

# HP OpenView Service Oriented Architecture Manager

## WSM .NET Agent Administrator Guide

Version: 2.11

Windows



Aug 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](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

<b>1 Introduction .....</b>	<b>1-1</b>
Document Overview .....	1-1
Audience .....	1-1
Prerequisites .....	1-1
Contextual Overview .....	1-2
Control Point .....	1-2
WS Management Web Services .....	1-3
Runtime Description .....	1-3
<b>2 Installing the WSM .NET Agent.....</b>	<b>2-1</b>
Software Requirements.....	2-1
Pre-Installation Steps .....	2-1
Running the Installer .....	2-2
Verifying the Installation .....	2-4
Uninstalling the WSM .NET Agent .....	2-4
<b>3 Configuring SOAP Extensions .....</b>	<b>3-1</b>
Overview .....	3-1
Creating a Service Configuration .....	3-1
Registering Configuration Handlers .....	3-2
Using Service Specific Configurations .....	3-3
Using a Default Service Configuration .....	3-3
Verifying Extension Types .....	3-4
Logging Extension.....	3-4
Example .....	3-5

Audit Extension ..... 3-5  
    Example ..... 3-5  
Business Metric Extension ..... 3-6  
    Example ..... 3-7

# Introduction

This chapter contains overview information about the WSM .NET Agent. In particular, this chapter includes the following sections:

- Document Overview
- Contextual Overview
- Runtime Description

## Document Overview

The WSM .NET Agent Administrator Guide provides instructions required to use the .NET Agent to manage Web services. The guide includes an overview of the Agent architecture and guidance for installing the agent.

The guide does not provide instructions for setting up an end-to-end management solution. The *HP OpenView Service Oriented Architecture (SOA) Manager Administrator Guide*, together with this guide, should be used when creating a complete Web services management solution.

## Audience

This guide is primarily intended for Systems Integrators, System Administrators, and Developers who are responsible for integrating and enabling management components in their application environment.

## Prerequisites

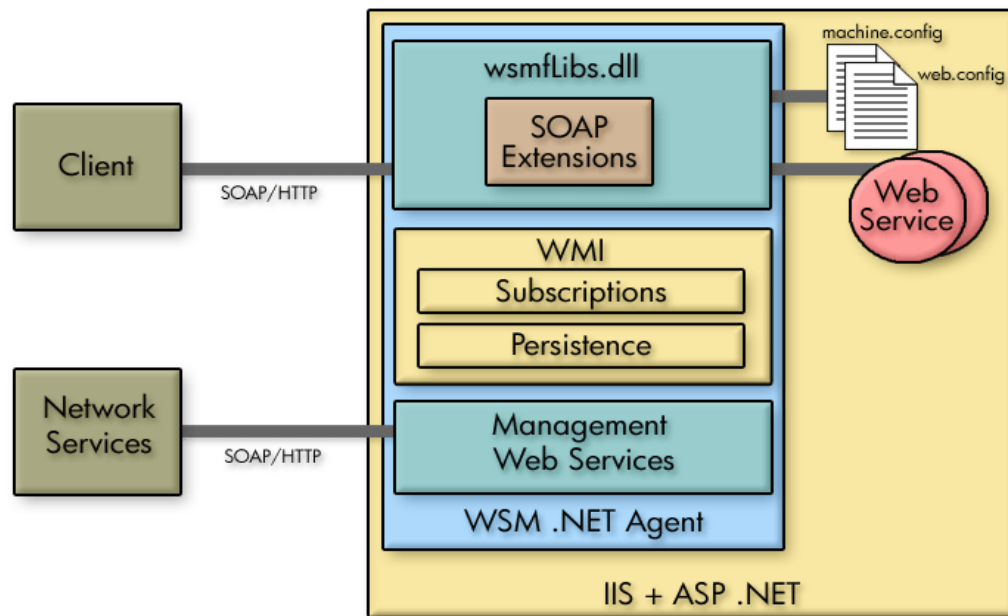
Users of this guide should be familiar with Web services management in general and with the .NET platform specifically.

## Contextual Overview

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

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

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



**Figure 1-1: Contextual Overview**

### Control Point

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



## WS Management Web Services

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



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

## Runtime Description

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

**Table 2-1: .NET Agent Runtime Description**

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



## Installing the WSM .NET Agent

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

► The .NET Agent has been tested on Windows 2003 Server.

### Software Requirements

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

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

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

### Pre-Installation Steps

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

To configure a Web proxy:

- 1 Using a text editor, open `machine.config` located in `<winnt>\Microsoft .NET\Framework\v1.1.4322\CONFIG`

- 2 Set the `usesystemdefault` attribute for the `<proxy>` element to `false`.
- 3 Set the `proxyaddress` attribute to the name of your proxy server including the port number. For example:

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

## Running the Installer

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



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

To run the installer:

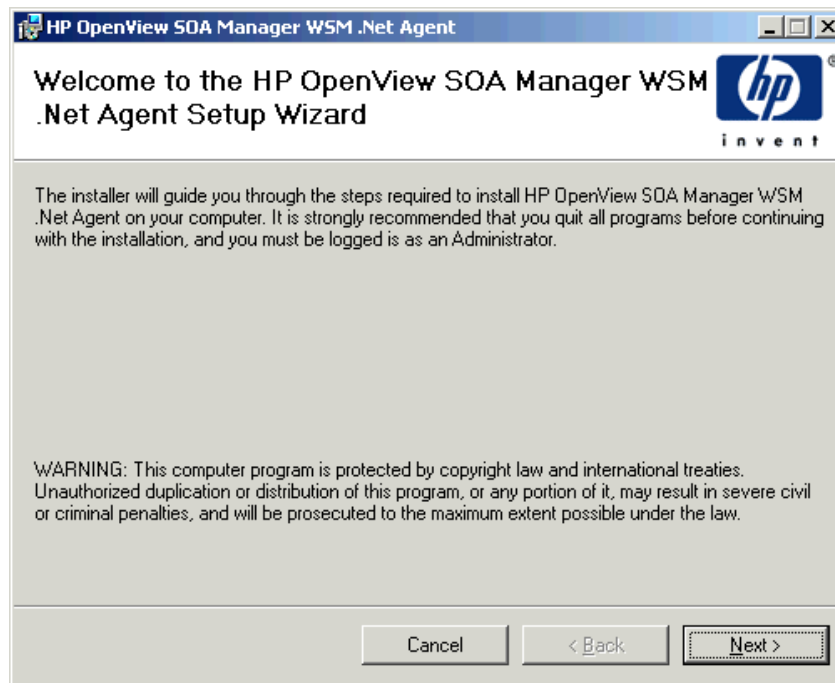
- 1 Log in to the system as a root/admin user.
- 2 Place the SOA Manager CD into the CD ROM drive.
- 3 Exit the Auto Run screen.
- 4 From a command prompt, change directories on the CD to `\Autopass\win`.
- 5 Run the following command:

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

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

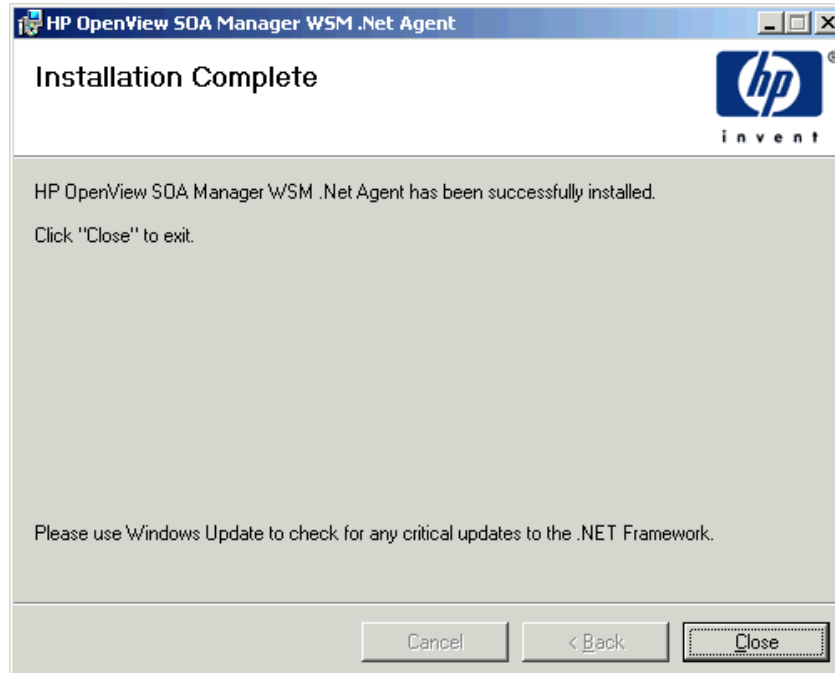
- 6 From the command line, change directories on the CD to `\WSM_Agents\DotNet`.
- 7 Double click on `HPWsmAgentInstall.msi`.

The installation wizard start screen displays.



**Figure 2-1: Install Wizard Start Screen**

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



**Figure 2-2: Install Wizard Completion Screen**

## Verifying the Installation

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

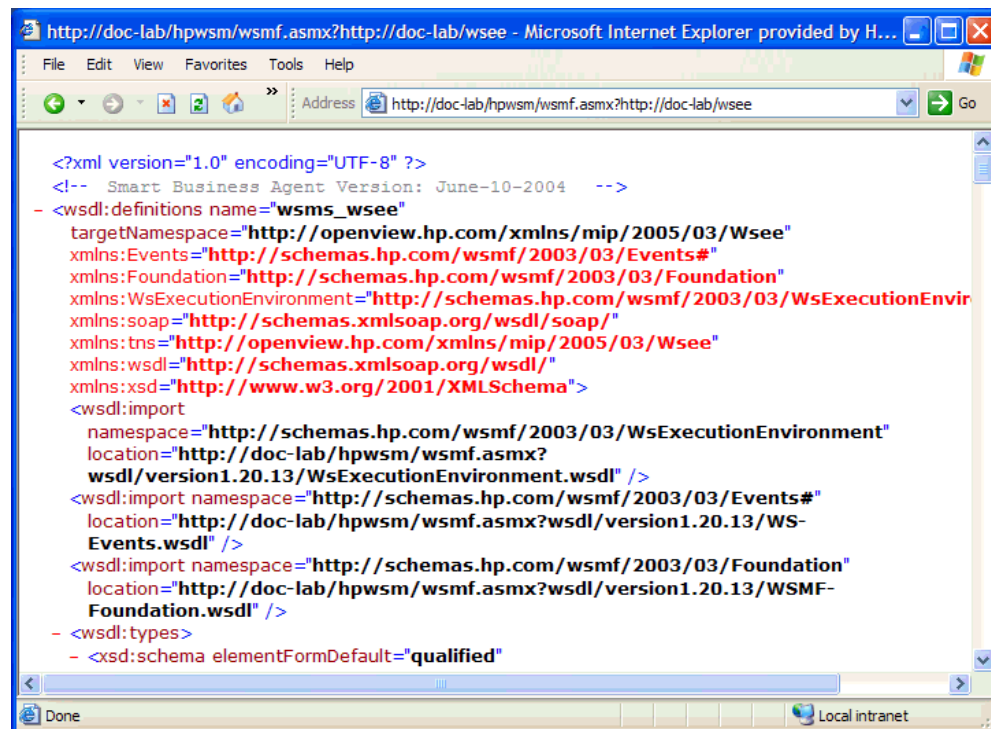
To verify the installation:

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

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

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

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



## Uninstalling the WSM .NET Agent

To uninstall the WSM .NET Agent

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

# Configuring SOAP Extensions

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

## Overview

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



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

## Creating a Service Configuration

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

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

## Registering Configuration Handlers

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



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

To register the .NET Agent's Configuration handlers:

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

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

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

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



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

- 4 Save the file.



## Using Service Specific Configurations

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

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

## Using a Default Service Configuration

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



Service specific configurations override default service configurations.

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

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

```

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

```

## Verifying Extension Types

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

```

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

```



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

## Logging Extension

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

To configure the Logging extension:

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

```

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

```

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

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

## Example

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

## Audit Extension

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

To configure the Logging extension:

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

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

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

## Example

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

## Business Metric Extension

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



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

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

To configure a Business Metric extension:

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

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

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

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

- 3 Save and close the file. Business Metric Alerts are viewed in the BSE.

## Example

The following example can be used to create a business metric alert for the FinanceService sample Web service.

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

