport>
```

<routing>

The <routing> element is contained within the <service> element and is used to bind the brokered service to a Web service endpoint. The dispatcher is used to send the request to final endpoint. The following example demonstrates the routing definition:

```
<routing>
  <entry binding="ns2:FinanceServiceSoapSoapBinding">
    <ns3:endpoint
      type="com.hp.wsm.sn.common.dispatcher.
        endpoint.http.HttpEndpointProvider">
      <ns1:property name="priority" value="primary"/>
      <ns4:address>
        http://15.40.235.105:8080/axis/services/FinanceServiceSoap
      </ns4:address>
    </ns3:endpoint>
  </entry>
</routing>
```

Handler Definitions

Each handler that you want to include for the brokered service must be referenced in the brokered service definition and be contained within the <service> element. Any properties for the handler must be included as attributes. The following example demonstrates the audit handler definition:

```
<audit
  classname="com.hp.wsm.sn.router.xml.handlers.audit.
    AuditHandlerFactory"
  includeProfiling="true"
  payload-filter="ALL"
  payload-option="REQUEST-RESPONSE"
</audit>
```



A reference of all the handlers can be found in the *WSM Broker Administrator Guide*.

Creating a Custom Handler

The following example demonstrates a custom handler:

```
Package com.mycompany.CustomHandler

import com.hp.wsm.sn.router.common.message.MessageServiceException;
import com.hp.wsm.sn.router.xml.XmlOperation;
import com.hp.wsm.sn.router.xml.handlers.BaseXmlHandler;
import java.io.IOException;

public class CustomHandler extends BaseXmlHandler {
  public void onRequest(XmlOperation operation) throws
    MessageServiceException {
    try {
      // Custom code goes here.
    }
    catch (IOException e) {
      throw new MessageServiceException.RequestReadException(e);
    }
  }
}
```



An `onResponse` method is also available when processing a response message.

An operation can be retrieved by using `getOperation()` or `getOperationAsString()` on the `XmlOperation` object.

Accessing Attachments for SOAP with Attachments

The Broker provides an API that can be used in custom handlers to access attachments that are part of a SOAP with Attachments (SwA) message. This API is part of the `com.hp.wsm.sn.router.xml.SoapOperation` class and is therefore only available for SOAP services.

To access the soap attachments of a service request in a custom broker handler:

- 1 Create a Handler class which extends `BaseXmlHandler` and overwrites the methods:


```
public void onRequest(XmlOperation operation)
public void onResponse(XmlOperation operation)
```
- 2 In the `onRequest` and/or `onResponse` method, verify that the `XmlOperation` parameter is an instance of `SoapOperation`.

- 3 If it is a `SoapOperation`, the attachments can be accessed by casting the `XmlOperation` to a `SoapOperation` and calling:

```
public Attachments getRequestAttachments()
public Attachments getResponseAttachments()
```

The `com.hp.wsm.sn.router.soap.Attachments` class that is returned from these operations has methods to retrieve body parts by index:

```
public BodyPart get(int i)
```

Or by name:

```
public BodyPart getAttachmentByPartName(String partName)
```

The returned type is `javax.mail.BodyPart`. Below is an example of a handler that accesses attachments from a SwA message:

```
package ...

import com.hp.wsm.sn.router.xml.handlers.BaseXmlHandler;
import com.hp.wsm.sn.router.xml.XmlOperation;
import com.hp.wsm.sn.router.xml.SoapOperation;
import com.hp.wsm.sn.router.common.message.
    MessageServiceException;
import com.hp.wsm.sn.router.soap.Attachments;

public class SoapAttachmentHandler extends BaseXmlHandler {
    public void onRequest(XmlOperation operation)
        throws MessageServiceException {
        // check that the operation is an instance of SoapOperation
        if (operation instanceof SoapOperation) {
            SoapOperation soapOp = (SoapOperation)operation;
            Attachments attachments =
                soapOp.getRequestAttachments();
        }
        // add code to manipulate the request attachment parts.
    }

    public void onResponse(XmlOperation operation)
        throws MessageServiceException.InternalError {
        // check that the operation is an instance of SoapOperation
        if (operation instanceof SoapOperation) {
            SoapOperation soapOp = (SoapOperation)operation;
            Attachments attachments =
                soapOp.getResponseAttachments();
        }
        // add code to manipulate the response attachment parts.
    }
}
```

Adding Custom Handlers to a Brokered Service Definition

Custom brokered services allow you to add your own custom handlers to a brokered service's handler chain. In order to add a custom handler, you must first create the custom brokered service and then edit the service's definition file located in the brokered service jar file.

To add a custom handler:

- 1 Uncompress `<install_dir>\conf\broker\<broker_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 (i.e., `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 brokered service including the custom handler class and any dependent classes.

Deploying a Brokered Service Jar

Brokered service JARs are placed in the `<install_dir>/conf/broker` directory. Any brokered service JARs that are in this directory are automatically deployed by the WSM Broker. When the JAR is deployed, it is automatically extracted to `<install_dir>/conf/broker` and placed in a directory that is named using the IP address of the host computer.



Never edit the extracted files as they will be overwritten when a new JAR is deployed.

Producing Custom SLA Reports

This chapter describes how to create SLA audit reports using the Network Service's audit database. The chapter provides a brief overview of the integration architecture and includes a reference of the schema and data dictionary of the audit database. An example of an SLA audit report is also provided. For more information on using the Web services API to access auditing information see Chapter 5 "Web Services API Reference".

Overview

SOA Manager has the ability to capture audit information about Web service messages into a central audit database. When a WS Container or WS Intermediary is registered with the Services Network, the Audit Service registers an event callback listener. Subsequently, the WS Container or WS Intermediary posts Audit Trace message lists back to the Audit Service at a configured interval. The Audit Service inserts these trace messages into the Audit database. This information in conjunction with the model relationships which are also populated into the database (such as which Business Service contains which Web services) can be used to enable reporting and analytical applications such as SLA reporting, billing, non-repudiation, forecasting, etc.

This chapter provides instructions for producing custom SLA reports based on the management data that is stored in the Audit database. For more information on using the API to access audit data, see Chapter 5 "Web Services API Reference".

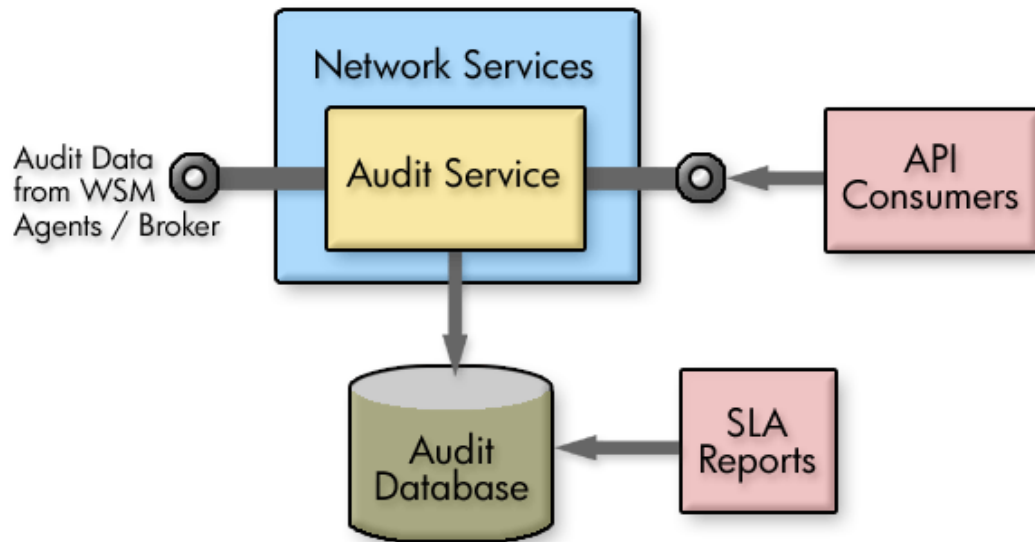


Figure 3-1: Audit Integration Architecture

Database Schema Reference

There are two tables that are used to create SLA Audit reports:

- MESSAGE_TRACE
- MESSAGE

MESSAGE_TRACE Table

Name	Type	Description
sequenceId	xsd:string	<p>An identifier that links together MessageTrace instances that represent hops between nodes (i.e. brokers and agents) for the same message call. This information is currently passed between nodes in the HTTP header SequenceHeader.</p> <p>If a node receives a message that has a SequenceHeader, it should use that value to populate the sequenceId field of the MessageTrace for that message.</p> <p>If a node receives a message that does not have a SequenceHeader, the node should generate a new globally unique sequenceId. It should use this value to populate the sequenceId field of the MessageTrace for that message.</p> <p>In either case, the node should also add a</p>

Name	Type	Description
		SequenceHeader HTTP header to any outgoing messages associated with this call and populate it with the appropriate sequenceId.
traceSourceId	xsd:string	A unique identifier that identifies the source of the message trace – the broker or agent that is sending the message trace to the CollectionService. The value is the host and port of the router/agent sending the trace.
traceSourceRole	xsd:string	A string indicating the role being played by the sender of this message trace with regards to the message being traced. Restricted to sender (if the agent is sending trace information regarding a message being sent to another service) or receiver (if the agent is sending trace information regarding a message that it has received).
serviceName	xsd:string	The name of the service being invoked. The value of this field is the name of the service element in the WSDL for this service.
serviceVersion	xsd:string	The version of the service being invoked. The value of this field is the namespace of the service element in the WSDL for this service.
portType	xsd:string	The WSDL portType being invoked.
operation	xsd:string	The WSDL operation being invoked.
binding	xsd:string	The WSDL binding being invoked.
serviceType	xsd:string	An indicator if the service is managed internally (proxy, agent) or is outside the service network. This field is restricted to the values internal and external.
transportType	xsd:string	The transport on which the message traveled. This is typically http or jms.
consumerSecurityPrincipal	xsd:string	The Principal invoking the service.
providerSecurityPrincipal	xsd:string	The Principal used by the services network on behalf of the consumer.
senderURI	xsd:string	A URI identifying the message source.
receiverURI	xsd:string	A URI identifying the message destination.

Name	Type	Description
duration	xsd:long	The duration in milliseconds of the call from the time the request was received by this node to when it was completed.
timestamp	xsd:dateTime	The time at which the call was received. This should be in UTC format ('Z' extension) with Millisecond (.sss) precision.
errorType	xsd:string	An indicator for the type of error that occurred, if any. This field is restricted to the values none, application, transport, timeout, marshalling, and soap fault.
requestSize	xsd:int	The size of the request in bytes.
responseSize	xsd:int	The size of the response in bytes

MESSAGE Table

This table contains actual message payloads.

Name	Type	Description
sequenceId	xsd:string	The UUID representing this message. This ID gets generated at each agent and populated into Audit trace record and is used to co-relate audit traces to other kinds of alerts.
requestMessage	CLOB	The payload body of a request message.
responseMessage	CLOB	The payload body of a response message.

Example SLA Report

You can build SLA reports from the SOA Manager Audit database using a reporting package like Crystal Reports. Refer to the Audit database schema and data dictionary to understand the source of this information.

From: MM/DD/YY HH:MM:SS

To: MM/DD/YY HH:MM:SS

Compute over interval: MONTH / WEEK / DAY / HOUR

Consumer: <securityPrincipal>

Consumed Service: <serviceNS>:<serviceName>

Number of Requests made:

Number of Requests processed successfully:

Number of Requests failed:

%age Availability over interval: (Number of Requests processed successfully / Number of Requests made) X 100

Max Response time:

Average Response time:

Sample SLA Reports for Month of Nov 2004

From: 11/01/04 00:00:00

To: 11/30/04 00:00:00

Compute over : DAY

Consumer: joebob

Consumed Service: http://wsm.hp.com/finance:financeServiceProxy

Time Window	Number of Requests	Number of Successes	Number of failures	% Availability	Ave Response (ms)	Max Response (ms)
11/01/04 – 11/02/04	35	24	11	68.57	274	360
11/02/04 – 11/03/04	45	44	1	97.77	248	423
...						

Consumed Service: http://wsm.hp.com/mobile:mobileServiceProxy

Time Window	Number of Requests	Number of Successes	Number of failures	% Availability	Ave Response (ms)	Max Response (ms)
11/01/04 – 11/02/04	29	29	0	100	128	175
11/02/04- 11/03/04	13	13	0	100	110	155
...						

Security Customizations

This chapter provides instructions for customizing the security features that are provided by the WSM Broker Agent. If you are not familiar with the Broker's security features, see Chapter 7 "Using the Broker's Security Features" in the *WSM Broker Administrator Guide* before completing these instructions. In addition, some of the steps in this chapter require creating custom handlers and custom services, for more information on these topics see Chapter 2 "General Integration".

Overview

It is possible to customize the WSM Broker Agent's security features including the integration with Select Access. In general, there are two use cases that are covered in this chapter:

- Passing custom query parameters as part of a Select Access authorization query
- Checking personalization attributes that are returned as a result of an authentication query

Both use cases can be implemented either at the message level or the transport level. Typically, the customization occurs within the policy handler chain by either creating custom handlers or customizing existing handlers.

Implementing Custom Authentication based on User Profile

It is possible to customize the WSM Broker Agent's security features including the integration with Select Access. In general, there are two use cases that are covered:

- Passing custom query parameters as part of a Select Access authorization query
- Checking personalization attributes that are returned as a result of an authentication query

Both use cases can be implemented either at the message level or the transport level. Typically, the customization occurs within the handler chain by either creating custom handlers or customizing existing handlers.

Customizing the RealmAuthorizationHandler

The `RealmAuthorizationHandler` interface offers callback methods that are used to add custom authorization policies when using HTTP transport level security. See Chapter 7 “Using the Broker's Security Features” in the *WSM Broker Administrator Guide* for instructions on setting up transport-level security.

There are four steps discussed in this section:

- Create a class that implements the `RealmAuthorizationHandler` interface.
- Add your implementation class to the `<install_dir>/lib/ext` directory.
- Configure the implementation class in the Broker's `MipServer.xml` configuration file.
- Verify the integration.

Create an Implementation Class

Create a class that implements the `RealmAuthorizationHandler` interface. Details about the interface are provided below, as well as an example.

Interface: RealmAuthorizationHandler

```
com.hp.wsm.sn.common.security.auth.jetty.RealmAuthorizationHandler
```


Methods

Method	Description
getContextParameters	<p>Context parameters are passed to security provider as context to an authorization request. This method returns string name value pairs representing context parameters.</p> <pre>public java.util.Properties getContextParameters (com.hp.wsm.sn.common.security.auth.ServiceRequest serviceRequest, org.mortbay.http.HttpServletRequest httpRequest)</pre> <ul style="list-style-type: none"> • serviceRequest – only the service name will be known at the transport level. • httpRequest – the current http request.
onAuthorize	<p>The security provider has successfully authorized the subject. Implement any custom authorization rules here, such as checking subject attributes. This method returns true if the subject is authorized to access the service.</p> <pre>public boolean onAuthorize (javax.security.auth.Subject subject, com.hp.wsm.sn.common.security.auth.ServiceRequest serviceRequest, org.mortbay.http.HttpServletRequest httpRequest)</pre> <ul style="list-style-type: none"> • subject - the authorized subject. • serviceRequest - only the service name will be known at the transport level. • httpRequest - the current http request.

Example

```
package com.hp.wsm.examples.auth;
import com.hp.wsm.sn.common.security.auth.jetty.
    RealmAuthorizationHandler;
import com.hp.wsm.sn.common.security.auth.ServiceRequest;
import com.hp.wsm.sn.common.security.auth.
    SubjectAttributesCredential;
import java.util.Properties;
import java.util.Set;
import org.mortbay.http.HttpServletRequest;
import javax.security.auth.Subject;

public class ExampleRealmAuthenticationHandler implements
    RealmAuthorizationHandler {
    public Properties getContextParameters(ServiceRequest serviceReq,
        HttpRequest request) {
        Properties context = new Properties();
        context.setProperty("contentLength",
            String.valueOf(request.getContentLength()));
    }
}
```

```

        return context;
    }

    public boolean onAuthorize(Subject subject, ServiceRequest
        serviceReq, HttpRequest request) {
        Set attributeCredentials = subject.getPublicCredentials
            (SubjectAttributesCredential.class);
        if (!attributeCredentials.isEmpty()) {
            SubjectAttributesCredential attributes =
                (SubjectAttributesCredential)
                    attributeCredentials.iterator().next();
            String roles = attributes.getSubjectAttributes()
                .getProperty("groups");
            if (roles != null) {
                return roles.indexOf("gold") > -1;
            }
        }
        return false;
    }
}

```

Add the Implementation Class

The implementation class must be located by the Security Provider at runtime. Typically, this means creating a JAR that contains the implementation class and adding it to the `<install_dir>/lib/ext` directory. This ensures that the class is located in the classpath of the Broker Agent.

Configuring the Broker Agent

The Broker Agent must be configured to use the implementation class at runtime. The configuration entry is added in the Broker's `mipServer.xml` file.

To configure the implementation class within the Broker Agent:

- 1 Use a text editor to open `<install_dir>/conf/broker/mipServer.xml`.
- 2 Add the property `"com.hp.mip.security.realmAuthHandler"` and set it to the fully qualified implementation class name. For instance, using the above example, the following entry would be used:

```

<entry name=com.hp.mip.security.realmAuthHandler>
    com.hp.wsm.examples.auth.ExampleAuthorizationHandler
</entry>

```

- 3 Save and close the File.

Verify the Integration

To verify the integration, you must create a custom brokered service that is configured to use transport-level security. See Chapter 7 “Using the Broker’s Security Features” in the *WSM Broker Administrator Guide* for instructions on setting up transport-level security.

Creating Custom Security Handlers

When using message-level security, custom handlers can be created to pass custom query parameters and check personalization attributes. The handlers are included as part of a custom XML brokered service's handler chain.

In this scenario two handlers are created. A context handler is used to setup context information before an authorization call (i.e., to Select Access) and an authorization handler is used to check any returned attributes. An example of these handlers is provided below. They are named `ExampleContextHandler` and `ExampleAuthorizationHandler` respectively.

There are two steps discussed in this section:

- Create the security handlers.
- Add the handlers to a custom brokered service definition

Create the Handlers

Create two handlers that extend `com.hp.wsm.sn.router.xml.handlers.BaseXmlHandler`. The handlers use the method `AuthorizationContext` available from the `XmlOperation` method that is passed into the `onRequest` method for an XML Handler. An example of both handlers is provided below:

ExampleContextHandler

```
package com.hp.mip.test.auth;

import com.hp.wsm.sn.router.common.message.MessageServiceException;
import com.hp.wsm.sn.router.xml.XmlOperation;
import com.hp.wsm.sn.router.xml.handlers.BaseXmlHandler;

import java.io.IOException;

public class ExampleContextHandler extends BaseXmlHandler {

    public void onRequest(XmlOperation operation) throws
        MessageServiceException {
        super.onRequest(operation);
        try {
            int length = operation.getAuditRequestData().getBytes().length;
            operation.getSecurityContext().getAuthorizationContext()
                .getContextProperties().setProperty("contentLength",
                    String.valueOf(length));
        }
        catch (IOException e) {
            throw new MessageServiceException.RequestReadException(e);
        }
    }
}
```

ExampleAuthorizationHandler

```

package com.hp.mip.test.auth;

import
com.hp.wsm.sn.common.security.auth.SubjectAttributesCredential;

import com.hp.wsm.sn.router.common.message.MessageServiceException;
import com.hp.wsm.sn.router.xml.XmlOperation;
import com.hp.wsm.sn.router.xml.handlers.BaseXmlHandler;

import javax.security.auth.Subject;
import java.util.Set;

public class ExampleAuthorizationHandler extends BaseXmlHandler {

    public void onRequest(XmlOperation operation) throws
        MessageServiceException {
        Subject subject = operation.getSecurityContext().getSubject();
        Set attributeCredentials =
subject.getPublicCredentials(SubjectAttributesCredential.class);
        String email = null;
        if (!attributeCredentials.isEmpty()) {
            SubjectAttributesCredential attributes =
                (SubjectAttributesCredential)
                    attributeCredentials.iterator().next();
            email =
                attributes.getSubjectAttributes().getProperty("email");
        }
        if (email == null || email.indexOf("hp.com") == -1) {
            throw new
                MessageServiceException.BadRequest.NotAuthorized("Subject is
                not from hp!", null);
        }
    }
}

```

Add the Handlers to a Custom Brokered Service

The custom security handlers must be added to a custom brokered service that is configured to use message-level security. In particular, the custom brokered service must contain the handler `ServiceSecurityInboundHandler`.

When adding the handlers to the handler chain, the context handler (i.e. `ExampleContextHandler`) must be located before `ServiceSecurityInboundHandler` and the authorization handler (i.e., `ExampleAuthorizationHandler`) must be located after `ServiceSecurityInboundHandler` in the handler chain.

Creating an XML Introspection Service

An XML Introspection service provides the ability to authorize services not only by user credentials but also by the request XML content, such as request XML message body or any XML documents. An XML Introspect service:

- Authenticates an XML service with HTTP Basic Authentication Only by using the Select Access Validator;
- Adds an XML message to service authorization context by a custom XML Introspection handler;
- Authorizes the service by sending the authenticated credential (which will be a SecureToken if returned by the Select Access Validator in the first authentication or the username and password credential) and the XML message in the authorization context to Select Access Validator.

Create an XML Introspection Handler Class

XML introspection is completed using a custom handler. Like all custom handlers, the handler must extend `BaseXmlHandler`. The handler is responsible for retrieving the XML document and applying security methods. The following example demonstrates an Introspection handler:

```
package com.hp.wsm.sn.common.security.auth.selectaccess;
import com.hp.wsm.sn.router.xml.handlers.BaseXmlHandler;
import com.hp.wsm.sn.router.xml.XmlOperation;
import com.hp.wsm.sn.router.common.message.MessageServiceException;
import com.hp.wsm.sn.common.security.common.SecurityContext;
import com.hp.wsm.sn.common.security.common.AuthorizationContext;
import org.w3c.dom.Document;
import java.util.Map;

public class XmlIntrospectionHandler extends BaseXmlHandler {
    /**
     * Handle service request.
     * @param operation
     * @throws MessageServiceException
     */
    public void onRequest(XmlOperation operation) throws
        MessageServiceException {

        super.onRequest(operation);
        // find the request XML Document
        final Document requestDoc = operation.getRequestDocument();
        final SecurityContext securityContext =
            operation.getSecurityContext();
        final AuthorizationContext authorizationContext =
            securityContext.getAuthorizationContext();
        Map params = authorizationContext.getServiceParameters();

        // put an XML document into authorization parameter list with
        // name "XML" which will be picked up by SelectAccess Security
        // Provider when service is being authorized against
        // SelectAccess validator.

        params.put("XML", requestDoc);
    }
}
```

Create an XML Introspection Brokered Service

The steps in this section create a brokered service and add the XML Introspection handler to the brokered service. If you are not familiar with custom brokered service and adding custom handlers to a brokered service, see Chapter 2 “Manually Implementing Brokered Services” section.

To create an XML introspection brokered service:

- 1 Open the Broker Configurator console, and create an XML Service. For example: `XmlIntrospectionService`.
- 2 When defining the service, select **Basic Authentication** and select the check box for **Authentication Only** in the Inbound Transport Authorization section.



Ensure that you have completed the steps for integrating Select Access and setting up basic authentication only for user authentication with the Broker.

- 3 Select any features as needed and click **finish**.
- 4 Convert the XML service to a Custom XML service and from the Edit Custom Service screen, select **Service Security Inbound Handler** from the Add a new handler drop-down list.
- 5 Click the **OK** to save the brokered service.
- 6 Find and extract the service JAR file under `<install_dir>/conf/broker` (i.e., `XmlIntrospectionService.jar`).
- 7 Edit the `service.xml` file and add the XML Introspection handler to the handler list before `ServiceSecurityInboundHandler`. For example:

```
<handler classname="com.hp.wsm.sn.router.xml.handlers.
LogHandler"/>
```

```
<handler classname="com.hp.wsm.sn.router.xml.handlers.inbound.
XmlContractHandler"/>
```

```
<handler classname="com.hp.wsm.sn.router.xml.handlers.outbound.
XmlDispatchHandler"/>
```

```
<handler classname="com.hp.wsm.sn.common.security.auth.selectaccess.
XmlIntrospectionHandler" />
```

```
<handler classname="com.hp.wsm.sn.router.xml.handlers.inbound.
ServiceSecurityInboundHandler" />
```

- 8 Re-package the service JAR including the XML Introspection handler. The handler must have the same fully qualified name as entered in the `service.xml` file.
- 9 Undeploy and deploy the brokered service for the changes to take effect.

Web Services API Reference

This chapter describes the management Web services interfaces that are available when performing integrations with the SOA Manager. The interfaces discussed in this chapter are considered northbound interfaces. Northbound interfaces are used to interact with the SOA Manager's management model.



This chapter does not cover southbound interfaces. Contact the developer alliance team to get detailed documentation about the southbound interfaces.

All management Web services are described using WSDL and adhere to the WSMF specification, which defines a method for exposing management data as Web services. At runtime, the WSDLs are published and available at `http://<host>:5002/wsmf/services`. Each Web service may contain multiple port types which in turn may contain multiple operations.

This chapter is organized by features that are provided by the following Web service port types:

- `ServicePerformanceSnapshotPT`
- `AuditQueryPT`
- `ImportExportPT`
- `LifecyclePT`
- `AlertService`

Retrieving SLO Data

The BSE is used to view performance metrics for a Web service that is contained in a business service. The BSE queries a business service's management Web service that contains operations for obtaining the performance metrics.

Any client can be used to obtain performance metrics. A business service's management Web service is typically used when:

- You do not want to use the BSE to view Web services performance metrics.
- You want to customize a different management console to aggregate Web services metrics.
- You want to use Web services metrics in a broader context such as a performance analysis program.

This section provides a description of the `ServicePerformanceSnapshotPT` port type which is used to get a Web service's metrics. The port type is located in a business service's management Web service. You can access a business service's management Web service using the following URL:

```
http://<host>:5002/wsmf/services/
runtime$service=WsmfBusinessService$name=<name>?wsdl
```

You must replace `<name>` with a business service name.



The `ServicePerformanceSnapshotPT` is only applicable if a business service contains a single Web service and cannot be used if a business service contains multiple Web services.

ServicePerformanceSnapshotPT

The `ServicePerformanceSnapshotPT` port type includes the following operations.

queryPerformanceData

This operation will return a summary of the performance of the Business Service. It includes performance statistics for the last polling interval, the lifetime of the service, the SLO interval and the intervals that are displayed in the BSE application. This operation takes no arguments.

Since a business service itself does not have performance metrics related to it, performance data is determined using the following algorithm:

- Check if the Business Service has an entry point. If the endpoint corresponds to a Web service or a Web service operation configuration, the performance metrics for this Web service or Web service operation is reported.
- If the endpoint is a Web service container configuration or a Web service intermediary configuration, the metrics corresponding to an arbitrary Web service configuration under this configuration is reported.

- If the business service has no endpoint, or has an endpoint that does not pertain to Web services (MOM, GRID, or Database), a Web service intermediary configuration is selected under the business service. If an intermediary is not available, a Web service container configuration is selected. From this container/intermediary, an arbitrary Web service is selected and its performance metrics are reported.
- If there are no container/intermediary configurations or Web services under the business service, the performance metrics are returned empty.



This method will not produce valid values until the SOA Manager has finished initializing and gathered enough data from the service. This will typically be after three complete polling loops.

Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ns1:queryPerformanceData xmlns:ns1=
      "http://openview.hp.com/xmlns/mip/2005/03/BusinessService" />
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

queryPerformanceDataResponse

```
<xsd:element name="queryPerformanceDataResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref=" mip-monitoring.xsd:PerformanceData"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The schema for the PerformanceData type is shown below and is broken down later in this document.

```
<s:complexType name="PerformanceWindowType">
  <s:sequence>
    <s:element name="SizeInMillis" type="s:long" />
    <s:element name="StartTime" type="s:dateTime" />
    <s:element name="EndTime" type="s:dateTime" />
    <s:element name="SuccessSamples" type="s:long" />
    <s:element name="SuccessMillisTotal" type="s:long" />
    <s:element name="FailureSamples" type="s:long" />
    <s:element name="FailureMillisTotal" type="s:long" />
    <s:element name="Min" type="s:long" />
    <s:element name="Max" type="s:long" />
    <s:element name="SecurityViolationCount" type="s:long" />
    <s:element name="AvailabilityPercentage" type="s:long" />
    <s:element name="SubWindow" type="tns:PerformanceWindowType"
      minOccurs="0" maxOccurs="unbounded" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:anyAttribute namespace="##other" />
</s:complexType>
```

```

<s:complexType name="PerformanceDataType">
  <s:sequence>
    <s:element name="Available" type="s:boolean" />
    <s:element name="UptimeMillis" type="s:long" />
    <s:element name="TimeOfLastPoll" type="s:long" />
    <s:element name="LastPoll" type="tns:PerformanceWindowType" />
    <s:element name="Lifetime" type="tns:PerformanceWindowType" />
    <s:element name="Slo" type="tns:PerformanceWindowType" />
    <s:element name="Window" type="tns:PerformanceWindowType"
      minOccurs="0" maxOccurs="unbounded" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
<s:anyAttribute namespace="##other" />
</s:complexType>

<s:element name="PerformanceWindow"
  type="tns:PerformanceWindowType"/>
<s:element name="PerformanceData" type="tns:PerformanceDataType"/>

```

Sample Response

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:BusinessService=
    http://openview.hp.com/xmlns/mip/2005/03/BusinessService
  xmlns:mip-monitoring.xsd=
    "http://openview.hp.com/xmlns/mip/2005/03/mip-monitoring.xsd">
  <soapenv:Body>
    <BusinessService:queryPerformanceDataResponse>
      <mip-monitoring.xsd:PerformanceData>
        <mip-monitoring.xsd:Available>
          true
        </mip-monitoring.xsd:Available>
        <mip-monitoring.xsd:UptimeMillis>
          54969
        </mip-monitoring.xsd:UptimeMillis>
        <mip-monitoring.xsd:TimeOfLastPoll>
          1119982577484
        </mip-monitoring.xsd:TimeOfLastPoll>
        <mip-monitoring.xsd:LastPoll>
          <mip-monitoring.xsd:SizeInMillis>
            60000
          </mip-monitoring.xsd:SizeInMillis>
          <mip-monitoring.xsd:StartTime>
            2005-06-28T11:15:17.484-07:00
          </mip-monitoring.xsd:StartTime>
          <mip-monitoring.xsd:EndTime>
            2005-06-28T11:16:17.484-07:00
          </mip-monitoring.xsd:EndTime>
          <mip-monitoring.xsd:SuccessSamples>
            0
          </mip-monitoring.xsd:SuccessSamples>
          <mip-monitoring.xsd:SuccessMillisTotal>
            0
          </mip-monitoring.xsd:SuccessMillisTotal>
          <mip-monitoring.xsd:FailureSamples>
            0
          </mip-monitoring.xsd:FailureSamples>
          <mip-monitoring.xsd:FailureMillisTotal>

```

```

    0
    </mip-monitoring.xsd:FailureMillisTotal>
    <mip-monitoring.xsd:Min>-1</mip-monitoring.xsd:Min>
    <mip-monitoring.xsd:Max>-1</mip-monitoring.xsd:Max>
    <mip-monitoring.xsd:SecurityViolationCount>
    0
    </mip-monitoring.xsd:SecurityViolationCount>
    <mip-monitoring.xsd:AvailabilityPercentage>
    100
    </mip-monitoring.xsd:AvailabilityPercentage>
  </mip-monitoring.xsd>LastPoll>
  <mip-monitoring.xsd:Lifetime>
  ...
  </mip-monitoring.xsd:Lifetime>
  <mip-monitoring.xsd:Slo>
  ...
  </mip-monitoring.xsd:Slo>
  <mip-monitoring.xsd:Window>
    <mip-monitoring.xsd:SizeInMillis>
    360000
    </mip-monitoring.xsd:SizeInMillis>
    <mip-monitoring.xsd:StartTime>
    2005-06-28T11:10:10-07:00
    </mip-monitoring.xsd:StartTime>
    <mip-monitoring.xsd:EndTime>
    2005-06-28T11:16:10-07:00
    </mip-monitoring.xsd:EndTime>
    <mip-monitoring.xsd:SuccessSamples>
    0
    </mip-monitoring.xsd:SuccessSamples>
    <mip-monitoring.xsd:SuccessMillisTotal>
    0
    </mip-monitoring.xsd:SuccessMillisTotal>
    <mip-monitoring.xsd:FailureSamples>
    0
    </mip-monitoring.xsd:FailureSamples>
    <mip-monitoring.xsd:FailureMillisTotal>
    0
    </mip-monitoring.xsd:FailureMillisTotal>
    <mip-monitoring.xsd:Min>-1</mip-monitoring.xsd:Min>
    <mip-monitoring.xsd:Max>-1</mip-monitoring.xsd:Max>
    <mip-monitoring.xsd:SecurityViolationCount>
    0
    </mip-monitoring.xsd:SecurityViolationCount>
    <mip-monitoring.xsd:AvailabilityPercentage>
    100
    </mip-monitoring.xsd:AvailabilityPercentage>
  <mip-monitoring.xsd:SubWindow>
    <mip-monitoring.xsd:SizeInMillis>
    60000
    </mip-monitoring.xsd:SizeInMillis>
    <mip-monitoring.xsd:StartTime>
    2005-06-28T11:15:10-07:00
    </mip-monitoring.xsd:StartTime>
    <mip-monitoring.xsd:EndTime>
    2005-06-28T11:16:10-07:00
    </mip-monitoring.xsd:EndTime>
    <mip-monitoring.xsd:SuccessSamples>
    0
    </mip-monitoring.xsd:SuccessSamples>
    <mip-monitoring.xsd:SuccessMillisTotal>
    0
    </mip-monitoring.xsd:SuccessMillisTotal>
    <mip-monitoring.xsd:FailureSamples>

```

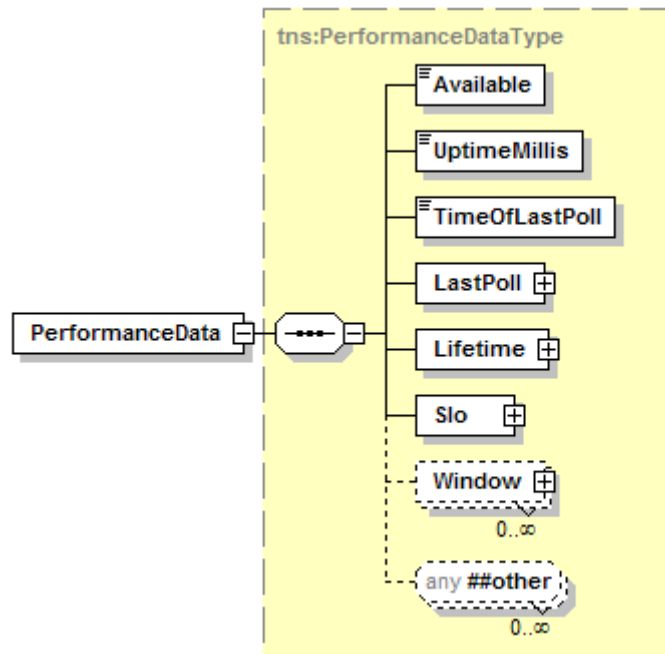
```

0
</mip-monitoring.xsd:FailureSamples>
<mip-monitoring.xsd:FailureMillisTotal>
0
</mip-monitoring.xsd:FailureMillisTotal>
<mip-monitoring.xsd:Min>
-1
</mip-monitoring.xsd:Min>
<mip-monitoring.xsd:Max>
-1
</mip-monitoring.xsd:Max>
<mip-monitoring.xsd:SecurityViolationCount>
0
</mip-monitoring.xsd:SecurityViolationCount>
<mip-monitoring.xsd:AvailabilityPercentage>
100
</mip-monitoring.xsd:AvailabilityPercentage>
</mip-monitoring.xsd:SubWindow>
...
</mip-monitoring.xsd:Window>
...
</mip-monitoring.xsd:PerformanceData>
</BusinessService:queryPerformanceDataResponse>
</soapenv:Body>
</soapenv:Envelope>

```

Schema Types

PerformanceDataType



```

<s:complexType name="PerformanceDataType">
  <s:sequence>
    <s:element name="Available" type="s:boolean" />
    <s:element name="UptimeMillis" type="s:long" />
    <s:element name="TimeOfLastPoll" type="s:long" />
    <s:element name="LastPoll" type="tns:PerformanceWindowType" />

```

```

    <s:element name="Lifetime" type="tns:PerformanceWindowType" />
    <s:element name="Slo" type="tns:PerformanceWindowType" />
    <s:element name="Window" type="tns:PerformanceWindowType"
      minOccurs="0" maxOccurs="unbounded" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
<s:anyAttribute namespace="##other" />
</s:complexType>

```

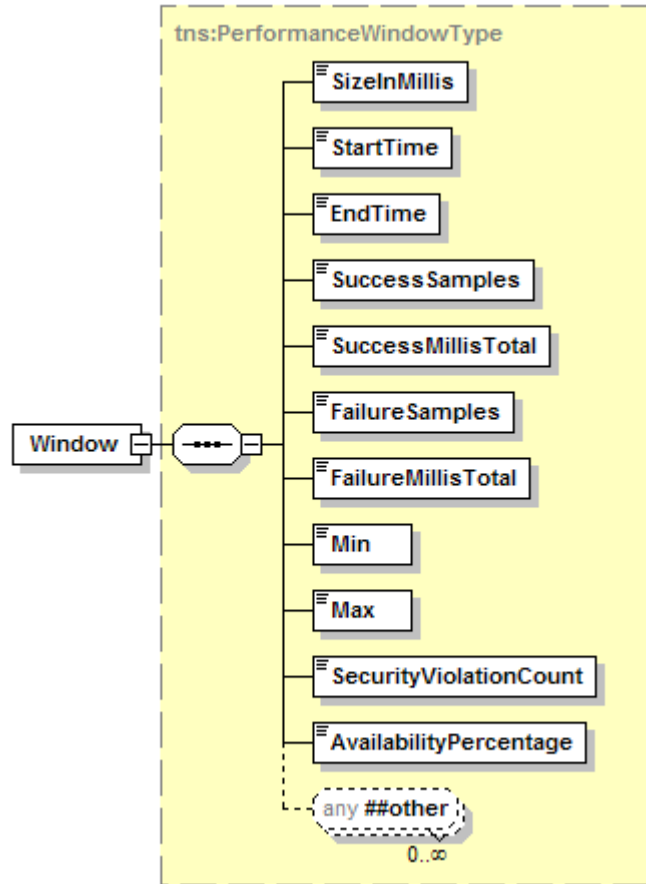
The `PerformanceDataType` element contains the following properties.

- **Available** – Indicates if the service is currently online.
- **UptimeMillis** – How long the service has been online.
- **TimeOfLastPoll** – The timestamp for the last poll.

The rest of the message contains multiple `PerformanceWindowType` elements.

- **LastPoll** – All the metrics measured in the last polling loop.
- **Lifetime** – Summary of all metrics gathered since running the server.
- **Slo** – All metrics gathered within the SLO time window.
- **Window** – The final `Window` property represents summaries of all metrics captured in the time windows shown in the BSE application performance grid. Each `Window` has six subwindows, each representing an equal percentage of the parent window.

PerformanceWindowType



```

<s:complexType name="PerformanceWindowType">
  <s:sequence>
    <s:element name="SizeInMillis" type="s:long" />
    <s:element name="StartTime" type="s:dateTime" />
    <s:element name="EndTime" type="s:dateTime" />
    <s:element name="SuccessSamples" type="s:long" />
    <s:element name="SuccessMillisTotal" type="s:long" />
    <s:element name="FailureSamples" type="s:long" />
    <s:element name="FailureMillisTotal" type="s:long" />
    <s:element name="Min" type="s:long" />
    <s:element name="Max" type="s:long" />
    <s:element name="SecurityViolationCount" type="s:long" />
    <s:element name="AvailabilityPercentage" type="s:long" />
    <s:element name="SubWindow" type="tns:PerformanceWindowType"
      minOccurs="0" maxOccurs="unbounded" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:anyAttribute namespace="##other" />
</s:complexType>

```

PerformanceWindowType contains the following properties.

- **SizeInMillis** – The size of this window in milliseconds.
- **StartTimeMillis** – The time, in milliseconds of the first sample in the dataset.
- **EndTimeMillis** – The time, in milliseconds of the last sample in the dataset.

► The start and end times may be the same if the window includes data from only one poll.

- **SuccessSamples** – The number of successful invocations on this service.
- **SuccessMillisTotal** – The duration in milliseconds of all the successful calls.

► Average duration per call is $\text{SuccessMillisTotal} / \text{SuccessSamples}$.

- **FailureSample** – The number of failed invocations on this service.
- **FailureMillisTotal** – The duration in milliseconds of all the failed calls.

► Average duration per call is $\text{FailureMillisTotal} / \text{FailureSamples}$.

- **Min** – The duration of the fastest call in milliseconds.
- **Max** – The duration of the slowest call in milliseconds.
- **SecurityViolationsCount** – The number of security violations.
- **AvailabilityPercentage** – Percentage of time this service was available.

Ranges from 0 – 100%

- **SubWindow** (optional) – Some windows are further broken down into subwindows. For example, the `Window` element in `PerformanceDataType` has six `SubWindows` representing a further breakdown of data. A one hour window may have six 10 minute subwindows.

Modifying Business Service SLO Policy

This WSMF API can be used to set the SLO policy for a business service. The policy of a business service corresponds to the policy of its endpoint. The SLO policy includes "breach" and "warning" values for each performance metric and also includes the ability to assign alert categories for SLO alerts.

ServiceMonitoringPolicyPT

The port type `ServiceMonitoringPT` includes the following operations.

setPolicy

This operation will set the SLO values for a business service's performance metrics. As part of the policy definition, you can also assign an alert category for SLO alerts.



Alert recipients for an alert category can only be defined using the BSE console.

Sample Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
  "http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ns1:setPolicy xmlns:ns1=
      "http://openview.hp.com/xmlns/mip/2005/03/BusinessService">
      <ns1:Policy xmlns:ns1=
        "http://openview.hp.com/xmlns/mip/2005/03
        /mip-monitoring.xsd">
        <ns1:SloPolicy>
          <ns1:Metric>TotalMessageCount</ns1:Metric>
          <ns1:Category>
            service.performance.Default_Breach
          </ns1:Category>
          <ns1:Enabled>true</ns1:Enabled>
          <ns1:Severity>
            http://schemas.hp.com/notification/severity/minor
          </ns1:Severity>
          <ns1:Type>LessThanOrEqualTo</ns1:Type>
          <ns1:Value>5</ns1:Value>
        </ns1:SloPolicy>
        <ns1:AlertPolicy>
          <ns1:Type>BusinessContent</ns1:Type>
          <ns1:Category>
            service.performance.Default_Breach
          </ns1:Category>
        </ns1:AlertPolicy>
      </ns1:Policy>
    </ns1:setPolicy>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response

```
<soapenv:Envelope xmlns:soapenv=
  http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:BusinessService=
    "http://openview.hp.com/xmlns/mip/2005/03/BusinessService">
  <soapenv:Body>
    <BusinessService:setPolicyResponse xsi:nil="true" />
  </soapenv:Body>
</soapenv:Envelope>
```

getPolicy

This operation returns the current SLO values that are set for a business service's performance metrics. This operation takes no properties.

Sample Request

```
<SOAP-ENV:Envelope
```



```

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <ns1:getPolicy xmlns:ns1="http://openview.hp.com/xmlns/mip
    /2005/03/BusinessService" />
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Sample Response

```

<soapenv:Envelope
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:BusinessService=http://openview.hp.com/xmlns/mip
    /2005/03/BusinessService
  xmlns:mip-monitoring.xsd="http://openview.hp.com/xmlns/mip
    /2005/03/mip-monitoring.xsd">
  <soapenv:Body>
    <BusinessService:getPolicyResponse>
      <mip-monitoring.xsd:Policy>
        <mip-monitoring.xsd:SloPolicy>
          <mip-monitoring.xsd:SloPolicy>
            <mip-monitoring.xsd:Metric>
              TotalMessageCount
            </mip-monitoring.xsd:Metric>
            <mip-monitoring.xsd:Category>
              service.performance.Default_Breach
            </mip-monitoring.xsd:Category>
            <mip-monitoring.xsd:Enabled>
              true
            </mip-monitoring.xsd:Enabled>
            <mip-monitoring.xsd:Severity>
              http://schemas.hp.com/notification
                /severity/minor
            </mip-monitoring.xsd:Severity>
            <mip-monitoring.xsd:Type>
              LessThanOrEqualTo
            </mip-monitoring.xsd:Type>
            <mip-monitoring.xsd:Value>
              5
            </mip-monitoring.xsd:Value>
          </mip-monitoring.xsd:SloPolicy>
        </mip-monitoring.xsd:SloPolicy>
      <mip-monitoring.xsd:SloPolicy>
        <mip-monitoring.xsd:SloPolicy>
          <mip-monitoring.xsd:Metric>
            TotalMessageCount
          </mip-monitoring.xsd:Metric>
          <mip-monitoring.xsd:Category>
            service.performance.Default_Breach
          </mip-monitoring.xsd:Category>
          <mip-monitoring.xsd:Enabled>
            true
          </mip-monitoring.xsd:Enabled>
          <mip-monitoring.xsd:Severity>
            http://schemas.hp.com/notification
              /severity/major
          </mip-monitoring.xsd:Severity>
          <mip-monitoring.xsd:Type>
            LessThanOrEqualTo
          </mip-monitoring.xsd:Type>
          <mip-monitoring.xsd:Value>
            10
          </mip-monitoring.xsd:Value>
        </mip-monitoring.xsd:SloPolicy>
      </mip-monitoring.xsd:SloPolicy>
    </BusinessService:getPolicyResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

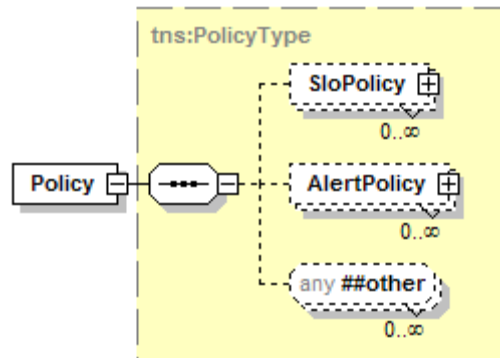
```

        </mip-monitoring.xsd:Value>
    </mip-monitoring.xsd:SloPolicy>
</mip-monitoring.xsd:SloPolicy>
<mip-monitoring.xsd:SloPolicy>
    <mip-monitoring.xsd:SloPolicy>
        <mip-monitoring.xsd:Metric>
            Available
        </mip-monitoring.xsd:Metric>
        <mip-monitoring.xsd:Category>
            service.performance.Default_Breach
        </mip-monitoring.xsd:Category>
        <mip-monitoring.xsd:Enabled>
            false
        </mip-monitoring.xsd:Enabled>
        <mip-monitoring.xsd:Severity>
            http://schemas.hp.com/notification
            /severity/major
        </mip-monitoring.xsd:Severity>
        <mip-monitoring.xsd:Type>
            EqualTo
        </mip-monitoring.xsd:Type>
        <mip-monitoring.xsd:Value>
            true
        </mip-monitoring.xsd:Value>
    </mip-monitoring.xsd:SloPolicy>
</mip-monitoring.xsd:SloPolicy>
<mip-monitoring.xsd:AlertPolicy>
    <mip-monitoring.xsd:AlertPolicy>
        <mip-monitoring.xsd:Type>
            BusinessContent
        </mip-monitoring.xsd:Type>
        <mip-monitoring.xsd:Category>
            service.performance.Default_Business_Content
        </mip-monitoring.xsd:Category>
    </mip-monitoring.xsd:AlertPolicy>
</mip-monitoring.xsd:AlertPolicy>
<mip-monitoring.xsd:AlertPolicy>
    <mip-monitoring.xsd:AlertPolicy>
        <mip-monitoring.xsd:Type>
            Error
        </mip-monitoring.xsd:Type>
        <mip-monitoring.xsd:Category>
            service.performance.Default_Breach
        </mip-monitoring.xsd:Category>
    </mip-monitoring.xsd:AlertPolicy>
</mip-monitoring.xsd:AlertPolicy>
<mip-monitoring.xsd:AlertPolicy>
    <mip-monitoring.xsd:AlertPolicy>
        <mip-monitoring.xsd:Type>
            Warning
        </mip-monitoring.xsd:Type>
        <mip-monitoring.xsd:Category>
            service.performance.Default_Breach
        </mip-monitoring.xsd:Category>
    </mip-monitoring.xsd:AlertPolicy>
</mip-monitoring.xsd:AlertPolicy>
</mip-monitoring.xsd:Policy>
</BusinessService:getPolicyResponse>
</soapenv:Body>
</soapenv:Envelope>

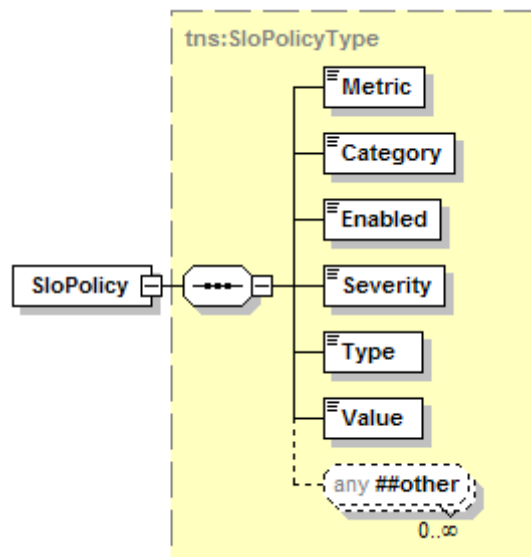
```

Schema Types

The PolicyType contains both the sloPolicy and the alertPolicy types.



SloPolicy



```
<s:complexType name="SloPolicyType">
  <s:sequence>
    <s:element name="Metric" type="s:string" />
    <s:element name="Category" type="s:string" />
    <s:element name="Enabled" type="s:boolean" />
    <s:element name="Severity" type="s:string" />
    <s:element name="Type" type="s:string" />
    <s:element name="Value" type="s:string" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:anyAttribute namespace="##other" />
</s:complexType>
```

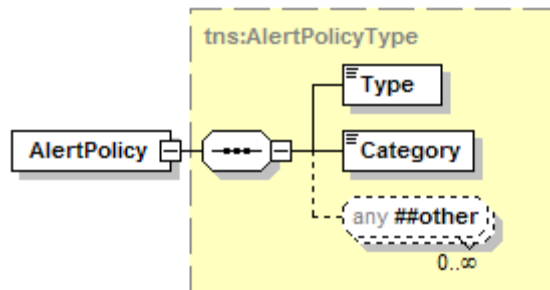
The `sloPolicyType` type element contains the following properties.

- **Metric** – The performance metric name to be set:
 - SuccessMessageCount

- AvailablePercentage
- MinResponseTime
- FailureMessageCount
- SecurityViolationCount
- TotalMessageCount
- MaxResponseTime
- AverageResponseTime
- UptimePercentage
- Available
- **Category** – The alert category that the SLO alert is sent to. Any alert category that is configured in the Network Services can be used. For example:
 - service.performance.Default_Breach
 - service.performance.Default_Warning

Alert recipients for an alert category can only be defined using the BSE.
- **Enabled** – Indicates whether this SLO is enabled. Valid entries are true and false.
- **Severity** – Whether this SLO definition is for a Warning or a SLO Breach:
 - http://schemas.hp.com/notification/severity/minor (*Warning*)
 - http://schemas.hp.com/notification/severity/major (*Breach*)
- **Type** – Each metric has a single comparator type which is one of the following:
 - LessThanOrEqualTo
 - GreaterThanOrEqualTo
 - EqualTo
- **Value** – The SLO Value you want to set for the given metric. The MinResponseTime, MaxResponseTime, and AverageResponseTime are entered as milliseconds. The AvailablePercentage and UptimePercentage metric are entered as a percentage.

AlertPolicyType



```

<s:complexType name="AlertPolicyType">
  <s:sequence>
    <s:element name="Type" type="s:string"/>
    <s:element name="Category" type="s:string"/>
  
```

```
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:anyAttribute namespace="##other" />
</s:complexType>
```

- **Type** – The SLO alert type (i.e., Warning).
- **Category** – The SLO alert category.

Querying Audit Message Traces

The Network Services server stores all audit trace messages in the SOA Manager database. The BSE is used to view the audit traces and also generates several reports based on the trace messages. Chapter 3 “Producing Custom SLA Reports” discusses using the database to create custom reports. The following section discusses how to retrieve audit trace messages programmatically using the Audit Web service.

Any client can use the Audit Web service to query audit trace messages. The service is typically used when:

- You do not want to use the BSE view audit trace messages.
- You want to customize a different management console to aggregate audit trace messages.
- You want to use audit trace messages in a broader context such as a billing or a non-repudiation program.

This section provides a description of the `AuditQueryPT` port type. The port type is located in the Audit Web service. You can access the Audit Web service using the following URL:

```
http://<host>:5002/wsmf/services/Runtime$service=AuditService?wsdl
```

AuditQueryPT

The `AuditQueryPT` port type includes the following operations which are detailed in this section:

- `queryAuditTraces`
- `retrieveMessageTrace`
- `retrieveProfile`
- `queryServiceReport`

queryAuditTraces

The schema elements for the request and response of this operation are shown below.

Request Schema

```
<xsd:element name="queryAuditTraces">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="mip-auditing.xsd:AuditQuery"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Response Schema

```
<xsd:element name="queryAuditTracesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="mip-auditing.xsd:MessageTraceResultSet" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Because auditing may generate a large number of message traces, the audit query has several options for filtering results. Also, the entire set of audit messages are not returned, but a cursor-like system is used to iterate through the large number of results that may be generated by a query.

Example request and response messages are shown below. The `AuditQuery` and `MessageTraceResultSet` types are explained in more detail the “Schema Types” section for this port type.

Sample Request

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ns1:queryAuditTraces
      xmlns:ns1="http://openview.hp.com/xmlns/mip/2005/03/AuditService">
      <ns1:AuditQuery
        xmlns:ns1="http://openview.hp.com/xmlns/mip/2005/03/mip-auditing.xsd">
        <ns1:Successes>true</ns1:Successes>
        <ns1:Failures>>false</ns1:Failures>
        <ns1:StartTime>2005-06-01T00:00:00</ns1:StartTime>
        <ns1:EndTime>2005-06-30T00:00:00</ns1:EndTime>
        <ns1:StartRow>0</ns1:StartRow>
        <ns1:NumRows>2</ns1:NumRows>
        <ns1:IncludeProfileTrace>>false</ns1:IncludeProfileTrace>
        <ns1:IncludePayload>>false</ns1:IncludePayload>
      </ns1:AuditQuery>
    </ns1:queryAuditTraces>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response

```
<soapenv:Envelope
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:AuditService=http://openview.hp.com/xmlns/mip/2005/03/AuditService
  xmlns:mip-auditing.xsd="http://openview.hp.com/xmlns/mip/2005/03/mip-auditing.xsd">
  <soapenv:Body>
    <AuditService:queryAuditTracesResponse>
      <mip-auditing.xsd:MessageTraceResultSet>
        <mip-auditing.xsd:StartRow>0</mip-auditing.xsd:StartRow>
        <mip-auditing.xsd:TotalRows>57</mip-auditing.xsd:TotalRows>
        <mip-auditing.xsd:MessageTraceList>
          <mip-auditing.xsd:MessageTrace>
            <mip-auditing.xsd:SequenceId>
```

```

152446098cd26d1041fa0239e15</mip-auditing.xsd:SequenceId>
  <mip-auditing.xsd:TraceSourceId />
  <mip-auditing.xsd:TraceSourceRole>
    traceSourceRole
  </mip-auditing.xsd:TraceSourceRole>
  <mip-auditing.xsd:ServiceName>
    FinanceService
  </mip-auditing.xsd:ServiceName>
  <mip-auditing.xsd:ServiceVersion>
    http://wsm.hp.com/finance
  </mip-auditing.xsd:ServiceVersion>
  <mip-auditing.xsd:PortType>
    FinanceServiceSoap
  </mip-auditing.xsd:PortType>
  <mip-auditing.xsd:Operation>
    QuoteRequest
  </mip-auditing.xsd:Operation>
  <mip-auditing.xsd:Binding />
  <mip-auditing.xsd:ServiceType />
  <mip-auditing.xsd:TransportType>
    HTTP/1.0
  </mip-auditing.xsd:TransportType>
  <mip-auditing.xsd:ConsumerSecurityPrincipal />
  <mip-auditing.xsd:ProviderSecurityPrincipal />
  <mip-auditing.xsd:SenderURI>
    15.244.60.98
  </mip-auditing.xsd:SenderURI>
  <mip-auditing.xsd:ReceiverURI>
    15.244.60.98:7001/FinanceService/FinanceService
  </mip-auditing.xsd:ReceiverURI>
  <mip-auditing.xsd:Duration>0</mip-auditing.xsd:Duration>
  <mip-auditing.xsd:Timestamp>
    2005-06-01T08:26:53.786-07:00
  </mip-auditing.xsd:Timestamp>
  <mip-auditing.xsd:ErrorType>
    none
  </mip-auditing.xsd:ErrorType>
  <mip-auditing.xsd:RequestSize>
    412
  </mip-auditing.xsd:RequestSize>
  <mip-auditing.xsd:ResponseSize>
    663
  </mip-auditing.xsd:ResponseSize>
  <mip-auditing.xsd:RequestPayload />
  <mip-auditing.xsd:ResponsePayload />
  <mip-auditing.xsd:ProfileTraceList />
</mip-auditing.xsd:MessageTrace>
  ...
</mip-auditing.xsd:MessageTraceList>
</mip-auditing.xsd:MessageTraceResultSet>
</AuditService:queryAuditTracesResponse>
</soapenv:Body>
</soapenv:Envelope>

```

retrieveMessageTrace

This operation is used to retrieve the details of a single message trace, identified by its `sequenceId`. The request and response schemas are shown below as well as example request and response messages.

Request Schema

```
<xsd:element name="retrieveMessageTrace">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="includeProfileTrace"
        type="xsd:boolean"/>
      <xsd:element name="sequenceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The `retrieveMessageTrace` type element contains the following properties:

- **retrieveMessageTrace/includeProfileTrace** – Set true to include detailed profiling with the returned message traces.
- **retrieveMessageTrace/sequenceId** – The `sequenceId` that identifies the requested message traceResponse Schema.

Response Schema

```
<xsd:element name="retrieveMessageTraceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="mip-auditing.xsd:MessageTrace"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Sample Request

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ns1:retrieveMessageTrace xmlns:ns1=
      "http://openview.hp.com/xmlns/mip/2005/03/AuditService">
      <ns1:includeProfileTrace>false</ns1:includeProfileTrace>
      <ns1:sequenceId>
        dea9a10107cfad78d0f94bb4689eac7e@ovh006.cup.hp.com
      </ns1:sequenceId>
    </ns1:retrieveMessageTrace>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response

```
<soapenv:Envelope
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:AuditService=
    http://openview.hp.com/xmlns/mip/2005/03/AuditService
  xmlns:mip-auditing.xsd=
    "http://openview.hp.com/xmlns/mip/2005/03/mip-auditing.xsd">
  <soapenv:Body>
    <AuditService:retrieveMessageTraceResponse>
      <mip-auditing.xsd:MessageTrace>
        <mip-auditing.xsd:SequenceId>
          dea9a10107cfad78d0f94bb4689eac7e@ovh006.cup.hp.com
        </mip-auditing.xsd:SequenceId>
      </mip-auditing.xsd:MessageTrace>
    </AuditService:retrieveMessageTraceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```

</mip-auditing.xsd:SequenceId>
<mip-auditing.xsd:TraceSourceId />
<mip-auditing.xsd:TraceSourceRole>
  traceSourceRole
</mip-auditing.xsd:TraceSourceRole>
<mip-auditing.xsd:ServiceName>
  financeServiceProxy
</mip-auditing.xsd:ServiceName>
<mip-auditing.xsd:ServiceVersion>
  http://wsm.hp.com/finance
</mip-auditing.xsd:ServiceVersion>
<mip-auditing.xsd:PortType>
  {http://wsm.hp.com/finance}FinanceServiceSoap
</mip-auditing.xsd:PortType>
<mip-auditing.xsd:Operation>
  getQuote
</mip-auditing.xsd:Operation>
<mip-auditing.xsd:Binding>
  FinanceServiceSoap
</mip-auditing.xsd:Binding>
<mip-auditing.xsd:ServiceType />
<mip-auditing.xsd:TransportType />
<mip-auditing.xsd:ConsumerSecurityPrincipal>
  bob
</mip-auditing.xsd:ConsumerSecurityPrincipal>
<mip-auditing.xsd:ProviderSecurityPrincipal />
<mip-auditing.xsd:SenderURI>
  15.244.60.98
</mip-auditing.xsd:SenderURI>
<mip-auditing.xsd:ReceiverURI>
  http://ovh006.cup.hp.com:9032/financeServiceProxy
</mip-auditing.xsd:ReceiverURI>
<mip-auditing.xsd:Duration>1525</mip-auditing.xsd:Duration>
<mip-auditing.xsd:Timestamp>
  2005-06-03T10:49:46.104-07:00
</mip-auditing.xsd:Timestamp>
<mip-auditing.xsd:ErrorType>
  none
</mip-auditing.xsd:ErrorType>
<mip-auditing.xsd:RequestSize>
  922
</mip-auditing.xsd:RequestSize>
<mip-auditing.xsd:ResponseSize>
  700
</mip-auditing.xsd:ResponseSize>
<mip-auditing.xsd:RequestPayload>
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><soap:Header><wsa:Action>http://wsm.hp.com/f
inance/getQuote</wsa:Action><wsa:From><wsa:Address>
http://schemas.xmlsoap.org/ws/2003/03/addressing/role/anonymous</ws
sa:Address><wsa:From><wsa:MessageID>uuid:dd624794-
e042-417a-8a76-
f445e3d09f5f</wsa:MessageID><wsa:To>http://ovh006.cup.hp.
com:9032/financeServiceProxy?WSDL</wsa:To><wsu:Timestamp>
<wsu:Created>2005-06-
03T17:41:39Z</wsu:Created><wsu:Expires>2005-06-
03T17:46:39Z</wsu:Expires><wsu:Timestamp></soap:Heade
r><soap:Body><QuoteRequest

```

```

xmlns="http://wsm.hp.com/Finance/Request"><symbol>HPQ</symbol></QuoteRequest></soap:Body></soap:Envelope>
</mip-auditing.xsd:RequestPayload>
  <mip-auditing.xsd:ResponsePayload><?xml version="1.0"
encoding="UTF-8"?>
<env:Envelope
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><env:Body><n1:QuoteResponse
xmlns:n1="http://wsm.hp.com/Finance/Response"><n1:getQuoteResult
xmlns:n1="http://wsm.hp.com/Finance/Response"><n1:change
xmlns:n1="http://wsm.hp.com/Finance/Response">-
0.25</n1:change><n1:percentChange
xmlns:n1="http://wsm.hp.com/Finance/Response">-
20.0</n1:percentChange><n1:price
xmlns:n1="http://wsm.hp.com/Finance/Response">1.0</n1:price>
</n1:getQuoteResult></n1:QuoteResponse></env:Body><
lt;/env:Envelope></mip-auditing.xsd:ResponsePayload>
  <mip-auditing.xsd:ProfileTraceList />
  </mip-auditing.xsd:MessageTrace>
</AuditService:retrieveMessageTraceResponse>
</soapenv:Body>
</soapenv:Envelope>

```

retrieveProfile

This operation is used to retrieve detailed profile traces generated from the processing of a single message. The request and response schemas are shown below as well as example request and response messages. The `ProfileTraceType` is explained in more detail in the “Schema Types” section for this port type.

Request Schema

```

<xsd:element name="retrieveProfile">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="correlationId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

The `retrieveProfile` type element contains the following properties:

- **retrieveProfile/correlationId** – The `correlationId` shared by profile traces from the same message. This value is the same as the `sequenceId` of the message trace from the same message.
- **retrieveProfileResponse/ProfileTraceList** – List of profile traces.

Response Schema

```

<xsd:element name="retrieveProfileResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="mip-auditing.xsd:ProfileTraceList"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

Sample Request

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ns1:retrieveProfile xmlns:ns1=
      "http://openview.hp.com/xmlns/mip/2005/03/AuditService">
      <ns1:correlationId>
        315e738b467da0c4be90a098954554c9@ovw017.cup.hp.com
      </ns1:correlationId>
    </ns1:retrieveProfile>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Sample Response

```

<soapenv:Envelope
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:AuditService=
    http://openview.hp.com/xmlns/mip/2005/03/AuditService
  xmlns:mip-auditing.xsd=
    "http://openview.hp.com/xmlns/mip/2005/03/mip-auditing.xsd">
  <soapenv:Body>
    <AuditService:retrieveProfileResponse>
      <mip-auditing.xsd:ProfileTraceList>
        <mip-auditing.xsd:ProfileTrace>
          <mip-auditing.xsd:ProfileId>
            5e643a3bb7da14c7b6d836420b9d69d5
          </mip-auditing.xsd:ProfileId>
          <mip-auditing.xsd:CorrelationId>
            315e738b467da0c4be90a098954554c9@ovw017.cup.hp.com
          </mip-auditing.xsd:CorrelationId>
          <mip-auditing.xsd:StartTime>
            2005-06-28T15:05:30-07:00
          </mip-auditing.xsd:StartTime>
          <mip-auditing.xsd:EndTime>
            2005-06-28T15:05:32-07:00
          </mip-auditing.xsd:EndTime>
          <mip-auditing.xsd:Actor>
            com.hp.wsm.sn.router.http.handlers.
            message.MessageDispatchHandler
          </mip-auditing.xsd:Actor>
          <mip-auditing.xsd:Success>
            true
          </mip-auditing.xsd:Success>
          <mip-auditing.xsd:Context />
        </mip-auditing.xsd:ProfileTrace>
        ...
        <mip-auditing.xsd:ProfileTrace>
          <mip-auditing.xsd:ProfileId>
            808519471718ab7f9ee5e792e48c72b1
          </mip-auditing.xsd:ProfileId>
          <mip-auditing.xsd:ParentId>
            5e643a3bb7da14c7b6d836420b9d69d5
          </mip-auditing.xsd:ParentId>
          <mip-auditing.xsd:CorrelationId>
            315e738b467da0c4be90a098954554c9@ovw017.cup.hp.com
          </mip-auditing.xsd:CorrelationId>
          <mip-auditing.xsd:StartTime>
            2005-06-28T15:05:32-07:00
          </mip-auditing.xsd:StartTime>

```

```

        <mip-auditing.xsd:EndTime>
            2005-06-28T15:05:32-07:00
        </mip-auditing.xsd:EndTime>
        <mip-auditing.xsd:Actor>
            com.hp.wsm.sn.router.xml.XmlMessageWriter
        </mip-auditing.xsd:Actor>
        <mip-auditing.xsd:Success>
            true
        </mip-auditing.xsd:Success>
        <mip-auditing.xsd:Context />
    </mip-auditing.xsd:ProfileTrace>
</mip-auditing.xsd:ProfileTraceList>
</AuditService:retrieveProfileResponse>
</soapenv:Body>
</soapenv:Envelope>

```

queryServiceReport

This operation queries the audit table based on the request and returns the response. The request and response schemas are shown below as well as example request and response messages.

Request Schema

```

<s:complexType name="ServiceReportQueryType">
  <s:sequence>
    <s:element name="FromTime" type="s:dateTime"/>
    <s:element name="ToTime" type="s:dateTime"/>
    <s:element name="Consumer" type="s:string"/>
    <s:element name="ServiceName" type="s:string"/>
    <s:element name="ServiceNamespaceURI" type="s:string"
      nillable="true"/>
    <s:element name="ComputationInterval" type="s:string"/>
  </s:sequence>
</s:complexType>

```

ServiceReportQueryType contains the following properties:

- **FromTime** – Start date for the audit query.
- **ToTime** – End date for the audit query.
- **Consumer** – Consumer Name. Empty element is equivalent to all consumers.
- **ServiceName** – Service Name. Empty element is equivalent to all service.
- **ServiceNamespaceURI** – NamespaceURI of the service.
- **ComputationInterval** – Valid values are DAY, MONTH, WEEK, HOURS.

Response Schema

The response schema has 4 sections.

- **ReportEntry** – There will be more than one ReportEntry depending on FromTime, ToTime, and ComputationInterval. ReportEntry contains AverageResponseTime, MaxResponse in milliseconds, NumberofSuccess, NumberofFailures, and NumberofRequest.
- **ReportEntryList** – ReportEntryList is a container element for ReportEntry.

- **ServiceReportEntry** – There can be more than one ServiceReportEntry element. If the query is for all the services, then there will be a ServiceReportEntry element for each service. The ServiceReportEntry contains ServiceName and ReportEntryList.
- **ServiceReportList** – This is the top level element. There is always one ServiceReportList irrespective of whether the query fetched any result or not.

```

<s:complexType name="ReportEntryType">
  <s:sequence>
    <s:element name="FromTime" type="s:dateTime" />
    <s:element name="ToTime" type="s:dateTime" />
    <s:element name="NumberOfRequests" type="s:long" />
    <s:element name="NumberOfSuccess" type="s:long" />
    <s:element name="NumberOfFailures" type="s:long" />
    <s:element name="PercentageAvailability" type="s:decimal" />
    <s:element name="AverageResponseTime" type="s:long" />
    <s:element name="MaxResponseTime" type="s:long" />
    <s:element name="MinimumResponseTime" type="s:long" />
  </s:sequence>
</s:complexType>
<s:complexType name="ReportEntryListType">
  <s:sequence>
    <s:element name="ReportEntry" type="tns:ReportEntryType"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
</s:complexType>
<s:complexType name="ServiceReportEntryType">
  <s:sequence>
    <s:element name="ServiceName" type="s:string" />
    <s:element name="ReportEntryList"
      type="tns:ReportEntryListType" />
  </s:sequence>
</s:complexType>
<s:complexType name="ServiceReportListType">
  <s:sequence>
    <s:element name="ServiceReportEntry"
      type="tns:ServiceReportEntryType" maxOccurs="unbounded" />
  </s:sequence>
</s:complexType>
<s:element name="ServiceReportList"
  type="tns:ServiceReportListType" />

```

Sample Request Message

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ns1:queryServiceReport xmlns:ns1=
      "http://openview.hp.com/xmlns/mip/2005/03/AuditService">
      <ns1:ServiceReportQuery xmlns:ns1=
        "http://openview.hp.com/xmlns/mip
          /2005/03/mip-auditing.xsd">
        <ns1:FromTime>2005-06-01T00:00:00</ns1:FromTime>
        <ns1:ToTime>2005-06-30T00:00:00</ns1:ToTime>
        <ns1:Consumer />
        <ns1:ServiceName />
        <ns1:ServiceNamespaceURI />
        <ns1:ComputationInterval>MONTH</ns1:ComputationInterval>
      </ns1:ServiceReportQuery>
    </ns1:queryServiceReport>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Sample Response Message

```

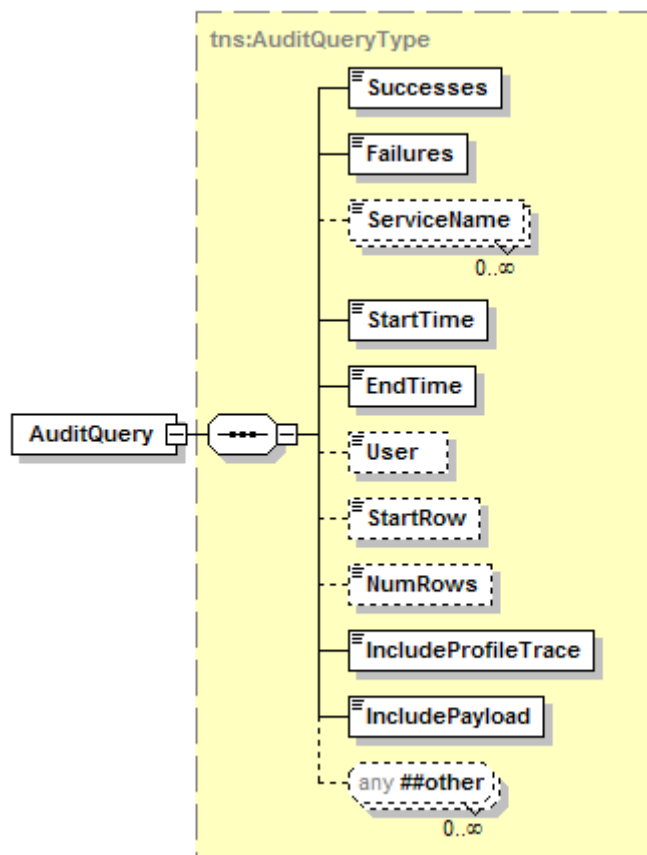
<soapenv:Envelope
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:AuditService="http://openview.hp.com/xmlns
    /mip/2005/03/AuditService"
  xmlns:mip-auditing.xsd="http://openview.hp.com/xmlns/mip
    /2005/03/mip-auditing.xsd">
  <soapenv:Body>
    <AuditService:queryServiceReportResponse>
      <mip-auditing.xsd:ServiceReportList>
        <mip-auditing.xsd:ServiceReportEntry>
          <mip-auditing.xsd:ServiceName>
            financeServiceProxy
          </mip-auditing.xsd:ServiceName>
          <mip-auditing.xsd:ServiceNamespaceURI>
            http://wsm.hp.com/finance
          </mip-auditing.xsd:ServiceNamespaceURI>
          <mip-auditing.xsd:ReportEntryList>
            <mip-auditing.xsd:ReportEntry>
              <mip-auditing.xsd:FromTime>
                2005-05-31T17:00:00-07:00
              </mip-auditing.xsd:FromTime>
              <mip-auditing.xsd:ToTime>
                2005-06-29T17:00:00-07:00
              </mip-auditing.xsd:ToTime>
              <mip-auditing.xsd:NumberofRequests>
                35
              </mip-auditing.xsd:NumberofRequests>
              <mip-auditing.xsd:NumberofSuccess>
                32
              </mip-auditing.xsd:NumberofSuccess>
              <mip-auditing.xsd:NumberofFailures>
                3
              </mip-auditing.xsd:NumberofFailures>
              <mip-auditing.xsd:PercentageAvailability>
                91
              </mip-auditing.xsd:PercentageAvailability>
              <mip-auditing.xsd:AverageResponseTime>
                534
              </mip-auditing.xsd:AverageResponseTime>
              <mip-auditing.xsd:MaxResponseTime>
                2652
              </mip-auditing.xsd:MaxResponseTime>
              <mip-auditing.xsd:MinimumResponseTime>
                15
              </mip-auditing.xsd:MinimumResponseTime>
            </mip-auditing.xsd:ReportEntry>
          </mip-auditing.xsd:ReportEntryList>
        </mip-auditing.xsd:ServiceReportEntry>
      </mip-auditing.xsd:ServiceReportList>
    </AuditService:queryServiceReportResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Schema Types

AuditQueryType

Auditing may generate a large number of message traces saved in the database. The AuditQueryType is used to filter audit queries to return result sets of a usable size. The query type includes several types that are optional.



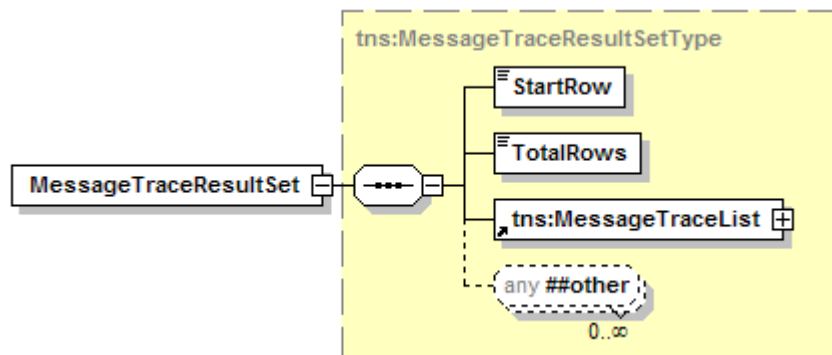
```
<s:complexType name="AuditQueryType">
  <s:sequence>
    <s:element name="Successes" type="s:boolean" />
    <s:element name="Failures" type="s:boolean" />
    <s:element name="ServiceName" type="s:string"
      minOccurs="0" maxOccurs="unbounded" />
    <s:element name="StartTime" type="s:dateTime" />
    <s:element name="EndTime" type="s:dateTime" />
    <s:element name="User" type="s:string" minOccurs="0" />
    <s:element name="StartRow" type="s:int" minOccurs="0" />
    <s:element name="NumRows" type="s:int" minOccurs="0" />
    <s:element name="IncludeProfileTrace" type="s:boolean" />
    <s:element name="IncludePayload" type="s:boolean" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:anyAttribute namespace="##other" />
</s:complexType>
```

This type contains the following properties:

- **AuditQueryType/Successes** – Set true to return messages that returned successfully.
- **AuditQueryType/Failures** – Set true to return messages that generated faults. Both successes and failures can be returned at once to return all messages.
- **AuditQueryType/ServiceName** – This element can be used to select audit traces generated by one or more services, specifying their WSDL service names.
- **AuditQueryType/StartTime** – Filter the audit query to only include messages that were processed after the specified time.
- **AuditQueryType/EndTime** – Filter the audit query to only include messages that were processed before the specified time.
- **AuditQueryType/User** – Only return authenticated messages that sent by the specified principal.
- **AuditQueryType/StartRow** – If the query generates multiple results, return the result set starting at the specified row.
- **AuditQueryType/NumRows** – The maximum number of message traces to return.
- **AuditQueryType/IncludeProfileTrace** – Set true to include detailed profiling with the returned message traces. This will significantly increase the size of the response, and the length of time the query takes to execute.
- **AuditQueryType/IncludePayload** – Set true to include the message request and response payloads. As with including profiling, this will have a significant impact on response times.

MessageTraceResultSet

This type is used to return large sets of message traces returned from audit queries. It provides a “cursor” mechanism for iterating through large results sets with acceptable performance.



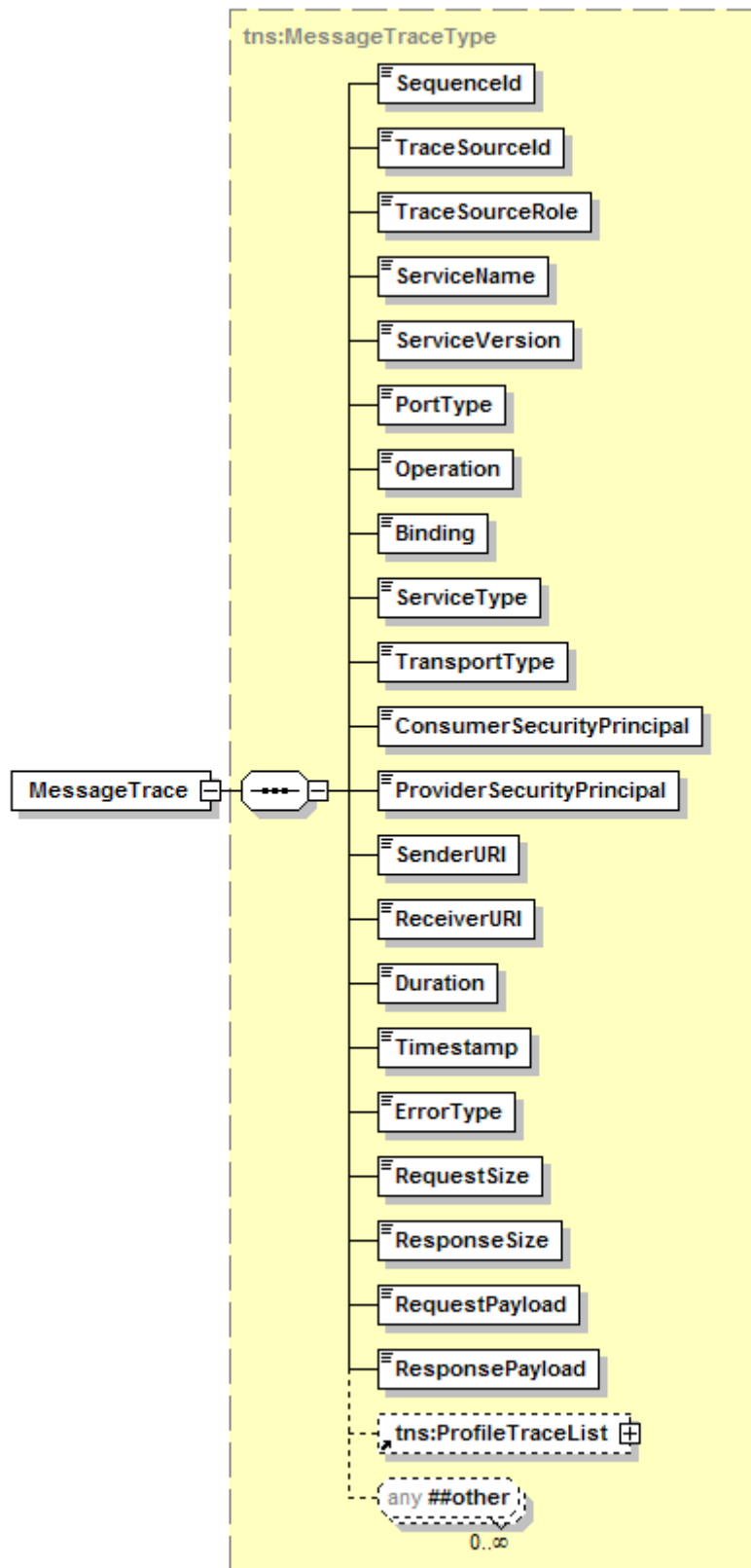
```
<s:complexType name="MessageTraceResultSetType">
  <s:sequence>
    <s:element name="StartRow" type="s:int" />
    <s:element name="TotalRows" type="s:int" />
    <s:element ref="tns:MessageTraceList" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:anyAttribute namespace="##other" />
</s:complexType>
```

This type contains the following properties:

- **MessageTraceResultSet/StartRow** – The index that the result set begins at.
- **MessageTraceResultSet/TotalRows** – This specifies the total number of rows in the result. The result set is a subset of these message traces.
- **MessageTraceResultSet/MessageTraceList** – The list of message traces.

MessageTraceType

This type contains contextual information relating to a single message.



```

<s:complexType name="MessageTraceType">
  <s:sequence>
    <s:element name="SequenceId" type="s:string" />

```

```

    <s:element name="TraceSourceId" type="s:string" />
    <s:element name="TraceSourceRole" type="s:string" />
    <s:element name="ServiceName" type="s:string" />
    <s:element name="ServiceVersion" type="s:string" />
    <s:element name="PortType" type="s:string" />
    <s:element name="Operation" type="s:string" />
    <s:element name="Binding" type="s:string" />
    <s:element name="ServiceType" type="s:string" />
    <s:element name="TransportType" type="s:string" />
    <s:element name="ConsumerSecurityPrincipal" type="s:string" />
    <s:element name="ProviderSecurityPrincipal" type="s:string" />
    <s:element name="SenderURI" type="s:string" />
    <s:element name="ReceiverURI" type="s:string" />
    <s:element name="Duration" type="s:long" />
    <s:element name="Timestamp" type="s:dateTime" />
    <s:element name="ErrorType" type="s:string" />
    <s:element name="RequestSize" type="s:int" />
    <s:element name="ResponseSize" type="s:int" />
    <s:element name="RequestPayload" type="s:string" />
    <s:element name="ResponsePayload" type="s:string" />
    <s:element ref="tns:ProfileTraceList" minOccurs="0" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:anyAttribute namespace="##other" />
</s:complexType>

```

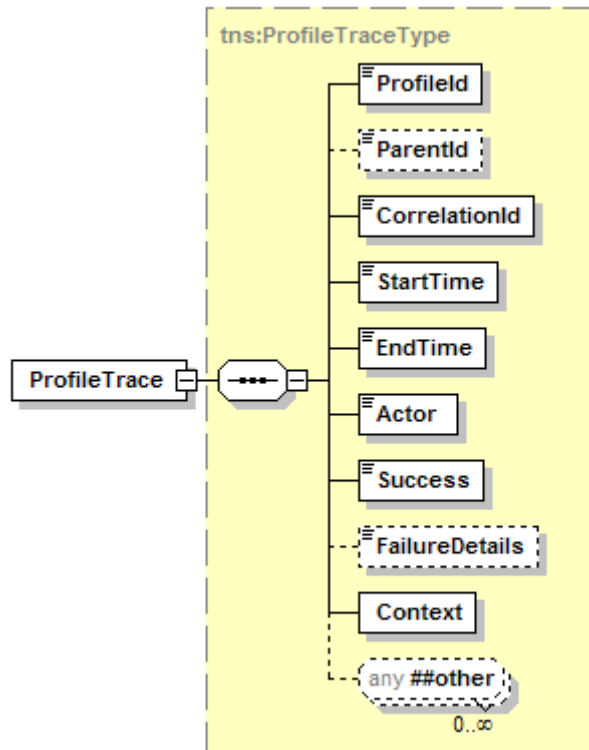
This type contains the following properties:

- **MessageTraceType/SequenceId** – UUID representing this message. This ID gets generated at each agent and populated into Audit trace record and is used to correlate audit traces to other kinds of alerts.
- **MessageTraceType/TraceSourceId** – Unique identifier that identifies the source of the message trace – the router or agent that is sending the message trace to the Audit Service.
- **MessageTraceType/ServiceName** – The name of the service being invoked. The value of this field is populated using the local name of the service element in the WSDL for this service.
- **MessageTraceType/ServiceVersion** – The namespace of the service being invoked. The value of this field is the namespace of the service element in the WSDL for this service.
- **MessageTraceType/PortType** – The WSDL port type being invoked.
- **MessageTraceType/Operation** – The WSDL operation being invoked.
- **MessageTraceType/Binding** – The WSDL binding being invoked.
- **MessageTraceType/ConsumerSecurityPrincipal** – The Principal invoking the service.
- **MessageTraceType/SenderURI** – Remote IP address of the message sender.
- **MessageTraceType/Duration** – The duration in milliseconds of the call from the time the request was received by this node to when response was returned.
- **MessageTraceType/Timestamp** – Time at which the message was processed.
- **MessageTraceType/ErrorType** – Indicates the type of error that occurred, if any. This field is restricted to the values ‘none’, ‘application’, ‘transport’, ‘internal’.
- **MessageTraceType/RequestSize** – The size of the request in bytes.

- **MessageTraceType/ResponseSize** – The size of the response in bytes.
- **MessageTraceType/RequestPayload** – Payload body of request message.
- **MessageTraceType/ResponsePayload** – Payload body of the response message.

ProfileTraceType

A profile trace is a detailed trace from a single actor, for example an XML Handler, taking part in the processing of a message. A single message usually generates multiple profile traces. The parentId and startTime elements can be used to reconstruct the structure and order of related profile traces.



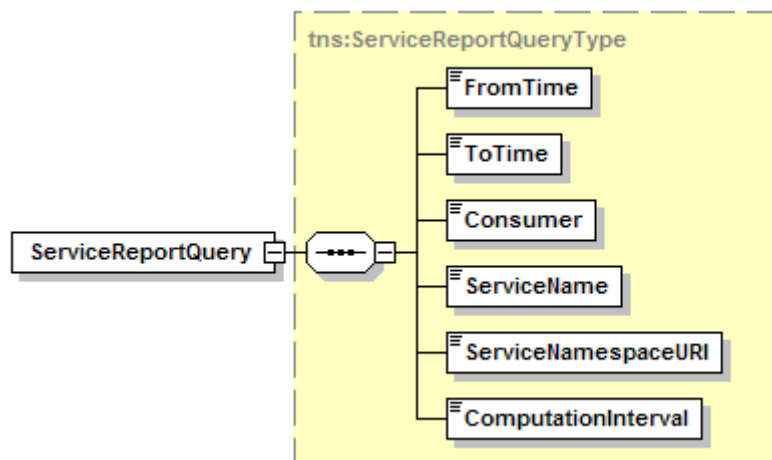
```
<s:complexType name="ProfileTraceType">
  <s:sequence>
    <s:element name="ProfileId" type="s:string" />
    <s:element name="ParentId" type="s:string" minOccurs="0" />
    <s:element name="CorrelationId" type="s:string" />
    <s:element name="StartTime" type="s:dateTime" />
    <s:element name="EndTime" type="s:dateTime" />
    <s:element name="Actor" type="s:string" />
    <s:element name="Success" type="s:boolean" />
    <s:element name="FailureDetails" type="s:string"
      minOccurs="0" />
    <s:element name="Context" type="common:PropertiesType" />
    <s:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:anyAttribute namespace="##other" />
</s:complexType>
```

This type contains the following properties:

- **ProfileTraceType/ProfileId** – UUID for this profile trace.

- **ProfileTraceType/ParentId** – The ID of the parent profile trace if this trace executes within the context of another profile trace.
- **ProfileTraceType/CorrelationId** – Correlates profile traces from the same message. The sequenceId from the message trace is typically used as the correlationId.
- **ProfileTraceType/StartTime** – The time the specified actor begins executing the message.
- **ProfileTraceType/EndTime** – The time the specified actor finishes executing the message.
- **ProfileTraceType/Actor** – The class or component being profiled.
- **ProfileTraceType/Success** – Returns true if the profile was successful, false if an exception was thrown.
- **ProfileTraceType/FailureDetails** – Contains detailed information of any failure.
- **ProfileTraceType/Context** – Any context for this profile as name/value pairs.

ServiceReportQueryType



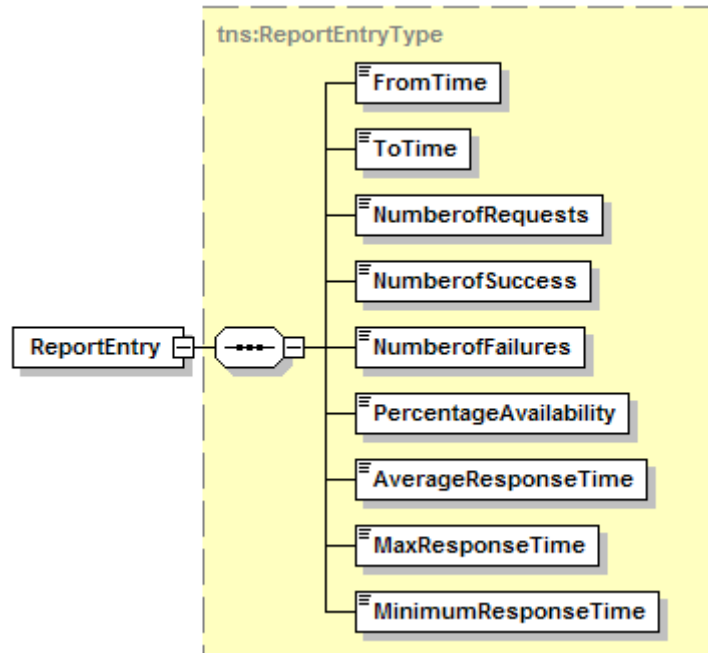
```
<s:complexType name="ServiceReportQueryType">
  <s:sequence>
    <s:element name="FromTime" type="s:dateTime"/>
    <s:element name="ToTime" type="s:dateTime"/>
    <s:element name="Consumer" type="s:string"/>
    <s:element name="ServiceName" type="s:string"/>
    <s:element name="ServiceNamespaceURI" type="s:string"
      nillable="true"/>
    <s:element name="ComputationInterval" type="s:string"/>
  </s:sequence>
</s:complexType>
```

ServiceReportQueryType contains the following properties:

- **FromTime** – Start date for the audit query.
- **ToTime** – End date for the audit query.
- **Consumer** – Consumer Name. Empty element is equivalent to all consumers.
- **ServiceName** – Service Name. Empty element is equivalent to all service.

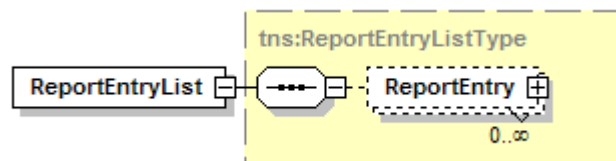
- **ServiceNamespaceURI** – NamespaceURI of the service.
- **Computation Interval** – Valid values are DAY, MONTH, WEEK, HOURS.

ReportEntryType



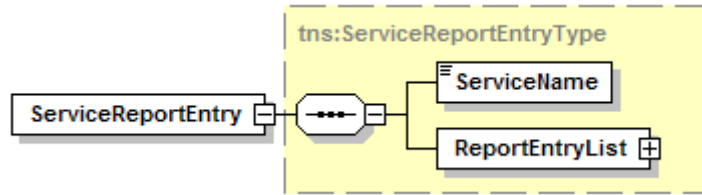
```
<s:complexType name="ReportEntryType">
  <s:sequence>
    <s:element name="FromTime" type="s:dateTime"/>
    <s:element name="ToTime" type="s:dateTime"/>
    <s:element name="NumberOfRequests" type="s:long"/>
    <s:element name="NumberOfSuccess" type="s:long"/>
    <s:element name="NumberOfFailures" type="s:long"/>
    <s:element name="PercentageAvailability" type="s:decimal"/>
    <s:element name="AverageResponse" type="s:decimal"/>
    <s:element name="MaxResponse" type="s:decimal"/>
    <s:element name="MinResponseTime" type="s:decimal"/>
  </s:sequence>
</s:complexType>
```

ReportEntryListType



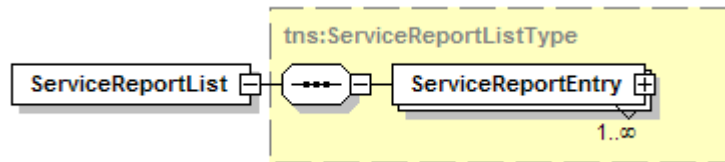
```
<s:complexType name="ReportEntryListType">
  <s:sequence>
    <s:element name="ReportEntry" type="tns:ReportEntryType"
      minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

ServiceReportEntryType



```
<s:complexType name="ServiceReportEntryType">
  <s:sequence>
    <s:element name="ServiceName" type="s:string"/>
    <s:element name="ReportEntryList"
      type="tns:ReportEntryListType"/>
  </s:sequence>
</s:complexType>
```

ServiceReportListType



```
<s:complexType name="ServiceReportListType">
  <s:sequence>
    <s:element name="ServiceReportEntry"
      type="tns:ServiceReportEntryType" minOccurs="0"
      maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```


Importing/Exporting a Business Service

The Network Services server maintains all service models in the Service Catalog. The BSE is typically used to create a service model and also contains a feature for importing and exporting service models to and from the Service Catalog. This section discusses how to import or export service models using the Service Catalog Web service.

Any client can use the Service Catalog Web service to import or export a service model. The service is typically used when:

- You create a service model using a tool other than the BSE.
- You do not want to use the BSE to import or export service models.
- You want to customize a different management console to aggregate service models.
- You want to reuse a service model in a broader management application.

This section provides a description of the `ImportExportPT` port type. The port type is located in the Service Catalog Web service. You can access the Service Catalog Web service using the following URL:

```
http://<host>:5002/wsmf/services/Runtime$service=WsmfServiceCatalog.<host>?wsdl
```

ImportExportPT

The operations in this port type are used for importing and exporting a business service. The following operations are available for this port type:

exportBusinessService

This operation takes `businessService` name as input, and returns the URL of the exported business Service.

Request Schema

```
<xsd:element name="exportBusinessService">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="param0" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

- **exportBusinessService/param0** – Business Service name

Response Schema

```
<xsd:element name="exportBusinessServiceResponse"
  type="xsd:string"/>
```

- **exportBusinessServiceResponse** – This element is of type `xsd:string`. It contains the URL of the exported Business Service.

Sample Request Message

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ns1:exportBusinessService xmlns:ns1=
      "http://openview.hp.com/xmlns/mip/2005/03/ServiceCatalog">
      <ns1:param0>FinanceBS</ns1:param0>
    </ns1:exportBusinessService>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response Message

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ServiceCatalog=
    "http://openview.hp.com/xmlns/mip/2005/03/ServiceCatalog">
  <soapenv:Body>
    <ServiceCatalog:exportBusinessServiceResponse>
      http://ovw017.cup.hp.com:5002/businessServiceRepository/
      export/FinanceBS
    </ServiceCatalog:exportBusinessServiceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

importBusinessService

This operation takes URL from where the businessService can be imported.

Request Schema

```
<xsd:element name="importBusinessService">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="param0" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

- **importBusinessService/param0** – URL of the Business Service to be imported

Response Schema

```
<xsd:element name="importBusinessServiceResponse"
  type="xsd:string"/>
```

- **importBusinessServiceResponse** – This element is of type `xsd:string`. If the operation succeeds, then it returns "success", otherwise it returns `SOAP:Fault`.

Sample Request Message

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ns1:importBusinessService xmlns:ns1=
      "http://openview.hp.com/xmlns/mip/2005/03/ServiceCatalog">
      <ns1:param0>
        http://ovw017.cup.hp.com:5002/businessServiceRepository
        /export/FinanceBS
      </ns1:param0>
    </ns1:importBusinessService>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Sample Response Message

```

<soapenv:Envelope
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ServiceCatalog=
    "http://www.hp.com/mso/mip/2004/02/ServiceCatalog">
  <soapenv:Body>
    <ServiceCatalog:importttBusinessServiceResponse>
      success
    </ServiceCatalog:importttBusinessServiceResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Deleting a Business Service

The Network Services server maintains all service models in the Service Catalog. The BSE is typically used to delete service models from the Service Catalog. This section discusses how to delete service models using a business service's management Web service.

Any client can use a business service's management Web service to delete a service model. The service is typically used when:

- You do not want to use the BSE to delete service models.
- You want to customize a different management console to delete service models.

This section provides a description of the LifecyclePT port type. The port type is located in a business service's management Web service. You can access the management Web service using the following URL:

```
http://<host>:5002/wsmf/services/  
runtime$service=WsmfBusinessService$name=<name>?wsdl
```

You must replace <name> with a business service name.

LifecyclePT

The LifecyclePT port type includes the following operation.

delete

This operation will delete the Business Service, along with any contained brokered or managed services and their associated alerts. It will also remove the Business Service from UDDI registry.

Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope  
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/  
  xmlns:xsd=http://www.w3.org/2001/XMLSchema  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  <soapenv:Body>  
    <delete xmlns="http://openview.hp.com/xmlns/mip  
      /2005/03/BusinessService" />  
  </soapenv:Body>  
</soapenv:Envelope>
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope  
  xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/  
  xmlns:xsd=http://www.w3.org/2001/XMLSchema  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
```

```
xmlns:BusinessService=  
  "http://openview.hp.com/xmlns/mip/2005/03/BusinessService">  
<soapenv:Body>  
  <BusinessService:deleteResponse xsi:nil="true" />  
</soapenv:Body>  
</soapenv:Envelope>
```

Managing Alerts

The Network Services server contains an alerting infrastructure. Among other things, the alerting infrastructure stores all alert messages in the SOA Manager database and is used to send and acknowledge alert messages. This section discusses how to manage alerts using the Alert Web service.

Any client can use the Alert Web service to manage alert messages. The service is typically used when:

- You do not want to use the BSE to manage alert messages.
- You want to customize a different management console to manage the SOA Manager alerts.
- You want to leverage the SOA Manager's alerting infrastructure in other applications.
- You want to generate your own alerts and send them to the SOA Manager.
- You want to use the SOA Manager alerts in a broader context such as SLA monitoring.

This section provides a description of the `AlertService` port type. The port type is located in the Alert Web service. You can access the Alert Web service using the following URL:

```
http://<host>:5002/wsmf/services/Runtime$service=AlertService?wsdl
```

AlertService

The `AlertService` port type uses the following methods, data structures, and operations.

Methods

```
[document/literal]
void sendAlert(AlertType Alert)

[document/literal]
void deleteAlerts(AlertQueryType AlertQuery)

[document/literal]
AlertList queryAlerts(AlertQueryType AlertQuery)

[document/literal]
PropertyList queryAlertProperties(int param0)

[document/literal]
void deleteAlertsByIds(AlertIdListType AlertIdList)
```

Data Structures

```
NameValuePairType
{
```

```

        string          Name
        string          Value
    }

NameValuePairType  NameValuePair

PropertiesType
{
    NameValuePairType[]  NameValuePair
}

PropertiesType  Properties

AlertType
{
    string          Name
    anyURI          Source
    string          Category
    string          Message
    string          Severity
    dateTime        CreationTime
    int             Id
    PropertiesType  Properties
    any[]
}

AlertType  Alert

AlertListType
{
    AlertType[]  Alert
}

AlertQueryType
{
    anyURI          Source
    string          Severity
    dateTime        StartTime
    dateTime        EndTime
    int             StartRow
    int             NumRows
    any[]
}

AlertIdListType
{
    int[]          Id
}

SelectExpression
{
    string          SelectExpression
}

AlertListType  AlertList

AlertListType  AlertList

```

sendAlert(AlertType Alert)

This operation sends an alert to the AlertService. The alert will be saved in the database and be available for subsequent queries. SendAlert has no response.

Alerts can be sent for any managed object. The state of that managed object will change based on the alert severity. If you send a severe alert to an object it will be considered in a severe state until the alert is acknowledged or a recovery alert is received.

Below is a sample alert for a fictional server at `http://some_server`. This alert is flagging the alert system that the `server_status` for `http://some_server` is now severe due to an `alert_down_alert` event.

Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:mrns0="http://openview.hp.com/xmlns/mip
    /2005/03/mip-monitoring.xsd"
  xmlns:mrns1="http://openview.hp.com/xmlns/mip
    /2005/03/mip-common.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tns="http://openview.hp.com/xmlns/mip/2005/03/AlertService"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <tns:sendAlert>
      <mrns0:Alert>
        <mrns0:Name>server_down_alert</mrns0:Name>
        <mrns0:Source>http://some_server</mrns0:Source>
        <mrns0:Category>default</mrns0:Category>
        <mrns0:Message>The server is down</mrns0:Message>
        <mrns0:Severity>http://schemas.hp.com
          /notification/severity/critical</mrns0:Severity>
        <mrns1:Properties>
          <mrns1:NameValuePair>
            <mrns1:Name>alert.sequence</mrns1:Name>
            <mrns1:Value>server_status</mrns1:Value>
          </mrns1:NameValuePair>
        </mrns1:Properties>
      </mrns0:Alert>
    </tns:sendAlert>
  </soap:Body>
</soap:Envelope>
```

The above request can be broken down as follows:

- **Name/MessageId** – `server_down_alert`

The name/messageId of the alert is a key that identifies a unique message type. It is used as a key on the server to reformat or replace the original message for display. So if your message is “The server has gone down”, a good alert name is “`server_down_alert`”. The actual message may get changed to “svr dn” when sent to a small display such as pager or cell phone.

Assume you are tracking a server and reporting two events, server up, and server down. Server up alerts would have the name “`server_up_alert`”. Server down alerts would be named “`server_down_alert`”.

- **Source** – `http://some_server`

Source is the entity that the alert applies to. Conceptually, it is the object that will own the alert once it is in the database. In this case, the server at `http://some_server`.

- **Category** – default

Category is used for routing the alert to different alert targets set up in the BSE application. It works just like a vanilla logging package.

- **Message** – “The server is down”

This is a default message to be used if no formatters are found on the alert server. The final message is determined by the name/messageId and settings in the alert subsystem. As such, you should make sure any details in the message are also put into the properties object.

Example: Our message “The server is down” contains no extra information that is not conceptually contained in the name/messageId “server_down_alert”. But if our message was more specific such as “Server foo is down” we would need to include some extra information in the properties object so the formatting system could access it. Adding a “server.name” property with value “foo” would be fine because now the formatters could reconstruct the original message by using the key “server_down_alert” to look up a localized pattern “Server \${server.name} is down” and substitute the property server.name=foo.

- **Severity** – “http://schemas.hp.com/notification/severity/critical”.

There are seven severity levels defined in the WSFM standard.

- http://schemas.hp.com/notification/severity/normal
- http://schemas.hp.com/notification/severity/info
- http://schemas.hp.com/notification/severity/warn
- http://schemas.hp.com/notification/severity/minor
- http://schemas.hp.com/notification/severity/major
- http://schemas.hp.com/notification/severity/critical
- http://schemas.hp.com/notification/severity/unknown

- **Property alert.sequence** – “server_status”.

Broadly speaking, there are two different kinds of alerts. There are alerts for one time events. These do not recover. Examples include an alert for a very large product order or a security violation. The other sort of alert represents a transient state within the system. These alerts do recover and represent things like servers going down (and then back up) or SLOs being violated (and then fulfilled). In order to recover, an alert must identify itself using a sequenceId. This new alert effectively overrides any previous alerts with the same sequenceId. In this example I have created a critical alert with a sequence called “server_status”. The status of http://some_server will remain critical until I send another alert with the same sequenceId and a status of normal (which I will do in the next example). By contrast, a one shot alert does not clear its status until acknowledge/deleted by the user. If you are sending a one shot alert, don’t include an “alert.sequence” property in your properties bundle.

Recovering From an Alert

Below is a sample alert for a fictional server at http://some_server. This alert is flagging the alert system that the server_status for http://some_server is now normal due to a server_up_alert event. This alert will cancel the previous alert (shown above) and is known as a recovery alert.

SendAlert Sample Request

```
<soap:Envelope
```

```

xmlns:mrns0=http://openview.hp.com/xmlns/mip
  /2005/03/mip-monitoring.xsd
xmlns:mrns1=http://openview.hp.com/xmlns/mip
  /2005/03/mip-common.xsd
xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
xmlns:tns=http://openview.hp.com/xmlns/mip/2005/03/AlertService
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<soap:Body>
  <tns:sendAlert>
    <mrns0:Alert>
      <mrns0:Name>server_up_alert</mrns0:Name>
      <mrns0:Source>http://some_server</mrns0:Source>
      <mrns0:Category>default</mrns0:Category>
      <mrns0:Message>Server is up</mrns0:Message>
      <mrns0:Severity>http://schemas.hp.com/
        notification/severity/normal</mrns0:Severity>
      <mrns1:Properties>
        <mrns1:NameValuePair>
          <mrns1:Name>alert.sequence</mrns1:Name>
          <mrns1:Value>server_status</mrns1:Value>
        </mrns1:NameValuePair>
      </mrns1:Properties>
    </mrns0:Alert>
  </tns:sendAlert>
</soap:Body>
</soap:Envelope>

```

The above request is identical to the previous request except for the following.

- **Name/messageId** is now server_up_alert
- **Message** = Server is up
- **Severity** = http://schemas.hp.com/notification/severity/normal

Because the alert.sequence property is the same as the previous example the status of http://some_server is now normal.

queryAlerts(AlertQueryType query)

This operation returns a list of alerts that match a certain set of criteria. Only unacknowledged/unresolved alerts are returned. Alerts are returned in reverse chronological order (newest first).

Below is a sample query for a fictional server at http://some_server. This query is requesting all pending alerts for the resource.

Sample Request

```

<soap:Envelope
  xmlns:mrns0=http://openview.hp.com/xmlns/mip
    /2005/03/mip-monitoring.xsd
  xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:tns=http://openview.hp.com/xmlns/mip/2005/03/AlertService
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <tns:queryAlerts>
      <mrns0:AlertQuery>
        <mrns0:Source>http://some_server</mrns0:Source>
      </mrns0:AlertQuery>
    </tns:queryAlerts>
  </soap:Body>

```

```
</soap:Envelope>
```

Pseudo code for AlertQueryType

```
AlertQueryType
{
    anyURI           Source
    string           Severity
    dateTime         StartTime
    dateTime         EndTime
    int              StartRow
    int              NumRows
}
```

- **Source** – Only return alerts for this managed object.
- **Severity** – Only return alerts matching this Severity.
- **StartTime/EndTime** – Only return alerts occurring in the specified time window.
- **StartRow** – Skip over the first <int> alerts. This is primarily used for paging through large numbers of alerts.

To view the first ten alerts you would send `alertQuery(startRow:0, numRows:10)`. To see the next ten you would send `alertQuery(startRow:10, numRows:10)`.

- **NumRows** – Indicates the maximum number of alerts returned.

All parameters are optional. An `AlertQueryType` with no contents will retrieve all unresolved alerts.



Alerts returned with this operation do not include their extended properties so `AlertType.Properties` will be empty. To retrieve the `Properties` object, use `queryAlertProperties()` which is described below.

Sample Response (pseudo code)

```
AlertList
{
    Alert
    {
        Name = server_up_alert,
        Source = http://some_server,
        Category = default,
        Message = Server is up,
        Severity = http://schemas.hp.com/notification/severity/normal,
        CreationTime = 2005-05-11T21:11:15.109-07:00,
        Id = 43,
        Properties = empty
    }
    Alert
    {
        Name = server_down_alert,
        Source = http://some_server,
        Category = default,
        Message = Server is down,
        Severity = http://schemas.hp.com/notification
            /severity/critical,
        CreationTime = 2005-05-11T21:10:33.559-07:00,
        Id = 42,
        Properties = empty
    }
}
```

queryAlertProperties(int alertId)

This operation will retrieve the extended properties object for the alert matching alertId.

Below is a sample operation for retrieving the alert properties for alert #43 (the recovery alert from a previous example). This example assumes the alert ID has been obtained from a previous call to `queryAlerts(...)` or some other mechanism.

Sample Request

```
<soap:Envelope
  xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:tns=http://openview.hp.com/xmlns/mip/2005/03/AlertService
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <tns:queryAlertProperties>
      <tns:param0>43</tns:param0>
    </tns:queryAlertProperties>
  </soap:Body>
</soap:Envelope>
```

Sample Response (pseudo code)

```
return_value
{
  NameValuePair
  {
    Name = alert.sequence,
    Value = server_status
  }
  NameValuePair
  {
    Name = category,
    Value = default
  }
  NameValuePair
  {
    Name = date,
    Value = 5/11/05
  }
  NameValuePair
  {
    Name = datetime,
    Value = 5/11/05 9:11 PM
  }
  NameValuePair
  {
    Name = id,
    Value = 43
  }
  NameValuePair
  {
    Name = message,
    Value = Server is up
  }
  NameValuePair
  {
    Name = name,
    Value = server_up_alert
  }
  NameValuePair
```

```

    {
      Name = severity,
      Value = NORMAL
    }
  NameValuePair
  {
    Name = source,
    Value = http://some_server
  }
  NameValuePair
  {
    Name = status,
    Value = New
  }
  NameValuePair
  {
    Name = time,
    Value = 9:11 PM
  }
  NameValuePair
  {
    Name = timestamp,
    Value = 1115871075109
  }
  NameValuePair
  {
    Name = wsmf-severity,
    Value = http://schemas.hp.com/notification/severity/normal
  }
}

```

deleteAlerts(AlertQueryType query)

This operation will resolve/acknowledge all alerts that match the AlertQuery. This method has no response.

Below is a sample operation for a fictional server at http://some_server. This operation will delete all pending alerts for the managed object.

Sample Request

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:mrns0=http://openview.hp.com/xmlns/mip
    /2005/03/mip-monitoring.xsd
  xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:tns=http://openview.hp.com/xmlns/mip/2005/03/AlertService
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <tns:deleteAlerts>
      <mrns0:AlertQuery>
        <mrns0:Source>http://some_server</mrns0:Source>
      </mrns0:AlertQuery>
    </tns:deleteAlerts>
  </soap:Body>
</soap:Envelope>

```

See the `queryAlerts()` example for a discussion on `AlertQueryType`.



Alerts are never physically deleted from the database; they are only marked as acknowledged/resolved.

deleteAlertsByIds(AlertIdListType AlertIdList)

This operation will resolve/acknowledge all alerts with IDs contained in the AlertIdList. This method has no response.

Below is a sample operation for a fictional delete operation. This operation will delete alerts #42 and #43. This example assumes alert IDs have been obtained from a previous call to `queryAlerts(...)` or some other mechanism.

Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:mrns0=http://openview.hp.com/xmlns/mip
    /2005/03/mip-monitoring.xsd
  xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:tns=http://openview.hp.com/xmlns/mip/2005/03/AlertService
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <tns:deleteAlertsByIds>
      <mrns0:AlertIdList>
        <mrns0:Id>42</mrns0:Id>
        <mrns0:Id>43</mrns0:Id>
      </mrns0:AlertIdList>
    </tns:deleteAlertsByIds>
  </soap:Body>
</soap:Envelope>
```



Alerts are never physically deleted from the database; they are only marked as acknowledged/resolved.

UDDI Data Mappings Reference

The SOA Manager's assets that are published to a UDDI registry can be leveraged by any application that can integrate with UDDI. This section provides a reference of how the assets are mapped in UDDI.

In particular, this section includes mappings for:

- Business Services
- Configuration
- WS Intermediary and WS Container IT Service
- Database, Host, and MOM IT Service
- Dependency Relationships
- Managed Endpoint Relationships
- Organizational Contact Information

Overview

When you publish a business service, WS Intermediary/Container IT service, or DB IT service, the following information is created in a UDDI registry:

- The WSDL for the management web service. The WSDL is published according to: *Using WSDL in a UDDI Registry, Version 2.0.2 – Technical Note*. The following UDDI entries are created:
 - UDDI business service for the WSDL service
 - UDDI binding template for the WSDL port
 - UDDI tModel for each binding and port type
- The WSDL for the web service that is managed by the business service. The web service WSDL is published to the UDDI registry if it does not exist in the registry already. It is published according to the technical note cited above.
- Relationships between the services. The following types of relationships are published:

- **Dependency** – An SOA Manager’s business service has a keyed reference to each of its consumed web services, WS Configurations, and IT services. A WS Configuration contains a keyed reference to the IT services it configures.
- **Managed endpoint** – An SOA Manager’s business service contains a keyed reference to the managed web service.
- Organizational contact information. The following types of contact information are published:
 - **Relationship** – A Business or IT service contains a keyed reference for a contact’s role: supports, owns, or consumes.
 - **Name** – A Business or IT service contains a keyed reference for a contact’s name.
 - **Distinguished name** – A Business or IT service contains a keyed reference for a contact’s directory distinguished name.
 - **Email address** – A Business or IT service contains a keyed reference for a contact’s email address.

Business Service Mappings

The following is published for an SOA Manager business service:

- The WSDL for the management web service. The WSDL is published according to: *Using WSDL in a UDDI Registry, Version 2.0.2 – Technical Note*. The following UDDI entries are created:
 - UDDI business service for the WSDL service
 - UDDI binding template for the WSDL port
 - UDDI tModel for each binding and port type
- The WSDL for the Web service that is managed by the business service. The Web service WSDL is published to the UDDI registry if it does not exist in the registry already. It is published according to the technical note cited above.
- Relationships. The relationships are published as keyed references in the UDDI business service category bag. The relationships between an SOA Manager’s business service and other services are described further in the sections on relationships below. The following types of relationships are published:
 - **Dependency** – A business service contains a keyed reference to consumed Web services, WS Configurations, and IT services it contains.
 - **Managed endpoint** – A business service contains a keyed reference to its managed web services.
- Organizational contact information. Organizational contact information is published as keyed references in the UDDI business service category bag. The contact information is explained in more detail in the “Organizational Contact Information Mappings” section. The following types of contact information are published:
 - **Relationship** – A business service contains a keyed reference for a contact’s role: supports, owns, or consumes.
 - **Name** – A business service contains a keyed reference for the name of an organizational contact.

- **Distinguished name** – A business service contains a keyed reference for the directory distinguished name of an organizational contact.
- **Email address** – A business service contains a keyed reference for the email address of an organizational contact.

An example is shown below:

```

    <businessService serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
    bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>Finance Business Service</name>
- <bindingTemplates>
- <bindingTemplate bindingKey="uddi:8d3b1fc0-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
  <accessPoint useType="endPoint">http://KS725366.americas.
  cpqcorp.net:5002/wsmf/services/Runtime$service=
  WsmfBusinessService$name=Finance$2BBusiness$2BService
  </accessPoint>
- <tModelInstanceDetails>
- <tModelInstanceInfo tModelKey="uddi:384b5980-bc2e-11d9-817c-
    0f2c03c0817a">
- <instanceDetails>
  <instanceParams>BusinessServiceAlertSourcePort</instanceParams>
  </instanceDetails>
  </tModelInstanceInfo>
  <tModelInstanceInfo tModelKey="uddi:321c4010-bc2e-11d9-817c-
    0f2c03c0817a" />
  </tModelInstanceDetails>
  </bindingTemplate>
+ <bindingTemplate bindingKey="uddi:8d8e96f0-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:8dc0a370-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:8e4fc410-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:8e88fc80-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:8ece1bd0-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:8f133b20-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:8f4ebd80-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:8f8f21e0-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:8fcf8640-bc2e-11d9-817c-
    0f2c03c0817a" serviceKey="uddi:8d176b20-bc2e-11d9-817c-
    0f2c03c0817a">
  </bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="uddi:2f3362e0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Business service manages functional
    business service endpoint" keyValue="808be6b0-bc2e-11d9-817c-
    0f2c03c0817a" />

```

```

<keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
0f2c03c0817a" keyName="Dependency between business service and
IT service configuration" keyValue="0548af20-bc2d-11d9-817c-
0f2c03c0817a" />
<keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
0f2c03c0817a" keyName="Dependency between business service and
IT service configuration" keyValue="7cf133c0-bc2e-11d9-817c-
0f2c03c0817a" />
<keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
0f2c03c0817a" keyName="Dependency between business service and
IT service configuration" keyValue="709c76c0-bc2e-11d9-817c-
0f2c03c0817a" />
<keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
0f2c03c0817a" keyName="Dependency between business service and
business service" keyValue="3c1cd3e0-bc2e-11d9-817c-
0f2c03c0817a" />
<keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
0f2c03c0817a" keyName="Finance Management: name"
keyValue="Finance Management" />
<keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
0f2c03c0817a" keyName="Finance Management: relationship"
keyValue="wsmf:owns-and-provides" />
<keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
0f2c03c0817a" keyName="Finance Management: email"
keyValue="finance.mgmt@acme.com" />
<keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
0f2c03c0817a" keyName="Finance Management: distinguished name"
keyValue="en=Finance Management,ou=IT
Services,o=Company,dc=acme,de=com" />
<keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
Entity type" keyValue="service" />
<keyedReference tModelKey="uddi:uddi.org:xml:namespace"
keyName="XML namespace"
keyValue="http://openview.hp.com/xmlns/mip/2005/
03/BusinessService" />
<keyedReference tModelKey="uddi:uddi.org:xml:localname"
keyName="XML local name" keyValue="Runtime$service=
WsmfBusinessService$name=Finance$2BBusiness$2BService" />
</categoryBag>
</businessService>

```

Configuration Mappings

The following information is published for a WS Intermediary/Container, database, MOM and Host Configuration:

- The WSDL for the management web service. The WSDL is published according to *Using WSDL in a UDDI Registry, Version 2.0.2 – Technical Note*. These UDDI entries are created:
 - UDDI business service for the WSDL service
 - UDDI binding template for the WSDL port
 - UDDI tModel for each binding and port type.
- Relationships. The relationships are published as keyed references in the UDDI business service category. The relationships between the WS Intermediary/Container Configuration and other services are described further in the sections on relationships. The following types of relationships are published:

- Dependency. A Configuration contains a keyed reference to the WS IT services that it configures.



Host, MOM, and database configurations do not have Management Endpoint relationships. In addition, intermediary and container configuration can have more than one Management Endpoint relationship.

An example for a WS intermediary configuration is shown below:

```
- <businessService serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>Intermediary For Finance Service</name>
- <bindingTemplates>
- <bindingTemplate bindingKey="uddi:70d82030-bc2e-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a">
  <accessPoint useType="endPoint">http://KS725366.americas.cpqcorp.
    net:5002/wsmf/services/Runtime$service=WsmfItService$name=
    Intermediary$2BFor$2BFinance$2BService$BusinessService=
    Finance$2BBusiness$2BService</accessPoint>
- <tModelInstanceDetails>
- <tModelInstanceInfo tModelKey="uddi:bd5e0a20-bc2c-11d9-817c-
  0f2c03c0817a">
- <instanceDetails>
  <instanceParms>FoundationManagedObjectIdentityPort</instanceParms>
</instanceDetails>
</tModelInstanceInfo>
<tModelInstanceInfo tModelKey="uddi:b765b820-bc2c-11d9-817c-
  0f2c03c0817a" />
</tModelInstanceDetails>
</bindingTemplate>
+ <bindingTemplate bindingKey="uddi:711d3f80-bc2e-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:715da3e0-bc2e-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:719b9740-bc2e-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:71dbfba0-bc2e-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:7217a510-bc2e-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:72559870-bc2e-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:7295fcd0-bc2e-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:709c76c0-bc2e-11d9-817c-
  0f2c03c0817a">
</bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
  0f2c03c0817a" keyName="Dependency between IT service
  configuration and IT service" keyValue="eb480990-bc2c-11d9-817c-
  0f2c03c0817a" />
  <keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
  Entity type" keyValue="service" />
```

```

<keyedReference tModelKey="uddi:uddi.org:xml:namespace"
  keyName="XML namespace"
  keyValue="http://openview.hp.com/xmlns/mip/2005/03/ITService" />
<keyedReference tModelKey="uddi:uddi.org:xml:localname"
  keyName="XML local name"
  keyValue="Runtime$service=WsmfItService$name=
  Intermediary$2BFor$2BFinance$2BService$BusinessService=
  Finance$2BBusiness$2BService" />
</categoryBag>
</businessService>

```

WS Intermediary/Container IT Service Mappings

The following information is published for a WS Intermediary/Container IT service:

- The WSDL for the management web service. The WSDL is published according to *Using WSDL in a UDDI Registry, Version 2.0.2 – Technical Note*. These UDDI entries are created:
 - UDDI business service for the WSDL service
 - UDDI binding template for the WSDL port
 - UDDI tModel for each binding and port type.
- Organizational contact information. Organizational contact information is published as keyed references in the UDDI business service category bag. The contact information is explained in more detail in the “Organizational Contact Information Mappings” section. The following types of contact information are published:
 - **Relationship** – A WS Intermediary/Container IT service contains a keyed reference for a contact’s role: supports, owns, or consumes.
 - **Name** – A WS Intermediary/Container IT service contains a keyed reference indicating the name of an organizational contact.
 - **Distinguished name** – A WS Intermediary/Container IT service contains a keyed reference specifying the directory distinguished name of an organizational contact.
 - **Email address** – A WS Intermediary/Container IT service contains a keyed reference with the email address of an organizational contact.

An example is shown below:

```

- <businessService serviceKey="uddi:eb480990-bc2c-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>brokerWsees</name>
- <bindingTemplates>
- <bindingTemplate bindingKey="uddi:eb649240-bc2c-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:eb480990-bc2c-11d9-817c-
  0f2c03c0817a">
  <accessPoint useType="endPoint">http://KS725366.americas.cpqcorp.
  net:5002/wsmf/services/Runtime$service=
  WsmfWseeItService$IntermediaryWseeItService=
  brokerWsees</accessPoint>
- <tModelInstanceDetails>
- <tModelInstanceInfo tModelKey="uddi:bd5e0a20-bc2c-11d9-817c-
  0f2c03c0817a">
- <instanceDetails>

```

```

<instanceParms>FoundationManagedObjectIdentityPort</instanceParms>
</instanceDetails>
</tModelInstanceInfo>
<tModelInstanceInfo tModelKey="uddi:b765b820-bc2c-11d9-817c-
  0f2c03c0817a" />
</tModelInstanceDetails>
</bindingTemplate>
+ <bindingTemplate bindingKey="uddi:eba03bb0-bc2c-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:eb480990-bc2c-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:ebde2f10-bc2c-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:eb480990-bc2c-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:ec1c4980-bc2c-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:eb480990-bc2c-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:ec6f99a0-bc2c-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:eb480990-bc2c-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:ed05e630-bc2c-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:eb480990-bc2c-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:ed4b0580-bc2c-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:eb480990-bc2c-11d9-817c-
  0f2c03c0817a">
</bindingTemplates>
- <categoryBag>
<keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
  0f2c03c0817a" keyName="Broker Support: name" keyValue="Broker
  Support" />
<keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
  0f2c03c0817a" keyName="Broker Support: relationship"
  keyValue="wsmf:operates-and-supports" />
<keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
  0f2c03c0817a" keyName="Broker Support: email"
  keyValue="broker@acme.com" />
<keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
  0f2c03c0817a" keyName="Broker Support: distinguished name"
  keyValue="en=Broker Support,ou=IT Services,o=Company,dc=acme,
  de-com" />
<keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
  Entity type" keyValue="service" />
<keyedReference tModelKey="uddi:uddi.org:xml:namespace"
  keyName="XML namespace"
  keyValue="http://openview.hp.com/xmlns/mip/2005/
  03/WSEEITService" />
<keyedReference tModelKey="uddi:uddi.org:xml:localname"
  keyName="XML local name"
  keyValue="Runtime$service=
  WsmfWseeItService$IntermediaryWseeItService=brokerWsees" />
</categoryBag>
</businessService>

```

IT Service Mappings

The following information is published for an SOA Manager database, MOM, and Host IT service:

- The WSDL for the management web service. The WSDL is published according to *Using WSDL in a UDDI Registry, Version 2.0.2 – Technical Note*. These UDDI entries are created:
 - UDDI business service for the WSDL service
 - UDDI binding template for the WSDL port
 - UDDI tModel for each binding and port type.
- Organizational contact information. Organizational contact information is published as keyed references in the UDDI business service category bag. The contact information is explained in more detail in the “Organizational Contact Information Mappings” section. The following types of contact information are published:
 - **Relationship** – An SOA Manager IT service contains a keyed reference for a contact’s role: supports, owns, or consumes.
 - **Name** – An SOA Manager IT service contains a keyed reference indicating the name of an organizational contact.
 - **Distinguished name** – An SOA Manager IT service contains a keyed reference specifying the directory distinguished name of an organizational contact.
 - **Email address** – An SOA Manager IT service contains a keyed reference with the email address of an organizational contact.

An example of a database IT service mapping is shown below:

```
- <businessService serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>oracleDB</name>
- <bindingTemplates>
- <bindingTemplate bindingKey="uddi:056a19d0-bc2d-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a">
  <accessPoint useType="endPoint">http://KS725366.americas.cpqcorp.
    net:5002/wsmf/services/Runtime$service=WsmfDbItService$name=
    oracleDB</accessPoint>
- <tModelInstanceDetails>
- <tModelInstanceInfo tModelKey="uddi:bd5e0a20-bc2c-11d9-817c-
  0f2c03c0817a">
- <instanceDetails>
  <instanceParms>FoundationManagedObjectIdentityPort</instanceParms>
</instanceDetails>
</tModelInstanceInfo>
<tModelInstanceInfo tModelKey="uddi:b765b820-bc2c-11d9-817c-
  0f2c03c0817a" />
</tModelInstanceDetails>
</bindingTemplate>
+ <bindingTemplate bindingKey="uddi:05a80d30-bc2d-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:05e3b6a0-bc2d-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a">
```

```

+ <bindingTemplate bindingKey="uddi:0623f3f0-bc2d-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:06620e60-bc2d-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:067c4d20-bc2d-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:06ba4080-bc2d-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a">
+ <bindingTemplate bindingKey="uddi:06f833e0-bc2d-11d9-817c-
  0f2c03c0817a" serviceKey="uddi:0548af20-bc2d-11d9-817c-
  0f2c03c0817a">
</bindingTemplates>
- <categoryBag>
<keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
  0f2c03c0817a" keyName="DB Support: name"
  keyValue="DB Support" />
<keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
  0f2c03c0817a" keyName="DB Support: relationship"
  keyValue="wsmf:operates-and-supports" />
<keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
  0f2c03c0817a" keyName="DB Support: email"
  keyValue="dbSupport@acme.com" />
<keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
  0f2c03c0817a" keyName="DB Support: distinguished name"
  keyValue="en=DB Support,ou=IT Services,o=Company,dc=acme,
  de-com" />
<keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
  Entity type" keyValue="service" />
<keyedReference tModelKey="uddi:uddi.org:xml:namespace"
  keyName="XML namespace"
  keyValue="http://openview.hp.com/xmlns/mip/2005/
  03/DbITService" />
<keyedReference tModelKey="uddi:uddi.org:xml:localname"
  keyName="XML local name"
  keyValue="Runtime$service=WsmfDbItService$name=oracleDB" />
</categoryBag>
</businessService>

```

Dependency Relationships Mappings

The following dependency relationships will be represented in a UDDI registry:

- An SOA Manager business service dependency on another, consumed, SOA Manager business service.
- An SOA Manager business service dependency on an SOA Manager WS Intermediary/Container Configuration.
- An SOA Manager business service dependency on an SOA Manager IT service.
- An SOA Manager WS Intermediary/Container Configuration dependency on an SOA Manager WS Intermediary/Container IT service.

The dependency relationship categorization system is used to establish a dependency relationship between two business services.

tModel Definition

Field	Description
Name	hp-com:mip:dependency
Description	Taxonomy, which links two structures. KeyValue must be existing UDDI key of specified structure.
Categorization	Categorization
Checked	No

tModel Structure

```
<tModel tModelKey="uddi:xxxxxxxxxx" xmlns="urn:uddi-org:api_v3">
  <name>hp-com:mip:dependency</name>
  <description>Taxonomy, which links two structures. KeyValue must
    be existing UDDI key of specified structure.</description>
  <categoryBag>
    <keyedReference tModelKey="uddi:uddi.org:categorization:types"
      keyName="uddi-org:types" keyValue="categorization" />
  </categoryBag>
</tModel>
```

Valid Values

The key name of the keyed reference is one of:

- Dependency between business service and business service
- Dependency between business service and IT service configuration
- Dependency between IT service configuration and IT service

The key value is the UDDI service key of the SOA Manager service that is a dependency.

Example

```
- <businessService serviceKey="uddi:8d176b20-bc2e-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>Finance Business Service</name>
+ <bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="uddi:2f3362e0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Business service manages functional
    business service endpoint" keyValue="808be6b0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="0548af20-bc2d-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
```



```

    IT service configuration" keyValue="7cf133c0-bc2e-11d9-817c-
    0f2c03c0817a" />
<keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="709c76c0-bc2e-11d9-817c-
    0f2c03c0817a" />
<keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    business service" keyValue="3c1cd3e0-bc2e-11d9-817c-
    0f2c03c0817a" />
<keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: name"
    keyValue="Finance Management" />
<keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: relationship"
    keyValue="wsmf:owns-and-provides" />
<keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: email"
    keyValue="finance.mgmt@acme.com" />
<keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: distinguished name"
    keyValue="en=Finance Management,ou=IT
    Services,o=Company,dc=acme,de-com" />
<keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
    Entity type" keyValue="service" />
<keyedReference tModelKey="uddi:uddi.org:xml:namespace"
    keyName="XML namespace"
    keyValue="http://openview.hp.com/xmlns/mip/2005/
    03/BusinessService" />
<keyedReference tModelKey="uddi:uddi.org:xml:localname"
    keyName="XML local name"
    keyValue="Runtime$service=WsmfBusinessService$name=
    Finance$2BBusiness$2BService" />
</categoryBag>
</businessService>

```

Managed Endpoint Relationships Mappings

The managed endpoint relationship categorization system is used to establish a managed endpoint relationship between a business service and the web services it manages.

tModel Definition

Field	Description
Name	hp-com:mip:managedEndpoint
Description	Category system that establishes a managedEndpoint relationship between a businessservice entity and the endpoints it manages. The keyValue must be a pre-existing UDDI key of a businessservice.
Categorization	Categorization
Checked	No

tModel Structure

```
<tModel tModelKey="uddi:xxxxxxxxxxxx" xmlns="urn:uddi-org:api_v3">
  <name>hp-com:mip:managedEndpoint</name>
  <description>Category system that establishes a managedEndpoint
    relationship between a businessService entity and the endpoint
    it manages. The keyValue must be a pre-existing UDDI key of a
    businessService.</description>
  <categoryBag>
    <keyedReference tModelKey="uddi:uddi.org:categorization:types"
      keyName="uddi-org:types" keyValue="categorization" />
  </categoryBag>
</tModel>
```

Valid Values

The key name of the keyed reference is: Business service manages functional business service endpoint.

Example

```
- <businessService serviceKey="uddi:8d176b20-bc2e-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>Finance Business Service</name>
+ <bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="uddi:2f3362e0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Business service manages functional
    business service endpoint" keyValue="808be6b0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="0548af20-bc2d-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="7cf133c0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="709c76c0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    business service" keyValue="3c1cd3e0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: name"
    keyValue="Finance Management" />
  <keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: relationship"
    keyValue="wsmf:owns-and-provides" />
  <keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: email"
    keyValue="finance.mgmt@acme.com" />
  <keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: distinguished name"
```

```

        keyValue="en=Finance Management,ou=IT
        Services,o=Company,dc=acme,de-com" />
<keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
        Entity type" keyValue="service" />
<keyedReference tModelKey="uddi:uddi.org:xml:namespace"
        keyName="XML namespace"
        keyValue="http://openview.hp.com/xmlns/mip/2005/
        03/BusinessService" />
<keyedReference tModelKey="uddi:uddi.org:xml:localname"
        keyName="XML local name"
        keyValue="Runtime$service=WsmfBusinessService$name=
        Finance$2BBusiness$2BService" />
</categoryBag>
</businessService>

```

Organizational Contact Information Mappings

Organizational contacts are associated with SOA Manager Business and IT services. An organizational contact has the following attributes:

- Name
- Directory distinguished name
- Email address
- Relationship to an SOA Manager service:
 - Owns
 - Supports
 - Consumes

In the UDDI registry, keyed references contain organizational contact information. There is a keyed reference for every contact attribute. An SOA Manager business/IT service category bag contains keyed references for each of its contacts.

Here is an example of keyed references for an organizational contact named IIS Support Team:

```

<categoryBag>
  <keyedReference
    tModelKey="<key for contact name tModel>"
    keyName="IIS Support Team: name"
    keyValue="IIS Support Team"/>
  <keyedReference
    tModelKey="<key for contact distinguished name tModel>"
    keyName="IIS Support Team: distinguished name"
    keyValue="cn=IIS Support Team,ou=IT
    Services,o=Company,dc=acme,dc=com"/>
  <keyedReference
    tModelKey="<key for contact email tModel>"
    keyName="IIS Support Team: email"
    keyValue="iissupport@acme.com"/>
  <keyedReference
    tModelKey="<key for contact relationship tModel>"
    keyName="IIS Support Team: relationship"
    keyValue="wsmf:operates-and-supports"/>
</categoryBag>

```

Contact Name

An SOA Manager business/IT service contains organizational contact information including contact names.

The contact name categorization system is used to specify contact names for an SOA Manager business/IT service.

tModel Definition

Field	Description
Name	hp-com:mip:property:contact:name:eng
Description	English language name of an organizational contact.
Categorization	Categorization
Checked	No

tModel Structure

```
<tModel tModelKey="uddi:xxxxxxxxxxxx" xmlns="urn:uddi-org:api_v3">
  <name>hp-com:mip:property:contact:name:eng</name>
  <description>English language name of an organizational
    contact.</description>
  <categoryBag>
    <keyedReference tModelKey="uddi:uddi.org:categorization:types"
      keyName="uddi-org:types" keyValue="categorization" />
  </categoryBag>
</tModel>
```

Valid Values

The key name of the keyed reference contains the contact name followed by “: name”. An example is “ISS Support Team: name”.

The key value is the contact name.

Example

```
- <businessService serviceKey="uddi:8d176b20-bc2e-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>Finance Business Service</name>
+ <bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="uddi:2f3362e0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Business service manages functional
    business service endpoint" keyValue="808be6b0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="0548af20-bc2d-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
```

```

    IT service configuration" keyValue="7cf133c0-bc2e-11d9-817c-
    0f2c03c0817a" />
<keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="709c76c0-bc2e-11d9-817c-
    0f2c03c0817a" />
<keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    business service" keyValue="3c1cd3e0-bc2e-11d9-817c-
    0f2c03c0817a" />
<keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: name"
    keyValue="Finance Management" />
<keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: relationship"
    keyValue="wsmf:owns-and-provides" />
<keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: email"
    keyValue="finance.mgmt@acme.com" />
<keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: distinguished name"
    keyValue="en=Finance Management,ou=IT
    Services,o=Company,dc=acme,de-com" />
<keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
    Entity type" keyValue="service" />
<keyedReference tModelKey="uddi:uddi.org:xml:namespace"
    keyName="XML namespace"
    keyValue="http://openview.hp.com/xmlns/mip/2005/
    03/BusinessService" />
<keyedReference tModelKey="uddi:uddi.org:xml:localname"
    keyName="XML local name"
    keyValue="Runtime$service=WsmfBusinessService$name=
    Finance$2BBusiness$2BService" />
</categoryBag>
</businessService>

```

Contact Distinguished Name

An SOA Manager business/IT service contains organizational contact information including contact directory distinguished names.

The contact directory distinguished name categorization system is used to specify contact directory distinguished names for an SOA Manager business/IT service.

tModel Definition

Field	Description
Name	hp-com:mip:property:contact:dn
Description	Directory DN of an organizational contact.
Categorization	Categorization
Checked	No

tModel Structure

```
<tModel tModelKey="uddi:xxxxxxxxxxxx" xmlns="urn:uddi-org:api_v3">
  <name>hp-com:mip:property:contact:dn</name>
  <description>Directory DN of an organizational
    contact.</description>
  <categoryBag>
    <keyedReference tModelKey="uddi:uddi.org:categorization:types"
      keyName="uddi-org:types" keyValue="categorization" />
  </categoryBag>
</tModel>
```

Valid Values

The key name of the keyed reference contains the contact name followed by “: distinguished name”. An example is “ISS Support Team: distinguished name”.

The key value is the contact distinguished name.

Example

```
- <businessService serviceKey="uddi:8d176b20-bc2e-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>Finance Business Service</name>
+ <bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="uddi:2f3362e0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Business service manages functional
    business service endpoint" keyValue="808be6b0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="0548af20-bc2d-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="7cf133c0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="709c76c0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    business service" keyValue="3c1cd3e0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: name"
    keyValue="Finance Management" />
  <keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: relationship"
    keyValue="wsmf:owns-and-provides" />
  <keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: email"
    keyValue="finance.mgmt@acme.com" />
  <keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: distinguished name"
    keyValue="en=Finance Management, ou=IT
    Services, o=Company, dc=acme, de=com" />
```

```

<keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
  Entity type" keyValue="service" />
<keyedReference tModelKey="uddi:uddi.org:xml:namespace"
  keyName="XML namespace"
  keyValue="http://openview.hp.com/xmlns/mip/2005/
  03/BusinessService" />
<keyedReference tModelKey="uddi:uddi.org:xml:localname"
  keyName="XML local name"
  keyValue="Runtime$service=WsmfBusinessService$name=
  Finance$2BBusiness$2BService" />
</categoryBag>
</businessService>

```

Contact Email

An SOA Manager business/IT service contains organizational contact information including contact email addresses.

The contact email categorization system is used to specify contact email addresses for an SOA Manager business/IT service.

tModel Definition

Field	Description
Name	hp-com:mip:property:contact:email:smtp
Description	SMTP mail address of an organizational contact.
Categorization	Categorization
Checked	No

tModel Structure

```

<tModel tModelKey="uddi:xxxxxxxxxxxx" xmlns="urn:uddi-org:api_v3">
  <name>hp-com:mip:property:contact:email:smtp</name>
  <description>SMTP mail address of an organizational
    contact.</description>
  <categoryBag>
    <keyedReference tModelKey="uddi:uddi.org:categorization:types"
      keyName="uddi-org:types" keyValue="categorization" />
  </categoryBag>
</tModel>

```

Valid Values

The key name of the keyed reference contains the contact name followed by “: email”. An example is “ISS Support Team: email”.

The key value is the contact email address.

Example

```

- <businessService serviceKey="uddi:8d176b20-bc2e-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>Finance Business Service</name>
+ <bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="uddi:2f3362e0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Business service manages functional
    business service endpoint" keyValue="808be6b0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="0548af20-bc2d-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="7cf133c0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    IT service configuration" keyValue="709c76c0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
    business service" keyValue="3c1cd3e0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: name"
    keyValue="Finance Management" />
  <keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: relationship"
    keyValue="wsmf:owns-and-provides" />
  <keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: email" |
    keyValue="finance.mgmt@acme.com" />
  <keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: distinguished name"
    keyValue="en=Finance Management,ou=IT
    Services,o=Company,dc=acme,de-com" />
  <keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
    Entity type" keyValue="service" />
  <keyedReference tModelKey="uddi:uddi.org:xml:namespace"
    keyName="XML namespace"
    keyValue="http://openview.hp.com/xmlns/mip/2005/
    03/BusinessService" />
  <keyedReference tModelKey="uddi:uddi.org:xml:localname"
    keyName="XML local name"
    keyValue="Runtime$service=WsmfBusinessService$name=
    Finance$2BBusiness$2BService" />
</categoryBag>
</businessService>

```

Contact Relationship

An SOA Manager business/IT service contains organizational contact information such as contact relationship.

The contact relationship categorization system is used to specify contact relationships to an SOA Manager business/IT service.

tModel Definition

Field	Description
Name	hp-com:mip:contact:relationship
Description	Category system used to declare a relationship of a contact with a UDDI entity.
Categorization	Categorization
Checked	No

tModel Structure

```
<tModel tModelKey="uddi:xxxxxxxxxxxxx" xmlns="urn:uddi-org:api_v3">
  <name>hp-com:mip:contact:relationship</name>
  <description>Category system used to declare a relationship of a
    contact with a UDDI entity.</description>
  <categoryBag>
    <keyedReference tModelKey="uddi:uddi.org:categorization:types"
      keyName="uddi-org:types" keyValue="categorization" />
  </categoryBag>
</tModel>
```

Valid Values

The key name of the keyed reference contains the contact name followed by “: relationship”. An example is “ISS Support Team: relationship”.

The key value is one of the following: wsmf:consumes, wsmf:operates-and-supports, or wsmf:owns-and-provides.

Example

```
- <businessService serviceKey="uddi:8d176b20-bc2e-11d9-817c-
  0f2c03c0817a" businessKey="uddi:2ac538d0-b5d5-11d9-8851-
  bc2844308851" xmlns="urn:uddi-org:api_v3">
  <name>Finance Business Service</name>
+ <bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="uddi:2f3362e0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Business service manages functional
      business service endpoint" keyValue="808be6b0-bc2e-11d9-817c-
      0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
      IT service configuration" keyValue="0548af20-bc2d-11d9-817c-
      0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
      IT service configuration" keyValue="7cf133c0-bc2e-11d9-817c-
      0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
      IT service configuration" keyValue="709c76c0-bc2e-11d9-817c-
      0f2c03c0817a" />
  <keyedReference tModelKey="uddi:2e9acc60-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Dependency between business service and
```

```
    business service" keyValue="3c1cd3e0-bc2e-11d9-817c-
    0f2c03c0817a" />
  <keyedReference tModelKey="uddi:30838990-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: name"
    keyValue="Finance Management" />
  <keyedReference tModelKey="uddi:30fd2660-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: relationship"
    keyValue="wsmf:owns-and-provides" />
  <keyedReference tModelKey="uddi:302dc870-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: email"
    keyValue="finance.mgmt@acme.com" />
  <keyedReference tModelKey="uddi:2facffb0-bc2c-11d9-817c-
    0f2c03c0817a" keyName="Finance Management: distinguished name"
    keyValue="en=Finance Management,ou=IT
    Services,o=Company,dc=acme,de-com" />
  <keyedReference tModelKey="uddi:uddi.org:wSDL:types" keyName="WSDL
    Entity type" keyValue="service" />
  <keyedReference tModelKey="uddi:uddi.org:xml:namespace"
    keyName="XML namespace"
    keyValue="http://openview.hp.com/xmlns/mip/2005/
    03/BusinessService" />
  <keyedReference tModelKey="uddi:uddi.org:xml:localname"
    keyName="XML local name"
    keyValue="Runtime$service=WsmfBusinessService$name=
    Finance$2BBusiness$2BService" />
</categoryBag>
</businessService>
```

A

- alert service management web service, 5-40
- alerts, 5-40
- AlertService interface, 5-40
- attachments, 2-5
- audit
 - architecture, 3-2
 - database, 3-1, 3-2
 - message trace, 5-16
 - messages, 3-1
 - service, 3-1
- audit integration, 3-2
- audit management web service, 5-16
- AuditQueryPT interface, 5-16
- AuditQueryTypr type, 5-26
- authentication, 4-2
 - basic, 4-6, 4-8
- authorization, 4-2, 4-7
- average response time metric, 5-13

B

- BaseXmlHandler interface, 2-5
- billing, 3-1
- Broker Configurator, 1-3, 2-3
- brokered services, 1-3
 - add custom handler, 2-7
 - definition, 2-3
 - deploy, 2-7
 - jar, 2-3
 - manually implement, 2-2
 - xml introspection, 4-8
- BSE, 1-3
- business service management web service, 5-2, 5-38
- business services, 1-3
 - delete, 5-38
 - export, 5-35

- import, 5-36
- UDDI mappings, 6-2

C

- contact distinguished name, 6-15
- contact email, 6-17
- contact name, 6-14
- contact relationship, 6-18
- custom policy handler, 2-5
 - add, 2-7
- custom security, 4-2
 - handlers, 4-5

D

- database
 - message table, 3-4
 - message trace table, 3-2
 - schema, 3-2
- database integrations, 1-4
- database IT service
 - UDDI mappings, 6-8
- delete
 - business services, 5-38
- delete operation, 5-38
- deleteAlerts operation, 5-47
- deleteAlertsbByIds operation, 5-48
- dependency relationships UDDI mappings, 6-9
- document overview, 1-1

E

- exportBusinessService operation, 5-35

F

- forecasting, 3-1

G

- general integration, 1-2

H

handler. *See* policy handler

I

importBusinessService operation, 5-36

ImportExportPT interface, 5-35

integration points, 1-2

interfaces

AlertService, 5-40

AuditQueryPT, 5-16

BaseXmlHandler, 2-5, 4-5, 4-7

ImportExportPT, 5-35

LifecyclePT, 5-38

northbound, 1-5

overview, 1-5

RealmAuthoizationHandler, 4-2

ServicePerformanceSnapshotPT, 5-2

southbound, 1-5

startup tasks, 2-1

introspection service, 4-6

IT service

UDDI mappings, 6-6

IT service configurations

UDDI mappings, 6-4

J

Java API integrations, 1-3

L

LifecyclePT interface, 5-38

M

management integration

database, 1-4

general, 1-2, 2-1

Java API, 1-3, 1-4

web services, 1-4

management web service

alert service, 5-40

audit, 5-16

business service, 5-2, 5-38

service catlog, 5-35

mappings, 6-2, 6-4, 6-6, 6-8, 6-9, 6-13, 6-15, 6-17, 6-18

message table, 3-4

message trace table, 3-2

MessageTraceResultSet type, 5-27

MessageTraceType type, 5-28

N

non-repudiation, 3-1

northbound interfaces, 1-5

O

operations

delete, 5-38

deleteAlerts, 5-47

deleteAlertsbByIds, 5-48

exportBusinessService, 5-35

importBusinessService, 5-36

queryAlertProperties, 5-46

queryAlerts, 5-44

queryAuditTraces, 5-16

queryPerformanceData, 5-3

queryPerformanceDataResponse, 5-3

queryServiceReport, 5-23

retrieveMessageTrace, 5-18

retrieveProfile, 5-21

sendAlert, 5-41

organizational contact information UDDI mappings, 6-13

P

performance metrics, 5-2

PerformanceDataType type, 5-7

PerformanceWindowType type, 5-7

policy handler

add custom, 2-7

custom, 1-3, 2-5

custom security, 4-5

definition, 2-4

xml introspection, 4-7

prerequisites, 1-2

ProfileTraceType type, 5-31

Q

query audit messages, 5-16

queryAlertProperties operation, 5-46

queryAlerts operation, 5-44

- queryAuditTraces operation, 5-16
- queryPerformanceData operation, 5-3
- queryPerformanceDataResponse operation, 5-3
- queryServiceReport operation, 5-23

R

- RealmAuthoizationHandler interface, 4-2
- ReportEntryListType type, 5-33
- ReportEntryType type, 5-33
- reports, 1-4, 3-1, 3-4, 3-5
- retrieveMessageTrace operation, 5-18
- retrieveProfile operation, 5-21

S

- security
 - custom, 4-2
 - custom handlers, 4-5
 - customization, 4-1
- security customization, 1-4
- Select Access, 4-1, 4-5, 4-6
- sendAlert operation, 5-41
- server.xml, 2-2
- service catalog management web service, 5-35
- service security inbound handler, 4-8
- service.wsdl, 2-3
- service.xml, 2-3
- ServicePerformanceSnapshotPT interface, 5-2
- ServiceReportEntryType type, 5-34
- ServiceReportListType type, 5-34
- ServiceReportQueryType type, 5-32
- SLA
 - reports, 3-1
 - reports example, 3-4, 3-5
- SLO
 - data, 5-2
- SOAP attachments, 2-5
- southbound interfaces, 1-5
- startup, 1-3, 2-1

T

- tModel, 6-10, 6-11, 6-14, 6-17, 6-18
- trace messages, 3-1
- types
 - AuditQueryType, 5-26
 - MessageTraceResultSet, 5-27
 - MessageTraceType, 5-28
 - PerformanceDataType, 5-7
 - PerformanceWindowType, 5-7
 - ProfileTraceType, 5-31
 - ReportEntryListType, 5-33
 - ReportEntryType, 5-33
 - ServiceReportEntryType, 5-34
 - ServiceReportListType, 5-34
 - ServiceReportQueryType, 5-32

U

- UDDI
 - data mapping reference, 6-1
 - role of, 1-6
- UDDI mappings
 - business services, 6-2
 - database IT service, 6-8
 - dependency relationships, 6-9
 - IT service configurations, 6-4
 - IT services, 6-6
 - organizational contact information, 6-13

W

- web services
 - API reference, 5-1
 - integrations, 1-4
 - interfaces, 1-5
- WSDL, 1-4, 5-1, 6-2, 6-4, 6-6, 6-8
- WSDM, 1-4
- WSMF, 1-4

X

- xml introspection service, 4-6

