

HP OpenView Smart Plug-in for WebLogic Integration

for the HP-UX and Solaris Operating Systems

Software Version: A.01.03.000

User Guide

December 2005



Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

© Copyright 2005 Hewlett-Packard Development Company, L.P.

No part of this document may be copied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Trademark Notices

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

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

UNIX® is a registered trademark of The Open Group

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

Support

Please visit the HP OpenView support 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 valuable 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>

Contents

1	Introduction	9
	Document Overview	9
	Audience	9
	Prerequisites	10
	Chapters Summary	10
	Related Documents	11
	Overview of WLI Management	12
	WLI SPI Conceptual Overview	13
	WLI SPI SBA Component	14
	WLI SPI Frontend Subagent	16
	WLI SPI Backend Service	16
	Resource Explorer	16
	WLI SPI Reports	16
	Deployment Scenarios	17
	Consolidated Scenario	17
	Scenario 1: Co-located Components Scenario	18
	Scenario 2: Remote Frontend Subagent	18
	Scenario 3: Fully Distributed (Recommended)	19
	WSDM Overview	21
2	Install and Configure the WLI SPI	23
	Installation Process Summary	24
	Requirements	25
	Third Party License Review	25
	Software Requirements	25
	Hardware Requirement	26
	Installing and Configuring the WLI SPI Components	27
	Install the WLI SPI Depot	28

Install the WLI SPI Depot on HP-UX	28
Install the WLI SPI Depot on Solaris	29
Configure the Management Server	29
Install the WLI SPI Frontend Subagent Component	30
Install the WLI SPI SBA Component	33
Configure the Backend Service and the Frontend Subagent	41
Assign WLI SPI Templates	45
Configure Data Sources for Metric Data Collection	45
Assign Backend Service Templates	46
Assign Frontend Subagent Templates	47
Verify Deployed Templates	48
Start the Backend and the Frontend	49
Restarting the Backend and the Frontend	50
Stop the Backend and the Frontend	51
Checking the WLI SPI OVO Components' Status	52
Verify a Start Status	52
Verify a Stop Status	53
Install the Java Console	55
Starting the Java Console	55
Install the HP Resource Explorer	57
Starting the HP Resource Explorer from the Java Console	58
Starting the HP Resource Explorer form the Command Line	58
Install WLI SPI Reports	59
Uninstall the WLI SPI Components	60
Uninstalling the WLI SPI SBA	60
Uninstalling the Frontend Subagent	60
Remove the WLI SPI Bits on HP-UX	61
Remove the WLI SPI Bits on Solaris	62
Remove WLI SPI Message Groups	62
Remove the WLI SPI Application Group	62
Remove the WLI SPI User Profile	62
Remove the WLI SPI User	63
Remove WLI SPI Message Templates	63
Reinstall the Templates, Monitors, Commands, and Actions	64
Remove the WLI SPI Node Groups From the OVO Database	64

Configuring WLI SPI to Run as Non-Root User	64
3 Working with the WLI SPI	67
Service Management	68
Viewing WLI Managed Resources	68
Filtering Unwanted MO Types	69
Event Management	70
Viewing WLI Events	70
Monitoring and Data Collection	71
Metric Definition Configuration File	71
Modifying Data Collection	73
Collecting Data for Specific Metrics	75
Changing the Metric Data Collection Interval	75
Reporting and Performance Graphs	77
Using the OVO Embedded Performance Component (Coda)	77
Embedded Performance Component (Coda) Logging	77
View WLI SPI Reports	78
Defining a User Friendly Name for an MO	79
Monitoring Performance Metrics with OVPM	80
Configure OVPM	80
Create an Average Execution Time OVPM Graph for Business Processes	80
View the Business Process Average Execution Time OVPM Graph	81
Modify Logging and Tracing Levels	82
Changing Log Levels	82
Changing Trace Levels	83
4 Implementing Application-Specific Managed Objects	85
MO Implementation Class	85
Model MBean Interface Class	87
MBean Proxy Class	87
Discovery Class	87
SBA Configuration File	88
Complex Types	89
5 Troubleshooting	91
WLI SPI Backend Service	91

Backend Service is Not Found	91
Performance Agent Does Not Start Up	91
WLI SPI Frontend Subagent	92
Frontend Subagent Does Not Stop	92
Frontend Subagent Does Not Start	92
WLI Service is Not Visible	92
Missing Operational Notification	92
Errors in Frontend Logfile	98
WLI SPI SBA	100
Cannot View WSMF Services	100
Get Out of Memory Error Message When Running Ant Script	101
WLI SPI Configuration Editor	102
WLI SPI Configuration Editor Does Not Start	102
WLI SPI Configuration Editor Does Not Display Content	102
WLI SPI Configuration Editor Fails to Transfer WConfig.xml	102
WLI SPI Configuration Editor Throws AWTEException	103
Using the Self-Healing Info Application	104
Editing WConfig.xml	107
Using the WLI SPI Configuration Editor Tool	107
WConfig.xml Configuration Parameters	108
<FrontendSection>	108
<BackendSection>	112
<WLIEmailNotificationConfig>	112
Message Groups	116
Applications	116
Templates	117
Performance Metrics	119
Reports	122
Glossary	123
Index	127
Index	128

1 Introduction

This chapter provides an overview of the HP OpenView Smart Plug-In for the WebLogic Integration platform (WLI SPI). High level architectural and contextual information about the SPI's components is also provided.


Document Overview

The WLI SPI User Guide provides detailed information for setting up, configuring, and using WLI SPI to manage the BEA WebLogic Integration platform.

Instructions in this guide for using OVO-U, the OVO Console, and WLI (including WebLogic Server), are only specific to the WLI SPI and are not meant as a comprehensive reference. For documentation specific to these products, follow the link provided in the section [Related Documents](#) on page 11.

Audience

This User Guide is intended for anyone who is responsible for operating and administering OVO. In particular, the guide is for OVO administrators, OVO operators, system administrators, IT operators, and WLI application managers. It is expected that WLI application managers will work as OVO operators and will use the Java console to manage the WLI platform.

- 
- 1 MeasureWare Agent (MWA) is now known as OpenView Performance Agent (OVPA).
 - 2 Vantage Point Operations (VPO) is now known as OpenView Operations (OVO).
 - 3 Coda is now known as the embedded performance component.

Prerequisites

Users of this guide should have basic knowledge in the following areas:

- OpenView Operations for UNIX (HP-UX or Solaris)
- OpenView management solutions including SPIs
- OVO Management Console for Windows
- WLI software environment
- Java

Chapters Summary

The information in this guide is organized in a layered approach from generic instructions about the WLI SPI to detailed instructions about WLI management. Once you have completed and are familiar with the instructions in the guide, you can refer to specific instructions as needed.

- **Chapter 1 – Introduction:** Contains overview and contextual information about the WLI SPI and the WLI integration, including a brief introduction to WSDM.
- **Chapter 2 – Install and Configure the WLI SPI:** Contains detailed steps needed to install the components of the WLI SPI.
- **Chapter 3 – Working with the WLI SPI:** Contains instructions that are used to perform basic management tasks using the WLI SPI and additional OV products.
- **Chapter 4 – Implementing Application-Specific Managed Objects:** Contains instructions for creating customized Managed Objects for the WLI environment.

- **Chapter 5 – Troubleshooting:** Contains general procedures used to troubleshoot the WLI SPI and also provides solutions to common errors encountered when using the WLI SPI.
- **Appendix A – WLI SPI Configuration:** Contains a reference of all configuration parameters used to configure the OVO components of the WLI SPI.
- **Appendix B – List Templates and Reports:** Contains a reference of the templates and the reports for the WLI SPI.

Related Documents

The WLI SPI integrates with several OV components. OV product guides can be downloaded from the product manuals page on the OpenView web site.

BEA WLI and WebLogic Server (WLS) documentation is available at:

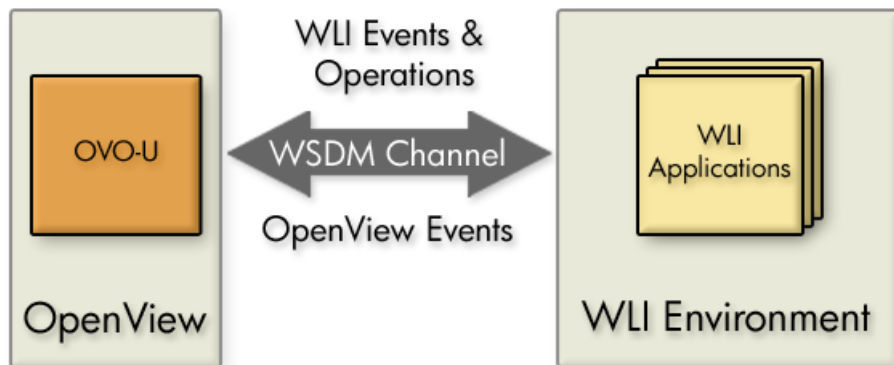
<http://edocs.bea.com/>

Overview of WLI Management

HP OpenView Operations for UNIX (OVO-U) provides a fully integrated management solution for networks, systems, databases, and applications found in heterogeneous distributed IT environments. This comprehensive product suite represents a complete set of tools enabling IT organizations to improve overall availability and reliability, maintain the highest degree of management flexibility, and establish management control over virtually all aspects of an enterprise environment.

The WLI SPI bridges OVO-U and the WLI platform through a common channel. As a result, the WLI SPI enables OpenView to manage WLI specific applications, as well as other elements in the user's environment, such as computing and network infrastructure. [Figure 1](#) shows a high level view of the integration:

Figure 1 Integration Overview



HP's strategic focus is to enable an adaptive infrastructure for our users. Our efforts together with BEA extends OpenView's core competency of service management to include critical business process information and the ability to adaptively manage the enterprise. The integration allows administrators to take immediate and appropriate action to maintain uptime and keep mission-critical applications running efficiently.

WLI SPI Conceptual Overview

The WLI SPI consists of five components. The components are:

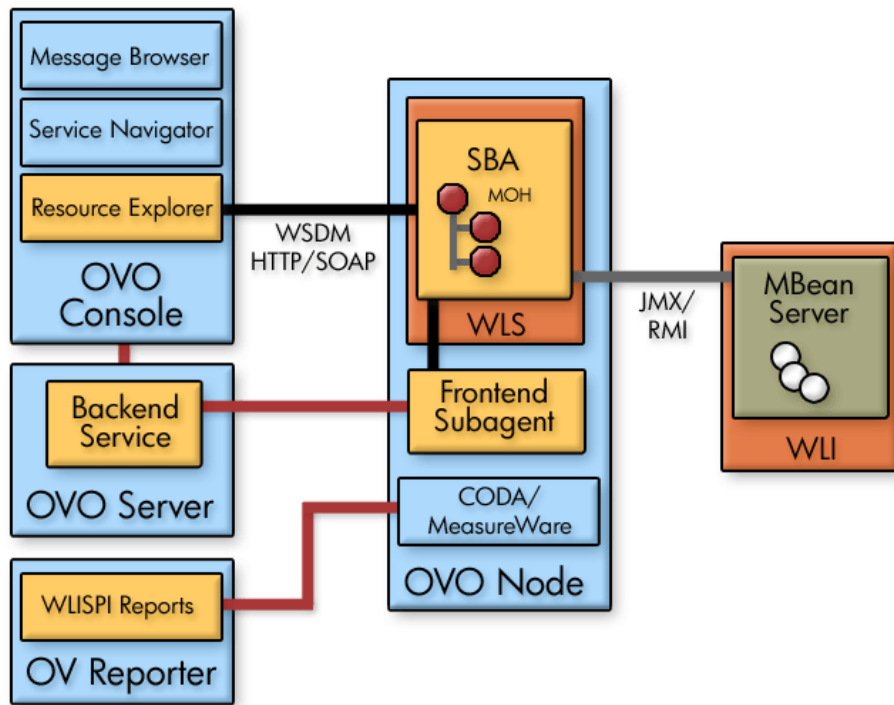
- WLI SPI Smart Business Agent (SBA)
- WLI SPI Frontend Subagent
- WLI SPI Backend Service
- Resource Explorer
- WLI SPI Reports

These components are distributed and are used to provide end-to-end management of the WLI environment and its applications. [Figure 2](#) shows a conceptual overview of the WLI SPI components and the interaction between them.



[Figure 5](#) shows a typical deployment scenario that emphasizes the distributed nature of the WLI SPI solution. For additional deployment scenario options, see the section [Deployment Scenarios](#) on page 17.

Figure 2 Conceptual Overview of the WLI SPI Components



WLI SPI SBA Component

The WLI SPI SBA is the foundation for Web Services Distributed Management (WSDM) enablement. This component of WLI-SPI is responsible for exposing the WLI manageability interface as WSDM-compliant Simple Object Access Protocol (SOAP) web services. These web services are exposed using the HTTP transport protocol by a WebLogic Server (WLS) instance. The SBA is a Java program that runs within a web container. The SBA can be located with WLI or can be on a remote system as shown in [Figure 2](#).

Agent Controller

The agent controller is the main component of the SBA. It is the entry point and lifecycle manager of the WSDM implementation.

The controller can support existing JMX MBeans, it wraps existing MBeans (as a proxy) with WSDM-compliant ModelMBeans. For this, the controller must be configured with one or more auto-discovery strategies, as discussed in the next section. In the WLI application, the controller wraps the WLI JMX MBeans to collect management information and invoke management operations on the WLI application platform.

The agent controller also supports WSDM-ready ModelMBeans. It provides a base class implementation that aids in the development of WSDM-enabled ModelMBeans. For this, the controller must be configured with one or more custom discovery strategies as discussed in [Chapter 4, Implementing Application-Specific Managed Objects](#).

Auto Discovery

The auto-discovery mechanism allows the SBA controller to automatically discover and manage your MBeans and create the required Web Services Definition Language (WSDL) files to expose your managed objects to a WSDM-enabled client.

WSDM Services

Regardless of whether the controller is using proxy MBeans found during auto-discovery or WSDM-enabled ModelMBeans, Managed Objects (MOs) are exposed as Web services.

The JAX-RPC handler is installed within the SOAP platform to redirect SOAP requests to operations on the ModelMBeans managed by the agent controller.

Developer Kit

The SBA also provides a developer kit so that you can implement WSDM-enabled ModelMBeans to manage your applications. It provides the `JmxService` base class, which is a complete `RequiredModelMBean`-compliant implementation. It provides caching, persistence, and targeting of operations to other objects. The required WSDM operations are automatically implemented.

See [Chapter 4, Implementing Application-Specific Managed Objects](#) for detailed information about creating and integrating new MOs for specific applications running on your WLI platform.

WLI SPI Frontend Subagent

The Frontend Subagent is responsible for communicating with the WLI SPI SBA component to gather management data and information. The Frontend Subagent is essentially a WSDM client able to communicate with MOs exposed as Web services by the SBA.

WLI SPI Backend Service

The backend software receives management data and information from the Frontend Subagent, converts the information to an OpenView recognized form, and allows OpenView to manage a WLI environment using traditional OpenView tools. The Backend Service must be located on the OVO-U management server.



Both the Frontend and Backend can be located on the same system. If the Frontend and Backend are on the same system, they must run on the OVO management server system. This may be a likely scenario during testing.

Resource Explorer

The Resource Explorer is a UI tool that is used to view and interact with MOs. The tool is available only for Windows. The tool can be either started from the OVO Java Console (if installed on Windows) or as a standalone application from the Windows command line.

WLI SPI Reports

The WLI SPI provides various customized reports that provide performance data of the WLI Environment. The reports are implemented using the HP OpenView Reporter product, which can interface with both MeasureWare and the OVO embedded performance component (Coda).

Deployment Scenarios

As discussed in the section [WLI SPI Conceptual Overview](#) on page 13, the WLI SPI is a distributed solution. As such, its components can be distributed in several different deployment configurations. This section provides some common deployment scenarios and does not represent every possible configuration.

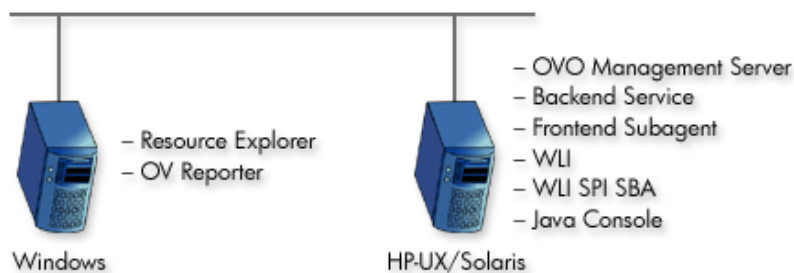
Consolidated Scenario

It is possible to have a single computer that hosts all of the WLI SPI components except for the Resource Explorer and OV Reporter, which are only available for Windows. In this scenario, the Resource Explorer must be started from the Windows command line and can not be started from within the Java Console. This scenario uses minimal hardware, but results in heavy loads on a single computer. Therefore, it is not recommended for production environments.



You may need to reconfigure the default ports used by the components if port conflicts occur.

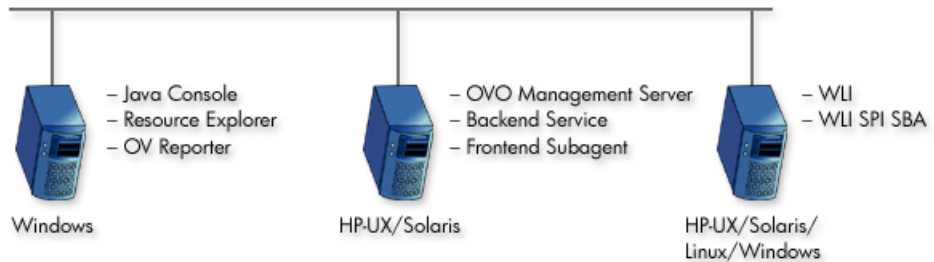
Figure 3 Consolidated Scenario



Scenario 1: Co-located Components Scenario

In this scenario, both the WLI SBA component and WLI platform are located on the same system, and the WLI SPI Frontend Subagent and Backend Service are located on the OVO management server. Application and management processing is concentrated on a single computer. This scenario is good for testing and is also applicable for less demanding production environments.

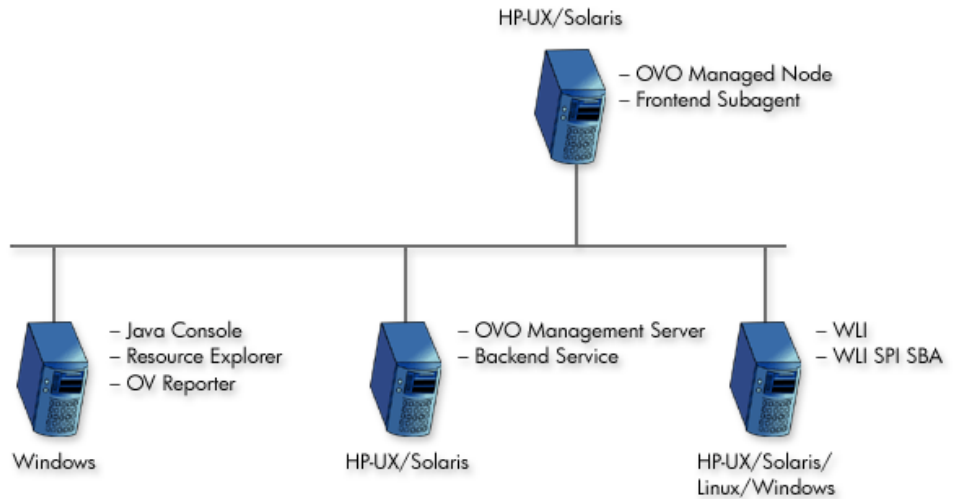
Figure 4 Co-located Component Scenario



Scenario 2: Remote Frontend Subagent

In this scenario, both the WLI SBA component and WLI platform are located on the same system, and the Frontend Subagent is remote from the OVO Managed Server computer. This scenario effectively separates management processing between a managed node and the management server. This scenario is ideal for a production environment.

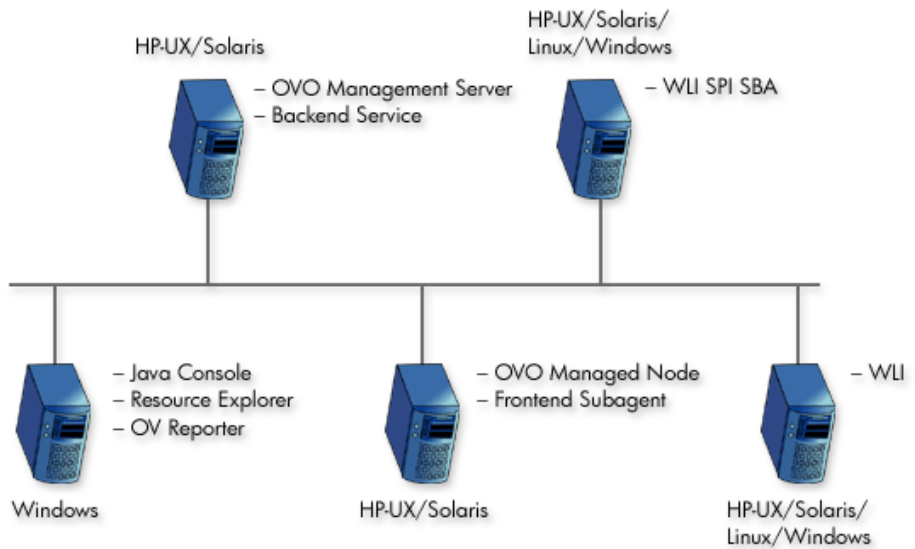
Figure 5 Remote SBA and Frontend Subagent Scenario



Scenario 3: Fully Distributed (Recommended)

In this scenario, the WLI SPI components are separated on different computers. Each computer is relegated to a single task. This scenario provides the most efficient processing and resource utilization. However, this scenario introduces the maximum overhead. Maximum hardware is used and maintaining the solution can become cumbersome. This scenario is ideal for a production environment.

Figure 6 Fully Distributed Scenario



WSDM Overview

WSDM is a management specification for managing application resources using Web services technology as well as managing Web services using Web services. The specification allows a common integration channel between an ISV and a management station such as OVO. The WSDM specification is defined by the WSDM Technical Community (TC) at OASIS. More information on the standard can be found at the WSDM TC site at OASIS.



The WSDM implementation in the WLI SPI is based on an HP-authored preliminary version of WSDM known as the Web Service Management Framework (WSMF).

WSDM provides a way to represent and realize management models and is based on Managed Objects (MOs) and the relationships between them. MOs provide management capabilities by implementing management interfaces. The management interfaces are described using Web Services Definition Language (WSDL). Hence, WSDM provides the architecture for defining management interfaces for MOs.

The foundational interfaces defined by WSDM can be extended in order to better manage resources in specific domains. In the WLI SPI, there are two types of extensions: OpenView domain extensions that provide OpenView data and events to WLI applications and WLI domain extensions that enable OpenView to access WLI specific manageability.

XML interfaces

WSDM defines a set of domain agnostic management interfaces, as well as domain specific management interfaces for Web services. Vendors are free to extend WSDM to their specific domain in order to manage resources effectively.

WSDM interfaces are categorized. Each category of interfaces corresponds to one port type in WSDLs.

- Discovery – a set of interfaces for discovering MOs and their relationships in the managed environments. There is a set of default relationship types defined in WSDM, and vendors can extend these relationship types as well.
- Configuration – a set of interfaces that allows a manager to find out the configuration information about a managed object.

- Control – a set of interfaces to control MOs / resources.
- Performance – a set of interfaces that allows a manager find out performance information about MOs.
- EventPush & Event Pull – a set of interfaces for subscribing to events in either push or pull fashion.
- EventCallback – a set of interfaces for listening to events after subscription.

2 Install and Configure the WLI SPI

This chapter provides instructions for installing and configuring the various components of the OpenView WLI SPI. Some of the SPI components are tightly integrated with OpenView Operations (OVO) for UNIX. If you are not familiar with OVO, refer to the OVO for UNIX documentation when completing the instructions in this chapter. In addition, it is recommended that a development-time installation of OVO be used to test the WLI SPI before installing and running on a production installation of OVO.

Installation Process Summary

The following list highlights the installation and configuration process and provides links to the individual installation and configuration tasks. The installation process proceeds as follows:

- 1 Review the [Deployment Scenarios](#) on page 17 in Chapter 1 and select a deployment scenario.
- 2 Review the [Requirements](#) section for hardware and software requirements.
- 3 Install the [WLI SPI Depot](#).
- 4 Configure the [Management Server](#).
- 5 Install the [WLI SPI Frontend Subagent Component](#).
- 6 Install the [WLI SPI SBA Component](#)
- 7 Configure the [Backend Service](#) and the [Frontend Subagent](#).
- 8 Assign [WLI SPI Templates](#).
- 9 Start the [Backend](#) and the [Frontend](#).
- 10 Install the [Java Console](#).
- 11 Install the [HP Resource Explorer](#).
- 12 Install [WLI SPI Reports](#).

Requirements

This section details the hardware and software requirements for installing and running the components of the WLI SPI. Make sure the requirements are met before beginning the installation.

- ▶ Before installing the WLI SPI components, make sure your WLI environment is installed and operational.

Third Party License Review

The WLI SPI utilizes third party code. You must review the third party licenses before installing the WLI SPI. The licenses are located on the CD in the `OV_DOC/A.01.03.000/Licenses` directory. If you disagree with a particular license, you must not install the WLI SPI.

Software Requirements

The following recommended software requirements should be met when installing and running the WLI SPI.

Table 1 WLI SPI Required Software

Software Requirements	Platforms
WebLogic Platform 8.1.2	HP-UX 11.11, 11.23 Solaris 8.0 Red Hat Linux AS 3.0 Windows 2000
OVO-U 7.10	HP-UX 11.11, 11.23 Solaris 2.8, 5.8, 8.0, 9.0
OVO-U 8.x	HP-UX 11.11, 11.23, 11.23 IA 64 bit, Solaris 2.8, 5.8, 8.0, 9.0, 10.0

Table 1 WLI SPI Required Software

Software Requirements	Platforms
Java Console A.07.00 (UNIX version included with OVO)	HP-UX 11.11, 11.23, Windows 2000
DSI2DDF A.01.23 for OVO 7.x and DSI2DDF A.02.00.00 for OVO 8.x	HP-UX 11.11, 11.23 Solaris 2.8, 5.8, 8.0, 9.0, 10.0
JRE 1.4	HP-UX 11.11, 11.23, Solaris 2.8, 5.8, 8.0, 9.0, 10.0
OV Reporter 3.5 (required to view WLI SPI performance reports and graphs)	Windows 2000

Hardware Requirement

The following recommended hardware requirements should be met when installing and running the WLI SPI.

Table 2 Hardware Requirements

Hardware	Requirements
RAM	<ul style="list-style-type: none"> • 512 MB: System running Frontend Subagent including the WLI SPI SBA. • 512 MB: System running the Backend Service. 512 MB is sufficient to run both the Frontend Subagent and Backend Service if they are located on the same system.
Disk Space	<ul style="list-style-type: none"> • 50 MB: Front End Subagent. • 250 MB: Backend Service. • 50MB: WLI SPI SBA.

Patches

It is important that all recommended patches for OVO-U and Java are installed prior to running any components of the WLI SPI. To find the recommended OVO patches, go to <http://support.openview.hp.com/patches>. In the Browse By Product Version section, select **operations for UNIX** and click the >> button. To find the recommended Java patches for HP-UX, go to <http://www.hp.com/products1/unix/java>. From the left side of the screen, click **Patches**. To find the recommended Java patches for Solaris, go to <http://sunsolve.sun.com/pubpatch> for the latest J2SE patch cluster.



Patches must be installed on all systems on which the WLI SPI components are running. Keep your system updated with latest patches, check for latest available patches regularly.

Kernel Parameters

Verify that the kernel parameters on both the Backend Service and Frontend Subagent nodes meet the recommended settings to run a Java program. If you are running both the Backend Service and Frontend Subagent on an HP-UX node, you can use the `HPjconfig` tool, which provides the recommended kernel parameter settings for your system. To download the `HPjconfig` tool, go to <http://www.hp.com/products1/unix/java/java2/hpjconfig>.

Installing and Configuring the WLI SPI Components

This section includes steps for installing the Backend Service on the OVO management server, as well as installing and configuring the WLI SPI SBA component, and Frontend Subagent. You must have OVO for UNIX installed before following the steps in this section. To install OVO for UNIX, refer to the OVO documentation. In addition, you must be logged onto the management server and managed nodes as **root** when completing the instructions in this section.



Make sure that JRE 1.4 is installed on both the management server and the managed node. On the management server, set an environment variable `JAVA_HOME` to the JRE installation directory.

Install the WLI SPI Depot

The WLI SPI Depot contains all of the WLI SPI components. The WLI SPI depot has to be installed on an OVO management server and it automatically installs the WLI SPI Backend Service. However, the WLI SPI Frontend Subagent, WLI SPI SBA, and Resource Explorer must be installed after the WLI SPI Depot is installed. In addition, if you want to install the Frontend Subagent on a managed node, you can do so from the management server after you install the Depot.

- ▶ Both the Frontend and Backend can be located on the same system. If the Frontend and Backend are on the same system, they must run on the OVO management server system. This may be a likely scenario during testing.

Install the WLI SPI Depot on HP-UX

To install the WLI SPI Depot:

- 1 From a command prompt on the OVO management server, run:
swinstall.
- 2 In the Specify Source dialog box, Source Depot Path field, enter the full path to the HP-UX SPI depot on the product CD. For example:
/<mount_point>/OV_DEPOT/11.0HPUX.sdtape
- 3 Click **OK**.
- 4 In the SD Install - Software Selection dialog box, select **SPIWLI**.
- 5 Select **Actions** → **Install**. The Install Analysis dialog box displays.
- 6 Click **OK**. An Install Window dialog box displays.

- ▶ The WLI SPI may take a few minutes to install. The installation status should report that the installation completed successfully. If the installation status is “Completed with Warnings”, this is expected and you can proceed.

- 7 Click the **Done** button.
- 8 In the SD Install - Software Selection dialog box, select **File** → **Exit**.

Install the WLI SPI Depot on Solaris

When installing the WLI SPI on Solaris run `swinstall` from the command line and provide the complete path to the Solaris SPI depot on the product CD. In addition, include the WLI SPI SD install name (`SPIWLI`) in the command. For example:

```
swinstall -x reinstall=true -s /<mount_point>/OV_DEPOT/  
SOLARIS.sdtape SPIWLI
```

Configure the Management Server

This section provides instructions to configure the OVO management server to use the WLI SPI. All the steps in this section are to be performed from the OVO management server.

Configure the `opc_adm` operator

The WLI SPI user profile must be assigned to the OVO administrator. This step is required in order for the administrator to see the WLI SPI notifications.

To configure the `opc_adm` operator:

- 1 Launch the OVO administrator GUI on the OVO management server and log on as **opc_adm**.
- 2 From OVO administrator GUI, select **Window** → **User Bank**.
- 3 From the User Bank window, right-click **opc_adm** and select **Modify** from the popup menu.
- 4 In the Modify User: `opc_adm` window, click the **Profiles** button.
- 5 In the Profiles of User: `opc_adm` window, verify that the WLISPI User Profile icon appears. If it is not displayed in the window:
 - a Open the User Profile Bank window by selecting **Window** → **User Profile Bank**.
 - b Drag the WLISPI User Profile from the OVO User Profile Bank window and drop it into the Profiles of User: `opc_adm` window.
 - c Close the OVO User Profile Bank window.

- 6 Close the Profiles of user: `opc_adm` window.
- 7 Close the Modify User: `opc_adm` window by clicking the **OK** button.
- 8 Close the User Bank window.

Register the Backend Service

To register the Backend Service:

- 1 From the OVO administrator GUI, select the OVO management server in the Node Bank window.
- 2 Select **Actions** → **Agents** → **Install / Update SW & Config**.
- 3 In the Install / Update OVO Software and Configuration dialog box, select **Monitors, Commands** and **Actions** in the Components frame.
- 4 Select **Force Update** in the Options frame.
- 5 Click the **OK** button.
- 6 On the OVO management server, register the Backend Service by running:

```
/opt/OV/bin/wli-perl /opt/OV/bin/wli-register-backend add
```

Install the WLI SPI Frontend Subagent Component

The Frontend Subagent collects management data from the WLI SPI SBA and makes the data available to the Backend Service. The Frontend Subagent can run on the management server system or on a remote system configured as a managed node. For detailed steps on installing and configuring OVO managed nodes, consult the OVO documentation.



If you are running the Frontend Subagent on the management server, you can skip the first two sections: [Add a Managed Node to the WLISPI-UNIX Node Group](#) and [Install the Monitors, Commands, and Actions on a Managed Node](#).

Add a Managed Node to the WLISPI-UNIX Node Group

If you are installing the Frontend Subagent on a managed node, the node must be configured as part of the WLISPI-UNIX Node Group.

▶ If you are installing the Frontend Subagent on the management server, you can skip this section.

To add a managed node to the WLISPI-UNIX Node Group:

- 1 From the OVO administrator GUI, open the Node Bank window by selecting **Window** → **Node Bank**.
- 2 From any OVO window, open the Node Group Bank window by selecting **Window** → **Node Group Bank**.
- 3 From the Node Group window, double-click the **WLISPI-UNIX** node group.
- 4 Drag the node where the Frontend Subagent will be installed from the Node Bank and drop it onto the Node Group: WLISPI-UNIX window.
- 5 Click the **Close** icon in the Node Group: WLISPI-UNIX window to close the window.

Install the Monitors, Commands, and Actions on a Managed Node

If you are installing the Frontend Subagent on a managed node, the node must be configured to use monitors, commands, and actions.

▶ If you are installing the Frontend Subagent on the management server, you can skip this section.

To install monitors, commands, and actions, on a managed node:

- 1 From the Node Bank window, select the node where the Frontend Subagent will be installed.
- 2 Select **Actions** → **Agent** → **Install / Update SW & Config**.
- 3 In the Install / Update OVO Software and Configuration dialog, select **Monitors, Commands** and **Actions** in the Components frame.
- 4 Select **Force Update** in the Options frame.
- 5 Click the **OK** button.

Install the Frontend Subagent

To install the Frontend Subagent:

- 1 From the Node Bank window, select the node where you want to install the Frontend Subagent.
- 2 Select **Actions** → **Subagents** → **Install/Update**.
- 3 In the Install / Update Subagents dialog box, select the **WLI SPI Subagent** in the Subagents frame on the left.

Click the **OK** button. The Install Subagent Packages window is displayed. It might take a couple of minutes for the Frontend Subagent to be installed. After installation, the following message is displayed:

```
Subagent successfully installed and configured on Managed
Node xxx.
```

- 4 Press **Enter**.



You cannot start the Frontend Subagent until you configure it. See [Configure the Backend Service and the Frontend Subagent](#) on page 41. You might get the following critical message in the OVO message browser:

```
WLI SPI Subagent (Frontend) of subagent 26 aborted; process
did an exit 0 (OpC30-1040).
```

You can ignore this message since the Frontend Subagent cannot start until it is configured.

Verifying the Frontend Subagent Install

You can verify that the Frontend Subagent is installed and registered. On the Frontend system, if it's an OVO7.x or OVO8.x DCE agent node, run:

```
opcagt -status -id 26
```

The following message is displayed:

```
OVO Managed Node status:
```

```
-----
---
Control Agent   /opt/OV/bin/OpC/opcctl  (1792) is running
Message Agent  /opt/OV/bin/OpC/opcmgsa (26036) is running
```



```
BBC Local Location Broker /opt/OV/bin/llbserver (26037) is
running Subagent 26:
```

```
WLI SPI Subagent (Frontend) /opt/OV/bin/wli-perl /opt/OV/bin/
wli-start-frontend isn't running
```

If the Frontend Subagent is installed on an OVO8.x HTTPS agent node, run:

```
/opt/OV/bin/ovc -status -id WLI
```

The following message appears:

```
wli-frontend WLI SPI Frontend Subagent      WLI      Stopped
```

Install the WLI SPI SBA Component

This section provides detailed steps for installing the WLI SPI SBA component. An InstallAnywhere GUI installer is provided for installing the WLI SPI SBA. The SBA component can be installed on either a UNIX or Windows platform. Installers for both platforms are provided in `/opt/OV/bin`. Regardless of the platform, the steps are identical. For HP-UX and Solaris platforms, use the `wli-sba-install.bin` executable. For the Windows platform, use the `wli-sba-install.exe` executable.



Make sure the SBA component is installed on a computer where WebLogic Administration Server will be hosted for the cluster

To install the WLI SPI SBA component:

- 1 If you are installing the WLI SPI SBA on a remote computer, ftp the WLI SPI SBA Installer, located on the management server at `/opt/OV/bin`, to the remote computer. This step does not have to be completed if the WLI SPI SBA is being installed on the management server.
- 2 Execute the platform-appropriate installer. For UNIX, type the installer command on the command line. For Windows, double-click on the file name. This opens the Introduction screen.
- 3 Click **Next**. This opens the Choose Install Folder screen. The default location for the installation on UNIX is under the `/opt` directory. For Windows, it is `C:\opt`. If you do not want to use the default location, click **Choose** to browse to another location.

- 4 Click **Next**. This opens the Choose Shortcut Folder screen. This allows you to create product icons. If you do not want to use the default values, click **Choose** to browse to another location.
- 5 Click **Next**. This opens the Pre-Installation Summary screen. Review the information.
- 6 Click **Install** if everything appears to be correct. When the installation is complete, the Install Complete screen is displayed.
- 7 Click **Done**.

Configuring the WLI SPI SBA

Minimal configuration is required if any. Review the values contained in the `wli.properties` file located in the SBA home directory. Check if all the settings such as, host, port, user, password, specified for the WLS instance hosting the SBA and the WLS instance hosting the WLI application to be managed are appropriate for your setup. Modify them as necessary.

As you can see from [Listing 1-1: The wli.properties File](#), by default the WLS instance hosting the SBA is configured to listen at port 8001. And the WLS instance hosting WLI is configured to listen at localhost on port 7001.



The system environment variable `WL_HOME` must be set to the directory where WebLogic Platform 8.1 is installed (for example, `C:\bea\weblogic81`).

Listing 1-1: The wli.properties File

```
###-----###
### WLI-SPI SBA Configuration Properties ###
###-----###

wl.home=${env.WL_HOME}

#
# Settings for JVM and WLS instance hosting the SBA
#
sba.wls.host=localhost
```

```
sba.wls.port=8001
sba.wls.user=weblogic
sba.wls.password=weblogic
sba.wls.domain.dir=${project.dir}/internal/wsmfdomain
sba.wls.server.name=wsmfserver
sba.wls.min.heap.size=128M
sba.wls.max.heap.size=128M
sba.wls.max.new.size=32M

sba.wls.jpda.enabled=true
sba.wls.jpda.opts=transport=dt_socket,address=6006,server=y,susp
end=n

# how often, in milliseconds, to poll WLI for updates to its
MBeans

sba.wli.poll.frequency=20000

#
# Info for the WLI instance being managed
#
wli.wls.host=localhost
wli.wls.port=7001
wli.wls.user=weblogic
wli.wls.password=weblogic
```

Starting the WLI SPI SBA

There are two steps involved when starting the SBA agent.

- 1 You must start the WLI application server hosted in its WLS instance. To start WLI, use the standard `startWebLogic` script found in your WLI domain directory.

This WLS is configured by default to listen at port 7001. You know the server is fully started when you see the following message in the console window:

```
<Thread "ListenThread.Default" listening on port 7001, ip address *.*>
```

- 2 You must start the SBA hosted in its separate WLS instance.



Please note that on a UNIX system, the `JAVA_HOME` environment variable must be set in the `/etc/rc.config.d/wlisba` script

The following script is provided in the `bin/` subdirectory to start the SBA:

```
start-sba
```

This WLS is configured by default to listen on port 8001. This is the URL you will use to view the deployed Web services. See the section [Checking the WLI SPI SBA Status](#) on page 38. You know the server is fully started when you see the following message in the console window:

```
[java] --> 04-14-04 16:04:22 INFO [Thread-1] servlet: *** MO discovery completed successfully. ***
```

```
[java] --> 04-14-04 16:04:22 INFO [hp.thread.10.initial-publisher-refresh]controller: Smart Business Agent Controller has completed initial discovery; it is now fully initialized and started!
```

```
[java] <Apr 14, 2004 4:04:22 PM PDT> <Notice> <WebLogicServer> <BEA-000329> <Started WebLogic Admin Server "myserver" for domain "mydomain" running in Production Mode>
```

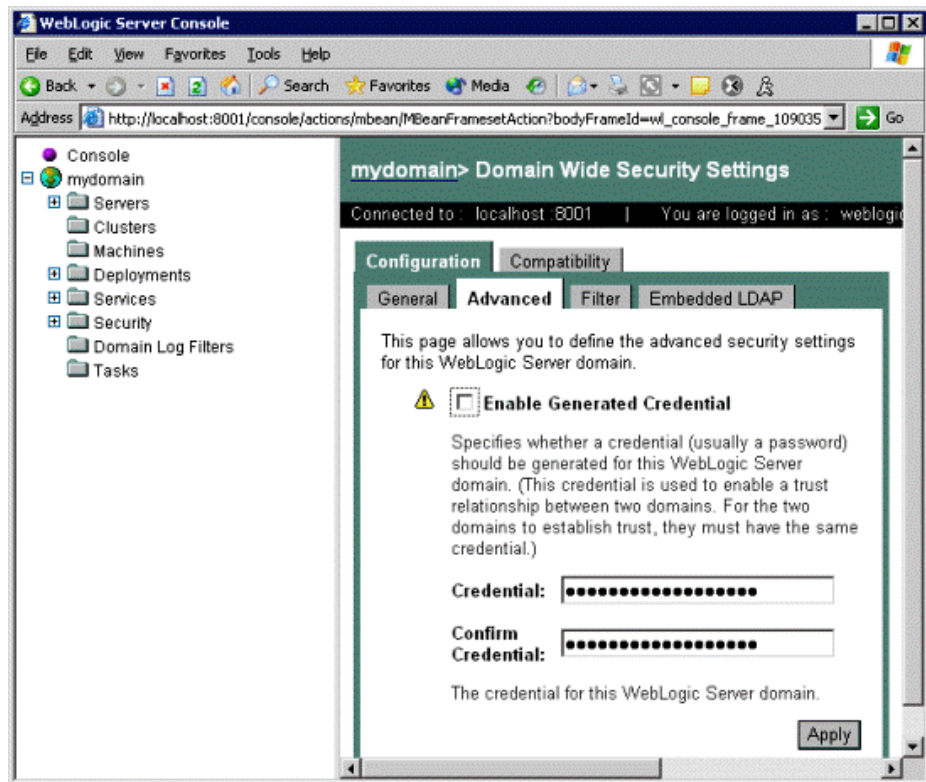
```
[java] <Apr 14, 2004 4:04:22 PM PDT> <Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>
```

```
[java] <Apr 14, 2004 4:04:22 PM PDT> <Notice> <WebLogicServer> <BEA-000355> <Thread "ListenThread.Default" listening on port 8001, ip address *.*>
```

Configuring the WebLogic Trust Relationship

After installing the WLI-SPI SBA and starting WLI and the SBA, the WebLogic trust relationship must be configured for the SBA to function properly. To configure the trust relationship, perform the following steps:

- 1 Log into the WLS console. For example, browse to **http://machine:port/console**.
- 2 Select **Security** in the left pane.
- 3 Select **Advanced** tab in the right pane.



- 4 Uncheck **Enable Generated Credential**.
- 5 Specify a fixed string in the Credential: text field (for example, **wlisp**) and confirm as indicated.
- 6 Click **Apply**.

- 7 Shutdown and restart the WLS instance.
- 8 Perform steps 1 through 7 for all WLS instances that need to be in the same trusted domain. This needs to be done once for the `WLI` WLS instance and once for the `SBA` WLS instance.

The specified credential is used by the WLS instance to authenticate itself to all other WLS instances.

Stopping the WLI SPI SBA

You can stop the WLS instance hosting the SBA by using the following script provided in the `bin\` subdirectory:

```
stop-sba
```

The WLS instance hosting WLI can be stopped by running the standard `stopWebLogic` script found in your WLI domain directory.

Checking the WLI SPI SBA Status

To check if the WLI SPI SBA has started successfully go to the URL:

`http://<host>:8001/wsmf/services`

Replace `<host>` with either your computer's DNS or IP address. The WSMF deployed services on this location are display as shown below:

15 Deployed WSMF Services

1. OrderNew_processtype [[WSDL](#)]
2. OrderRequisition_processtype [[WSDL](#)]
3. WorkflowBilling_processtype [[WSDL](#)]
4. WorkflowShipping_processtype [[WSDL](#)]
5. adapter_e2eWorkflow_AppView1__Default [[WSDL](#)]
6. archiver [[WSDL](#)]
7. eventgen_timer_Test [[WSDL](#)]
8. msgbroker [[WSDL](#)]
9. msgchannel_.deadletter.rawData_end2endServer [[WSDL](#)]
10. msgchannel_.deadletter.string_end2endServer [[WSDL](#)]
11. msgchannel_.deadletter.xml_end2endServer [[WSDL](#)]
12. msgchannel_end2end.StartChannel_end2endServer [[WSDL](#)]
13. msgchannel_end2end.billingshipping_end2endServer [[WSDL](#)]
14. server [[WSDL](#)]
15. worklist [[WSDL](#)]

Starting and Stopping the WLI SPI SBA Using the RC Script

For UNIX-based systems, the WLI SPI SBA installation provides an RC script that starts the SBA at system startup and stops the SBA at system shutdown. The script is located at `/sbin/init.d/wlisba` and should not be modified.

By default, the RC script is disabled and is not used during system startup or shutdown. The `/etc/rc.config.d/wlisba` configuration file contains the configuration settings for the `wlisba` RC script and is used to configure the RC script.

To start and stop the WLI SPI SBA using the RC script:

- 1 Using a text editor, open `/etc/rc.config.d/wlisba`.
- 2 Modify the following variables:
 - `WLISBA` - Set this variable to 1 to enable the RC script's execution at system startup/shutdown; by default, the RC script is disabled.
 - `WLISBA_HOME` - Set this variable to the directory where the SBA is installed.
 - `JAVA_HOME` - Set this variable to the directory where JDK 1.4 is installed; you may set this to the JDK that is bundled with WebLogic.
 - `WL_HOME` - Set this variable to the directory where WebLogic Server 8.1, SP2 or later is installed.
- 3 Save and close the file.
- 4 Reboot the system to ensure the RC script runs and that the SBA automatically starts.

[Table 3](#) lists all of the available variables in `/etc/rc.config.d/wlisba`:

Table 3 List of the variables available in /etc/rc.config.d/wlisba

Variable	Description
WLISBA	A flag used to enable the WLI SPI SBA subsystem; set to 1 to enable or 0 to disable; default value is 0. WLISBA=1
WLISBA_HOME	The path where WLI SPI SBA is installed; default value is /opt/wlisba1.0. WLISBA_HOME=/opt/wlisba1.0
WLISBA_LOG	The path to a file to which the SBA's output should be redirected; default value is /var/opt/wlisba1.0/log.txt. WLISBA_LOG=/var/opt/wlisba1.0/log.txt
JAVA_HOME	The path where JDK 1.4 or later is installed; no default value. JAVA_HOME=/opt/java1.4
WL_HOME	The path where WLS 8.1.2 or later is installed; no default value. WL_HOME=/opt/bean/weblogic81
ANT_OPTS	The options to pass to Ant when executing the start or stop targets in wli.xml. ANT_OPTS=-verbose

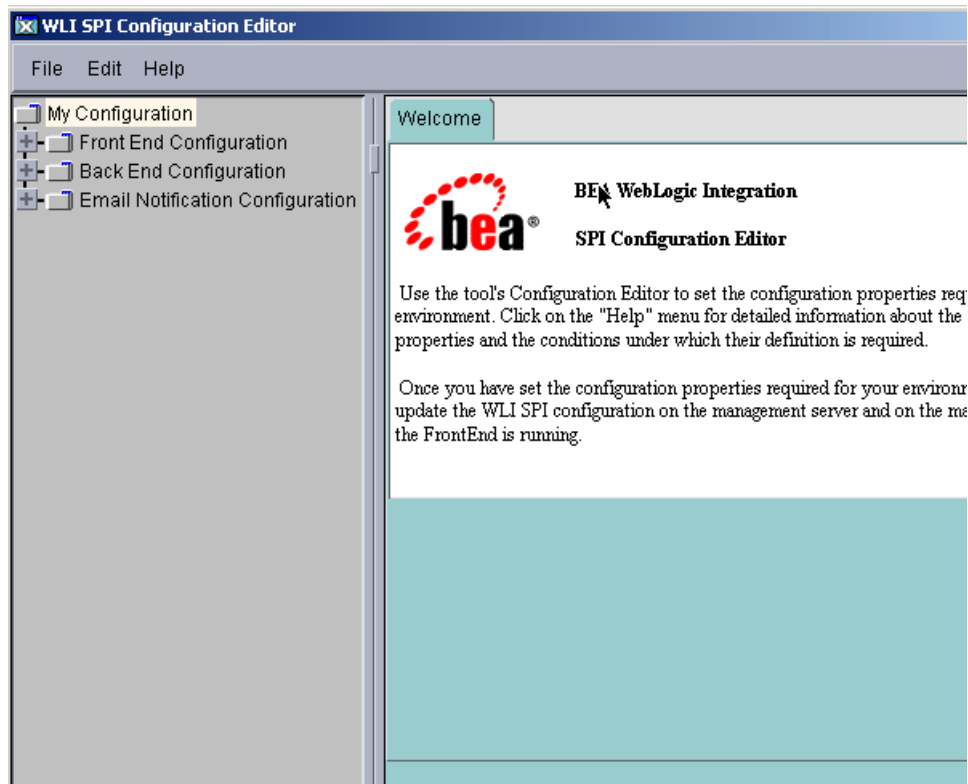
Configure the Backend Service and the Frontend Subagent

To configure the Frontend Subagent and Backend Service, you must define various configuration options. You can define these options using the WLI SPI Configuration Editor that is present in the **WLI SPI Tools** application group.

To configure the Backend Service and the Frontend Subagent:

- 1 From the OVO administrator GUI, open the Application Bank window by selecting **Window** → **Application Bank**.
- 2 In the Application Bank window, double-click the **WLI SPI Tools** application group.
- 3 In the Application Group: WLI SPI Tools window, right-click **Configure WLI SPI** application and select **Execute** from the popup menu.
- 4 The WLI SPI Configuration Editor window appears.

Figure 1 WLI SPI Configuration Editor



- 5 In the WLI SPI Configuration Editor, expand the **Back End Configuration** node.
- 6 Select **OVO Management Server Info**.

- 7 On the right pane, modify the Host Name and other fields if required.
- 8 Expand the **Front End Configuration** node.
- 9 Select **Managed Object Information**.
- 10 Add the WSDL location for each object you want the Frontend Subagent to monitor. For example:
http://<wli_sba_host>:8001/wsmf/services/server?wsdl



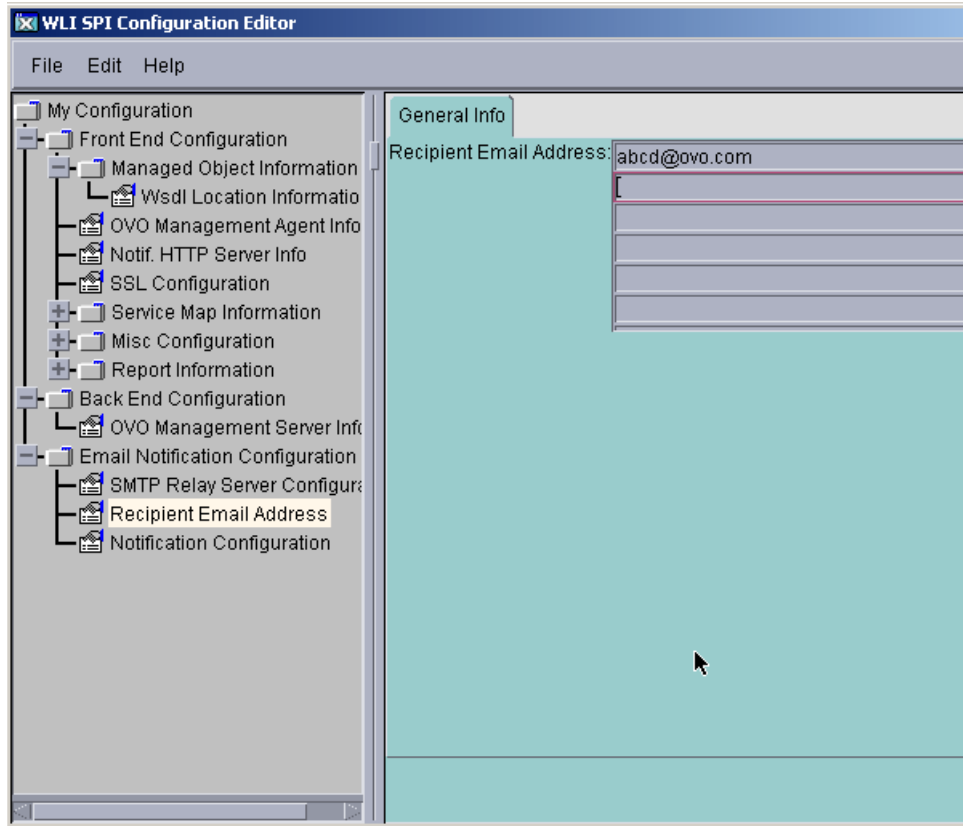
The WSDL location that is configured in the Managed Object Information area must be accessible before starting the Frontend Subagent. To ensure that the WSDL is accessible make sure that the WLI SPI SBA is running and the MO discovery is completed.

- 11 Select **OVO Management Agent Info**.
- 12 Modify the HostName, JRE Home, and any other fields if required.
- 13 Select **Email Notification Configuration**.
- 14 On the right side specify an email address that will be used as the Sender's email address when an email notification is sent.



If you are using an SMTP Gateway Server that requires user authentication for sending email, then you must provide a valid email address as the sender's address, which is allowed to send email using the SMTP Gateway Server selected.

- 15 Expand **Email Notification Configuration**.
- 16 Select **SMTP Relay Server Configuration**.
- 17 In the SMTP Relay Host Name field, enter the SMTP Gateway hostname to use. If the Gateway requires authentication, specify the user ID and password.
- 18 Select **Recipient Email Address**.



- 19 On the right pane, specify all the email addresses where email notifications are sent.
- 20 If you do not want to get email Notifications, select **Notification Configuration**.
- 21 On the right pane, select **disable** for SLA Violation Notification and Overdue Task Notification.
- 22 Select **File** → **Save** to save your changes to the configuration file.
- 23 Select **File** → **Exit**. In the output of application window, the following message appears

```
'SPI configuration change is completed. Please start/restart the SPI for the configuration change to take effect.'
```

It may take a few minutes for the message to appear. Do not close the output of application window until you see the completion message.

24 Click the **Close** button in the Output of Application window.



Configuration changes do not become effective until the SPI is started or restarted.

Assign WLI SPI Templates

OVO-based templates are used to capture management data and metrics. The templates are specific to the WLI SPI and are included with the WLI SPI installation. In general, the templates collect and monitor data. The data includes metric data, WLI managed environment messages, logging data, and server process data. The data is used to effectively manage the WLI environment within OVO. See [Appendix B, List Templates and Reports](#) for a complete reference of the templates included with the WLI SPI.

The WLI SPI templates are organized in two groups:

- WLISPI-Backend Template Group – templates that are specific to the Backend Service and must be assigned to the management server node.
- WLISPI-Frontend Template Group – templates that are specific to the Frontend Subagent. If the Frontend Subagent is on a managed node, the templates must be assigned and deployed to the managed node.

Before assigning WLI SPI templates, you must configure the data sources that are used to store the collected data metrics.

Configure Data Sources for Metric Data Collection

The metrics collected by the Frontend Subagent templates are stored in a data source. A script is provided that creates and configures the data source.

To configure data source for metric data collection:

- 1 From the command prompt, change directories to `/opt/OV/bin`.
- 2 Run the command:
`./wli-perl wli-create-datasources`

Assign Backend Service Templates

The Backend Service templates must be assigned to the management server before the templates can be used.

To assign the Backend Service templates on the management server:

- 1 From any OVO window, open the Node Bank window by selecting **Window** → **Node Bank**.
- 2 In the Node Bank window, select the management server node where the Backend Service is running.
- 3 Select **Actions** → **Agents** → **Assign Templates**.
- 4 In the Define Configuration dialog box, click the **Add** button.
- 5 In the Add Configuration dialog box, click the **Open Template Window** button. The Message Source Templates dialog box displays.
- 6 From the Template Groups list on the left, select the **SPI for WLI** template group.
- 7 From the Template groups list on the right, select the **WLISPI-Backend** template group.
- 8 Switch to the Add Configuration dialog box and click the **Get Template Selections** button.
- 9 In the Add Configuration dialog box, click **OK**.
- 10 Close the Message Source Templates dialog box by selecting **Templates** → **Close Window**.
- 11 Click **OK** in the Define Configuration dialog box.

Deploy Backend Service Templates to the Management Server

To deploy the Backend Service templates to the management server:

- 1 From the OVO administrator GUI, open the Node Bank window by selecting **Window** → **Node Bank**.
- 2 In the Node Bank window, select the node where the management server is running.
- 3 Select **Actions** → **Agents** → **Install / Update SW & Config**.

- 4 In the Install / Update OVO Software and Configuration dialog box, select **Templates** in the Components frame.
- 5 Ensure that **Force Update** is selected in the Options frame.
- 6 Click the **OK** button.

Assign Frontend Subagent Templates

The Frontend Subagent template groups must be assigned and deployed to the node where the Frontend Subagent is installed before the templates can be used.

To assign Frontend template groups to a node:

- 1 From any OVO window, open the Node Bank window by selecting **Window** → **Node Bank**.
- 2 In the Node Bank window, select the node where the Frontend Subagent is running.
- 3 Select **Actions** → **Agents** → **Assign Templates**.
- 4 In the Define Configuration dialog box, click the **Add** button.
- 5 In the Add Configuration dialog box, click the **Open Template Window** button. The Message Source Templates dialog box displays.
- 6 From the Template Groups list on the left, select the **SPI for WLI** template group.
- 7 From the Template Groups list on the right, select the **WLISPI-Frontend** template group.
- 8 Switch to the Add Configuration dialog box and click the **Get Template Selections** button.
- 9 In the Add Configuration dialog box, click **OK**.
- 10 Close the Message Source Templates dialog box by selecting **Templates** → **Close Window**.
- 11 Click **OK** in the Define Configuration dialog box.

Deploy Frontend Templates to a Managed Node

If the Frontend Subagent is running on managed node, the templates must be deployed to the managed node. This can be done from the management server.

To deploy the Frontend Subagent templates to a managed node:

- 1 From the OVO administrator GUI, open the Node Bank window by selecting **Window** → **Node Bank**.
- 2 In the Node Bank window, select the node where the Frontend Subagent is running.
- 3 Select **Actions** → **Agents** → **Install / Update SW & Config**.
- 4 In the Install / Update OVO Software and Configuration dialog box, select **Templates** in the Components frame.
- 5 Ensure that **Force Update** is selected in the Options frame.
- 6 Click the **OK** button.

Verify Deployed Templates

The following procedure can be used to verify that WLI SPI template groups have been deployed. The procedure is completed on the Frontend Subagent node to verify the Frontend templates and on the Backend Service node to verify the Backend templates.

To verify deployed templates:

- 1 From a command prompt, change directories to `/opt/OV/bin/OpC`.
- 2 Run the command:
opctemplate
- 3 In the output, verify that the deployed WLISPI templates display.

Start the Backend and the Frontend

The OVO components can be started from the OVO Administration GUI. Before starting the components, you must have installed and configured the Frontend Subagent detailed in the previous sections.



Make sure you have set the `JAVA_HOME` variable to the JRE installation directory. See the section [Configure the Backend Service and the Frontend Subagent](#) on page 41. Also, make sure that WLI SPI SBA is running and has completed the MO discovery.

To start the backend and the frontend:

- 1 From the OVO administrator GUI, open the Application Bank window by selecting **Window** → **Application Bank**.
- 2 From the Application Bank window, double-click the **WLI SPI Tools** icon.

From the Application Group: WLI SPI Tools window, right-click **Start WLI SPI** and select **Execute** in the popup menu. An Output of Application window displays. A message similar to the following displays:

```
Node <machine>.<domain name>:
```

```
Starting OpC services...Done.
```

```
Starting up the backend...
```

```
The backend started.
```

```
Starting up the frontend...
```

The frontend startup is complete when the message 'Service Map Created' is emitted in the `/var/opt/OV/log/wli/frontend.log` file.

- 3 Use the steps in the [Checking the WLI SPI OVO Components' Status](#) section of this chapter to verify that the WLI SPI is started.

Restarting the Backend and the Frontend

To restart the backend and the frontend:

- 1 From the OVO administrator GUI, open the Application Bank window by selecting **Window** → **Application Bank**.
- 2 From the Application Bank window, double-click the **WLI SPI Tools** icon.

From the Application Group: WLI SPI Tools window, right-click **Restart WLI SPI** and select **Execute** in the popup menu. An Output of Application window displays. A message similar to the following displays:

```
Node <machine>.<domain name>:
```

```
Starting OpC services...Done.
```

```
The frontend was stopped successfully
```

```
The backend was stopped successfully
```

```
Starting up the backend...
```

```
The backend started.
```

```
Starting up the frontend...
```

The Frontend startup is complete when the message 'Service Map Created' is emitted in the `/var/opt/OV/log/wli/frontend.log` file.

- 3 Use the steps in the section [Checking the WLI SPI OVO Components' Status](#) on page 52 to verify that the WLI SPI is started.

Stop the Backend and the Frontend

To stop the backend and the frontend:

- 1 From the OVO administrator GUI, open the Application Bank window by selecting **Window** → **Application Bank**.
- 2 From the Application Bank window, double-click the **WLI SPI Tools** icon.
- 3 From the Application Group: WLI SPI Tools window, right-click **Stop WLI SPI** and select **Execute** in the popup menu. An Output of Application window displays a message similar to the following:
Node xxx:
Shutting down OpC services... Done.
The frontend was stopped successfully.
The backend was stopped successfully.
- 4 Click the **Close** button.
- 5 Use the steps in the section [Checking the WLI SPI OVO Components' Status](#) on page 52 to verify that the WLI SPI is stopped.

Checking the WLI SPI OVO Components' Status

You can check the status of the WLI SPI to see if it is started and operating successfully. You can also check the status to insure that the WLI SPI has been stopped successfully. This is helpful when debugging any problems.

To check the Status of the WLI SPI:

- 1 From the OVO administrator GUI, open the Application Bank window by selecting **Window** → **Application Bank**.
- 2 From the Application Bank window, double-click the **WLI SPI Tools** icon.
- 3 From the Application Group: WLI SPI Tools window, right-click **Check Status WLI SPI** and select **Execute** in the popup menu. The status of both the Backend and Frontend Subagent are displayed.
- 4 Click the **Close** button, after checking the status.

Verify a Start Status

If the WLI SPI is started without any errors, the following status displays when using the WLI SPI Check Status command:

```
object manager name: wli-backend
state:                RUNNING
PID:                  4063
last message:        -
exit status:         -
Node <machine>.<domain name>:
OVO Managed Node status:
-----
Control Agent        /opt/OV/bin/OpC/opcctl1a (2428) is running
Message Agent       /opt/OV/bin/OpC/opcm1sga (5944) is running
BBC Local Location Broker /opt/OV/bin/llbserver (5945) is
running
Subagent 26:
  WLI SPI Subagent (Frontend)
  /opt/OV/bin/wli-perl /opt/OV/bin/wli-start-frontend (4389)
is
  running
Done.
```

When using an OVO 8.x HTTPS agent node, the following status displays:

```
object manager name: wli-backend
state:                RUNNING
PID:                 4063
last message:        -
exit status:         -
Node <machine>.<domain name>:
OVO Managed Node status:
-----
ovcd          OV Control          CORE          (14629)
Running
ovbbccb      OV Communication Broker CORE          (14630)
Running
ovconfd     OV Config and Deploy    COREEXT       (14631)
Running
Subagent 26:
WLI SPI Frontend Subagent wli-frontend    (4389) is running
Done.
```



If the Frontend Subagent is not running, it may be still starting up. On the Frontend Subagent node, look in the `/var/opt/OV/log/wli/frontend.log` file for the 'Service Map Created' message. If the message isn't emitted, the Frontend Subagent has not completed its startup.

Verify a Stop Status

If the WLI SPI has stopped without any errors, the following status displays when using the WLI SPI Check Status command:

```
object manager name: wli-backend
state:                NOT_RUNNING
PID:                 -
last message:        -
exit status:         Exit(0)
Node <machine>.<domain name>:
OVO Managed Node status:
-----
Control Agent      /opt/OV/bin/OpC/opcctla (2428) is running
Message Agent     /opt/OV/bin/OpC/opcmgsa (5944) is running
BBC Local Location Broker /opt/OV/bin/llbserver (5945) is
running
```

```
Subagent 26:  
  WLI SPI Subagent (Frontend)  
    /opt/OV/bin/wli-perl /opt/OV/bin/wli-stop-frontend isn't  
running  
Done.
```

When using an OVO 8.x HTTPS agent node, the following status displays:

```
object manager name: wli-backend  
state:                NOT_RUNNING  
PID:                  4063  
last message:        -  
exit status:         Exit(0)  
Node <machine>.<domain name>:  
OVO Managed Node status:  
ovcd      OV Control          CORE          (14629)  
Running  
ovbbccb   OV Communication Broker CORE          (14630)  
Running  
ovconfd   OV Config and Deploy   COREEXT      (14631)  
Running  
Subagent 26:  
WLI SPI Frontend Subagent wli-frontend  isn't running  
Done.
```

Install the Java Console

OVO for UNIX includes a Java-based operator's console that is used to monitor an OVO managed environment. In regards to the WLI SPI, the Java console is used to:

- View the status of the WLI SPI and WLI services.
- Perform impact analysis and root cause analysis for the WLI SPI and WLI services.
- Interact with WLI SPI tools.
- View and manage WLI and WLI SPI messages.
- Access the HP Resource Explorer

The Java Console can run on any system where a JRE is installed. However, the instructions in this section are specific to installing the Java Console on Windows. See the OVO documentation or the Java Console's Online Help (once you have installed the Java console on Windows) for instructions on additional platform installation instructions.



To access the HP Resource Explorer from within the Java Console, you must install the Java Console on a Windows system. See the section [Install the HP Resource Explorer](#) on page 57.

To install the Java console:

- 1 `Ftp /opt/OV/www/htdocs/ito_op/ITO_JAVA.exe` from the OVO management server to the PC where you are installing the console.
- 2 Execute `ITO_JAVA.exe`. An install shield displays and guides you through the installation process.

Starting the Java Console

To start the Java Console:

- 1 Double-click the **Java Console** icon on the desktop.
- 2 Enter the following information in the HP OpenView Operations Login dialog box (if a different operator was configured, use that operator name instead of "WLISPI_Op"):
 - User Name: **WLISPI_Op**

- Password: **WLISPI_Op**
 - Management Server: <your OVO management server system>
- 3 Click **OK**. The Java Console starts and displays information for your OVO environment.
 - 4 From the Object Pane, expand the Services node to view WLI services.
 - 5 From the Object Pane, expand the Message Groups node to view the WLI and WLISPI message groups.
 - 6 From the Object Pane, expand the Applications node to view WLI SPI applications.



If you are new to the Java console, see the OVO documentation or the Java Console's Online Help.

Install the HP Resource Explorer

The HP Resource Explorer is used to interact with WLI Managed Objects (MOs) that are exposed for management purposes. The Resource Explorer interacts directly with the WLI SPI SBA component and allows the operator to perform many client management tasks. These include:

- Browsing MO published events
- Browsing relationships between MOs
- Browsing and editing MO attributes
- Invoking operations on MOs and editing their parameter values

The HP Resource Explorer must be installed on a Windows platform. In addition, when accessing the Resource Explorer from the Java Console, the Console and Resource Explorer must be installed on the same computer.

To install the HP Resource Explorer:

- 1 Ftp the HP Resource Explorer package (`wli-resource-explorer.zip`) located on the management server at `/opt/OV/bin`.
- 2 Unzip the HP Resource Explorer to any directory on the local machine.
- 3 Modify the user's `PATH` environment variable to include the directory where you installed the HP Resource Explorer. The `PATH` is used by the Java Console to locate and start the Resource Explorer.
- 4 Create a `JAVA_HOME` variable that contains the path to JRE 1.4.



The Resource Explorer does not work with JRE 1.5.

Starting the HP Resource Explorer from the Java Console

The HP Resource Explorer is typically started from the Java Console. You can start multiple concurrent Resource Explorer sessions.

To start the Resource Explorer from the Java Console:

- 1 Start the HP Java Console.
- 2 From the Object Pane, expand the Services node.
- 3 From the WLIService node, right-click a WLI service and then from the popup menu select **Start** → **Resource Explorer**. The Resource Explorer starts in its own window in a separate process.

Online help is included with the Resource Explorer. From the Explorer's menu bar, select **Help**. The Online Help includes instructions for browsing and editing MOs as well as invoking an MO's operations.

Starting the HP Resource Explorer form the Command Line

The Resource Explorer can also be started from the command line using the `hp_resource_explorer.bat` file. The file is located in the `root` directory where the Resource Explorer is installed.

As part of the command, you must pass the URL of the management Web service. Once invoked, this Web service becomes the root of the functional tree in the Resource Explorer. The following is an example command line:

```
hp_resource_explorer.bat -moUri  
http://<host>:8001/wsmf/services/server?wsdl
```

To view a list of all the available management Web services, use the following URL:

```
http://<host>:8001/wsmf/services
```

Install WLI SPI Reports

The WLI SPI includes an integration with HP OpenView Reporter, which is a Windows-based report management system. As part of the integration, a set of reports are included with the WLI SPI and used to view the performance of the WLI Applications.



HP OpenView Reporter must be installed prior to completing the instructions in this section. For detailed installation instructions, see the HP OpenView Reporter documentation. In addition, the Frontend Subagent system must be running either MeasureWare or the OVO embedded performance component (Coda). See [Chapter 3, Working with the WLI SPI](#) for more information on MeasureWare and on the OVO embedded performance component (Coda).

To install the WLI SPI Reports:

- 1 From the HP OpenView Smart Plug-ins for OVO/UNIX CD, change directories to `\OV_REPORTER\WLISPI_A.01.03`.
- 2 Double-click the **HP OpenView BEA WLI SPI Reports.msi**. The InstallShield Wizard is displayed.
- 3 Click **Next**. The Setup Type screen is displayed.
- 4 Accept the default option (Complete) and click **Next**. The Ready to Install program screen is displayed.
- 5 Click **Install** to initiate the installation.
- 6 After the installation is complete, click **Finish**.

Uninstall the WLI SPI Components

To uninstall the WLI SPI, use the procedures in this section. The procedures should be completed in the order in which they are listed.



Make sure that you uninstalled the Frontend Subagent before you remove the WLI SPI from the OVO management server. If you do not, you will encounter problems if you try to reinstall the WLI SPI.

Uninstalling the WLI SPI SBA

This component can be uninstalled by running the uninstaller executable found in the `UninstallerData` subdirectory of the SBA home directory.

Uninstalling the Frontend Subagent

The Frontend Subagent must be stopped before you can stop the WLI SPI.

To uninstall the Frontend Subagent:

- 1 On the node where the Frontend Subagent is installed, stop the Subagent by running:

```
opcagt -stop -id 26 (OVO 7.x)
```

```
ovc -stop WLI (OVO 8.x)
```

- 2 Launch the OVO administrator GUI on the OVO management server and log on as **opc_adm**.
- 3 In the Node Bank window, select the node where the Frontend Subagent is installed.
- 4 Select **Actions** → **Subagents** → **Deinstall**.
- 5 In the Deinstall Subagents dialog box, select **WLI SPI** Subagent in the Subagents frame on the left.

Click the **OK** button. The Install Subagent Packages dialog box is displayed. The following confirmation message is displayed:

```
"SubAgent successfully removed and unconfigured from Managed Node xxx."
```

- 6 Press **Enter**.
- 7 If the `opcagt` is not running on the node where the Frontend Subagent was installed, restart it by running.


```
opcagt -start -id 26 (OVO 7.x)
ovc -start WLI (OVO 8.x)
```
- 8 Verify that the Frontend Subagent is uninstalled by running the `opcagt` command on the node where the Frontend Subagent was running:


```
opcagt -status -id 26 (OVO 7.x)
ovc -status WLI (OVO 8.x)
```

The output on OVO 7.x should be:

```
Error opcctla (Control Agent)(xxx) : Subagent 26 not
registered
```

The output on OVO 8.x should be:

```
(ctrl-4) No component matches target.
```

Remove the WLI SPI Bits on HP-UX

- 1 Stop the Backend Service on the OVO management server by running **ovstop wli-backend**.
- 2 On the OVO management server, run **swremove**.
- 3 From the SD Remove - Software Selection dialog box, select the **SPIWLI**.
- 4 Select **Actions** → **Remove**.
- 5 From the Remove Analysis dialog box, click the **OK** button.
- 6 From the Remove Window dialog box, verify that the Status says that it completed successfully. If the status is "Completed with Warnings", this is expected and you can proceed. If the status indicates errors, click the **Logfile** button to view the errors.
- 7 Click the **Done** button.
- 8 Select **File** → **Exit** to Close the SD Remove - Software Selection dialog box.

Remove the WLI SPI Bits on Solaris

To remove the WLI SPI Bits on Solaris:

- 1 Stop the Backend Service on the OVO management server by running **ovstop wli-backend**.
- 2 On the OVO management server, run `swremove` from the command line and include the WLI SPI SD install name (SPIWLI) in the command.
swremove SPIWLI

Remove WLI SPI Message Groups

- 1 From any OVO window, select **Window** → **Message Group Bank**.
- 2 In the Message Group Bank window, right-click the **WLI** message group and select **Delete** from the popup menu.
- 3 Click the **Yes** button to confirm the delete.
- 4 Right-click **WLI SPI** message group and select **Delete** from the popup menu.
- 5 Click the **Yes** button to confirm the delete.
- 6 Click the **Close** icon to close the window.

Remove the WLI SPI Application Group

- 1 From any OVO window, open the Application Bank window by selecting **Window** → **Application Bank**.
- 2 From the Application Bank window, right-click **WLI SPI Tools** and select **Delete** from the popup menu.
- 3 Click the **Yes** button to confirm the delete.
- 4 Click the **Close** icon to close the window.

Remove the WLI SPI User Profile

- 1 From OVO window, select **Window** → **User Profile Bank**.
- 2 From the User Profile Bank window, right-click the **WLI SPI User Profile** and select **Delete** from the popup menu.

- 3 Click the **Yes** button to confirm the delete.
- 4 Click the **Close** icon to close the window.

Remove the WLI SPI User

- 1 From OVO window, select **Window** → **User Bank**.
- 2 From the User Bank window, right-click **WLISPI_Op** and select **Delete** from the popup menu.
- 3 Click the **Yes** button to confirm the delete.
- 4 Click the **Close** icon to close the window.

Remove WLI SPI Message Templates

- 1 From any OVO window, select **Window** → **Message Source Templates**.
- 2 In the Message Source Templates window, select **SPI for WLI** in the Template Groups frame on the left.
- 3 Select all of the template groups in the right frame.
- 4 Click the **Delete From All** button.
- 5 Click the **Yes** button to confirm the delete.
- 6 Select **SPI for WLI** again in the Template Groups frame.
- 7 Click the **Delete From All** button.
- 8 Click the **Yes** button to confirm the delete.
- 9 Select **[Toplevel]** in the Template Groups frame.
- 10 Select all of the template groups that start with **WLISPI-** in the right frame.
- 11 Click the **Delete From All** button.
- 12 Click the **Yes** button to confirm the delete.
- 13 Select **[Toplevel]** again in the Template Groups frame.
- 14 Select all of the templates that start with **WLISPI-** in the right frame.
- 15 Click the **Delete From All** button.
- 16 Click the **Yes** button to confirm the delete.

- 17 Close the window by selecting **Templates** → **Close Window**.

Reinstall the Templates, Monitors, Commands, and Actions

- 1 From the Node Group Bank window, select the **WLISPI-UNIX** node group.
- 2 Select **Action** → **Agents** → **Install / Update SW & Config**.
- 3 In the Install / Update OVO Software and Configuration dialog box, select **Templates, Monitors** and **Commands** in the Components frame on the left.
- 4 Select **Force Update** in the Options frame.
- 5 Click the **OK** button.
- 6 Click the **OK** button.

Remove the WLI SPI Node Groups From the OVO Database

- 1 From any OVO window, select **Window** → **Node Group Bank**.
- 2 In the Node Group Bank window, right-click the **WLISPI-UNIX** node group and select **Delete** from the popup menu.
- 3 Click the **Yes** button to confirm the delete.
- 4 Right-click **WLISPI-EXTERNAL** node group and select **Delete** from the popup menu.
- 5 Click the **Yes** button to confirm the delete.
- 6 Click the **Close** icon to close the window.

Configuring WLI SPI to Run as Non-Root User

OVO processes normally run as user root on UNIX systems. The root or administrative privileges enable the processes to perform the following:

- Access OpenView files and resources that are normally restricted to privileged access only.
- Switch user for application specific access rights.
- Access operating system resources such as log files and configuration files.

- Run application or operating system specific commands and executables.

WLI SPI processes function in a similar way. But there may be systems within IT environments where it is necessary to limit the number of processes that have root permissions to a small, well defined and tested group. WLI SPI on UNIX managed node can be configured to run under a user that does not have root permissions. This is often referred to as “running as non-root”. But while installing WLI SPI, you must be logged onto the management server and managed nodes as root user.



This feature is not supported if the Frontend Subagent and Backend Service are both located on the OVO management server.

Configuring WLI SPI to run as non-root user is supported on HTTPS agents with OVO. You can do this using the `ovswitchuser` tool.

Steps to configure SPI as non-root user are listed below:

- 1 Stop OV processes by executing the following command:

```
ovc -kill
```

- 2 Assign the permissions and ownership to non-root user for agent processes by running the command:

```
/opt/OV/bin/ovswitchuser.sh -existinguser <nonroot user>  
-existinggroup <nonroot group>
```

For example: `/opt/OV/bin/ovswitchuser.sh -existinguser wliuser -existinggroup wligroup`

- 3 Assign the permissions and ownership to non-root user for WLI SPI by running the command:

```
/var/opt/OV/bin/instrumentation/wli-switch-user.pl  
<nonroot user> <nonroot group>
```

- 4 Change BBC port from 383 (default value) to any available port greater than 1024, both on the systems where Backend Service and Frontend Subagent are running.

- On the managed node (where Frontend is installed) run the following command:

```
ovconfchg -ns bbc.cb.ports -set PORTS "<host name of the  
system where frontend is installed>:<port_num>"
```

For Example: `ovconfchg -ns bbc.cb.ports -set PORTS "ovnode.domain.com:8001"`

- b On the management server (system where Backend service is installed) run the following command:

```
ovconfchg -ns bbc.cb.ports -set PORTS "<host name of the management server>:<default port>,<host name of the machine where frontend is installed>:<new frontend port>"
```

```
For Example: ovconfchg -ns bbc.cb.ports -set PORTS "ovnode.domain.com:383,ovserver.domain.com:8001"
```

- 5 If the performance component used by OVO is the embedded performance component (Coda), change the port on the managed node from 381(default value) to any available port greater than 1024 by executing the command:

```
ovconfchg -ns coda.commm -set SERVER_PORT <port number>
```



If the performance component used by OVO is OV Performance Agent, then data collection is not supported when WLI SPI is running as a Non-root user.

- 6 Confirm the port change by executing the following command on the system where frontend is installed:

```
ovconfget
```

- 7 Start the agent as non-root user by running the following command on the system where frontend is installed:

```
su - <nonroot user>
```

- 8 Start `ovc` as non-root user by running the following command:

```
ovc -start (ensure that all processes are running)
```

For information on supported agent platforms, refer to *HP OpenView Smart Plug-ins for WLI Release Notes*.

Refer to the manual, *HP OpenView Operations HTTPS Agent Concepts and Configuration Guide*, for more information on configuring non-root user.

3 Working with the WLI SPI

This chapter provides some tasks that are typically performed when using the WLI SPI. In particular, the following sections are included:

- Service Management
- Event Management
- Monitoring and Data Collection
- Reporting and Performance Graphs
- Monitoring Performance Metrics with OVPM
- Modify Logging and Tracing Levels

Service Management

Service management is achieved using the Service Navigator that is included in the OVO Java Console and using the HP Resources Explorer plug-in to the Service Navigator. During runtime, the WLI SPI automatically discovers WLI managed resources and represents them as a Service Map. Any deployment changes that occur in the WLI environment are dynamically synchronized with the Service Map.

The Service Navigator and HP Resource Explorer also allow detailed management of the resources that are presented in the Service Map. The detailed management includes browsing the managed resource hierarchy, their relationships, attributes, metrics, and invocation of methods that are exposed by the resource.

Instructions for installing and starting the Java Console (including the Service Navigator) and the HP Resource Explorer are located in Chapter 2. In addition, Online Help is available for both the Java Console and HP Resource Explorer.



This section is intended only as a quick start reference and not a replacement for the Java Console or HP Resource Explorer documentation.

Viewing WLI Managed Resources

To view WLI managed resources:

- 1 Start and log in to the Java Console.
- 2 From the Object Pane, expand the Services node to view WLI resources. The resources are represented in a hierarchy. Expand any node to view any contained resources.
- 3 From the WLIService node, right-click a WLI resource and from the popup menu select **Start** → **Resource Explorer**. The Resource Explorer starts in its own window as a separate process. Like the service node, the Resource Explorer lists the WLI resources in a hierarchy form and allows you to select each resource, view its management data, and invoke any available operations.

Filtering Unwanted MO Types

You can define a list of MO types that you do not want to be displayed in the Java Console's Service Map. This feature is especially useful if many MOs are being exposed.

To filter out unwanted MO types:

- 1 From the OVO administrator GUI, open the Application Bank window by selecting **Window** → **Application Bank**.
- 2 In the Application Bank window, double-click the **WLI SPI Tools** application group.
- 3 In the Application Group: WLI SPI Tools window, right-click **Configure WLI SPI** application and select **Execute** from the popup menu.
- 4 In the WLI SPI Configuration Editor tool, click the icon next to Top of tree.
- 5 Click the icon next to Front End Configuration.
- 6 Click the icon next to Misc. Configuration.
- 7 Select **Managed Objects to Ignore**.
- 8 From the General Info section in the right frame, enter the MO type that you want to ignore. All the MOs of the specified MO type are ignored. For example, if you do not want the Archiver MOs to appear in the service Map, specify: **urn:WebLogicIntegration/archiver**
- 9 Select **File** → **Save** to save your changes to the configuration file.
- 10 Select **File** → **Exit**.
- 11 In the Output of Application window, you should see the message 'SPI configuration change is completed. Please start/restart the SPI for the configuration change to take effect.' It may take some time before this message appears. Do not close the Output of Application window until you see the message.
- 12 Restart the WLI SPI.
- 13 Verify that the Frontend Subagent startup has completed. Look for the 'Service map created' message in the `/var/opt/OV/log/wli/frontend.log` file.
- 14 Start and log in to the Java Console. The MO that you filtered is no longer listed in the service map.

Event Management

The WLI SPI provides event management that alerts an OpenView operator on any of the following conditions:

- SLA (on execution time) violations of business process instances
- Success ratio violations of business process type
- Notification on error from adapters, event generators, and message channels
- Overdue tasks



The WLI SPI can also send email notifications based on any event. See the section [Configure the Backend Service and the Frontend Subagent](#) on page 41 in Chapter 2 for setup instructions.

Viewing WLI Events

All events through WLI are captured and displayed in the Java Console's message browser.

To view WLI Events:

- 1 Start and log in to the Java Console.
- 2 From the Object Pane, expand the Message Groups node to view the WLI and WLI SPI message groups.
- 3 Right-click a message and select **Properties**. The Message Properties dialog box displays and lists additional details about the event.

Monitoring and Data Collection

The Frontend Subagent uses metric definitions to capture WLI management data for use in OVPM or OV Reporter (i.e., when generating alarms, graphs, and reports). This section provides instructions on how to customize what data is collected and how to change the default collection behavior. A brief overview of the how metrics are defined is also provided.

Metric Definition Configuration File

Metrics are defined in the `MetricDefinitions.xml` configuration file. The file is located on the managed node in `/var/opt/OV/conf/wli`. The elements for this file are described below.

- `<MetricDefinitions>` – The `MetricDefinitions` element is the top-level element within the document. It contains one collection of metrics, consisting of one or more metric definitions.
- `<Metric>` – The `Metric` element represents one metric. Each metric has a unique ID. If a user-defined metric is an alarming, graphing or reporting metric, the metric ID must be `WLISPI_0xxx` where `xxx` is a number from 700 through 799. However, if the metric is used only within the calculation of another metric, the metric ID must begin with a letter (case-sensitive) and can be followed by any combination of letters, numbers and underscores. The `Metric` element contains one more source elements that represent the metric data source.
- Two data sources are supported: WSM and calculations.

The following table lists attributes for the `Metric` element:

Table 1 Metric Element Attributes

Attribute	Type/ Values	Required	Default	Description
id	ID	yes	N/A	The metric ID
namea	text	no	no	The metric name, used for graphing and reporting. The name can be up to 20 characters in length.
alarm	yes no	no	no	If yes, the metric value is sent to the agent using opcmn.
report	yes no	no	no	If yes, the metric value is logged for reporting.
previous	yes no	no	yes	If yes, the metric value is saved in a history file so that deltas can be calculated. If you are not calculating deltas on a metric, set this to “no” for better performance.
graph	yes no	no	no	If yes, the metric is logged for graphing.
description	text	no	“”	A description of the metric.

- <WSM> – The WSM element is used when the data source of the metric is a WLI metric definition. The WSM element contains the following sub-elements:
 - <MetricName> – The WLI metric definition name.
 - <ObjectTypeList> – List of MO types for which the metric values will be collected.
 - <ObjectWSDLList> – List of MO instances for which the metric values will be collected.

- **<Calculation> and <Formula>** – The Calculation element is used when the data source of the metric is a calculation using other defined metrics. The Calculation element contains a Formula element which consists of a string that specifies the mathematical manipulation of other metric values to obtain the final metric value. The metrics are referred to in the calculation expression by their metric ID. The collector can perform calculations that combine one or more metrics to define a new metric. The result of the calculation is the metric value. Calculations must use the following syntax:
 - Operators supported are +, -, /, *, and unary minus.
 - Operator precedence and associativity follows the Java model.
 - Parentheses can be used to override the default operator precedence.
 - Allowable operands are metric IDs and literal doubles.

A metric ID can refer to either a WSM metric or another calculated metric. Literal doubles can be specified with or without the decimal notation. The metric ID refers to the id attribute of the Metric element in the metric definitions document. The calculation parser also supports the following functions. All function names are lowercase and take a single parameter which must be a metric ID:

- `delta` – returns the result of subtracting the previous value of the metric from the current value.
- `interval` – returns the time in milliseconds that has elapsed since the last time the metric was collected.

The following example defines a metric whose value is the ratio (as expressed as a percent) of Metric_1 to Metric_3:

```
<Formula>(Metric_1/Metric_3)*100</Formula>
```

The following example could be used to define a metric that is a rate (number of times per second) for Metric_1.

```
<Formula>(delta(Metric_1)/interval(Metric_1))*1000</Formula>
```

Modifying Data Collection

By default, data is collected for all WLI Business Processes, WLI Adapters, WLI Message channels, WLI Event Generators, and WLI System Archivers which are discovered in the WLI Managed environment. If you want to collect

data for a subset of the above mentioned WLI entities, you can modify the `/var/opt/OV/conf/wli/MetricDefinitions.xml` file. The idea is instead of collecting metric data on a type of MO, one may be interested in collecting metric data for specific instances of a type of MO. For example, if you want to collect `AvgExecTime` metric for only the Business Process:

`http://15.244.63.244:8001/wsmf/ services/OrderNew_processtype?wsdl`

Change:

```
<Metric id="WLISPI_0001" name="BP_AvrgExecTime" alarm="no"
      graph="yes" report="yes">
  <WSM>
    <MetricName>AvgExecTime</MetricName>
    <ObjectTypeList>
      <ObjectType>urn:WebLogicIntegration/processtype</
ObjectType>
    </ObjectTypeList>
  </WSM>
</Metric>
```

To:

```
<Metric id="WLISPI_0001" name="BP_AvrgExecTime" alarm="no"
      graph="yes" report="yes">
  <WSM>
    <MetricName>AvgExecTime</MetricName>
    <ObjectWSDLList>
      <WSDL>http://15.244.63.244:8001/wsmf/services/
      OrderNew_processtype?wsdl
    </WSDL>
    </ObjectWSDLList>
  </WSM>
</Metric>
```

The data for `AvgExecTime` metric is now collected only for the **`http://15.244.63.244:8001/wsmf/ services/OrderNew_processtype?wsdl`** **Business Process**

Restart the WLI SPI for the changes in `MetricDefintions.xml` file to take effect.

Collecting Data for Specific Metrics

The `WLISPI-Collect-Mon` monitor template is used to collect metric data. By default, the template is configured to collect data for all metrics defined in the `/var/opt/OV/conf/wli/MetricDefinitions.xml` file. However, you can modify the template to collect data for a subset of the available metrics. For example, to collect data for Metrics `WLISPI-0001 – WLISPI-0004` and `WLISPI_0017`, follow these steps:

- 1 From the OVO administrator GUI, open the Message Source Templates window by selecting **Window** → **Message Source Templates**.
- 2 Double click **SPI for WLI** in the Template Groups frame on the left.
- 3 Select **WLISPI-Frontend** template group from the left frame.
- 4 Select the `WLISPI-Metrics` template group in the Template Groups frame on the right.
- 5 Select the `WLISPI-Collect-Mon` template in the right frame and click the **Modify** button.
- 6 In the Modify Threshold Monitor dialog, modify the Monitor Program or MIB ID field to contain:
`wli-perl -S wli-collect-data -c WLISPI-Collect-Mon -m 1-4, 17`
- 7 Click the **OK** button in the Modify Threshold Monitor dialog.
- 8 Redeploy the templates by following the instructions in the section [Deploy Frontend Templates to a Managed Node](#) on page 48 in Chapter 2.

Changing the Metric Data Collection Interval

To change the metric data collection interval, edit the Polling Interval in the `WLISPI-Collect-Mon` and `WLISPI-Graph-Mon` templates:

- 1 From the OVO administrator GUI, open the Message Source Templates window by selecting **Window** → **Message Source Templates**.
- 2 Double click **SPI for WLI** in the Template Groups frame on the left.
- 3 Select **WLISPI-Frontend** template group from the left frame.
- 4 Select the `WLISPI-Metrics` template group in the Template Groups frame on the right.

- 5 Select the `WLISPI-Collect-Mon` template in the right frame and click the **Modify** button.
- 6 In the Modify Threshold Monitor dialog, modify the Polling Interval from 5m to 10m. Click the **OK** button.
- 7 Select the `WLISPI-Graph-Mon` template in the right frame and click the **Modify** button.
- 8 In the Modify Threshold Monitor dialog, modify the Polling Interval from 5m to 10m.
- 9 Click the **OK** button.
- 10 Redeploy the templates by following the instructions in the section [Assign Frontend Subagent Templates](#) on page 47 in Chapter 2.

Reporting and Performance Graphs

This section provides instructions for using the OVO embedded performance component (Coda) for generating reports and performance graphs. In addition, instruction for viewing reports and graphs in OV Reporter are also provided.

Using the OVO Embedded Performance Component (Coda)

The OVO embedded performance component (Coda) is a performance subagent that is bundled for free with the OVO agent in OVO for UNIX 7.0 and higher. It is a light-weight performance agent comparable to MeasureWare. The OVO embedded performance component (Coda) holds only 5 weeks worth of data where as MeasureWare can potentially hold years worth of data. The OVO embedded performance component (Coda) Database is stored in the `<OVAgentDataDir>/databases` directory. There is a `coda.db` and `coda##### logs`.

To check if the OVO embedded performance component (Coda) is running, from the command prompt run:

```
opcagt -status -id 12 (OVO-U 7.x)
```

```
ovc -status -id coda (OVO-U 8.x)
```

The output should verify that the Performance Agent `/opt/OV/bin/coda` is running. You can also run the OVO embedded performance component (Coda) utility program. On HP-UX, the command is

```
/opt/OV/bin/codautl -support (OVO-U 7.x)
```

```
/opt/OV/bin/ovcodautl -support (OVO-U 8.x)
```

The result is a list of the last logged interval for all of the standard metrics and their values.

Embedded Performance Component (Coda) Logging

The OVO embedded performance component (Coda) keeps up to 5 weeks of data. Every Sunday at 12:00am (midnight), a new log file is created. The OVO embedded performance component (Coda) will continue to create a new logfile each week until it has accrued 5 weeks of logfiles. When the sixth logfile is

created, the oldest file is deleted. The OVO embedded performance component (Coda) log file (`coda.log`) is located in `<OVAgentDataDir>/log/`. For example on HP-UX the file is `/var/opt/OV/log/coda.log`. To check if the OVO embedded performance component (Coda) is logging data, use the following procedure:

- 1 Open `/var/opt/OV/log/coda.log`
- 2 At the end of the `coda.log` file, you should see the Staring message, which files were opened, deleted and/or created and finally the following message
"Waiting for requests..." message
As the OVO embedded performance component (Coda) logs data, the timestamp for the newest `coda#####` log changes.

View WLI SPI Reports

The section provides steps to use HP OpenView Reporter and gather WLI SPI Metric data and generate WLI SPI reports. The section is intended only as a quick start reference and does not represent a replacement for the Reporter documentation.

To view WLI SPI reports:

- 1 From the Windows Taskbar, select **Start** → **Programs** → **HP OpenView** → **reporter** → **Reporter**.
- 2 From the left tree, right-click **Discover Area** and select the **Add Single System** command. The Add Single System dialog box appears.
- 3 In the System field, enter the full DNS name of the computer where the Frontend Subagent is installed (**hostname.mycompany.com**).
- 4 Click **Add**. The Reporter's discovery program runs, discovers the system, and automatically gathers metric data (collected by either MeasureWare, or the OVO embedded performance component (Coda)) from the system.
- 5 From the Main toolbar, click **Generate Reports**. The WLI SPI reports are generated. This may take several minutes to complete.
- 6 From the Main toolbar, click **Show Reports**. A browser displays and lists all of the WLI SPI reports. The reports are organized into 4 categories that show the metric data over different time ranges: WLISPI Full Range, WLISPI Last Full Month, WLISPI Last Full Week, and WLISPI Yesterday.

Defining a User Friendly Name for an MO

You can define a user friendly name that appears on graphs and reports to identify each MO. The name is also used as the name of the file where the data is logged for graphs before it is sent to MeasureWare or to the OVO embedded performance component (Coda). Therefore, the name needs to be a valid file name.

To define a user friendly name:

- 1 From the OVO administrator GUI, open the Application Bank window by selecting **Window** → **Application Bank**.
- 2 In the Application Bank window, double-click the **WLI SPI Tools** application group.
- 3 In the Application Group: WLI SPI Tools window, right-click **Configure WLI SPI application** and select **Execute** from the popup menu.
- 4 In the BEA WLI Configuration Editor tool, click the icon next to Top of tree.
- 5 Click the icon next to Front End Configuration.
- 6 Click **Report Information**.
- 7 In the menu bar, select **Edit** → **New**.
- 8 The ReportGroupWsdLocation field refers to the MO ID of an MO. For example: If the MO ID for OrderNew Business Process MO is:

```
http://15.244.63.244:8001/wsmf/services/  
OrderNew_processtype?wsdl
```

And you want to give a new name to the OrderNew Business Process MO, then:

```
ReportGroupWsdLocation =http://15.244.63.244:8001/wsmf/  
services/ OrderNew_processtype?wsdl
```

```
ReportGroupName = <any text is valid>
```

- 9 Repeat step 7 and 8 to add entries for each MO
- 10 Select **File** → **Save**.
- 11 Select **File** → **Exit**.
- 12 Click the **Close** button in the Output of Application window.

Monitoring Performance Metrics with OVPM

As mentioned earlier, you can monitor performance using OVPM. For detailed instructions on OVPM, see the OVPM documentation.



The following procedures must be completed in the order listed.

Configure OVPM

On the system where OVPM 4.x is installed, add the Frontend Subagent host name into the `<install_dir>/Data/system<CODA → MWA>.txt` file.

For example, if OVPM 4.x were installed on **<machine1>.<domain name>** in `C:\Program Files\HP OpenView`, then the Frontend Subagent is running on **<machine2>.<domain name>**. If you are using the OVO embedded performance component (Coda), you need to add **<machine2>.<domain name>** into the `C:\Program Files\HP OpenView\Data\systemCODA.txt` file on **<machine1>.<domain name>**.

For OVPM 5.0, modify the file `OVPMconfig.ini` in the directory `<OV-INSTALL_DIR>\data\conf\perf` as mentioned in the section 'OpenView Reporter Integrations' in the *HP OpenView Performance Manager Administrator's Guide*.

Create an Average Execution Time OVPM Graph for Business Processes

To create an OVPM graph (Average Execution Time) for a Business Process:

- 1 Start the Performance Manager GUI by Select **Start → Programs → HP OpenView → performance manager → Performance Manager**.
- 2 In the Performance Manager home page, click **Web Forms Interface**.
- 3 Click the **Design** tab and enter a title.
- 4 Select the **Frontend Subagent** node as the Data Source.
- 5 Select a **Date Range**. You can use 1 Hour if you're testing recently logged data.
- 6 Select **Ending Now** if you're testing recently logged data.

- 7 Select **WLISPI_METRICS:WLISPI_METRICS** as the class.
- 8 Select **B001_AVRGEXECTIME** as the metric and enter a label.
- 9 Scroll up and select **SERVERNAME** as the Metric Filter.
- 10 Select **=** and Fill in the Business Process name you want in the graph. To figure out the Business Process name, you can look in the `/var/opt/OV/conf/wli/SiteConfig` file on the Frontend Subagent node.
- 11 Scroll down and enter **WLI** as the User Template File.
- 12 Enter a name for the Save Graph Name.
- 13 Click the **Save Graph** button.
- 14 Scroll up and click the **Draw Graph** button.

View the Business Process Average Execution Time OVPM Graph

To view the Business Process Average Execution Time OVPM graph:

- 1 In the Performance Manager home page, click the **Web Forms Interface** button.
- 2 Select the **Display** tab if it is not already selected.
- 3 Select **UserWLI** for the Template File.
- 4 Select the **Business Process Average Execution Time** graph for the Graph Name.
- 5 Select the **Frontend Subagent** node in the System Names list box.
- 6 Select the desired date range information.
- 7 Click the **Draw Graph** button.

Modify Logging and Tracing Levels

This section provides instructions for changing log and trace levels for the WLI SPI. Logging and tracing levels can be changed for both the Frontend Subagent and the Backend Service and are useful for debugging and auditing purposes.

Changing Log Levels

The WLI SPI supports two log levels (`ERROR` and `INFO`) for both the Frontend Subagent and Backend Service log messages. By default the log level is set to `INFO` for both the Frontend and the Backend. Log levels can be customized by modifying the log property files. The property files are

- `/var/opt/OV/log/wli/frontend.properties`
- `/var/opt/OV/log/wli/backend.properties`

Changing Frontend Log Levels

To change the Frontend's log level from `INFO` to `ERROR`:

- 1 Open the `/var/opt/OV/conf/wli/Frontend.properties` file.
- 2 Set the `java.util.logging.FileHandler.level` to `ERROR`. For example:

```
java.util.logging.FileHandler.pattern = %h/frontend.log  
java.util.logging.FileHandler.level = ERROR
```

- 3 Save and close the file.
- 4 Restart the WLI SPI. The Frontend log file is `/var/opt/OV/log/wli/frontend.log`

Changing Backend Log Levels

To change the Backend's log level from `ERROR` to `INFO`:

- 1 Open the `/var/opt/OV/conf/wli/Backend.properties` file.
- 2 Set the `java.util.logging.FileHandler.level` to `ERROR`. For example:

```
java.util.logging.FileHandler.pattern = %h/backend.log
java.util.logging.FileHandler.level = ERROR
```

- 3 Save and close the file.
- 4 Restart the WLI SPI. The Backend log file is `/var/opt/OV/log/wli/backend.log`

Changing Trace Levels

The WLI SPI provides a mechanism that collects and stores all trace data in a trace file. By default, tracing is configured to show `INFO` messages. To get a more detail level of trace messages the trace level has to be set to `FINE`. Trace levels can be customized by modifying the trace property files. The property files are:

- `/var/opt/OV/log/wli/frontend.trace`
- `/var/opt/OV/log/wli/backend.trace`

Changing Frontend Trace Levels

To change Frontend's trace levels from `INFO` to `FINE`:

- 1 Open the `/var/opt/OV/conf/wli/Frontend.properties` file.
- 2 For the Frontend trace file, set the `java.util.logging.FileHandler.level` value to `FINE`. For example:

```
java.util.logging.FileHandler.pattern = %h/frontend.trace
java.util.logging.FileHandler.level = FINE
```

- 3 Save and close the file.
- 4 Restart the WLI SPI. The Frontend trace file is `/var/opt/OV/log/wli/frontend.trace`.

Changing Backend Trace Levels

To change the Backend's trace levels from `INFO` to `FINE`:

- 1 Open the `/var/opt/OV/conf/wli/Backend.properties` file.

- 2 For the Backend trace file, set the `java.util.logging.FileHandler.level` to `FINE`. For example:

```
java.util.logging.FileHandler.pattern = %h/backend.trace  
java.util.logging.FileHandler.level = FINE
```
- 3 Save and close the file.
- 4 Restart the WLI SPI. The Backend trace file is `/var/opt/OV/log/wli/backend.trace`.

4 Implementing Application-Specific Managed Objects

Additional management capabilities, specific to your WLI application, can be exposed to the OpenView management console. This is accomplished by writing one or more Managed Object (MO) classes, as well as a discovery class that is responsible for deploying these MOs. We provide an example that demonstrates how to expose an application-specific MBean that is registered in the WLI MBeanServer. You can find the source code for the example under the `examples/src` subdirectory of the WLI SPI SBA installation directory. The following sections describe the basic classes you need to understand, as well as some configuration issues.

MO Implementation Class

An example of an MO implementation class is

```
com.hp.wsm.impact.sba.integration.wli.example.mo.impl.ExampleAppMO.
```

It extends the `BaseManagedObject` base class. This class provides a number of basic MO methods such as `getName()`, `getCreatedOn()`, and so on. It also provides the infrastructure for emitting management notifications, as well as for describing relationships to other MOs. Your MOs should always extend the `BaseManagedObject` base class. There are several abstract methods in `BaseManagedObject` which you need to implement:

Listing 4-1: Get Operations Required in Implementation Class

```
/**
 * Returns a URI that uniquely represents the type of this
MO.
 */
public AnyURI getType( );
```

```

/**
 * Returns a description of the MO.
 */
public String getDescription( );

/**
 * Returns the name and/or email address of the owner of
this MO.
 */
public String getOwner( );

/**
 * Returns the vendor that owns this MO.
 */
String getVendor( );

/**
 * Returns the version of the resource being exposed by
this MO.
 */
String getResourceVersion( );

/**
 * Returns the version of this MO.
 */
String getManagedObjectVersion( );

/**
 * Returns the hostname of the machine hosting the
 * resource being exposed by this MO.
 */
String getResourceHostName( );

```

Sample implementations of the above `get*` methods are provided in the `ExampleAppMo` in the `examples` directory as mentioned above.

Model MBean Interface Class

The Model MBean interface class is

```
com.hp.wsm.impact.sba.integration.wli.example.mo.ExampleAppMO  
ModelMBean.
```

This interface class defines the MO methods that are to be exposed as the management operations, which are accessible from the OpenView console. You must write such an interface for each of your implementation MOs. The class name of the interface must adhere to the naming convention where the MO implementation class name is followed by the `ModelMBean` suffix. In our example, the name of the interface is `ExampleAppMOModelMBean`.

MBean Proxy Class

The MBean proxy class is

```
com.hp.wsm.impact.sba.integration.wli.example.accessor.AppAcc  
essor.
```

This class demonstrates how to use the `MBeanProxy` class that is included with the SBA to obtain a strongly-typed proxy to an application-specific MBean that is registered in the `WLI MBeanServer`. It provides a `getMBean()` method that is called by the MO to obtain the proxy. The MO can then easily invoke operations on the application-specific MBean.

Discovery Class

The discovery class is

```
com.hp.wsm.impact.sba.integration.wli.example.ExampleDiscover  
yStrategy.
```

Discovery classes are responsible for deploying one or more MOs as management Web services, and defining any desired relationships between these MOs. Your discovery class must always extend the

DiscoveryStrategy base class. You need to implement only the discover() method. The comments in the ExampleDiscoveryStrategy source file describe the steps performed by this method.

SBA Configuration File

To deploy your discovery strategy to the SBA, you must add some entries to the wli-sba-config.xml SBA configuration file. These include the <MBeanServer>, <DiscoveryObject>, and <StartupClass> element entries.

The <MBeanServer> element names the MBean server instance. This name is used in the <DiscoveryObject> and <StartupClass> elements to indicate the name of the MBean server to which they have access. The <DiscoveryObject> element specifies the discovery class (ExampleDiscoveryStrategy) that will deploy your MOs as Web services. The <StartupClass> element specifies the class (ExampleStartup) that you want to run before the MOs are discovered by your discovery object so that the startup class can perform any necessary setup prior to discovery.

The entries are the following:

Listing 4-2: Entries for Discovery Object in SBA Configuration File

```
<MBeanServer name="myapp">
  <Standard>
    <DefaultDomain>myapp</DefaultDomain>
  </Standard>
</MBeanServer>

      <DiscoveryObject name="wli-example-discovery">
        <Strategy>
          com.hp.wsm.impact.sba.integration.wli
            .example.ExampleDiscoveryStrategy
        </Strategy>
      <MBeanServerNames>
        <MBeanServerName>myapp</MBeanServerName>
        <MBeanServerName isSink="true">sba</MBeanServerName>
      </MBeanServerNames>
      <WaitForDiscoveryObjects>
        <DiscoveryObjectName>
```



```

        wli-discovery
    </DiscoveryObjectName>
    </WaitForDiscoveryObjects>
</DiscoveryObject>

<StartupClass>
    <Strategy>
        com.hp.wsm.impact.sba.integration.wli
        .example.ExampleStartup
    </Strategy>
    <MBeanServerNames>
        <MBeanServerName>myapp</MBeanServerName>
    </MBeanServerNames>
</StartupClass>

```

You have to specify the value of the `<Strategy>` element as the fully qualified name of your discovery class.

Also, note how the `<WaitForDiscoveryObjects>` element is configured. This ensures that the core WLI discovery strategy will run before the application-specific discovery one does. This is necessary so that the application-specific discovery strategy can get a handle on the `WLI Server MO` in order to define a containment relationship to it from each application-specific MO. The containment relationships ensure that the application-specific MOs will get discovered by the WLI SPI and included in the WLI MO tree displayed in Service Navigator.

Complex Types

Try to limit the parameter and return types of your MO to the fundamental Java types (or to arrays of such types) that have a direct analog to the XML Schema simple types. These would include `Integer`, `String`, `Date`, etc., which directly map to the XML Schema simple types of `xsd:integer`, `xsd:string`, `xsd:dateTime`, etc.

If you absolutely must use a complex type, it will be necessary for you to write serialization/deserialization code to convert it to and from the XML type. You will also need to configure a mapping for the type in the WSMF Web application's deployment descriptor `web-services.xml` file. For details on how to do this, refer to the WebLogic Server documentation.

5 Troubleshooting

This chapter provides instructions for common troubleshooting tasks when using the components of the WLI SPI. In addition, refer to the WLI SPI Release Notes for the latest information about the WLI SPI.

WLI SPI Backend Service

Backend Service is Not Found

After registering the Backend Service, the `ovstatus` and `ovstart` commands indicate that the `wli-backend` cannot be found.

Solution:

Stop and restart the OVO server:

```
ovstop
ovstart
```

Performance Agent Does Not Start Up

When you run `opcagt -start -id 12` and `opcagt -status` the output shows that the performance agent is not running.

Solution:

To verify if CODA really is not running by running the command:

```
ps -ef | grep coda
```

If CODA is running it means CODA was either started up standalone, or the OVO control agent died, was restarted and is longer the parent.

If CODA is not running, look in the `<OVAgentDataDir>/log/coda.log` file for any error messages.

WLI SPI Frontend Subagent

Frontend Subagent Does Not Stop

The Frontend subagent does not stop if there is a problem canceling the subscription.

Solution:

You must manually stop the Frontend Subagent Java processes.

Frontend Subagent Does Not Start

The Frontend Subagent is not starting up.

Solution:

To locate the error, look in the `/var/opt/OV/log/wli/frontend.log` file. Refer to the section [Errors in Frontend Logfile](#) on page 98 for instructions on resolving the error.

WLI Service is Not Visible

The WLI Service does not appear in the OVO Java Console.

Solution:

Verify that the user name used to log into the HP OVO Java Console is the same as the one that is configured as `<OVUserNAme>` in the `WCConfig.xml` file on the Frontend Subagent node.

Missing Operational Notification

Operational notifications are not appearing in the OVO message browser.

Solution:

Complete the following procedures:

Deploy the Templates

Deploy the templates for the Frontend node. You need to do this only if the WLISPI templates are not already configured.

- 1 In the Node Bank window, select the node where the Frontend Subagent is running.
- 2 Select **Actions** → **Agents** → **Install / Update SW & Config**.
- 3 In the Install / Update OVO Software and Configuration dialog box, select **Templates** in the Components frame.
- 4 Select **Force Update** in the Options frame.
- 5 Click the **OK** button.

Verify OVO Messages Display in Message Browser

Verify that an OVO test message appears in the OVO message browser.

- 1 Launch the OVO administrator GUI on the OVO management server and log on as **opc_adm** or **WLISPI_Op**.
- 2 From a window on the Frontend Subagent node, send a test message by running

```
opcmsg severity=minor application=test object=test
msg_text="my test" msg_grp=WLI
```
- 3 A message with minor severity should appear in the OVO message browser. If a message does not appear, follow the steps in the section [Verify Credentials](#) on page 95.

Verify Resource Host Name +

The node must be in the OVO Node Bank and in the WLISPI-External or WLISPI-UNIX node group.



The node value must match exactly with the Hostname for the node in the OVO Node Bank. The node cannot be an IP address and it must be accessible through DNS. Do not add the host name in the `/etc/hosts` file. The name should be in the DNS tables.

- 1 Launch the OVO administrator GUI on the OVO management server and log on as **opc_adm**.
- 2 From any OVO window, open the Node Bank window by selecting **Window** → **Node Bank**.
The node must be a node in the Node Bank or a node in the Holding Area.
- 3 Close the Node Bank window.
- 4 From any OVO window, open the Node Group Bank window by selecting **Window** → **Node Group Bank**.
- 5 In the Node Group Bank window, double-click the **WLISPI-External** node group.
- 6 If the node is in the WLISPI-External node group window, close the window and skip the following steps.
- 7 If the node is not in the WLISPI-External node group window, click the arrow pointing up to get back to the Node Group Bank window.
- 8 In the Node Group Bank window, double-click the **WLISPI-UNIX** node group.
- 9 If the node is not in the WLISPI-UNIX node group, add it to either the WLISPI-UNIX or WLISPI-External node group.

Verify Source Object

The source object that is assigned as the `service_id` value in the Frontend output must be one of the objects that the Frontend Subagent recognizes.

- 1 Start the HP VP Java Console.
- 2 Click the **+** icon in front of the services to expand the list until you get to the service that represents the object that is the `service_id` value
- 3 Right-click the service and select **Properties** from the popup menu.
- 4 In the Services Properties [xxx] dialog box, the value in the Name field must exactly match the `service_id` value.

- 5 Click the Close button to close the dialog box.

Verify Credentials

The OVO operator that you are logged on as must be able to see messages in the WLI message group. We assume for this example that the OVO operator is **opc_adm**.

- 1 Launch the OVO administrator GUI on the OVO management server and log on as **opc_adm**.
- 2 From any OVO window, select **Window** → **User Bank**.
- 3 From the User Bank window, right-click **opc_adm** and select **Modify** from the popup menu.
- 4 In the Modify User: **opc_adm** window, click the **Profiles** button.
- 5 In the Profiles of User: **opc_adm** window, verify that the WLISPI User Profile icon appears. If it does not appear:
 - a Open User Profile Bank window by selecting **Window** → **User Profile Bank**.
 - b Drag the WLISPI User Profile from the OVO User Profile Bank window and drop it into the Profiles of User: **opc_adm** window.
 - c Close the OVO User Profile Bank window.
- 6 Close the Profiles of user: **opc_adm** window.
- 7 In the Modify User: **opc_adm** window, click the **OK** button.
- 8 Restart the session, from any OVO window, select **Map** → **Restart Session**.
- 9 In the HP OpenView Windows WARNING dialog, click the **OK** button.

Verify Communication with the OVO Management Server

If the Frontend is receiving the notification, you must check to see if there are communication problems between the Frontend node and the OVO management server. For this, you must turn on the OVO message tracing.

- 1 On the OVO management server, add `OPC_TRACE TRUE` and `OPC_TRACE_AREA MSG` to the `/opt/OV/bin/OpC/install/opcsvinfo` file. Your file must look similar to:

```
OPC_INSTALLED_VERSION A.07.10
```

```
OPC_MGMT_SERVER <machine>.<domain name>
OPC_MGMTSV_CHARSET iso885915
OPC_INSTALLATION_TIME 03/25/03 15:28:11
OPC_SG FALSE
OPC_TRACE TRUE
OPC_TRACE_AREA MSG
```

- 2 **Run** `/opt/OV/bin/OpC/opcsv -trace`. The trace information is written to `/var/opt/OV/share/tmp/OpC/mgmt_sv/trace`.
- 3 **Run** `tail -f /var/opt/OV/share/tmp/OpC/mgmt_sv/trace` to see the trace messages as they are written.
- 4 **On the Frontend node**, add `OPC_TRACE TRUE` and `OPC_TRACE_AREA MSG` to the `/opt/OV/bin/OpC/install/opcinfo` file.
- 5 **Run** `/opt/OV/bin/OpC/opcagt -trace`. The trace information is written to `/var/opt/OV/tmp/OpC/trace`.
- 6 **Run** `tail -f /var/opt/OV/tmp/OpC/trace` to see the trace messages as they are written.
- 7 **Send a notification**. In the OVO agent trace file (`/var/opt/OV/tmp/OpC/trace`), the following messages appear:

```
02/11 12:18:20.517 opcmgs(9930:001) [MSG]: Queueing message:
WLI 'Status Change from h'
```

```
02/11 12:18:20.523 opcmgsi(1791:001) [MSG]: Sending message:
6f732be6-5ccf-71d8-0be9-0ff414390000 WLI 'Status Change from
h' 15.244.60.103
```

```
02/11 12:18:20.534 opcmgsa(1778:001) [MSG]: Message/Act.Resp.
received from agents: 6f732be6-5ccf-71d8-0be9-0ff414390000
WLI 'Status Change from h' 15.244.60.103
```

```
02/11 12:18:20.535 opcmgsa(1778:001) [MSG]: OpC mgr for msg:
6f732be6-5ccf-71d8-0be9-0ff414390000 WLI 'Status Change from
h' 15.244.60.103 opcmgr : <machine>.<domain name>
15.244.20.57
```

```
02/11 12:18:20.535 opcmgsa(1778:001) [MSG]: forwarding msg:
6f732be6-5ccf-71d8-0be9-0ff414390000 WLI 'Status Change from
h' 15.244.60.103 opcmgr : <machine>.<domain name>
15.244.20.57
```



```
02/11 12:18:20.538 opcmsga(1778:001) [MSG]: Sending msg (len = 554): Status Change from http://schemas.hp.com/wsmf/2003/03/Founda
```

```
02/11 12:18:20.543 opcmsga(1778:001) [MSG]: Message forwarded: 6f732be6-5ccf-71d8-0be9-0ff414390000 WLI 'Status Change from h' 15.244.60.103 opcmgr : <machine>.<domain name> 15.244.20.57
```

In the OVO management server trace file (/var/opt/OV/share/tmp/OpC/mgmt_sv/trace), the following messages appear:

```
02/11 12:18:20.542 opcmsgrd(1595:02f) [MSG]: Message received: 6f732be6-5ccf-71d8-0be9-0ff414390000 WLI 'Status Change from h' 15.244.60.103
```

```
02/11 12:18:20.543 opcmsgm(1611:001) [MSG]: Message received from message receiver: 6f732be6-5ccf-71d8-0be9-0ff414390000 WLI 'Status Change from h' 15.244.60.103
```

```
02/11 12:18:20.552 opcmsgm(1611:001) [MSG]: Message processing: ip_addr=15.244.60.103 (mapped), node_name='<machine>.<domain name>'
```

```
02/11 12:18:20.552 opcmsgm(1611:001) [MSG]: csm_db_msg_add called with msg 6f732be6-5ccf-71d8-0be9-0ff414390000.
```

```
02/11 12:18:20.592 opcmsgm(1611:001) [MSG]: csm_db_msg_add finished for msg 6f732be6-5ccf-71d8-0be9-0ff414390000. Last err: 0-0
```

```
02/11 12:18:20.593 opcmsgm(1611:001) [MSG]: Message forwarded to DM: 6f732be6-5ccf-71d8-0be9-0ff414390000 WLI 'Status Change from h' 15.244.60.103
```

- 8 Turn tracing off on the OVO management server by modifying OPC_TRACE to FALSE in the /opt/OV/bin/OpC/install/opcsvinfo file.**
- 9 Run /opt/OV/bin/OpC/opcsv -trace.**
- 10 On the Frontend node, turn tracing off by setting OPC_TRACE to FALSE in the /opt/OV/bin/OpC/install/opcinfo file.**
- 11 Run /opt/OV/bin/OpC/opcagt -trace.**

Cleanup the OVO Message Queues

If you do not see the above trace messages, then your OVO message queues maybe corrupt.

- 1 Stop `opcagt` by running `opcagt -kill`.
- 2 Remove the temporary files by running `rm -f /var/opt/OV/tmp/OpC/*`.
- 3 Restart the `opcagt` by running `opcagt -start`.
- 4 Close all OVO GUIs.
- 5 Stop the OVO management server by running `ovstop opc ovoacomm`.
- 6 Remove the temporary files by running

```
rm -f /var/opt/OV/share/tmp/OpC/mgmt_sv/*
```

Restart the OVO management server processes by running:
`opcsv -start`.

Errors in Frontend Logfile

The Frontend subagent's log file is located in `/var/opt/OV/log/wli/frontend.log`.

Error Starting Notification HTTP Server

`NotifMgr: Error: Problem starting notification HTTP server. Please make sure port is not already in use: XXXX.`

Solution:

Verify if the Frontend is already running:

```
opcagt -status -id 26 (OVO 7.x)
ovc -status WLI (OVO 8.x)
```

If the response comes back with “WLI SPI Subagent (Frontend) ... is running”, that means the Frontend is already running.

If the system response is “... isn't running”, `opcagt` has got an indication that the Frontend is not up, but the Frontend Java process may not have been stopped yet. Wait for few minutes for the port to be released and try to start WLI SPI again.

Error Adding Service to Map

Problem adding services to map: Connection refused to host: XXX;
nested exception is: java.net.ConnectException: Connection
refused

Solution:

Verify if the Backend process is running. On the OVO management server
system, run the command:

```
ovstatus wli-backend.
```

If outcome is “NOT_RUNNING”, start the Backend by running:

```
ovstart wli-backend.
```

Frontend does not Connect to the Backend

Frontend is not able to connect to the Backend.

Solution:

This message may appear just after installing the Frontend Subagent because
the WLI SPI has not been configured yet or the Backend Service is not
started.

Make sure that you configure the SPI using the WLI SPI Configuration Editor
located in the OVO Application Bank. See [Chapter 2, Install and Configure
the WLI SPI](#) for complete instructions.

After the configuration is complete, use the Start WLI SPI application located
in the OVO Application Bank.

Problem Logging Metric Data

```
WSF-0009: ddflog returned an error logging  
WLISPI_RPT_METRICS:256
```

This error may occur when the data sources for the WLI SPI metrics are not
registered successfully. Make sure you register the WLI SPI Metric data
sources by running the following commands on the Frontend Subagent
machine:

```
cd /opt/OV/bin  
./wli-perl wli-create-datasources
```

WLI SPI SBA

Cannot View WSMF Services

You can not view the WSMF services for the WLI application server.

Solution:

There are several things to consider if you cannot view the WSMF services for the WLI application server. They include the following:

- Did you start things in the correct order? As indicated in the [Chapter 2, Install and Configure the WLI SPI](#), the WLS hosting the WLI must be started before running the WLS hosting the SBA.
- Did you wait for the WLS hosting the WLI to be fully started before running the SBA? As noted in [Chapter 2, Install and Configure the WLI SPI](#), the WLI server is fully initiated when you see the following message in the console window:

```
<Thread "ListenThread.Default" listening on port 7001, ip address *.*>
```

- Did you wait for the WLS hosting the SBA to be fully started before browsing the URL to view the deployed WSMF services? As noted in [Chapter 2, Install and Configure the WLI SPI](#), the SBA is fully initiated when you see the following message in the console window:

```
<BEA-000360> <Server started in RUNNING mode>  
[java] <Apr 14, 2004 4:04:22 PM PDT> <Notice>  
<WebLogicServer> <BEA-000355> <Thread "ListenThread.Default"  
listening on port 8001, ip address *.*>
```

- Did you specify the correct URL for the deployed WSMF services? The URL is **http://sba_host:sba_port/wsmf/services** where *sba_host* is the host name specified as the value of the `sba.wls.host` property in the `wli.properties` file located in the SBA home directory, and *sba_port* is the port number specified as the value of the `sba.wls.port` property in the `wli.properties` file.

Enabling Debugging

To enable debugging, you must modify the `log4j.properties` file located in the `config/weblogic` directory under the home directory of the SBA.

Change:

```
log4j.category.com.hp.wsm=WARN
```

to the following:

```
log4j.category.com.hp.wsm=DEBUG
```

Save the file, restart the SBA.

Enabling Printing of Stack Traces

To enable printing of stack traces for all major exceptions, you must modify the `wli.properties` file located in the home directory of the SBA.

Change:

```
wsm.stackdumps.enabled=false
```

to the following:

```
wsm.stackdumps.enabled=true
```

Save the file, restart the SBA.

Get Out of Memory Error Message When Running Ant Script

When running the `wli.xml` Ant script, the following error message may appear:

```
/opt/wli_spi_sba-0.9.31.ER17/wli.xml:250: Following error  
occurred while executing this line java.lang.OutOfMemoryError
```

Solution:

Increase the maximum heap size for Ant by setting the `ANT_MAXMEM` environment variable to 256M or higher prior to running the `wli.xml` script. To do this, run the following command from a Bourne-flavor shell:

```
$ ANT_MAX_HEAP=256M export ANT_MAX_HEAP
```

WLI SPI Configuration Editor

WLI SPI Configuration Editor Does Not Start

The WLI SPI Configuration Editor application does not start.

Solution:

Delete `/var/opt/OV/conf/wli/appconfig.xml.ser` and restart the configuration tool.

WLI SPI Configuration Editor Does Not Display Content

The WLI SPI Configuration Editor starts, but no content displays.

Solution:

This problem occurs when you manually edit `WCConfig.xml` with a text editor and save it with non UTF-8 encoding. This file is expected to be in UTF-8.

Open the file in your text editor and ensure that you save it in UTF-8 encoding. If your editor does not have a **save as** option to do this and uses system default encoding, then make sure you set your session's `LANG` variable to UTF-8.

WLI SPI Configuration Editor Fails to Transfer WCConfig.xml

When you run the WLI SPI Configuration Editor, the Output of Application window has the message `Error by transfer file from /var/opt/OV/conf/wli//WCConfig.xml to /var/opt/OV/conf/wli/WCConfig.xml (OpC40-745)`.

Solution:

This error message appears if there is a problem of transferring the `WCCconfig.xml` file from the OVO management server to the Frontend Subagent node. You need to manually transfer the file and restart the Backend service and Frontend Subagent.

- 1 **Ftp the `/var/opt/OV/conf/wli/WCCconfig.xml` file from the OVO management server to `/var/opt/OV/conf/wli` on the Frontend Subagent node.**

- 2 **Restart the Backend Service by running the command:**

```
ovstop wli-backend
ovstart wli-backend
```

- 3 **Restart the Frontend Subagent.**

On OVO 7.x machine:

```
opcagt -stop -id 26
opcagt -start -id 26
```

On OVO 8.x machine:

```
ovc -stop WLI
ovc -start WLI
```

WLI SPI Configuration Editor Throws `AWTException`

When you close the WLI SPI Configuration Editor, you get the following exception in the Application Output window:

```
java.awt.AWTException: cannot open XIM
```

Solution:

By default, you cannot input Asian (Japanese, Chinese, Korean) characters into the WLI SPI Configuration Editor. To input Asian characters in a Java GUI an input method is required. This input method can be a pure Java input method (independent of OS) or an input method provided by the OS. An input method is installed with Java 1.4. The WLI SPI Configuration Editor is a Java GUI. Therefore, you need to set the `JAVA_HOME` environment variable on the OVO server system to a JVM on the system that's configured to display Asian characters.

Using the Self-Healing Info Application

The Self-Healing Info application gathers system information as well as configuration, log and trace files of WLI SPI when a problem occurs in the WLI SPI.

All the gathered files and information are placed in a pre-defined output directory, thereby facilitating faster troubleshooting. Also, the data collector application is used to gather real-time data, which reduces the probability of troubleshooting with stale data.

Troubleshooting information is enhanced with cross-referencing capabilities available through the HP support site: http://support.openview.hp.com/self_healing.jsp. A link is then automatically set up to the self-healing services web site, where your SPI problem can be matched to similar SPI problems and solutions for increased troubleshooting effectiveness.

To greatly enhance troubleshooting, and access the search and cross-referencing capabilities of the HP Support web site, you can download and use the Self-Healing Services client software. Refer to the relevant section in the *HP OpenView Operations for UNIX SPI CD Installation Guide* for more information on how to download and use the software, and to set up the automatic link to the HP support web site.

If Self-Healing Services client is not installed and configured on the node being managed by the SPI, you can use the Self-Healing Info application to collect system information.

Whenever you encounter a problem with the WLI SPI, run the data collector by launching the Self-Healing Info application in the WLI SPI Supportability group.



Before using the Self-Healing Info application, turn on tracing and reproduce the problem, then run the application.

To launch the data collector on the node from where you want to gather data, drag the icon of the node and drop it on the Self-Healing Info application in the WLISPI Application Group window.



In case both the Frontend and Backend are running on two different machines, drag the icons of both the nodes and drop them on the Self-Healing Info application in the WLI SPI Application Group window.

The gathered output is placed as `/tmp/SPI_BEAWI_support.tar` on UNIX nodes. You can submit this file to HP Support for assistance or use this file to identify and correct the problem you encountered.



Depending on the Windows setting, the file may be a hidden file on some managed nodes. If you do not see the file, open Windows Explorer, from the Tools menu select the **View** tabbed page in the Folder Options. Under Hidden Files and Folders, select **Show Files and Folders**.

A WLI SPI Configuration

This appendix is a reference for the WLI SPI configuration parameters. In particular, this appendix focuses on the `WCConfig.xml` configuration file. The parameters are discussed throughout this guide; however, they are listed here in a reference style so they can be easily accessed.

Editing WCConfig.xml

The `WCConfig.xml` file is located in `/var/opt/OV/conf/wli/` on the system running the WLI SPI Front-End Subagent. The `WCConfig.xml` file is structured using XML. You can edit this file using the WLI SPI Configuration Editor tool, or you can manually edit this file using an XML or text editor. If you chose to manually edit `WCConfig.xml`, you must restart the Front-End Subagent before the changes take effect.

Using the WLI SPI Configuration Editor Tool

To edit parameters using the configuration tool:

- 1 Launch the OVO administrator GUI on the OVO management server and log on as **opc_adm**.
- 2 From any OVO window, bring up the Application Bank window by selecting **Window** → **Application Bank**.
- 3 In the Application Bank window, double-click the **WLI SPI Tools** application group.
- 4 In the Application Group: WLI SPI Tools window, right-click **Configure WLI SPI** application and select **Execute** from the popup menu.
- 5 Edit the available parameters.

- 6 Select **File** → **Save** to save your changes to the configuration file.

WCConfig.xml Configuration Parameters

The `WCConfig.xml` configuration file contains parameters that configure the Backend Service, Frontend Subagent, and the NNM integration. The root element of the configuration file is `<WcConfiguration>`. All elements must be children within the root element.

<FrontendSection>

This node contains parameters that are used to configure the Frontend Subagent. It contains six child nodes:

- `<ManagedObjectInfo>`
- `<OVOMgmtAgentInfo>`
- `<NotifHttpServerInfo>`
- `<SSLConfiguration>`
- `<ServiceMapInfo>`
- `<Miscellaneous>`
- `<ReportInfo>`

FrontendSection/ManagedObjectInfo

This element contains the WSDL location of the MO that you want the Frontend Subagent to monitor. You only need to add the 'root' MO's WSDL location because the Frontend subagent recursively discovers the children. It retrieves information for the children that are related to the parent MO by the relation configured in the Relations to Use list on the Service Map Information area. By default, all children that are related to the parent MO by **`http://schemas.hp.com/wsmf/2003/02/Relations/Contains`** and **`http://schemas.hp.com/wsmf/2003/03/Relations/DependsOn`** types are discovered. You can add more entries to the Relations to Use list but you need to be careful about circular dependencies.

The MOs configured in the Managed Object Information list must be accessible before starting up the Frontend subagent.

**FrontendSection/ManagedObjectInfo/WsdlLocation/
LocationCandidate**

MO's WSDL location.

FrontendSection/OVOMgmtAgentInfo

Parameters for the Frontend Subagent

FrontendSection/OVOMgmtAgentInfo/HostName

The Frontend subagent system's host name.

FrontendSection/OVOMgmtAgentInfo/JavaHome

The location of JAVA_HOME on Frontend subagent system.

FrontendSection/OVOMgmtAgentInfo/RmiPort

The Frontend subagent RMI port number.

FrontendSection/OVOMgmtAgentInfo/AgentType

The Frontend subagent system type. Choices are: UNIX or Windows. (Windows is not implemented yet.)

FrontendSection/NotifHttpServerInfo

Configures the HTTP server in the Frontend Subagent to receive WSMF notifications

FrontendSection/NotifHttpServerInfo/HTTPPort

The HTTP server's port number that receives WSMF notifications

FrontendSection/Service MapInfo

This node contains parameters that allow you to configure the MO Viewer Service Map

FrontendSection/Service MapInfo/PortTypesToBeIgnored

The MO Viewer does not display information about port types that are listed in this node.

FrontendSection/Service MapInfo/PortTypesToBeIgnored/PortType

This node contains parameters for specifying which port types will be ignored.

**FrontendSection/Service MapInfo/PortTypesToBeIgnored/PortType/
PortTypeName**

The name of the port type to be ignored

**FrontendSection/Service MapInfo/PortTypesToBeIgnored/PortType/
NamespaceURI**

Namespace of port type to be ignored

FrontendSection/Service MapInfo/ActionsToBeIgnored

The MO Viewer will ignore actions that are listed in this node.

FrontendSection/Service MapInfo/ActionsToBeIgnored/Action

This node contains parameters for specifying which actions will be ignored.

**FrontendSection/Service MapInfo/ActionsToBeIgnored/Action/
ActionName**

The name of the action to be ignored

**FrontendSection/Service MapInfo/ActionsToBeIgnored/Action/
NamespaceURI**

Namespace of the action to be ignored

FrontendSection/Service MapInfo/RelationsToBeUsed

List of relations that are used by the Frontend subagent to recursively discover children MOs.

FrontendSection/Service MapInfo/RelationsToBeUsed/Relation

Relation to use in discovery

FrontendSection/Service MapInfo/OVOUserForServiceMap

List of OVO users that can view the WLI service.

**FrontendSection/Service MapInfo/OVOUserForServiceMap/
OVOUserName**

OVO user name to associate with WLI service

FrontendSection/Service MapInfo/ServiceAliasMap

List of service names associated with WLISPI that can be substituted with another service name. For example if there is a DB service node as part of DB SPI, the WLISPI node connected to this DB can use the same service name as the DB service node.

FrontendSection/Service MapInfo/ServiceAliasMap/AliasInfo

**FrontendSection/Service MapInfo/ServiceAliasMap/AliasInfo/
ServiceName**

Service Name of the WLISPI

**FrontendSection/Service MapInfo/ServiceAliasMap/AliasInfo/
AliasName**

The service name to be replaced with this name.

FrontendSection/Service MapInfo/ServiceIconMap

This node contains parameters for defining icons for MO types.

FrontendSection/Service MapInfo/ServiceIconMap/ServiceIcon

List of icon/MO type. Displays the specified icon in the service map for an MO of the specified type

**FrontendSection/Service MapInfo/ServiceIconMap/ServiceIcon/
ServiceType**

MO type

**FrontendSection/Service MapInfo/ServiceIconMap/ServiceIcon/
IconFile**

Fully qualified name of the icon file

FrontendSection/miscellaneous/

This node is for miscellaneous parameters.

FrontendSection/miscellaneous/ManagedObjectsToBeIgnored/

This node allows you to list the MO types that you want to ignore. These MO types are not discovered and therefore will not appear in the service map.

**FrontendSection/miscellaneous/ManagedObjectsToBeIgnored/
ManagedObjectType**

MO type to ignore

FrontendSection/miscellaneous/RootServiceName

The root of the MO hierarchy

FrontendSection/ReportInfo

This node allows you to specify a user friendly name for MOs.

FrontendSection/ReportInfo/ReportGroupInfo

For each MO that you are collecting data for, you can define a user friendly name that appears on the graphs and reports to identify the MO. The name is also used as the name of the file where data is logged for graphs before it is sent to the performance agent. Therefore the name needs to be a valid file name.

FrontendSection/ReportInfo/ReportGroupInfo/ReportGroupObjectID

MO's WSDL location that data is being collected for

FrontendSection/ReportInfo/ReportGroupInfo/ReportGroupName

User friendly name of MO

<BackendSection>

This node contains parameters that are used to configure the Backend Service. It contains a single child node, <OVOMgmtServerInfo>.

BackendSection/OVOMgmtServerInfo

This node contains parameters that configure the Backend Service on the OVO management server.

BackendSection/OVOMgmtServerInfo/HostName

Backend Service system's host name

BackendSection/OVOMgmtServerInfo/JavaHome

JAVA_HOME on the Backend Service system

BackendSection/OVOMgmtServerInfo/RmiPort

Backend Service RMI port number

BackendSection/OVOMgmtServerInfo/NodeGroupName

OVO node group that the host name manager adds nodes to

<WLIEmailNotificationConfig>

This node allows you to configure an SMTP Gateway (such as recipient email address) to configure email/SMS notification. It has the following child nodes:

- <SMTPGatewayConfig>
- <RecipientEmailAddresses>

- <NotifConfig>

WLIEmailNotificationConfig/SMTPGatewayConfig

This node allows you to configure an SMTP gateway that we will be used for sending Email/SMS notification

WLIEmailNotificationConfig/SMTPGatewayConfig/SMTPRelayHost

The fully qualified name of the SMTP Gateway Host

WLIEmailNotificationConfig/SMTPGatewayConfig/SMTPRelayPort

The SMTP Gateway listen port, the default is 25

WLIEmailNotificationConfig/SMTPGatewayConfig/SMTPRelayUser

This parameter is the user name to access the SMTP Gateway if the SMTP Gateway specified requires user authentication

WLIEmailNotificationConfig/SMTPGatewayConfig/SMTPRelayPassword

This parameter is the password to access the SMTP Gateway if the SMTP Gateway specified requires user authentication

WLIEmailNotificationConfig/RecipientEmailAddresses

This node specifies the email addresses of the recipients of notifications.

WLIEmailNotificationConfig/RecipientEmailAddresses/RecipientEmailAddress

This parameter contains the email address of a recipient

WLIEmailNotificationConfig/NotifConfig

This node enables or disables Email/SMS notification

WLIEmailNotificationConfig/NotifConfig/SLAViolationNotif

This parameter is set to 'enable' to enable Email/SMS Notification for SLA Violation Notification and set to 'disable' to disable Email/SMS notification for SLA Violation Notification.

WLIEmailNotificationConfig/NotifConfig/OverdueTaskNotif

This parameter is set to 'enable' to enable Email/SMS Notification for Overdue Task Notification and set to 'disable' to disable Email/SMS notification for Overdue Task Notification.

WLIEmailNotificationConfig/SenderEmailAddress

This parameter contains the email address to be used as the sender's email address.

B List Templates and Reports

This appendix provides reference information for:

- Message Groups
- Applications
- Templates
- Performance Metrics
- Reports

Message Groups

Name	Description
WLI SPI	For messages coming from WLI SPI.
WLI	Messages coming from WLI managed environment

Applications

Name	Group	Description
WLI SPI Tools	Group	Contains all of the WLI SPI applications
Check Status WLI SPI	Application	Checks the status of the Backend and the Frontend
Configure WLI SPI	Application	Launches a GUI to modify the WLI SPI configuration file. Transfers the configuration file to the Frontend system and restarts the Backend and the Frontend.
Restart WLI SPI	Application	Restarts the Backend and the Frontend
Start WLI SPI	Application	Starts the Backend and the Frontend
Stop WLI SPI	Application	Stops the Backend and the Frontend

Templates

Name	Type	Group	Description
SPI for WLI	Group	N/A	Contains all of the WLI SPI template groups and templates.
WLISPI-Backend	Group	SPI for WLI	Contains the templates for the Backend Service node.
WLISPI-Frontend	Group	SPI for WLI	Contains the templates for the Frontend Subagent node.
WLISPI-Metrics	Group	WLISPI-Frontend	Contains the templates related to metric data collection and metric threshold monitoring.
WLISPI-Collect-Mon	Monitor	WLISPI-Metrics	Collects metric data for alarming, reports and graphs. Note that the metric data is logged into the report data source. For the graphs, the metric data is saved in text files in <code>/var/opt/OV/log</code> . It is logged into the graph data source by the WLISPI-Graph-Mon template.
WLISPI-Graph-Mon	Monitor	WLISPI-Metrics	Logs the metric data collected by the WLISPI-Collect-Mon template into the graph data source.
WLISPI_007	Monitor	WLISPI-Metrics	Reports the business process success ratio.

Name	Type	Group	Description
WLISPI-Backend- Log	Logfile	WLISPI-Backend	Monitors the Backend log file.
WLISPI-Frontend- Log	Logfile	WLISPI-Frontend	Monitors the Frontend log file.
WLISPI-Frontend- Mon	Monitor	WLISPI-Frontend	Monitors the WLISPI Frontend Subagent process.
WLISPI-Backend- Mon	Monitor	WLISPI-Backend	Monitors the WLISPI Backend service process.
WLISPI-Message-Interceptor	Message	WLISPI-Frontend	Template for intercepting messages from WLI managed environment.
WLISPI-Worklist- Mon	Monitor	WLISPI-Frontend	Monitors the WLI worklist for detecting overdue tasks.

Performance Metrics

Name	Metric Id	MO Type	Alarm	Graph and Report	Description
AvgExecTime	WLISPI_0001	urn:WebLogic Integration/ processtype	No	Yes	Average Execution Time for a Business Process
NumAborted Processes	WLISPI_0002	urn:WebLogic Integration/ processtype	No	Yes	Number of instances aborted for a Business Process
NumCompleted Processes	WLISPI_0003	urn:WebLogic Integration/ processtype	No	Yes	Yes
NumExceededSLA Processes	WLISPI_0004	urn:WebLogic Integration/ processtype	No	Yes	Number of instances of a Business Process that have Exceeded the SLA
NumTerminated Processes	WLISPI_0005	urn:WebLogic Integration/ processtype	No	Yes	Number of instances terminated for a Business Process
NumSuspended Processes	WLISPI_0006	urn:WebLogic Integration/ processtype	No	Yes	Number of instances suspended for a Business Process

Name	Metric Id	MO Type	Alarm	Graph and Report	Description
NumRunning Processes	WLISPI_0017	urn:WebLogic Integration/ processtype	No	Yes	Number of instances running for a Business Process
LastProcessStart Time	WLISPI_0008	urn:WebLogic Integration/ archiver	No	Yes	Last time the Archiver started archiving
LastProcessEnd Time	WLISPI_0009	urn:WebLogic Integration/ archiver	No	Yes	Last time the Archiver finished archiving
ArchivedRowCount	WLISPI_0011	urn:WebLogic Integration/ archiver	No	Yes	Number of rows archived during the last archiving
EventErrorCount	WLISPI_0012	urn:WebLogic Integration/ adapter	No	Yes	Event error count for an Adapter
ServiceErrorCount	WLISPI_0013	urn:WebLogic Integration/ adapter	No	Yes	Service error count for an Adapter

Name	Metric Id	MO Type	Alarm	Graph and Report	Description
ErrorCount	WLISPI_0014	urn:WebLogic Integration/eventgen	No	Yes	Error count for an Event Generator
DeadMessageCount	WLISPI_0015	urn:WebLogic Integration/msgchannel	No	Yes	Error Count for a Message Channel
MessageCount	WLISPI_0016	urn:WebLogic Integration/msgchannel	No	Yes	Number of messages in a Message Channel

Reports

Name	Description
Average Execution Time of a Business Process	Reports the average execution time of a particular Business process over a period of time.
SuccessRatio of a Business Process	Reports the SuccessRatio of a particular Business Process over a period of time.
Statistics of a Business Process	Reports the statistics of execution of a particular Business process. For example: how many instances failed, how many completed, how many aborted, how many are running, how many have violated SLA and so on for a specific Business Process type over a period of time.
Average Execution Time for System Archive	Reports the average time taken to complete System Archiving over a period of time.
Error Count in Event Generator	Reports the error count in an Event Generator over a period of time.
Error count in Adapter	Reports the service error count and event error count in an Adapter over a period of time.
Error Count in Message Channel	Reports the error count of a Message Channel over a period of time.
Message Queue Length of a Message Channel	Reports the queue length of a Message Channel over a period of time.

Glossary

Attributes

Represents information about an MO as a set of properties.

Backend Service

The Backend Service is a software component that receives data and information from the frontend subagent, converts the information to an OpenView recognized form, and allows OpenView to render the managed environment based on the management data.

Conversation

A managed object that implements the Conversation management interface which represents one service's view of a series of related messages.

Event

An event is a change in the state of the MO.

Event Manager

An Event Manager manages all events emitting from MOs. The Event Manager is responsible for storing, retrieving and (if persistence is implemented) recovering events.

Frontend Subagent

The Frontend Subagent is a software component that is responsible for communicating with the WLI SBA software to gather management data about a WLI environment and its hosted applications.

Managed Object (MO)

An MO is a management representation of a resource. An MO implements a management interface to provide a means to monitor and/or control the underlying resource. This term is taken out of WSMF specification. Usually

each managed resource will expose its manageability through an MO. OpenView manages all managed resources in WLI environment through their corresponding MOs. In the context of this document, when we talk about the MO, we also refer to the managed resource itself.

Management interface

A management interface exposes the management capabilities of a resource. A Management interface is presented as a set of attributes, operations, and notifications to be accessed through a set of WSDL `portTypes`.

Managed Resource

Any WLI application or abstract management notion such as a class of business process, is referred to as a managed resource.

Model

A model is a set of objects, properties, and their relations.

Namespaces

Namespaces are used to uniquely associate the port types for an interface with a URI. Namespaces are defined in an MO and used in the WSDL file for an MO.

Notification

A notification is a message that is sent or retrieved by one or more subscribers to inform that an event has occurred.

Notification Types

The set of exceptions and state changes that can be reported by an MO.

Operations

The set of functions that can be provided to support the management of an MO.

PortTypes

A `PortType` is the atomic unit of management functionality. In WSMF, MOs can choose which management `portTypes` to implement but cannot partially implement a `portType`. A `portType` is defined for each interface category and is used in the WSDL file for an MO.

Resource

A resource is a component of a deployed environment.

Resource View

An OpenView term that describes a UI representation of a computing environment from the system administrator point of view that starts with what applications are running on which hosts.

Relation

A relation is a type of association between MOs.

Relationship

A relationship specifies two managed objects and the relation to define how two specific objects are associated.

Service

An MO that implements the Service management interface which represents the management capabilities of a Web service. This Web service may be acting as the provider and/or the consumer of Web service messages.

Service View

The service view is a UI representation of a computing environment that is application-centric and describes all application dependencies. This view is the bases for root cause analysis of a failure condition.

Service Map

An OpenView term to describe a graphic view into a managed environment. This view shows relationships among MOs.

Simple Object Access Protocol (SOAP)

The standard for Web services messages. Based on XML, SOAP defines an envelope format and various rules for describing its contents. Seen (with WSDL and UDDI) as one of the three foundation standards of Web services, it is the preferred protocol for exchanging Web services.

Subscriber

A subscriber is an entity that is interested in selected notifications from MOs. These notifications contain information about the state change in an MO. For

scalability reason, subscription to notifications has an associated timeout. Subscription can be renewed before they expire.

Web Services Description Language (WSDL)

The standard format for describing a Web service. Expressed in XML, a WSDL definition describes how to access a Web service and what operations it can perform.

Web Services Execution Environment (WSEE)

An MO that implements the WSEE management interface which encapsulates the management capabilities of a Web service execution environment.



Index

A

- agent controller, 14
- agent node, 27
- application group, 62
- assign template groups, 47
- auto discovery, 15

B

- backend service, 16
 - architecture, 13
- business management
 - architecture, 12
- business process
 - monitoring, 80

C

- CODA
 - logging, 78
 - overview, 77
- coda.log, 78
- configuration
 - tool, 107
 - WLI SPI, 107
- configuration file
 - editing WCCconfig, 107
- create graphs, 80

D

- data collection, 74

- deployment scenarios, 17
 - co-located components, 18
 - consolidated, 17
 - fully distributed, 19
- deploy template groups, 46, 48
- domain extensions
 - OV, 21
 - WLI, 21

E

- event management, 70

F

- frontend subagent, 16
 - assigntemplate groups, 47
 - deploy template groups, 46, 48
 - uninstall, 60

G

- graphs, 77
 - creating, 80
 - viewing, 81

H

- hardware requirements, 26
- HPjconfig tool, 27

I

- installation
 - WLI SPI, 27

J

- Java console, 9
- Java console See resource explorer, 9

K

kernel parameters, 27

L

license

open source, 25

logging metrics, 71

log levels, 82

M

managed resources

viewing, 68

management categories

configuration, 21

control, 21

discovery, 21

event callback, 21

event push and event pull, 21

performance, 21

management console, 9

management interfaces, 21

management server, 27

MBean, 87

MeasureWare, 77

message group, 62

message groups

reference, 115

message templates, 63

MetricDefinitions.xml, 71

metric element attributes, 71

metrics, 71

configuration file, 71

modifying data collection, 74

MO

complex types, 90

create custom, 85

discovery class, 88

filtering unwanted, 69

implementation class, 85

model MBean interface, 87

resource explorer, 9

N

node groups, 64

O

open source license, 25

OpenView Operations, 12

OVO, 23

OVP, 80

business process graphs, 80

view graphs, 81

OV Reporter, 16

P

patches

software, 27

performance, 80

performance graphs, 77

performance metrics

reference, 115

R

reinstall

templates monitor commands and

actions, 64

- remove
 - WLI SPI application group, 62
 - WLI SPI bits, 61, 62
 - WLI SPI message group, 62
 - WLI SPI message templates, 63
 - WLI SPI node groups, 64
 - WLI SPI users, 63
- reporting, 77
- reports, 16
 - reference, 115
 - view, 78
- requirements
 - hardware, 26
 - software, 25
 - software patches, 27
 - WLI SPI installation, 25
- resource explorer, 9, 16
 - overview, 16

S

- SBA, 14
 - agent controller, 14
 - auto discovery, 15
 - configuration file, 88
 - developer kit, 15
- Self Healing Services, 104
- service management, 68
- service navigator, 9
- service reporter, 71
- software requirements, 25
 - patches, 27
- SPI (WLI) See WLI SPI, 12
- starting
 - WLI SPI, 49

T

- template groups
 - assign, 47
 - deploy, 46, 48
 - verify, 48
- templates
 - reference, 115
 - WLISPI-Collect-Mon, 75
 - WLISPI-Graph-Mon, 75
- templates monitor commands and actions, 64
- trace levels, 83

U

- uninstall
 - frontend subagent, 60
 - WLI SPI, 60
 - WLI SPI bits, 61, 62

V

- verify
 - template groups, 48
- view graphs, 81

W

- WCConfig
 - reference, 107
- WLI
 - business management overview, 12

- WLI SPI, 12
 - applications reference, 115
 - architecture overview, 12
 - backend service, 13, 16
 - business management overview, 12
 - configuration, 107
 - deployment scenarios, 17
 - frontend subagent, 16
 - hardware requirements, 26
 - installation HP-UX and SOLARIS, 27
 - installation requirements, 25
 - introduction, 12
 - message templates, 63
 - node groups, 64
 - remove bits, 61, 62
 - reports, 16
 - SBA See SBA, 14
 - software requirements, 25
 - starting, 49
 - uninstall, 60
 - user, 63
- WLISPI-Collect-Mon template, 75
- WLISPI-Graph-Mon template, 75
- WSDM
 - overview, 21
 - XML interfaces, 21
- WSDM channel, 12

