

HP Operations Smart Plug-in for IBM WebSphere Application Server

for HP Operations Manager for Windows®

Software Version: 5.30

Configuration Guide

Document Release Date: June 2008

Software Release Date: June 2008



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2003-2008 Hewlett-Packard Development Company, L.P.

Trademark Notices

UNIX® is a registered trademark of The Open Group.

Windows® is a US registered trademarks of Microsoft Corporation.

Java™ is a US trademark of Sun Microsystems, Inc.

Documentation Updates

This guide's title page contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP Software Support Online web site at:

<http://www.hp.com/go/hpsoftwaresupport>

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

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

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Contents

1	Introduction	9
2	Installing, Upgrading, and Removing the WebSphere SPI	11
	Installing the WebSphere SPI	11
	Upgrading the WebSphere SPI	11
	Policy Changes	11
	Prerequisites	11
	Upgrading the WebSphere SPI	12
	Removing WebSphere SPI	13
	Task 1: Remove All WebSphere SPI Policies from the Managed Nodes	13
	Task 2: Remove WebSphere SPI Node Groups on the Management Server	13
	Task 3: Remove the WebSphere SPI Software from the Management Server	13
3	Configuring the WebSphere SPI	15
	Configuration Prerequisites	15
	Task 1: Add Managed Nodes	15
	Task 2: Verify the Application Server Status	15
	Task 3: Collect WebSphere Login Information	17
	Task 4: Enable PMI	17
	Task 5: Connect using JSR 160	17
	Task 6: Update WebSphere's SDK	18
	Basic WebSphere SPI Configuration	19
	Configuration Prerequisite	19
	Task 1: Run Discover WebSphere	19
	Task 2: Verify the Discovery Process	21
	Additional WebSphere SPI Configuration	22
	Deploying a Different Policy Group	23
	WebSphere SPI in High Availability Environments	24
	Configuration Prerequisites	24
	Configuring WebSphere SPI for High Availability Environments	24
	Task 1: Create the WebSphere SPI monitoring configuration file	24
	Task 2: Create the clustered application configuration file	25
	Task 3: Configure WebSphere SPI for HTTPS or DCE Agent (Based on Requirement)	26
4	Customizing the WebSphere SPI Policies	27
	WebSphere SPI Policy Group and Types	27
	WebSphere SPI Policy Groups	27
	WebSphere SPI Policy Types	28
	Basic Policy Customizations	30
	Modifying Metric Policies	30

Threshold Level and Actions	30
Message and Severity	33
Advanced Policy Customizations	34
Creating New Policy Group	34
WebSphere SPI Collector/Analyzer Command with Parameters	35
Basic Collector Command Parameters	35
Using JMX Actions Command Parameters	36
Changing the Collection Interval for Scheduled Metrics	39
Changing the Collection Interval for Selected Metrics	39
Customizing the Threshold for Different Servers	40
Creating Custom, Tagged Policies	41
Restoring Default WebSphere SPI Policies	42
Viewing Text-Based Reports	42
Automatic Command Reports	42
Manually Generated Reports	43
WebSphere SPI Graphs	43
Monitoring WebSphere Application Server on Unsupported Platforms	44
Monitoring Remote Nodes (Running on Platforms Not Supported by WebSphere SPI)	44
Implementing Remote Monitoring	44
Configuring Remote System Monitoring	46
Task 1: Configure the Remote WebSphere System	46
Task 2: (Optional) Integrate HP Performance agent	46
Task 3: Deploy Policies to the Local Node	47
Configuring Remote Monitoring for Logfile (Optional)	47
Configuring the Logfile Policy for Remote Logfiles	47
Limitations in Remote Monitoring	48
5 Integrating HPOM Reporting and Graphing Features with the WebSphere SPI	49
Integrating the WebSphere SPI with HP Performance Agent	51
Integrating the WebSphere SPI with HP Reporter	52
Viewing Reports from the HPOM Management Console	54
Reports Generated by Reporter	55
Removing the WebSphere SPI Reporter Package	59
Integrating the WebSphere SPI with HP Performance Manager	59
Viewing Graphs that Show Alarm Conditions	59
Viewing Graphs that Show Past or Current Conditions	60
Viewing Graphs from the HP Performance Manager Console	60
WebSphere SPI Metrics Available for Graphs	61
Removing the WebSphere SPI Grapher Package	63
6 User Defined Metrics	65
Metric Definitions DTD	66
The MetricDefinitions Element	66
Example	66
The Metric Element	67
Example	67
The PMI Counter Element	68
Example	69

FromVersion and ToVersion Elements	69
Example	70
Calculation and Formula Elements	70
Syntax	70
Functions	70
Examples	71
Sample 1	71
Sample 2	71
Sample 3: Metric Definitions File	72
Creating User-Defined Metrics	74
Task 1: Disable Graphing (if Enabled)	74
Task 2: Create a metric definitions file	74
Task 3: Configure the Metric Definitions File Name and Location	74
Task 4: Create a UDM Policy Group and Policies	75
Task 5: Deploy the policy group	76
Task 6: Enable graphing.	76
7 Troubleshooting the WebSphere SPI	77
The Self-Healing Info Tool	77
Log and Trace Files	78
UNIX Managed Nodes	78
Windows Managed Nodes.	79
Troubleshooting the Discovery Process	81
Manually Deploying the Discovery Policies	82
Verifying the Node Name	82
Troubleshooting the Tools	83
Glossary	85
Index	91

1 Introduction

The HP Operations Smart Plug-in for IBM WebSphere Application Server (WebSphere SPI) allows you to manage WebSphere servers from an HP Operations Manager for Windows (HPOM) console. WebSphere SPI adds monitoring capabilities otherwise unavailable to HPOM. For more information on HPOM, see the HPOM console online help.

From the HPOM console, you can monitor the availability, use, and performance of WebSphere Application Servers running on HPOM managed nodes. You can integrate WebSphere SPI with other HP products like HP Reporter and HP Performance Manager to get consolidated reports and graphs which help you analyze trends in server usage, availability, and performance.

The WebSphere SPI online help provides valuable information about WebSphere SPI concepts and other topics that will help you understand the product.

This guide covers the following topics:

- [Installing, Upgrading, and Removing the WebSphere SPI](#)
- [Configuring the WebSphere SPI](#)
- [Customizing the WebSphere SPI Policies](#)
- [Integrating HPOM Reporting and Graphing Features with the WebSphere SPI](#)
- [User Defined Metrics](#)
- [Troubleshooting the WebSphere SPI](#)

2 Installing, Upgrading, and Removing the WebSphere SPI

Installing the WebSphere SPI

If HPOM is already installed, it is not necessary to stop the existing HPOM sessions before installing the WebSphere SPI. To install the WebSphere SPI on the management server, follow these steps:

- 1 Insert the HP Operations Smart Plug-ins DVD into the DVD drive of the management server system. The HP Operations Manager InstallShield Wizard opens.
- 2 Click **Next**. The Program Maintenance window opens.
- 3 Click **Install Products**. The Product Selection window opens.
- 4 From the options listed (there are three Product Selection windows), select the **IBM WebSphere** check box and click **Next**.

Complete the installation by following the instructions that appear as you proceed. For more information see the *HP Operations Smart Plug-ins DVD Installation and Upgrade Guide for Windows*.

Upgrading the WebSphere SPI

Detailed information about supported software, enhancements, fixes, and known problems and workarounds is available in the *HP Operations Smart Plug-in for WebSphere Application Server Release Notes* located on the HP Operations Smart Plug-ins DVD in `\Documentation\Releasenotes\WebSphere_AppServer_Releasenotes.html`.

Policy Changes

There are no policy changes in this release of WebSphere SPI. The policy structure is similar to the previous release.

Prerequisites

- 1 Back up the existing WebSphere SPI configuration file from the location:
`<OvOWShareInstallDir>\SPI-Share\wasspi\wbs\conf\SiteConfig`
On OVO for Windows 7.50, `<OvOWShareInstallDir>` can be `C:\Program Files\HP OpenView\Data\shared\`
- 2 Run the InstallShield Wizard (installer) to install the new version of the WebSphere SPI. If the installer detects that an older version of the WebSphere SPI is installed, it does the following to upgrade to the new version:

- Renames the existing SPI for WebSphere policy group to SPI for WebSphere - Saved Policies. The default policies you customized in the SPI for WebSphere policy group, are available in the SPI for WebSphere - Saved Policies policy group.
- Updates the WebSphere SPI instrumentation on the management server.
- Installs new tools, policies, and graph file on the management server.

Upgrading the WebSphere SPI

To upgrade the WebSphere SPI to version 5.10, follow these steps:

- 1 Install the WebSphere SPI version 5.10 software. See [Installing the WebSphere SPI](#) on page 11.
- 2 Deploy new instrumentation to the SPI for WebSphere node group:
 - a Right-click the **SPI for WebSphere** node group.
 - b Select **All Tasks** → **Deploy instrumentation**.
 - c Select **SPI for WebSphere** and **WBS SPI Discovery**.
 - d Verify that the “Remove Existing Instrumentation Before Deploying New Instrumentation” check box is clear.
 - e Click **OK**.



After upgrading WebSphere SPI, if you add an instance of WebSphere Application Server on a managed node, then you must run the Discover WebSphere tool on that node.

Removing WebSphere SPI

To completely remove the WebSphere SPI, delete all the WebSphere SPI program components and the WebSphere SPI policies.

Complete the tasks in the specified order.


Task 1: Remove All WebSphere SPI Policies from the Managed Nodes

- 1 In the console tree, select **Policy management** → **Policy groups**.
- 2 Right-click **SPI for WebSphere** and select **All Tasks** → **Uninstall from**. A node selection window appears.
- 3 Select the nodes on which the policies are installed.
- 4 Click **OK**.
- 5 Verify the policies are uninstalled. Check the status of the job in **Deployment jobs** under Policy groups. All WebSphere SPI policies must be uninstalled before you start the next task.

If you customized policies (copies of WebSphere SPI default policies) residing in other HPOM policy groups, you should remove them as well.

Task 2: Remove WebSphere SPI Node Groups on the Management Server

If you created the SPI for WebSphere node group (by running the Create WBSSPI Node Groups tool or manually), you must remove the group.

- 1 In the console tree, select **Nodes** → **SPI for WebSphere**.
- 2 Open the Node Configuration editor.
 - a Select the Nodes folder in the console tree.
 - b Click the node icon  on the Configuration toolbar to open the editor. A node list appears.
- 3 Select the name of the node group you want to delete and press the **Delete** key. You can also right-click the node group and select **Delete**. The Confirm Delete window opens.
- 4 Click **Yes**.
- 5 Click **OK** to close Configure managed nodes window.

Task 3: Remove the WebSphere SPI Software from the Management Server

- 1 Insert the HP Operations Smart Plug-ins DVD into the DVD drive of the management server. The HP Operations Manager InstallShield Wizard starts.
- 2 From the first screen, select **Next**. The Program Maintenance window opens.
- 3 Select **Remove products**. The Product Selection window opens.
- 4 Select the **IBM WebSphere** check box and click **Next**.
- 5 Complete the removal by following the instructions that appear as you proceed.

3 Configuring the WebSphere SPI

This chapter explains how to configure the WebSphere SPI for use with HP Operations Manager (HPOM). You must first complete all the configuration prerequisites. Then you must perform the basic configuration and complete additional configuration based on your environment.

Configuration Prerequisites

Complete the following tasks before configuring WebSphere SPI.

Task 1: Add Managed Nodes

For each WebSphere Application server you want to manage from HPOM, ensure that all nodes on which the WebSphere Application servers are running are configured in HPOM as managed nodes.

To add a UNIX managed node, follow these steps:

- 1 Install the HPOM agent on the node. See the HPOM console online help topic “Agent Installation on UNIX computers” for more information.
- 2 Specify each WebSphere Server node on UNIX to be managed. See the HPOM console online help topic “Configure Managed Nodes” for more information.

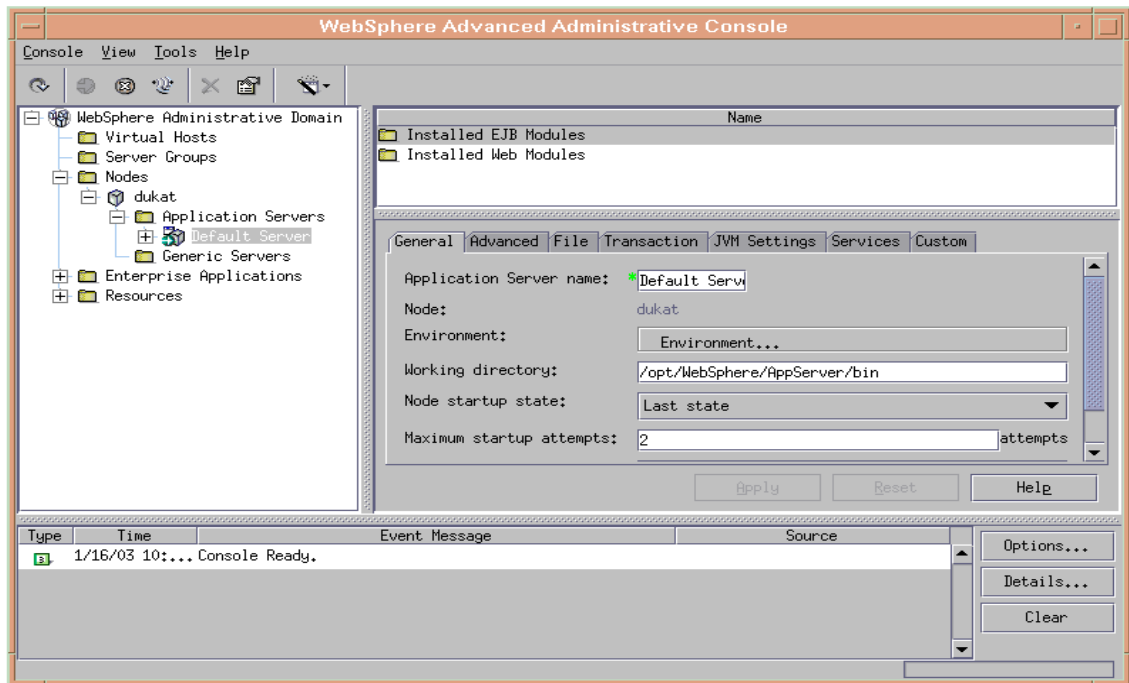
For a Windows managed node, do the following:

Specify each WebSphere node on Windows to be managed. See the HPOM console online help topic “Configure Managed Nodes” for more information (the HP Operations agent is automatically installed when you complete this step).

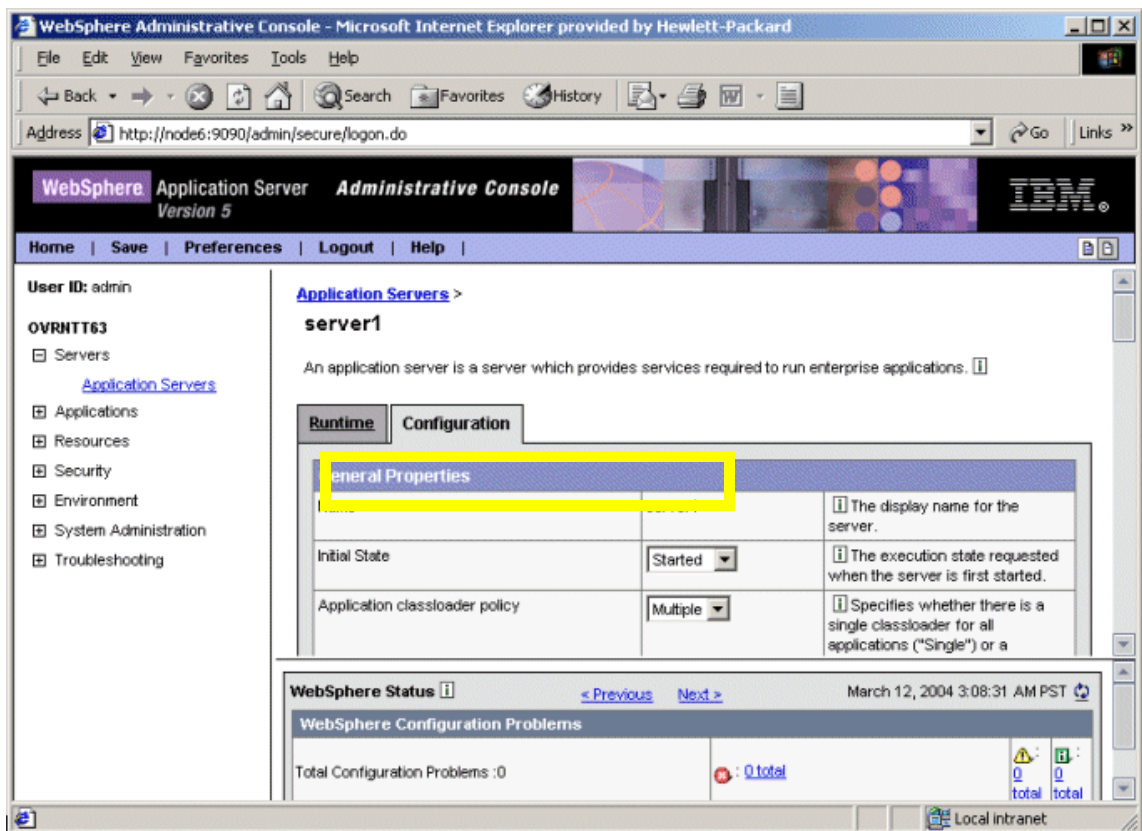
Task 2: Verify the Application Server Status

Verify that your WebSphere application servers are running.

For the WebSphere Server version 4, verify the server’s status from the WebSphere Advanced Administrative Console. A colored icon appears next to the Application Server name. A green icon means the server is running. A red icon means the server is not running. If the icon is red, start the server.



For the WebSphere Server version 5, check the server's status from the WebSphere Administrative Console



If you cannot verify the server's status using the Administrative Console, run the following commands on the managed node:

- UNIX: `<WebSphere_Install_Dir>/bin/serverStatus.sh -all`
For example: `/opt/WebSphere/AppServer/bin/serverStatus.sh -all`
- Windows: `<WebSphere_Install_Dir>\bin\serverStatus.bat -all`
For example: `C:\Program Files\WebSphere\AppServer\bin\serverStatus.bat -all`

Task 3: Collect WebSphere Login Information

If security is enabled on the WebSphere server, collect the username and password for each WebSphere Admin Server. The user must have the correct privileges assigned for the WebSphere Admin Server.

The WebSphere SPI discovery process uses the username (or Login) and password to gather basic configuration information and by the WebSphere SPI data collector to collect metrics.

Configuration of the WebSphere SPI is simplified if the username and password to access all WebSphere Admin Servers are the same.

If you are using WebSphere version 5.1.0 or earlier, you must use the default WebSphere Admin Server username and password (the username and password configured when the WebSphere application server was installed or configured).

If you are using WebSphere version 5.1.1 or later, you should be able to use the username and password for users or groups assigned to the administrator or operator role.

If you are using LDAP directory, then to access the WebSphere Console from LDAP, you must create a user account similar to the user account of LDAP in the local WebSphere instance. You must grant Administrator privileges to this user.

Task 4: Enable PMI

If you are running WebSphere Server version 5, enable PMI using the WebSphere Administrative Console and restart the server.

Task 5: Connect using JSR 160

You can configure the Websphere Application Server 6.1 or later to use JSR 160 connection to connect to the Websphere Application Server. By default, the JSR 160 connection is disabled.

To enable JSR 160 connection set the **JSR160** flag in the SPI Config file to **true**. By default, this flag is set to "false." The SPI Config file is present in the `<AgentDir>/wasspi/wbs/conf` directory.



If you use JSR 160 to connect to Websphere Application Server 6.1 or later, the application server can run in "security enabled" or "security disabled" mode. In the "security disabled" mode the collector can run in both Transient and Persistent mode, but in the "security enabled" mode the collector can only run in the Transient mode. To set the collector in Transient mode, add a line– **COLLECTOR_MODE=TRANSIENT** at the end of the SPIConfig file.

If you are using WebSphere Application Server 6.1 or greater, before starting the collector you must set the following values for the attributes in the `<Websphere_HOME>/profiles/<profile_name>/properties/sas.client.props` file. Set these values for all the profiles that you want to monitor.

- Set the value of `loginSource` attribute to **properties** (the default value of `loginSource` is **prompt**).

```
com.ibm.CORBA.loginSource=properties
```

- Set the value of `loginUserId` attribute to the WebSphere admin user id and `loginPassword` attribute to the WebSphere admin password:

```
com.ibm.CORBA.loginUserId=<admin_user>
```

```
com.ibm.CORBA.loginPassword=<admin_password>
```

If you do not update the `sas.client.props` file, the collector will fail.



After updating the `sas.client.props` file, you must restart the WebSphere Application Server, if it is running.

Task 6: Update WebSphere's SDK

For WebSphere Application Server 6.1 running on Windows nodes, you must update IBM Java SDK 1.5 to level SR4 or later (Java SDK 1.5 SR4 or later) or the collector may fail.

You can download Java SDK 1.5 SR4 or later from <http://www-1.ibm.com>.

Basic WebSphere SPI Configuration

To complete basic WebSphere SPI configuration, complete the following tasks.

Configuration Prerequisite

Before launching the Discover WebSphere tool deploy the following instrumentation files on the managed nodes:

- SHS Data Collector
- SPI Data Collector
- SPI for WebSphere
- WBSSPI Discovery

To deploy these instrumentation files, follow these steps:

- 1 From the HPOM console select **Operations Manager** → **Nodes**.
- 2 Right-click the managed node on which you want to run Discover WebSphere tool.
- 3 Select **All Tasks** → **Deploy instrumentation**. The Deploy Instrumentation window opens.
- 4 Select SHS Data Collector, SPI Data Collector, SPI for WebSphere, and WBSSPI Discovery from the list of instrumentation files and click **OK**.

To verify that these files deployed successfully, check Deployment Jobs under Policy management. There should be no error messages.

Task 1: Run Discover WebSphere

Discover WebSphere sets basic configuration properties needed for discovery and deploys the WebSphere SPI discovery policies, and updates the service map.

To run Discover WebSphere, follow these steps:

- 1 From the HPOM console, select **Operations Manager** → **Tools** → **SPI for WebSphere** → **SPI Admin**.
- 2 Double-click **Discover WebSphere**.
- 3 Select the managed nodes on which WebSphere Application servers are running.
- 4 Click **Launch**.


The Console Status window opens. After a few seconds for the Introduction window opens. This window contains brief information about the Discover WebSphere tool.

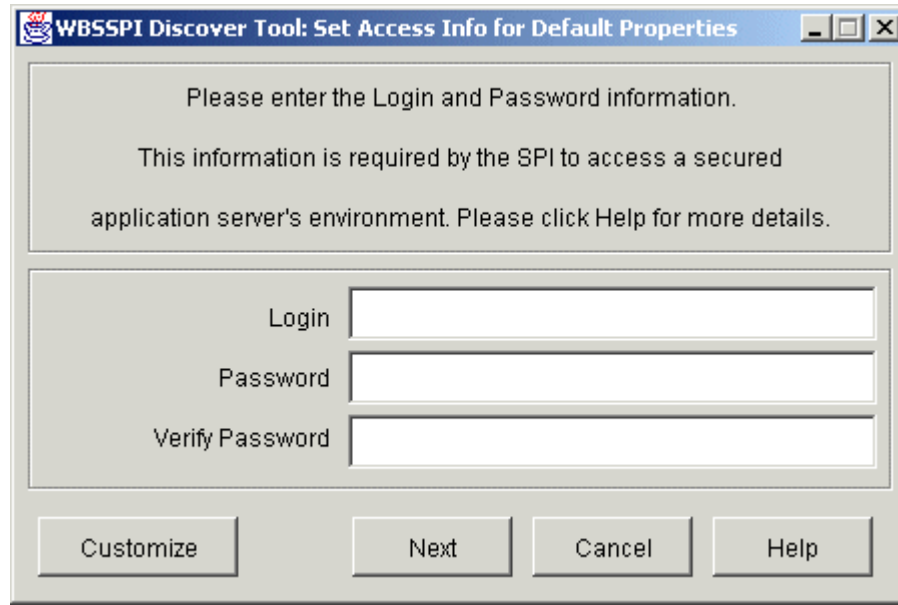
- 5 Click **Next**.

A second Introduction window opens. This window contains instructions about how to enter the WebSphere login and password information you collected.

Click **Next**.

- 6 If you did not set the WebSphere SPI LOGIN and PASSWORD properties, the Set Access Info for Default Properties window opens.

 If you have already set the LOGIN and PASSWORD properties, the configuration editor opens. Go to step 7.



Set the LOGIN and PASSWORD properties to the WebSphere login and password collected in [Task 3: Collect WebSphere Login Information](#) on page 17. The WebSphere Admin Server login information is required when security is enabled. If security is not enabled, leave these fields blank, Click **Next**, and go to step 8.

The LOGIN and PASSWORD properties set in this window are used as the default WebSphere Admin Server login and password (they are set at the global properties level). If no NODE level or server-specific LOGIN and PASSWORD properties are set, this WebSphere login and password are used by the WebSphere SPI to access all WebSphere Admin Servers. For more information about the configuration structure, see the topic – The configuration in the WebSphere SPI online help.

If the WebSphere Administration Server login and password are the same for all instances of WebSphere on all HPOM managed nodes, follow these steps:

- a Set the LOGIN and PASSWORD in the Set Access Info for Default Properties window.
- b Click **Next**.
- c Go to step 8.

If the WebSphere Admin Server login and password are different for different instances of WebSphere, you must customize the WebSphere SPI configuration by setting the LOGIN and PASSWORD properties at the NODE or server-specific level (for more information about the configuration structure, see the topic –The configuration in the WebSphere SPI online help).

- a Set LOGIN and PASSWORD to the most commonly used WebSphere login and password in the Set Access Info for Default Properties window.
 - b Click **Customize** to start the configuration editor.
- 7 From the configuration editor, set the configuration properties. For more information about using the configuration editor, see the WebSphere SPI online help.
 - 8 Click **Next** to save changes and exit the editor. The Confirm Operation window opens.

- 9 Click **OK**. The discovery policies are deployed to the selected managed nodes.
 - ▶ If you select **Cancel**, the discovery policies are not deployed. However, if you made changes to the configuration, those changes remain in the configuration on the management server. To make the changes to the selected managed nodes' configuration, you must start the Discover WebSphere tool, select those managed nodes, Click **Next** in the configuration editor, and then click **OK**.
- 10 Check the Console Status window for error messages. If none appear, click **Close**.
If the window displays an error message, see [Troubleshooting the Discovery Process](#) on page 81 to diagnose and troubleshoot.

Task 2: Verify the Discovery Process

Depending on the number of managed nodes in your environment, verification may take several minutes to complete.

To verify if the discovery process is successfully completed, follow these steps:

- 1 Check if the following message appears in the message browser of the managed node:

```
INFO - Updating the WBSSPI configuration data with discovered
information
```

Depending on the number of managed nodes in your environment, it may take several minutes for these messages to appear for all managed nodes.

If this message is present and the letter “S” (for successful) appears in the ‘A’ column of the message browser, the WBSSPI Discovery policies are successfully deployed.

If this message does not appear or if the message appears but the letter “F” appears in the ‘A’ column, check for error messages. To troubleshoot, see [Troubleshooting the Discovery Process](#) on page 81.

- 2 From the HPOM console, select **Operations Manager** → **Services** → **Applications** → **WebSphere**. The service map appears. It may take some time for the service map to appear completely.
- 3 Verify that the WebSphere, WebSphere Admin Server, and application server instances are represented correctly.

▶ After the discovery process is complete, the appropriate WebSphere SPI group policies are deployed on the managed nodes. 10 minutes after the policies are deployed, an automatic procedure to set up a managed node for WebSphere SPI operations starts.

- 4 Launch the Verify tool, 10 minutes after the service map appears, to verify the version of the policies installed on a managed node. To launch the Verify tool, follow these steps:
 - a From the HPOM console, select **Operations Manager** → **Tools** → **SPI for WebSphere** → **SPI Admin**.
 - b Double-click **Verify**. The Select Where to Launch This Tool window opens.
 - c Select the nodes on which you want to run the Verify tool.
 - d Click **Launch**. The Tool Status window opens.
The WebSphere SPI version is displayed. The version should be B.02.09 or later.
 - e Click **Close**.

Additional WebSphere SPI Configuration

After you successfully complete basic WebSphere SPI configuration, you must finish WebSphere SPI configuration by setting the properties that are not automatically discovered by the Discovery policies) and install and configure additional components. Setting some of these properties and configuring additional components depends on your environment.

See the WebSphere SPI online help for a complete definition of the properties.

Property	When to Set
START_CMD and STOP_CMD	To run the Start WebSphere and Stop WebSphere tools from the HPOM console.

- If you are configuring user-defined metrics, see the JMX Metric Builder online help for additional installation and configuration information.
- If HP Reporter is installed (must be purchased separately), see [Integrating the WebSphere SPI with HP Reporter](#) on page 52 for installation and configuration information.
- If HP Performance Manager is installed (must be purchased separately) and you want to view graphs, set the GRAPH_URL property. See [Integrating the WebSphere SPI with HP Performance Manager](#) on page 59 for additional installation and configuration information.

To update the configuration, follow these steps:

- 1 From the HPOM console, select **Operations Manager** → **Tools** → **SPI for WebSphere** → **SPI Admin**.
- 2 Double-click **Configure WBSSPI**. The Edit Parameter window opens.
- 3 Select the managed nodes to configure.
- 4 Click **Launch**. The Console Status window and then the Introduction window opens
- 5 Click **Next**. The configuration editor opens
- 6 Set the properties.
- 7 Click **Next** to save and exit the editor.

For a complete description of the WebSphere SPI properties and information about setting the properties using the configuration editor, see the section Configuration properties in the WebSphere SPI online help.

Deploying a Different Policy Group

The WBSSPI Discovery policy automatically deploys the Medium-Impact policy group to the managed node on which it discovers the presence of a WebSphere application server. To deploy a different set of policies (High-Impact, Low-Impact, or your own custom policies), follow these steps:

- 1 Remove the existing Medium-Impact policy group as follows:
 - a From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere**.
 - b Right click **Medium-Impact** and select **All Tasks** → **Uninstall from**.
 - c Select the nodes from which you want to remove the Medium-Impact policy group.
 - d Click **OK**.
- 2 Deploy the different policy group:
 - a From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere**.
 - b Right click the policy group to deploy and select **All Tasks** → **Deploy on**.
 - c Select the nodes from which you want to deploy the different policy group.
 - d Click **OK**.

The PMI level of a node is automatically adjusted to a higher level when a higher impact level policy group is deployed. For example, deploying the High-Impact policy group on a node would result in a PMI setting of “high” for the node. However, PMI levels do not automatically revert to lower impact levels, even after removing policies from a node and/or deploying a lower impact level policy group. To lower a PMI level for a node, you must manually re-set the PMI level within WebSphere. Monitoring settings can be changed using the WebSphere Resource Analyzer tool.



See IBM WebSphere documentation for more information about PMI.

WebSphere SPI in High Availability Environments

High availability is a general term used to characterize environments that are business critical and therefore are protected against downtime through redundant resources. Very often, cluster systems are used to reach high availability.

You can configure WebSphere SPI to accommodate cluster environments where failovers allow uninterrupted WebSphere server availability. WebSphere SPI monitoring, when synchronized with the cluster environment, can switch from the failed node to the active node.

Configuration Prerequisites

The prerequisites for using WebSphere SPI in high availability environments are:

- Management Server: HPOM for Windows 8.10, HPOM for Windows 8.00, or OVO for Windows 7.50
- Node: HP-UX MCSG cluster
- HPOM 8.x HTTPS and DCE Agent version (for details see Agent cluster support matrix)

Configuring WebSphere SPI for High Availability Environments

To configure WebSphere SPI for use in high availability environments complete the following tasks:

- Task 1: Create the WebSphere SPI monitoring configuration file
- Task 2: Create the clustered application configuration file
- Task 3: Configure WebSphere SPI

Task 1: Create the WebSphere SPI monitoring configuration file

WebSphere SPI uses a monitoring configuration file `<appl_name>.apm.xml` that works in conjunction with the clustered application configuration file.



`<appl_name>` is the namespace_name. For more information, see *HP Operations Manager for UNIX HTTPS Agent Concepts and Configuration Guide*.

The `<appl_name>.apm.xml` file lists all the WebSphere SPI templates on the managed node so that you can disable or enable these templates as appropriate, for inactive and active managed nodes.

To create this clustered application configuration file for your WBS environment, follow these steps:

- 1 Use the following syntax to create the `<appl_name>.apm.xml` file:

```
<?xml version="1.0"?>
<APMApplicationConfiguration>
  <Application>
    <Name> ... </Name>
    <Template> ... </Template>
    <StartCommand>wasspi_wbs_perl -S wasspi_wbs_clusterSvrApp -opt
startMonitor $instance</StartCommand>
```



```

        <StopCommand>wasspi_wbs_perl -S wasspi_wbs_clusterSvrApp -opt
        stopMonitor $instance</StopCommand>
    </Application>
</APMAApplicationConfiguration>

```

- 2 Enter the namespace_name within the <Name></Name> tag.
- 3 After the file is created, save it in the \$OvDataDir/bin/instrumentation directory for DCE agent. For HTTPS agent save it in the \$OvDataDir/bin/instrumentation/conf directory.

Sample <appl_name>.apm.xml file

```

<?xml version="1.0"?>
<APMAApplicationConfiguration>
    <Application>
        <Name>wbsspi</Name>
        <Template>WBSSPI Error Log</Template>
        <Template>WebSphere Activity Log</Template>
        <Template>WebSphere Logs</Template>
        <Template>WBSSPI-Performance</Template>
        <Template>WBSSPI-Messages</Template>
        <Template>WBSSPI-40-High-05min</Template>
        <Template>WBSSPI-40-Low-05min</Template>
        <Template>WBSSPI-40-Med-05min</Template>
        <Template>WBSSPI-ConfigCheck</Template>
        <Template>WBSSPI-Logfile-Monitor</Template>
        <Template>WBSSPI Service Discovery</Template>
        <StartCommand>wasspi_wbs_perl -S wasspi_wbs_clusterSvrApp -opt
        startMonitor $instance</StartCommand>
        <StopCommand>wasspi_wbs_perl -S wasspi_wbs_clusterSvrApp -opt
        stopMonitor $instance</StopCommand>
    </Application>
</APMAApplicationConfiguration>

```

To prevent the agent from running the policies on a passive node, you must mention the policy names within the <template></template> tag.



<appl_name>.apm.xml is dependent on the application namespace. It is not dependent on the instance level. Therefore, the start and stop actions are provided with the associated instance name as their first parameter when the start and stop actions are run at package switch time. The environment variable \$instanceName is set by CIAW when start or stop tasks are performed.

Task 2: Create the clustered application configuration file

The clustered application configuration file apminfo.xml, working in conjunction with the <appl_name>.apm.xml file of WebSphere SPI, allows you to associate WebSphere SPI monitored instances with cluster resource groups. As a result, when you move a resource group from one node to another, in the same cluster, monitoring stops on the failed node and starts on the new node.

To create the clustered application configuration file `apminfo.xml` follow these steps:

- 1 Use a text editor to create the file. The syntax is:

```
<?xml version="1.0" ?>
<APMClusterConfiguration>
  <Application>
    <Name>namespace_name</Name>
    <Instance>
      <Name><Instance Name></Name>
      <Package><Package Name></Package>
    </Instance>
  </Application>
</APMClusterConfiguration>
```

- 2 Enter `namespace_name` within the `<Name></Name>` tag.
- 3 Save the `apminfo.xml` file in the `$OvDataDir/conf/conf` directory for HTTPS Agent. For DCE Agent, save the `apminfo.xml` file in the `$OvDataDir/conf/OpC` directory.

Sample `apminfo.xml` file

```
<?xml version="1.0" ?>
<APMClusterConfiguration>
  <Application>
    <Name>namespace_name</Name>
    <Instance>
      <Name>instance_name</Name>
      <Package>test</Package>
    </Instance>
  </Application>
</APMClusterConfiguration>
```

Task 3: Configure WebSphere SPI for HTTPS or DCE Agent (Based on Requirement)

To configure WebSphere SPI for HTTPS or DCE agent, follow these steps:

- 1 Deploy instrumentation files and policies on the target cluster nodes.
- 2 Launch the Discover WebSphere tool with active cluster node as target. For details about launching the discovery tool, see the WebSphere SPI online help.
- 3 Launch the Configure WBSSPI tool with the active cluster node as target. The configuration editor opens.
- 4 Copy the `SiteConfig` file from active node to passive node. The file is located in the `$OvDataDir/wasspi/wbs/conf` directory for DCE agent and in the `$OvDataDir/conf/wbsspi` directory for HTTPS agent.

4 Customizing the WebSphere SPI Policies

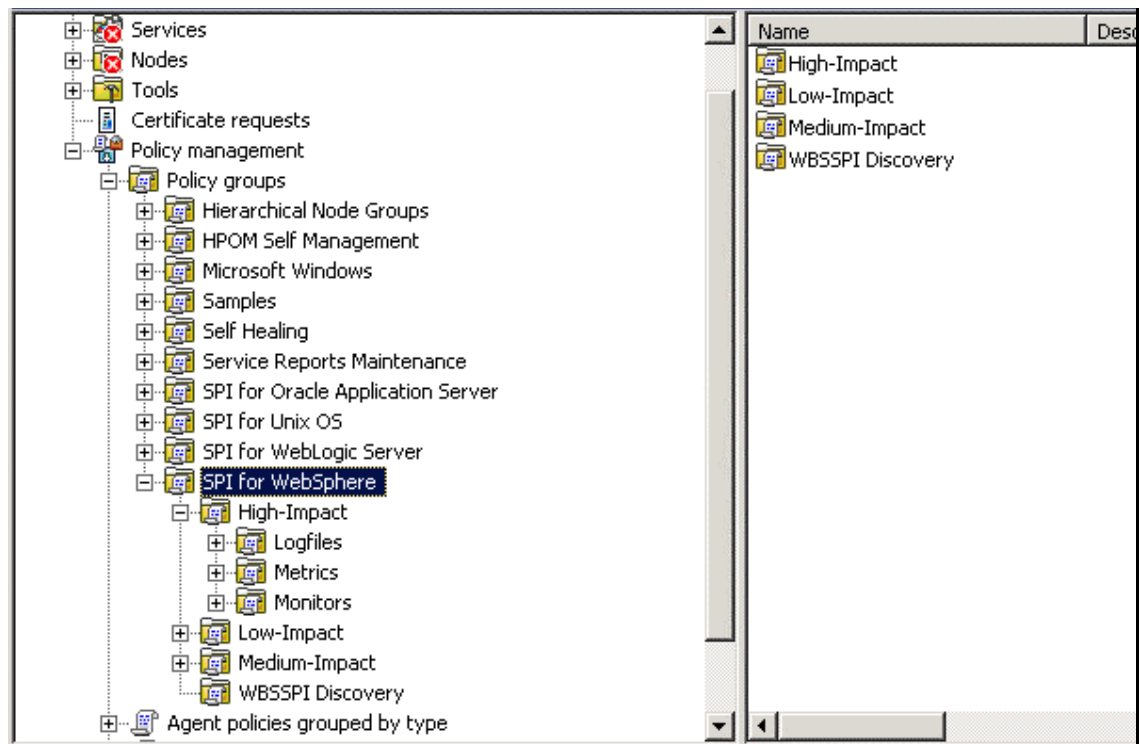
WebSphere SPI policies help you monitor the IBM WebSphere Application Servers. You can customize these policies depending on the requirements of your IT environment. This chapter includes general guidelines about the WebSphere SPI policies and explains how you can customize them. For more information see the Policies section in the WebSphere SPI online help.

WebSphere SPI Policy Group and Types

You can customize WebSphere SPI policies to suit the needs of your IT environment. However, these policies can also work without any modifications.

WebSphere SPI Policy Groups

The WebSphere SPI policies are organized under the top-level - SPI for WebSphere policy group (as shown in the following figure.)



WebSphere Policy Groups and System PMI levels: When you deploy a policy group on a managed node, the PMI level of the node is automatically adjusted to that of the policy group. For example, deploying the High-Impact policy group on a node would result in a PMI setting of “high” for the node.



PMI levels, once set, do not automatically revert to lower impact levels, even after removing policies from a node or deploying a lower impact level policy group. To lower a PMI level for a node you must manually re-set the PMI level within WebSphere. Change the monitoring settings with the WebSphere Resource Analyzer tool

The High-Impact, Medium-Impact, and Low-Impact subgroups contain metric, logfiles and collector policies that work as follows:

- **Metric policies** interpret incoming values of WebSphere’s performance levels and availability. Each value is evaluated according to the metric with which it is associated. If it is acceptable, it is ignored. If it is not, a message is sent to the HPOM Message browser and an automatic action may start.

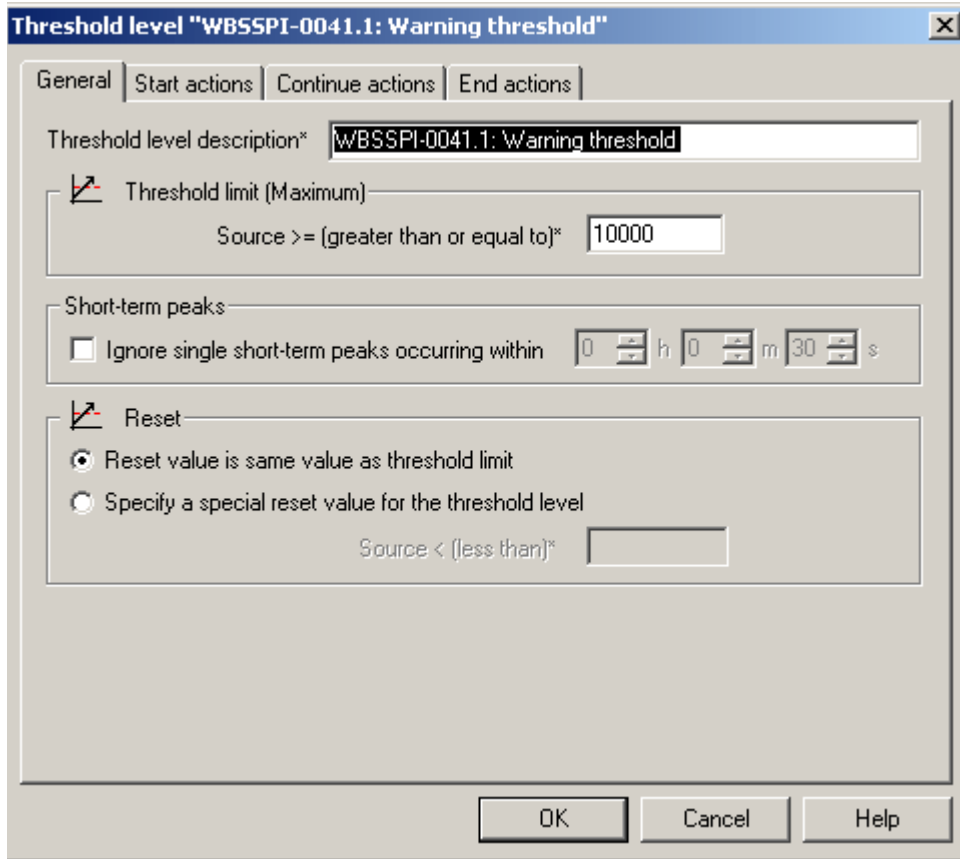
Metric policies contain defined thresholds that can trigger alerts/messages. Incoming values are compared against those thresholds and when a value exceeds a threshold, a message or alert is sent to the HPOM console.

- **Monitor policies** (collector policies) schedule when and what is collected. Specifically, the collector policy has two functions:
 - to run the collector/analyzer at each collection interval
 - to specify the metrics collected for that data collection interval.
- **Logfiles policies** monitor logfiles generated by WebSphere and WebSphere SPI. The information from these logfiles covers changes to WebSphere configurations and errors that occur in the operation of the WebSphere or the WebSphere SPI.

WebSphere SPI Policy Types

Metric policies define how data is collected for the individual metric and set a threshold value that, when exceeded, generates alerts or messages in the Message Browser. You can change the threshold within a policy by double-clicking on the policy, clicking the **Threshold levels** tab, and clicking on **Threshold level** in the Level summary pane.

Incoming values for metric WBSSPI-0041.1 are compared against its policy settings. In the following illustration, the default threshold is set at 10000.



Collector policies define all metrics for the WebSphere application that are scheduled for collection at the specified interval. Within the name of each collector policy is its collection interval (for example, WBSSPI-50-High-1h). When you open any collector policy, you see all metrics (by number) collected within the interval following the `-m` option of the collector/analyzer command `wasspi_wbs_ca`.

Basic Policy Customizations

This section covers basic policy customizations like changing threshold values, scheduling or deleting a metric from data collection, opening a metric policy or collector policy and so on.

Before you begin to customize any of the policies, make copies of the original policies so that the default policies remain intact.

Modifying Metric Policies

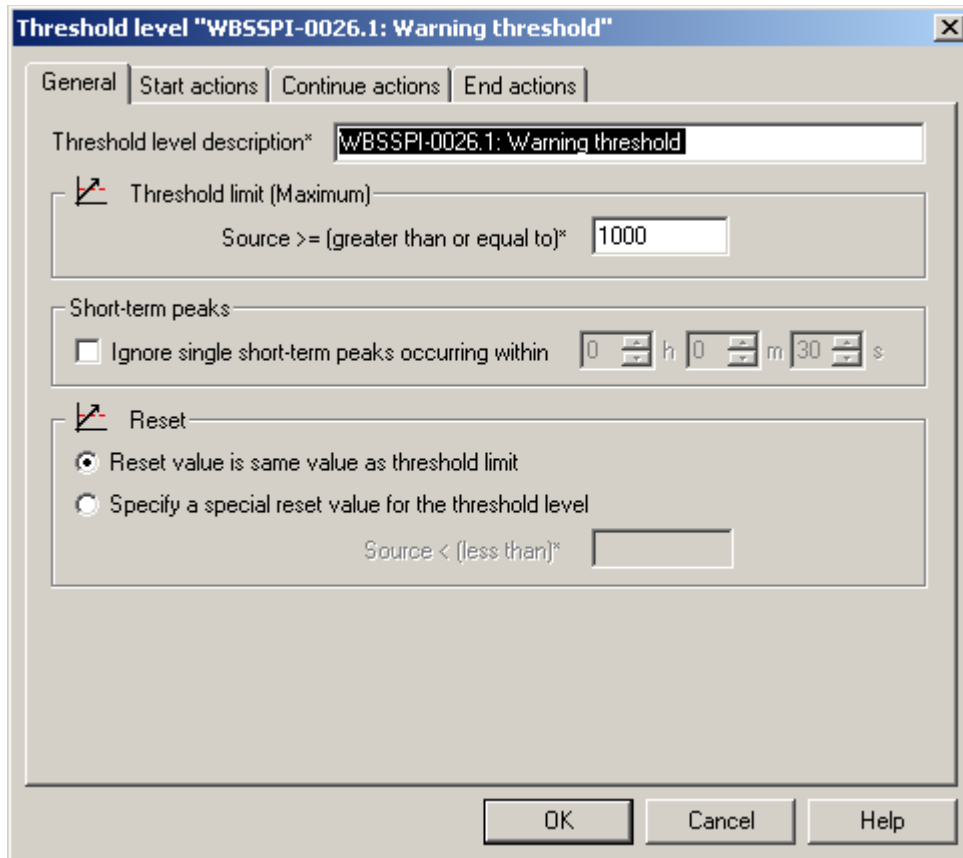
You can modify the metric attributes for all monitored instances of WebSphere. Some of these attributes are explained in the Configuration Properties section in the WebSphere SPI online help.

Threshold Level and Actions

To modify the threshold level and actions of a policy, follow these steps:

- 1 From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere** → *<Impact>*-**Impact** → **Metrics**.
- 2 Double-click a policy. The policy for which you want to change the threshold value. The policy window opens.
- 3 Select the Threshold levels tab. From the Level summary pane, click **Threshold level**. The Threshold level window opens.

The following figure shows the Threshold Level window.

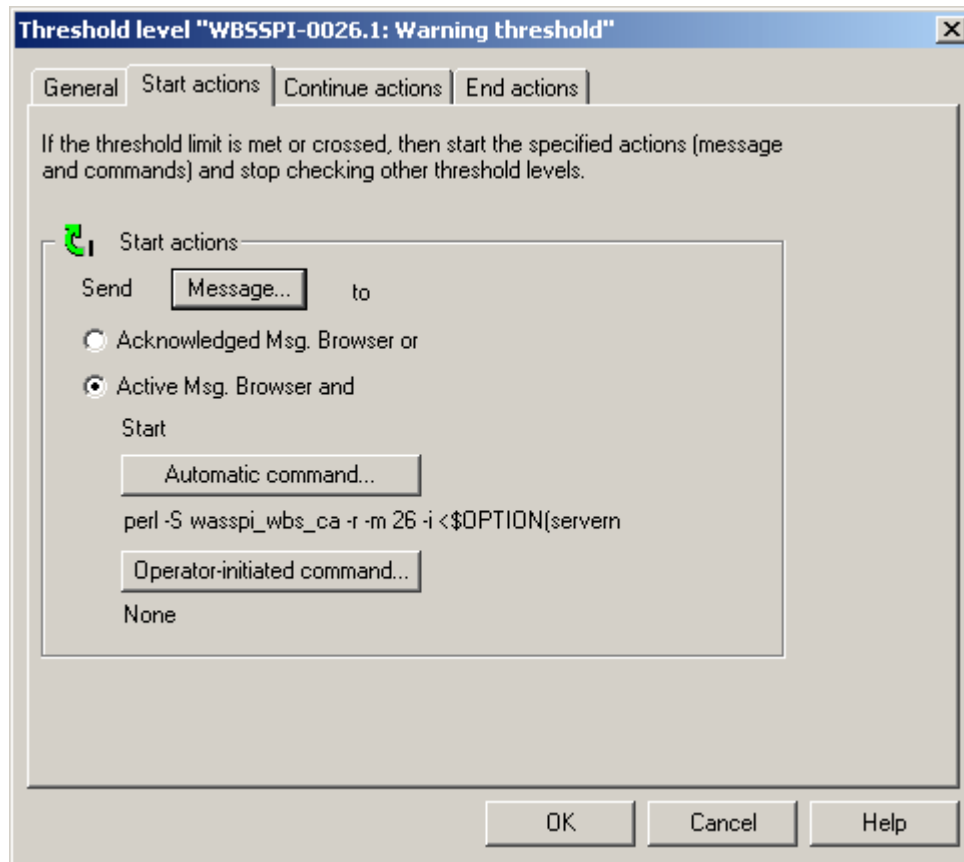


In the figure above, the threshold limit is set to 1000 for WBSSPI-0026. The incoming values for this metric show the total number of times per minute clients must wait for an available Enterprise Java bean. If the number exceeds 1000 the server response time slows down. This generates a Warning message.

You can modify the following attributes from the Threshold Level window:

- **Threshold limit.** If the threshold limit is met or crossed the WebSphere SPI triggers an alarm or message.
- **Short-term peaks.** A minimum time period over which the monitored value must exceed the threshold before generating a message. For a message to be sent, the value must be greater than the threshold each time the value is measured during a duration that you select. If the duration is set to 0 or the box is left empty, an alarm is generated as soon as HPOM detects that the threshold has been equaled or crossed.
- **Reset.** A limit below which the monitored value must drop or exceed (for minimum thresholds) to return the status of the monitored object to normal.

As the following figure shows, the Threshold Level window has the following three action tabs. You can click any of the action tabs to set the related actions.



- *Start actions*: Actions carried out the first time that the threshold is crossed
- *Continue actions*: Actions carried out at each subsequent polling interval if the reset value is not reached.
- *End actions*: Actions carried out after the threshold crosses the reset value.

In each of the actions tabs, you can set the type of actions to perform.

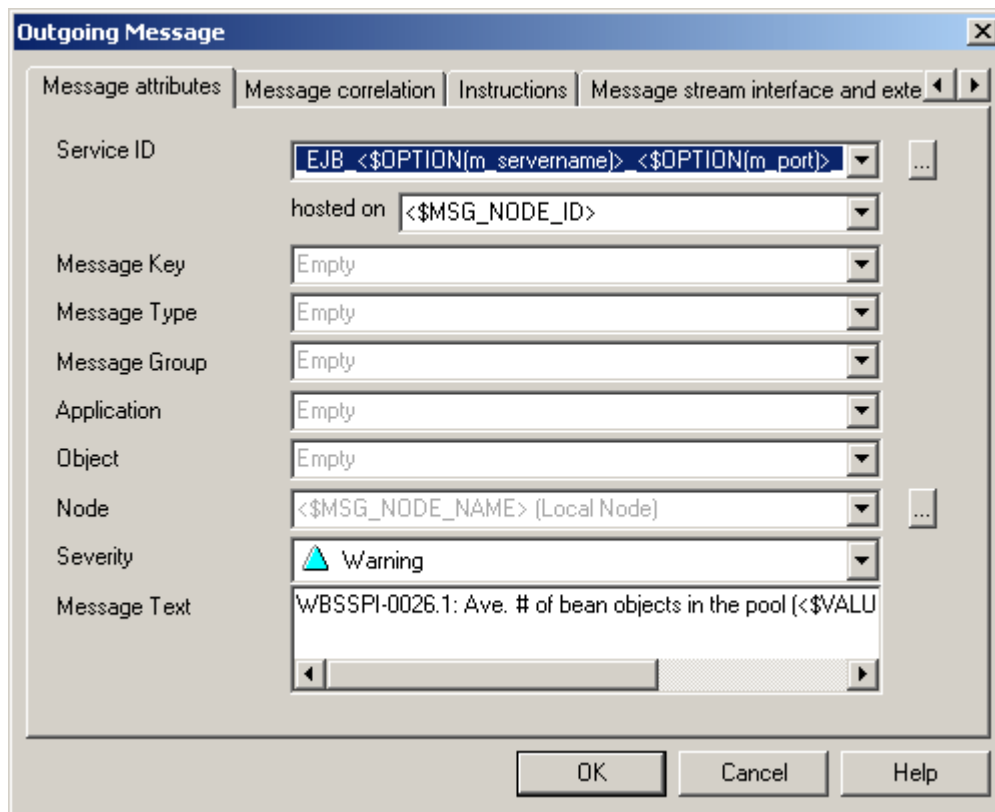
The WebSphere SPI provides the ability to generate graphs and reports, or to add custom programs. You can generate the reports and graphs through:

- *Automatic command*: An automatic command runs when the rule is matched. The automatic command that is delivered with the WebSphere SPI generates a snapshot report that shows the data values at the time the action was triggered because of an exceeded threshold. You can view the report in the message annotations.
- *Operator-initiated command*: An operator-initiated command is attached to the message that the rule sends to the message browser. You can run this command from the message browser. The operator-initiated command delivered with the WebSphere SPI lets you click **Perform Action** in the Message Properties window to view a graph of the metric whose exceeded threshold generated the message, along with other related metric values.

Message and Severity

To modify the message text and severity of a policy, follow these steps:

- 1 From the HPOM console select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere** → *<impact>*-**Impact** → **Metrics**.
- 2 Double-click a policy for which you want to modify the severity and message text. The Measurement Threshold window opens.
- 3 Select the **Threshold levels** tab.
- 4 Double-click the threshold level description (for example, WBSSPI-0071.1: Warning threshold). A new window opens. Click the **Start Actions** tab.
- 5 Click **Message**. The Outgoing Message window opens.
- 6 Click the **Message Attributes** tab and make the necessary modifications. Click **OK**.



The screenshot shows the 'Outgoing Message' dialog box with the 'Message attributes' tab selected. The dialog has a title bar with a close button (X) and a tabbed interface with 'Message correlation', 'Instructions', and 'Message stream interface and exte'. The 'Message attributes' tab contains the following fields:

- Service ID: EJB <\$OPTION(m_servername)> <\$OPTION(m_port)>
- hosted on: <\$MSG_NODE_ID>
- Message Key: Empty
- Message Type: Empty
- Message Group: Empty
- Application: Empty
- Object: Empty
- Node: <\$MSG_NODE_NAME> (Local Node)
- Severity: Warning (indicated by a blue triangle icon)
- Message Text: WBSSPI-0026.1: Ave. # of bean objects in the pool (<\$VALU

At the bottom of the dialog are three buttons: OK, Cancel, and Help.

In the Outgoing message defaults window you can modify the following attributes:

- *Severity*: Indicates the importance (severity) of the event that triggers this message.
 - *Message Text*: You can modify the text of the message but do *not* modify any of the parameters—beginning with \$ and surrounded by <> brackets—in a message.
- 7 Click **Save and Close** in the policy window to save the changes and exit.

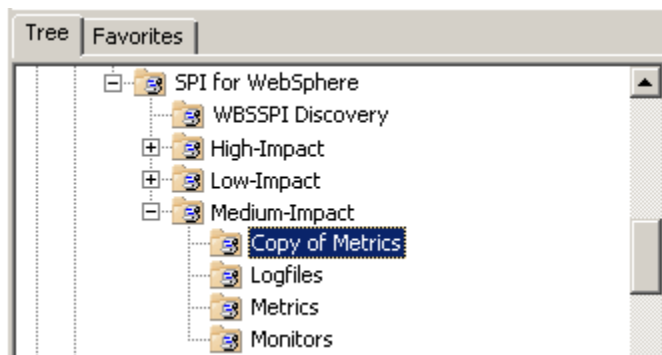
Advanced Policy Customizations

Advanced Policy customizations include making copies of default policy groups to customize a few settings and deleting whole groups of metrics within a policy's command line.

Creating New Policy Group

You can keep the custom policies that you create separate from the original default policies by creating new policy groups. Before you create a new policy group you must first determine the metrics and policies you want to modify. To create a new policy group, follow these steps:

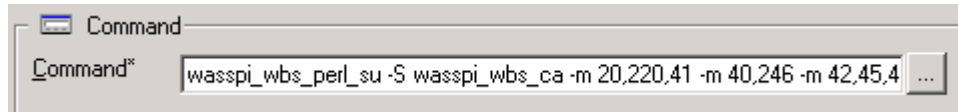
- 1 Create a new policy group:
 - a In the HPOM console select **Operations Manager** → **Policy management** → **Policy groups**.
 - b Right-click the policy group you want to copy and select **Copy**.
For example, right-click the **Metrics** policy group under Medium-Impact and select **Copy**.
 - c Right-click the group under which this policy group is located and select **Paste**.
For example, right-click **Medium-Impact** and select **Paste**.
 - d Right-click the new group and select **Rename**.
For example, right-click **Copy of Metrics** and select **Rename**.
 - e Type a new name in the box.



- 2 Rename the original policies within the new policy group:
 - a Double-click the new policy group to get a list of the policies.
 - b Double-click a policy. The policy window opens.
 - c Click **File** → **Save As**. The Save As window opens.
 - d Type a new policy name and click **OK**.
 - e Click **File** → **Exit** to close the policy window.
- 3 Delete all original policies within the new policy group. To do this, select the policies and press the **Delete** key. The Confirm multiple delete window opens.
Click **Yes** to confirm deletion; otherwise click **No**.
- 4 Alter the renamed policies within the new group as necessary.

WebSphere SPI Collector/Analyzer Command with Parameters

The `wasspi_wbs_perl -S wasspi_wbs_ca` command is used in every collector policy. You can view the default command line parameters within each collector policy in the Command box in HPOM console. Double-click the policy to open the policy window, the Command box is a part of this window



Basic Collector Command Parameters

The `wasspi_wbs_ca` command is required to start the WebSphere SPI data collection. The following table lists the parameters used by the default collector policies.

Parameter	Description	Syntax with Example
<code>-e</code>	(exclude) Allows you to exclude specific servers; may not be used with <code>-i</code> option.	Syntax: <code>-e <server_name></code> Example: <code>-e server1, server3</code>
<code>-i</code>	(include) Allows you to list specific servers to monitor. This option may not be used with <code>-e</code> option.	Syntax: <code>-i <server_name></code> Example: <code>-i server1, server3</code>
<code>-m</code>	(metric) Specifies the metric numbers or number ranges on which to collect data.	Syntax: <code>-m <metric_number,metric_number_range></code> Example: <code>-m 1, 3-5, 9-11, 15</code>
<code>-matchver</code>	(match version) Specifies the WebSphere application server version to monitor. This option may not be used with the <code>-minver</code> nor <code>-maxver</code> options. If no matching versions are found, the command does not run.	Syntax: <code>-matchver <version_number></code> Example: <code>-matchver 4</code>
<code>-maxver</code>	(maximum version) Specifies the highest WebSphere application server version to monitor. Use with <code>-minver</code> to specify a range of version. If no versions are found, the command does not run.	Syntax: <code>-maxver <version_number></code> Example: <code>-maxver 5</code>
<code>-minver</code>	(minimum version) Specifies the lowest WebSphere application server version to monitor. Use with <code>-maxver</code> to specify a range of version. If no versions are found, the command does not run.	Syntax: <code>-minver <version_number></code> Example: <code>-minver 4</code>

Parameter	Description	Syntax with Example
-r	(report) Generate an ASCII report for the specified metric(s).	Syntax: -r
-t	(tag) Allows you to create a new policy group by adding a prefix to an existing collector policy along with the metric numbers.	Syntax: wasspi_wbs_perl -S wasspi_wbs_ca -m <metric_number> -t <prefix>- Example: wasspi_wbs_perl -S wasspi_wbs_ca -m 220-223 -t DEV-
-x	Allows you to specify a property and value. Syntax: -x <property>=<property_value> where <property> can be one of the following: <ul style="list-style-type: none"> alarm: When off, overrides any default alarming defined for the metric. Example: -x alarm=off prefix: Default: JMXUDM_. Specify the prefix of the metric ID. Example: -x prefix=SALES_ print: When on, prints the metric name, instance name, and metric value to STDOUT in addition to any configured alarming or logging. Example: -x print=on graph: When off, prevents graphing function. Example: -x graph=off report: When off, prevents reporting function. Example: -x report=off 	

Examples

- To collect specific data on all configured servers:

```
wasspi_wbs_perl -S wasspi_wbs_ca -m 10-14,25,26
```
- To collect data from specific servers only:

```
wasspi_wbs_perl -S wasspi_wbs_ca -m 245,246,260 -i server1,server2
```
- To not collect data from specific servers:

```
wasspi_wbs_perl -S wasspi_wbs_ca -m 220-225 -e server1,server2
```

Using JMX Actions Command Parameters

This section describes the command parameters you can use to run JMX actions. JMX actions are one or more JMX calls (invoke, get, set) performed on an MBean instance or type. A single JMX call can be performed from the command line. Multiple JMX calls can be specified in an XML file or as a Metric sub-element in a UDM file.

Parameter	Description	Syntax with Example
-a Required	(action) Indicates a JMX action is performed.	Syntax: -a
-i	(include) Allows you to list specific servers on which to perform the JMX actions. If this parameter is not specified, the JMX actions are performed on all configured servers.	Syntax: -i <server_name> Example: -i server1,server3
-m	(metric) Specifies the metric ID containing the action to perform. This metric ID must be defined in a UDM file. This option may not be used with the -mbean or -xml options.	Syntax: -m <metric_id> Example: -m TestUDM_1000

Parameter	Description	Syntax with Example
-mbean	<p>Performs a JMX call on the specified MBeans. This option may not be used with the -m nor -xml options.</p> <p>Syntax: -mbean <objectname> <action></p> <p>Example: -mbean WebSphere:type=ThreadPool,* -get maximumSize</p> <p>where <action> (a JMX call) is one of the following:</p> <ul style="list-style-type: none"> -get: Returns the value of the a specified attribute. <p>Syntax: -mbean <objectname> -get <attribute></p> <p>Example: -get maximumSize</p> -invoke [-type]: Executes an MBean operation with the specified parameters. An operation may not require parameters (therefore, -type is not specified). A type parameter must be specified for operations which accept parameters. -type supports operation overloading. <p>Syntax: -mbean <objectname> -invoke <operation> [-type <parameter_type> <parameter_value>]...</p> <p><parameter_type> is one of the following: short, int, long, double, float, boolean, java.lang.Short, java.lang.Integer, java.lang.Long, java.lang.Double, java.lang.Float, java.lang.Boolean, and java.lang.String.</p> <p>Example: -invoke setInstrumentationLevel -type java.lang.String pmi=L -type boolean true</p> -set: Assigns the specified value to the specified attribute. <p>Syntax: -mbean <objectname> -set <attribute> <value></p> <p>Example: -set growable true</p> 	
-o	(object) Specifies an MBean instance.	<p>Syntax: -o <mbean_instance></p> <p>Example: -o exampleJMSSEServer</p>
-xml	Specifies the XML file that contains one or more JMX actions to perform. This option may not be used with the -m or -mbean options.	<p>Syntax: -xml <filename></p> <p>Example: -xml myJMXActions.xml</p>

Examples

- Set the maximum size for an alarming thread pool to 500 (<\$OPTION(instanceName)> specifies an alarming instance):

```
wasspi_wbs_perl -S wasspi_wbs_ca -a
-mbean WebSphere:type=ThreadPool,* -set maximumSize 500 -o
<$OPTION(instanceName)>
```

- Set the instrumentation levels to low on all PMI modules:

```
wasspi_wbs_perl -S wasspi_wbs_ca -a
-mbean WebSphere:type=Perf,* -invoke setInstrumentationLevel
-type java.lang.String pmi=L
```

- Use the sample UDM TestUDM_1000 in the `wbs_UDMMetrics-sample.xml` file:
`wasspi_wbs_perl -S wasspi_wbs-ca -a -m TestUDM_1000 -i <Servername>`

- Use the sample actions xml file:

```
wasspi_wbs_perl -S wasspi_wbs-ca -a
-xml /<wasspi_wbs_conf_dir>/JMXActions-sample.xml
-i <Servername>
```

Where, `<wasspi_wbs_conf_dir>` is `var/opt/OV/wasspi/wbs/conf` for DCE agent and `/var/opt/OV/conf/wbsspi` for HTTPS agent.

Changing the Collection Interval for Scheduled Metrics

You can change the collection interval for all scheduled metrics by changing the Polling Interval in the respective collector policy. For example, to change the collection interval of default metrics from 5 minutes to 10 minutes for the Medium-Impact policy WBSSPI-40-Med-05min, follow these steps:

- 1 From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere** → **Medium-Impact** → **Monitor**.
- 2 Right-click the collector policy WBSSPI-40-Med-05min and select **All Tasks** → **Edit**. The Measurement Threshold window opens.
- 3 Click **File** → **Save As**. The Save As window opens.
- 4 Change the existing name in the Name box to WBSSPI-40-Med-10min. Click **OK**.
- 5 Set the new interval.
 - a Click the **Schedule** tab.
 - b From the Schedule Task drop-down list select “Once per interval”.
 - c Set the interval to 10 minutes.
- 6 Click **Save and Close**.
- 7 Deploy the new policies.
 - a Right-click **WBSSPI-40-Med-10min** and select **All Tasks** → **Deploy on....**
 - b Select the nodes on which to deploy the policy.
 - c Click **OK**.

Changing the Collection Interval for Selected Metrics

You can change the collection interval of metrics, according to the requirements of your environment. For example, you can change the collection interval from 5 minutes to 10 minutes for metrics 72-73 of the collector policy WBSSPI-40-Med-05min. You can change the collection interval of any metric using these steps.

To change the collection interval, follow these steps:

- 1 Rename the selected metrics to reflect the new interval.
 - a From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere** → **Medium-Impact** → **Monitor**.
 - b Right-click the collector policy WBSSPI-40-Med-05min and select **All Tasks** → **Edit...** The policy window opens.

- c Click **File** → **Save As**. The Save As window opens.
 - d In the Name box change the existing name to **WBSSPI-40-Med-10min**. Click **OK** to confirm save or click **Cancel** to discard changes.
- 2 In the Command box, delete all metrics after the `-m` except `72-73`.
 - 3 Set the new interval.
 - a Click the **Schedule** tab.
 - b From the Schedule Task drop-down list select “Once per interval”.
 - c Set the interval to 10 minutes.
 - d Click **Save and Close** to save the changes.
 - 4 Edit the original policy to remove the modified metrics.
 - a Right-click the policy **WBSSPI-40-Med-05min** and select **All Tasks** → **Edit**. The policy window opens.
 - b In the Command text box, delete metrics `72-73` after `-m`.
 - c Click **Save and Close** to save the changes.
 - 5 Deploy the modified policies.
 - a Right-click **WBSSPI-40-Med-10min** and select **All Tasks** → **Deploy on....**
 - b Select the nodes on which you want to deploy the policy.
 - c Click **OK**.
 - d Right-click **WBSSPI-40-Med-05min** and repeat steps b-d.

Customizing the Threshold for Different Servers

You can set different threshold values for the same metric (by modifying the related metric policy) on different servers according to your needs. For example, you may want to set the threshold for metric `0212` at 100 for `SERVER_1` but let the threshold be 90 for all other servers. To do this you can copy the existing condition and modify it to serve as the exception. Follow these steps:

- 1 Double-click the metric policy (for example, double-click `WBSSPI-0212`). The Measurement Threshold window opens.
- 2 Select the **Threshold levels** tab.
- 3 Select the desired condition and click **Copy** to make a copy of the condition.
- 4 Name the condition– `WBSSPI-0212 . 2`.
- 5 Click **Specify instance filters...** The New Rule window opens.
- 6 Select the **Condition** tab and in the Object Pattern field, enter the following details:

```
<ServerName.var1>:<ServerPort.var2>:<NodeName.var3>:<*.var4>:<*.var5>:<*.var6>
```

For Example: If you want to set threshold for the application server `SERVER_1`, enter the following:

```
SERVER1:<*.var2>:<*.var3>:<*.var4>:<*.var5>:<*.var6>
```

`var1`, `var2`, `var3`, `var4`, `var5`, and `var6` are user defined variables. These variables must be different from the HPOM policy variables.

- 7 Click **OK**.

- 8 Double-click the condition **WBSSPI-0212.2**. The Threshold Level window opens.
- 9 Change the threshold limit to 100. Click **OK**.
- 10 In the Measurement Threshold window, click **Save and Close** to save the changes and exit.

Creating Custom, Tagged Policies

You can customize a policy by using the tag option (`-t` on the command line) that allows the collector/analyzer to recognize customized policies that have a tag attached to the name. This option gives you the flexibility of using more than a single set of policies to define conditions related to specific installations of WebSphere Application Server. It also preserves policies from being overwritten when an upgraded version of the WebSphere SPI is installed.

When multiple nodes are managed by a number of groups, you can use this option to create specially tagged policies that are separate from your original setup. In such a case, make copies of the policies, rename them with the tag, rework the collector policy to pick up the tagged names, and then assign them to various groups.

For example, you may create a group of policies and change each policy name to include **CLIENT01** in it. You may name a metric policy as **CLIENT01-WBSSPI_0212** (retaining the name of the metric used). You may name the collector policy as **FIRST_CLIENT-40_05min**. Similarly, you may set up another group for **SECOND_CLIENT** and modify the policy names to include **SECOND_CLIENT**.

To create the new policy group, follow these steps:

- 1 Copy the original policy group.
 - a Right-click the policy group you want to copy and select **Copy**.
For example, right-click the Metrics policy group under High-Impact and select **Copy**.
 - b Right-click the group under which this policy group is located and select **Paste**.
For example, right-click High-Impact and select **Paste**.
 - c Right-click **Copy of Metrics** and select **Rename**. Rename the new group to identify the new metric and collector policies.
For example, rename the group to **CLIENT01HighImpactMetrics**.
- 2 Rename the original policies within the new policy group.
The names of the metric policies in the new group must contain the new name followed by the original metric number. For example, you can rename a copy of **WBSSPI_0001** as **CLIENT01-WBSSPI_0001**.

The name you give to the new collector policy must also contain the identifying name. You must also modify the scheduled collection to include the new group by inserting the `-t` property in the Command box. The Command box is in the policy window that appears when you double-click the collector policy.

For example: `wasspi_wbs_ca -m 16 -t CLIENT01-`

- a Right-click the policy and select **All Tasks** → **Edit**. The policy window opens.
 - b Click **File** → **Save As**. The Save As window opens.
 - c Type a new policy name and click **OK**.
- 3 Select the original policies within the new policy group and press the **Delete** key to delete all the original policies. The Confirm Multiple Item Delete window opens.
 - 4 Click **Yes** to confirm delete.

Restoring Default WebSphere SPI Policies

To restore the default WebSphere policy groups on your management server, you must remove and then reinstall WebSphere SPI. For more information, see [Removing WebSphere SPI](#) on page 13 and [Installing the WebSphere SPI](#) on page 11.

Viewing Text-Based Reports

Some policies have actions defined with threshold violations or error conditions. These actions automatically generate reports. The reports are snapshots of data values collected from the server around the time that the alarm occurred.

- ▶ The reports discussed in this section are different from HP Reporter reports that show consolidated data generated as web pages in a management-ready presentation format. See [Integrating the WebSphere SPI with HP Reporter](#) on page 52.

Automatic Command Reports

Many metrics generate Automatic Command Reports. These reports are generated as soon as an alarm is triggered in HPOM. Automatic Command reports are generated for a single WebSphere Application Server instance with the exceeded threshold.

When an Automatic Command report is executed from HPOM, the server is queried for additional data. If you set the HPOM console message browser to display the SUIAON column, you can see an “S” under the “A” column (see the following figure), which indicates that a generated report is available in the Annotations area of the Message Properties.

Severity	S	U	I	A	O	Received	Group	
Normal	-	-	X	-	-	1/27/2003 3:21:42 PM	OpC	
Normal	-	-	X	-	-	1/27/2003 3:21:42 PM	OpC	
Normal	-	-	X	-	-	1/27/2003 3:21:42 PM	OpC	
Critical	-	-	X	-	-	1/27/2003 3:21:42 PM	OpC	
Critical	O	-	X	S	S	X	1/27/2003 3:21:43 PM	WebSphere
Normal	-	-	X	-	-	1/27/2003 3:21:43 PM	OpC	
Normal	-	-	X	-	-	1/27/2003 3:21:43 PM	OpC	

Summary: 1156 Critical, 0 Warning, 0 Error, 13 Info, 263 Normal, 0 Unknown, 0 Disabled, 0 Unavailable.

To view Automatic Command reports, do one of the following:

- Double-click a message in the HPOM message browser. The Message Properties window opens. Select the Annotations tab.
- Right-click a message and select **Annotations**. The Message Properties window opens.

The reports are available in the Message Properties window. These reports show data values of a single server. Column descriptions in the window provide further information.

Manually Generated Reports

Reports are generated for all WebSphere Application Server instances configured on the managed node. In contrast to Automatic Command reports that are generated for a single WebSphere Application Server instance, manually generated reports reflect the current state of all WebSphere Application Server instances on the managed node.

To manually generate a report, follow these steps:

- 1 From the HPOM console, select **Operations Manager** → **Tools** → **SPI for WebSphere** → **Metric Reports**.
- 2 Double-click the report you want to see. The Select Where to Launch This Tool window opens.
- 3 Select the managed node for which you want to see reports and click **Launch**. The Tool Status window opens.
- 4 View the report in the tool output field.
- 5 Click **Close** to close the window.

Performance Impact Ratings (PMI Levels) of Reporting Metrics

Low	5, 42, 222, 224, 247, 265
Medium	40, 221, 246, 262
High	41, 212, 213, 220, 261, 263, 264

Figure 1 The report generated for Metric 5 (I005)

```
Report for Application Server: Default server
Jan 29, 2003 11:25:50 AM
Metric I005_JVMMemUtilPct

Java Virtual Machine  Total Heap Memory  Free Heap Memory  Used Heap Memory
-----
jvmsRuntimeModule    23,842,816.0      17,217,696.0      6,625,120.0

Java Virtual Machine Profile
-----
No data available
```

WebSphere SPI Graphs

Some policies have operator actions associated with them that allow you to generate a graph. To view these graphs, follow these steps:

- 1 Double-click a message in the HPOM message browser. The Message Properties window opens.
- 2 Click the **Commands** tab. You can generate a graph if an operator-initiated command is configured and data is collected.
- 3 Click **Start** to generate the graph.

Monitoring WebSphere Application Server on Unsupported Platforms

The WebSphere SPI supports monitoring WebSphere systems running on HP-UX, Solaris, AIX, Windows 2000 and 2003, Red Hat Linux, and Suse Linux. It is also possible to configure the WebSphere SPI to monitor WebSphere systems running on unsupported platforms—systems we see as “remote systems.”

This section explains how to determine if your environment is favorable to setting up remote monitoring. If you determine that your environment meets the criteria described below, and you have some expertise in using the WebSphere SPI, this section offers an example to get you started.

Monitoring Remote Nodes (Running on Platforms Not Supported by WebSphere SPI)

For a WebSphere system running on an unsupported platform, you can use WebSphere SPI to monitor that remote system if the following conditions apply. The last condition is optional:

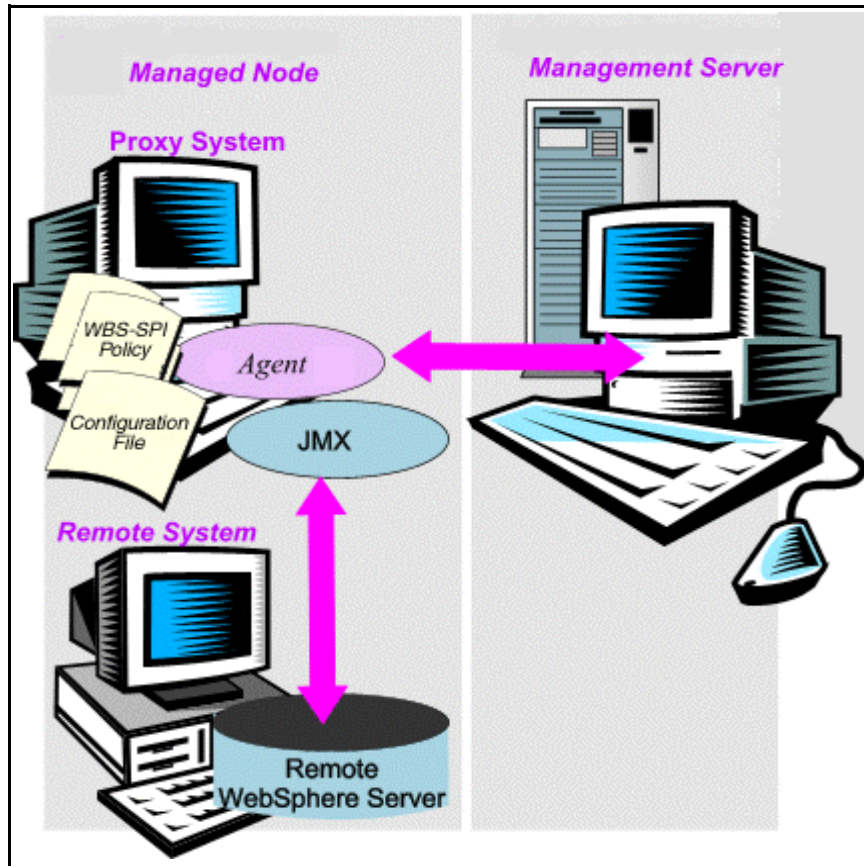
- The remote system is covered by a purchased license (using Tier 1 pricing).
- The WebSphere SPI runs on at least one managed node on a supported platform: HP-UX, Solaris, AIX, Windows 2000, Windows 2003, Red Hat Linux, or Suse Linux.
- The local/proxy system and remote system must be running the same version of WebSphere Server. For example, if the proxy system is running WebSphere Server version 5, the remote system must also be running WebSphere Server version 5.
- (Optional, for logfile monitoring) The remote system runs on a platform supported by the HP Operations agent software.

Implementing Remote Monitoring

In a standard configuration, WebSphere SPI programs or policies are deployed on the local, managed node. In a non-standard configuration, the local system is used as a proxy through which remote metric information becomes accessible.

Remote system data collection and interpretation relies on the local, managed node to act as the proxy on which data collection is configured.

In the following figure, the ‘Agent’ is HP Operations agent.



Configuration entries requirement:

Within the configuration, entries for both local and remote systems are included. You can include multiple remote system entries in a local system’s section. [Example Configuration](#) on page 46, shows how the remote entry appears with system IP address.

Policy deployment requirement:

Policies for the correct WebSphere PMI level should be deployed on the local node. If you need a separate policy group (for example, High-Impact or Medium-Impact) to cover a different level, you can copy and rename the existing policies and specify the WebSphere Server name on the command line using the `-i` or `-e` options. For more information on these command line parameters, see [WebSphere SPI Policy Group and Types](#) on page 27.

HP Operations agent deployment requirement (optional logfile monitoring):

To access remote WebSphere logfiles, the HP Operations agent software must be installed on the remote system. Using standard HPOM processes, you can modify the standard logfile policies included with the WebSphere SPI to specify the correct logfile names, then deploy them to the remote system.



Monitoring remote systems using logfile versioning is not supported

Configuring Remote System Monitoring

You can monitor WebSphere Application Servers on remote systems (running on operating systems other than HP-UX, Solaris, AIX, Windows 2000, Windows 2003, Red Hat Linux, or Suse Linux) by completing the following tasks.

Task 1: Configure the Remote WebSphere System

Using the Configure WBSSPI tool of the SPI Admin tools group, configure each local managed node that communicates with a remote WebSphere server. In the configuration, add entries for remote WebSphere servers.

- 1 Launch the Configure WBSSPI tool. For details, see the Tools section in the WebSphere SPI online help.
- 2 Select a WebSphere managed node from which to monitor the remote WebSphere server.
- 3 In the configuration, include an entry for each remote WebSphere system at the server-specific level:

`ADDRESS=<DNS server name or IP address>.`

The example configuration below shows how local and remote WebSphere servers are configured in the same file. For the remote servers the `ADDRESS=<IP_address>` line is added:

```
ADDRESS=15.75.27.109 or
ADDRESS=harley.hp.com
```

Example Configuration

```
#
#####
HOME=C:/WebSphere/AppServer
JAVA_HOME=C:/WebSphere/AppServer/java

SERVER1_NAME=classact
SERVER1_PORT=900

SERVER2_NAME=harley
SERVER2_PORT=901
SERVER2_ADDRESS=harley.hp.com
```

In this example, `SERVER1` is the local server, running on a Windows managed node. `SERVER2` is running on an HPOM managed node that is a system on a platform unsupported by WebSphere SPI. The remote system is configured similar to that of the local system but contains the new line `SERVER2_ADDRESS=harley.hp.com`.

Task 2: (Optional) Integrate HP Performance agent

The HP Performance agent collection occurs on the managed node, not the remote system. Therefore, if you use HP Performance Manager and want to graph the remote system data, ensure that HP Performance agent integration is enabled on the local managed node.

Task 3: Deploy Policies to the Local Node

Deploy a policy group to the local managed node. For example, deploy the High-Impact policy group on the local node if the local and remote managed nodes are to collect metrics that require the system be set at a high WebSphere PMI level.

- 1 From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere**.
- 2 Right-click a policy group and select **All Tasks** → **Deploy on**.
- 3 Select the local managed node.
- 4 Click **OK**.

Configuring Remote Monitoring for Logfile (Optional)

Monitoring remote system logfiles is supported if the following are true:

- 1 The HP Operations agent is running on the remote system.
- 2 The system does not re-version logfiles when they roll.

To set up logfile monitoring, in the HPOM console, copy the WebSphere SPI logfile policy and then configure, assign, and deploy the copied logfile policy to the remote system.

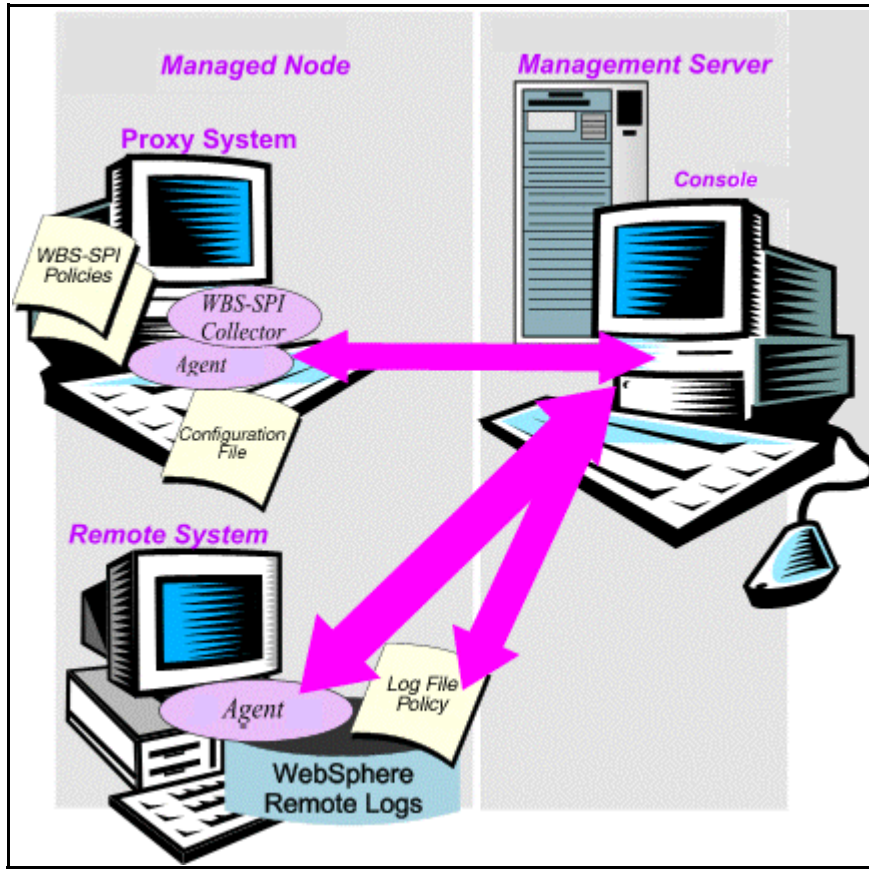
Configuring the Logfile Policy for Remote Logfiles

To configure the logfile policy for remote logfiles, follow these steps:

- 1 From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere** → **High-Impact**.
- 2 Select **Logfiles** and double-click the log policy.
- 3 In the Logfile pathname box, type the location of the logfile on the remote system:
/<path>/<file_name>.
- 4 Assign and deploy the logfile policy to the remote HPOM managed node.

WebSphere logfile monitoring is possible because of the Logfile policy and the HP Operations agent present on the remote system.

In the following figure 'Agent' is the HP Operations agent and the Console is HPOM Console.



Limitations in Remote Monitoring

- The WebSphere SPI and the HP Operations agent do not support access to logfiles that are re-versioned each time the logs are rolled.
- WebSphere logfiles on the remote system cannot be monitored if an HP Operations agent is not present on the remote system.
- You cannot run WebSphere SPI tools on remote systems.
- Same version of WebSphere Server must be running on both the proxy system and remote system.

5 Integrating HPOM Reporting and Graphing Features with the WebSphere SPI

The WebSphere SPI can be integrated with the following HP products. These products must be purchased separately.

- **HP Reporter**

Reporter produces management-ready, web page reports, showing historical and trends related information.

After you integrate HP Reporter with WebSphere SPI, Reporter generates a variety of reports, every night, that show consolidated information about the performance and availability of WebSphere Application Servers on configured managed nodes. See [Integrating the WebSphere SPI with HP Reporter](#) on page 52.

- **OpenView Performance Insight**

OpenView Performance Insight is a network management system that collects, processes, and reports data. This data is used to generate reports. For more information, see the *HP OpenView Performance Insight Administration Guide*.

For information on WebSphere SPI reports and integrating WebSphere SPI with OVPI, see the *Application Server Report Pack User Guide*.

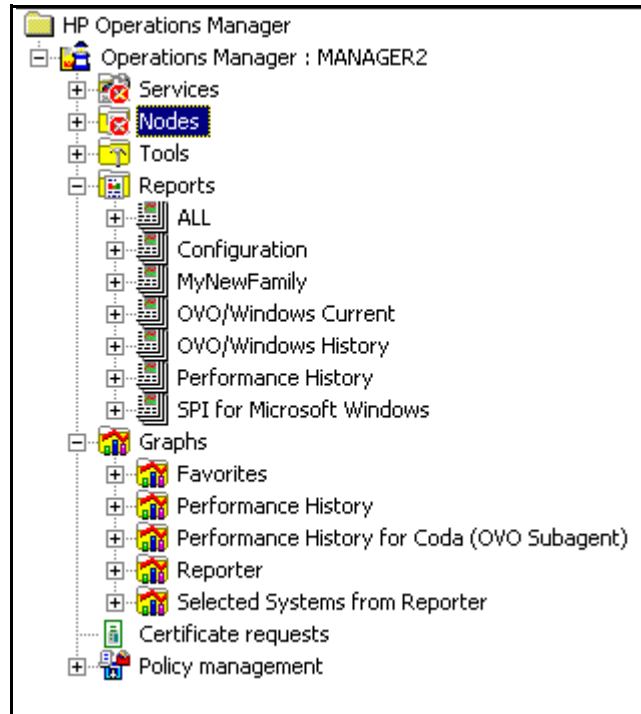
- **HP Performance Manager**

HP Performance Manager provides graphing capability.

After you integrate HP Performance Manager with the WebSphere SPI, you can view the graphs the following day. However, the graphs are available only if performance data is logged in the default performance subagent CODA or HP Performance agent. CODA is automatically deployed on all HPOM managed nodes.

See [Integrating the WebSphere SPI with HP Performance Manager](#) on page 59.

Figure 2 The Management Server Console Tree



Integrating the WebSphere SPI with HP Performance Agent

If your IT environment requires you to generate graphs and reports from historical data or to store large volumes of performance data, you may want to use the HP Performance agent to collect and store performance data. HP Performance agent is a product that must be purchased separately.

The data collected by HP Performance agent is used by Reporter, HP Performance Insight, and HP Performance Manager. The reporting and graphing features integrated with HPOM for Windows cannot use data collected by HP Performance agent and therefore can not work if you use Performance Agent.



If you are running HP Performance agent 4.x for Linux, you are not required to configure the WebSphere SPI data collector to use HP Performance agent. By default, the WebSphere SPI detects and uses this version of HP Performance agent to collect and store performance data.

To configure the WebSphere SPI data collector to use HP Performance agent, follow these steps:

- 1 Create a `nocoda.opt` file on the managed node, in the following directory:

Operating System	File Location
HP-UX, Linux, or Solaris	<code>/var/opt/OV/conf/dsi2ddf/</code>
AIX	<code>/var/lpp/OV/conf/dsi2ddf/</code>
Windows	<code>C:\Program Files\HP Openview\data\conf\dsi2ddf\</code>


If the directory `dsi2ddf` does not exist, create it.

- 2 Edit the `nocoda.opt` file to contain a single line:
ALL
- 3 Save the file.

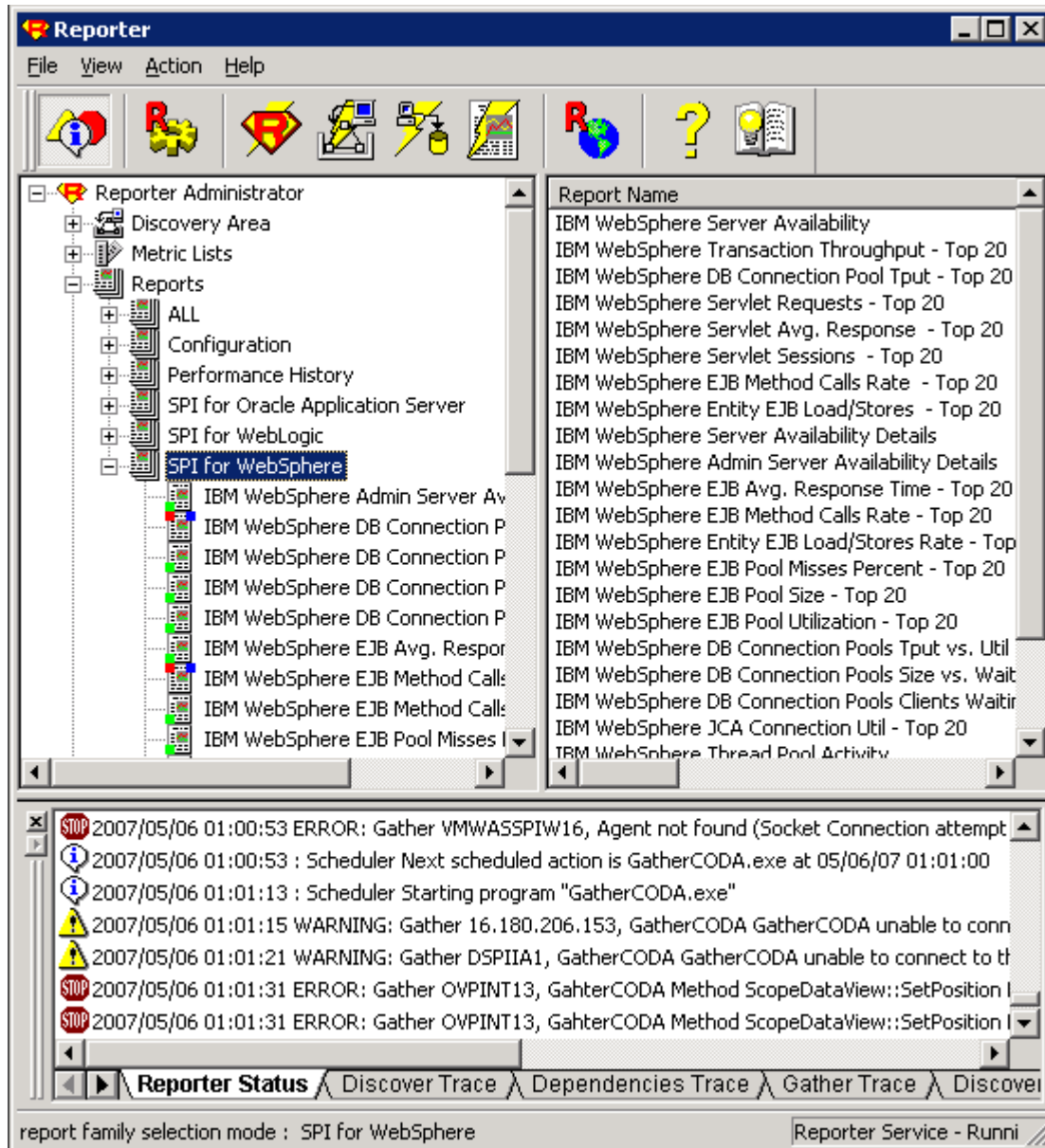
Integrating the WebSphere SPI with HP Reporter

Before integrating the WebSphere SPI with HP Reporter, you must configure the WebSphere SPI by deploying the software, configuring server connection, and assigning/deploying policies on target managed nodes.

To integrate the WebSphere SPI with Reporter, follow these steps:

- 1 Install the WebSphere SPI report package on the Windows system running HP Reporter.
 - a Insert the HP Operations Smart Plug-ins DVD into the DVD drive of the Windows system running Reporter. The HP Operations Manager InstallShield Wizard opens.
 - b Click **Next**. The Program Maintenance window opens. Click **Install Products**. The Product Selection window opens.
 - c From the options listed (there are three Product Selection windows), select the **Reporter** option of IBM WebSphere and click **Next**.
 - d Complete the installation by following the instructions that appear as you proceed.
 -  On Windows 2000 managed nodes, when installing the WebSphere SPI report package, you may get an error message indicating that the installer has detected an older version of the installer on your system. You can safely ignore the message and continue.
- 2 To see the Reporter window, click **Start** → **All Programs** → **HP OpenView** → **Reporter** → **Reporter**.
- 3 Check the Reporter window (see the illustration that follows) to note changes in the Reporter's configuration

In the Reporter status pane (at the bottom of the Reporter window), you can view information on programs that are running and any errors occurring on the managed nodes. You can check the status pane to see whether Reporter is updated with the WebSphere SPI reports.



In the Reporter Help, you can find instructions for assigning WebSphere SPI reports to the target nodes. To access Help, follow these steps:

- a Right-click **Reports** or **Discovered Systems** in the left panel of the Reporter main window.
 - b Select **Report Help** or **Discovered Systems Help**.
 - c Read the topic - To assign a report definition to a Discovered Systems Group.
- 4 Add group and single system reports by assigning reports as desired. For complete information, see the Reporter Help and the online *Concepts Guide*.

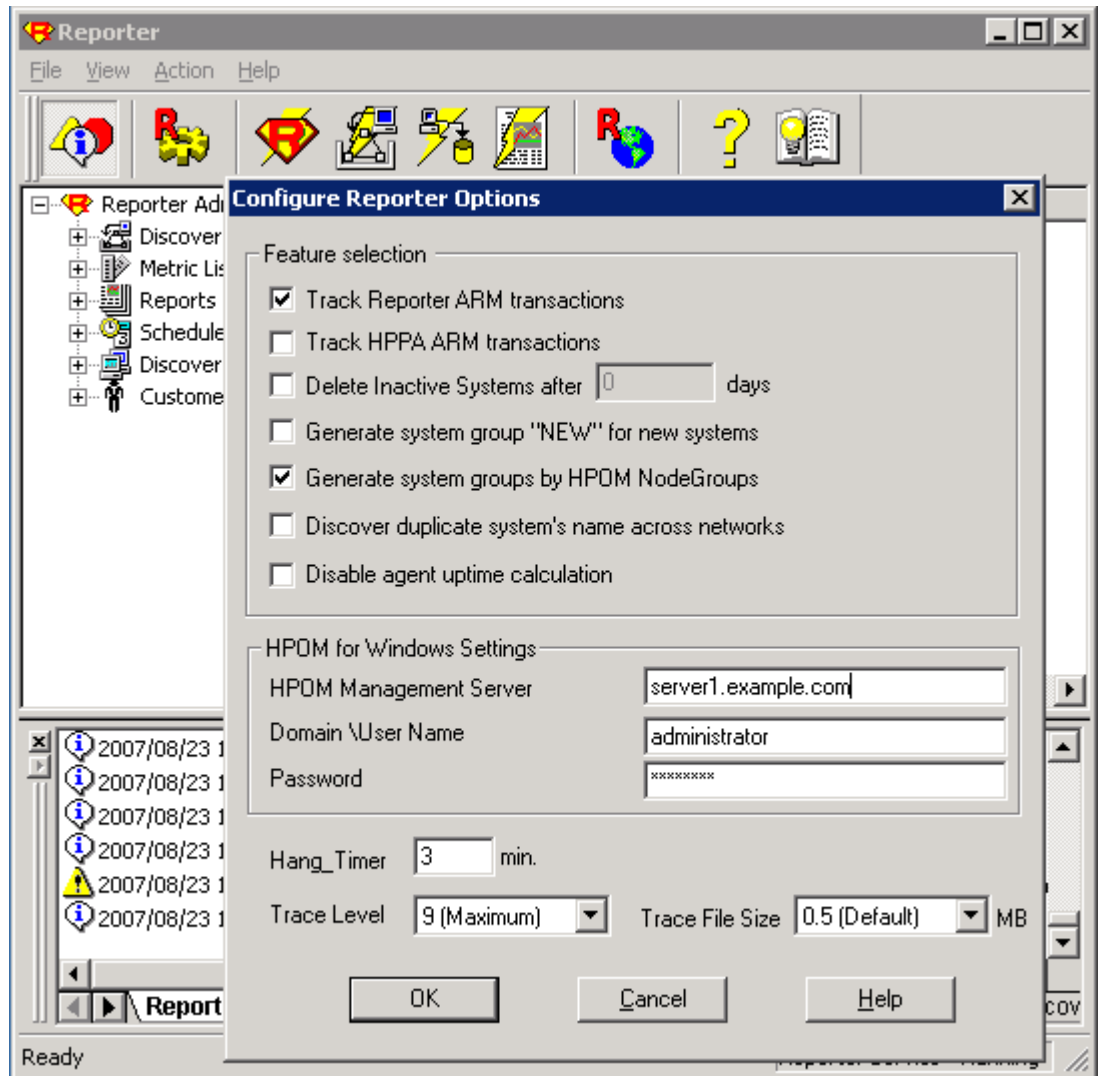


For group and single system WebSphere SPI reports you are required to identify systems by their full name. For example, **abc.xyz.com** is acceptable while **abc** is not.

Viewing Reports from the HPOM Management Console

To view WebSphere **Reports** from the HPOM Console, follow these steps:

- 1 Close the HPOM for Windows console (if it is open).
- 2 Open the HP Reporter window. Click **File** → **Configure** → **Options**. The Configure Reporter Options window opens.



- 3 In the HPOM for Windows Settings section, specify the name of the management server and user details. The user must be an HPOM administrator (a member of the HP-OVE-Admins group). Click **OK**.
- 4 From the menu bar, click **Action** → **Run** → **Run All**.
This will discover the node data from HPOM and generate the reports. This may take some time.
- 5 After the HP Reporter tasks complete, open the HPOM console. **Reports** will be visible in the console tree.

Reports Generated by Reporter

The reports available through the integration of HP Reporter and the WebSphere SPI show consolidated data on server performance and availability on all WebSphere systems. In addition, other reports show data for single systems. These reports are available one day after you install the WebSphere SPI report package on the Reporter Windows system. See [Integrating the WebSphere SPI with HP Reporter](#) on page 52 if you have not yet completed the report package installation.

The following tables show pre-defined reports.

Table 1 Reports for All Systems - WebSphere Performance

Report Title	Description	WebSphere Version
DB Connection Pool Throughput - Top 20	Shows the average number of connections allocated per day for the top 20 servers. The top 20 servers are selected based on the highest average number of connections allocated over the reporting period.	4.0, 5.0, 6.0, 6.1
EJB Method Calls Rate - Top 20	Shows the number of all EJB method calls per minute for the top 20 servers. The top 20 servers are selected based on the highest average method calls per minute over the reporting period.	4.0, 5.0, 6.0, 6.1
Entity EJB Load/Stores Rate - Top 20	Shows the number of all entity EJB loads and stores to/from the database per minute for the top 20 servers. The top 20 servers are selected based on the highest average loads and stores per minute over the reporting period.	4.0, 5.0, 6.0, 6.1
Servlet Average Response Time - Top 20	Shows the average response time for the top 20 servlets. The top 20 servlets are selected based on the highest average number of requests for the servlet per second over the reporting period.	4.0, 5.0, 6.0, 6.1

Table 1 Reports for All Systems - WebSphere Performance

Report Title	Description	WebSphere Version
Servlet Requests - Top 20	Shows the number of servlet requests per second by a server. The top 20 servers are selected based on the highest average number of servlet requests per second for the server over the reporting period.	4.0, 5.0, 6.0, 6.1
Servlet Sessions - Top 20	Shows the total number of servlet sessions being handled by the top 20 servers. The top 20 servers are selected based on the highest average number of sessions over the reporting period.	4.0, 5.0, 6.0, 6.1
Transaction Throughput - Top 20	Shows the average number of transactions processed per second for each server. The top 20 servers are selected based on the highest average number of transactions processed per second over the reporting period.	4.0, 5.0, 6.0, 6.1

Table 2 WebSphere Availability

Report Title	Description	WebSphere Version
Server Availability	<p>Contains a daily histogram showing the percentage of uptime. In addition, a trend line provides the number of measurements performed, indicating how much data was available to determine availability.</p> <p>Uptime and downtime are measured by the WebSphere SPI. A lower than expected trend line may indicate systems were unavailable or the data collection not running.</p>	4.0, 5.0, 6.0, 6.1

Table 3 Reports for Single System

Report Title	Description	WebSphere Version
Admin Server Availability Details	Contains spectrum graphs showing minutes of uptime by day and hour for a system. Uptime and downtime are measured by the WebSphere SPI. “No Data” may include system downtime or data collection not running. Graphs are based on measured uptime and downtime only (that is standby = down). The spectrum graphs use color to indicate the uptime percentage during each hour of each day.	4.0, 5.0, 6.0, 6.1
DB Connection Pools Clients Waiting vs. Timeout Rate	Compares the number of clients waiting vs. client timeout rate for the DB connection pools for a system.	4.0, 5.0, 6.0, 6.1
DB Connection Pools Size vs. Wait Time	Compares the average size vs. the average wait time for the DB connection pools on a server.	4.0, 5.0, 6.0, 6.1
DB Connection Pools Throughput vs. Utilization	Compares the throughput vs. the utilization of the DB connection pools on a server. Throughput is the number of connections allocated by a DB connection pool per second. The utilization of a connection pool is the number of connections being used as a percent of the maximum capacity configured for the pool.	4.0, 5.0, 6.0, 6.1
EJB Average Response Time - Top 20	Shows the average response time in milliseconds for the top 20 EJBs for a system. The top 20 EJBs are selected based on the highest average response time over the entire reporting period.	4.0, 5.0, 6.0, 6.1
EJB Method Calls Rate - Top 20	Shows the number of all EJB method calls per minute for the top 20 EJBs for a system. The top 20 EJBs are selected based on the highest average method calls per minute over the reporting period.	4.0, 5.0, 6.0, 6.1
EJB Pool Misses Percent - Top 20	Shows the percent of EJB retrievals that are not successful during the collection interval for a server. The top 20 EJBs are selected based on the highest average pool misses over the reporting period.	4.0, 5.0, 6.0, 6.1
EJB Pool Size - Top 20	Shows the average pool size for the top 20 EJBs for a system. The top 20 EJBs are selected based on the highest average pool size over the reporting period.	4.0, 5.0, 6.0, 6.1

Table 3 Reports for Single System (cont'd)

Report Title	Description	WebSphere Version
EJB Pool Utilization - Top 20	shows the utilization of an EJB pool as a percent of the number of EJB instances configured for the pool on a server. The top 20 EJBs are selected based on the highest average pool utilization over the reporting period.	4.0, 5.0, 6.0, 6.1
Entity EJB Load/Stores Rate by EJB - Top 20	Shows the number of all EJB loads and stores to/from the database per minute for the top 20 EJBs on a server for a system. The top 20 EJBs are selected based on the highest average loads and stores per minute over the reporting period.	4.0, 5.0, 6.0, 6.1
Server Availability Details	Contains spectrum graphs showing minutes of uptime by day and hour for a system. Uptime and downtime are measured by the WebSphere SPI. “No Data” may include system downtime or data collection not running. Graphs are based on measured uptime and downtime only (that is standby = down). The spectrum graphs use color to indicate the uptime percentage during each hour of each day.	4.0, 5.0, 6.0, 6.1
Servlet Average Response Time - Top 20	This report shows the average response time for the top 20 requested servlets for a system. The top 20 servlets are selected based on the highest average request rate over the entire reporting period.	4.0, 5.0, 6.0, 6.1
Servlet Requests - Top 20	Shows the number of requests per second for servlets on a server. The top 20 servlets are selected based on the highest average number of requests per second for the servlet over the reporting period.	4.0, 5.0, 6.0, 6.1
Thread Pool Activity	Shows the average size of thread pools against the average number of active threads, for all thread pools for a system.	4.0, 5.0, 6.0, 6.1
Transaction Throughput - Top 20	Shows the average number of transactions processed per second for each server. The top 20 servers for a system are selected based on the highest average number of transactions processed per second over the reporting period. The total height of the topmost line indicates the total transaction throughput for all the servers.	4.0, 5.0, 6.0, 6.1

Removing the WebSphere SPI Reporter Package

To remove the WebSphere SPI Reporter Package, follow these step:

- 1 Insert the HP Operations Smart Plug-ins DVD into the DVD drive of the Windows system running Reporter. The HP Operations Manager InstallShield Wizard opens.
- 2 Click **Next**. The Program Maintenance window opens.
- 3 Click **Remove Products**. The Product Selection window opens.
- 4 From the options listed (there are three Product Selection windows), select the **Reports** option of IBM WebSphere and click **Next** till the Remove the Selected Products window opens.
- 5 Click **Remove**.

Integrating the WebSphere SPI with HP Performance Manager

You must purchase and install HP Performance Manager, separately. To integrate the WebSphere SPI with HP Performance Manager, follow these steps:

- 1 Install the WebSphere SPI graph package on the Windows system running HP Performance Manager:
 - a Insert the HP Operations Smart Plug-ins DVD into the DVD drive of the Windows system running Reporter. The HP Operations Manager InstallShield Wizard opens.
 - b Click **Next**. The Program Maintenance window opens. Click **Install Products**. The Product Selection window opens.
 - c From the options listed (there are three Product Selection windows), select the **Graph** option of IBM WebSphere and click **Next**.
 - d Complete the installation by following the instructions as you proceed.
- 2 To graph any WebSphere metric, use the data source name: `WBSSPI_METRICS`.

Viewing Graphs that Show Alarm Conditions

For graphing purposes, the WebSphere SPI organizes metrics according to type. When a message is generated for any metric (listed in the tables in the following section), you can view a chart of that metric along with other metric values.

To view a graph associated with an alarm condition (operator-initiated action has been defined with the WebSphere SPI policy), complete these steps:

- 1 In the HPOM message browser, double-click the message for which you want to view the graph. The Message Properties window opens.
- 2 Click the **Commands** tab.
- 3 Click **Start** in the section Operator Initiated to start the operator-initiated command.
The operator action launches your web browser, where you can view the graph.

Viewing Graphs that Show Past or Current Conditions

To generate an available graph manually, follow these steps:


- 1 From the HPOM console, select **Operations Manager** → **Graphs** → **SPI for WebSphere**.
- 2 Double-click the graph you want to generate. A new window opens.
- 3 Select the nodes from which you want to retrieve data. Select the date range and the granularity for the graph.
- 4 Click **Finish**.



Graphs appears in the HPOM console tree only if you install HP Performance Manager on the same system as the HPOM management server.

Viewing Graphs from the HP Performance Manager Console

If you have not installed HP performance Manager on the same system as HPOM management server, you can view the WebSphere SPI Graphs from the HP Performance Manager console. Follow these steps:

- 1 Click **Start** → **All Programs** → **HP** → **HP BTO Software** → **Performance Manager** → **Performance Manager**. The Performance Manager console opens.
- 2 From the Select Nodes pane, select the node for which you want to see graph. If the node is not listed in the list, add the node:
 - a Click **Admin** in the menu bar. The Manage Nodes window opens.
 - b Click the **Add a Node**  icon. The Add a Node Window opens.
 - c Enter the node name and click **Add**.
 - d Click **Home** on the menu bar.
- 3 From the Select a Graph pane, select **SPI for WebSphere**.
- 4 Select the graph you want to see and click **Draw**.



If you have installed HP Performance Agent to collect performance data, you must select the **SPI for WebSphere - OVPA <version>** from the list in the Select a Graph pane.

WebSphere SPI Metrics Available for Graphs

The following tables show the graphs available for mapping the collected metric values. If you want to view the graph for any one of the metrics included in the following tables, you can use the View Graphs tool. A graph of the metric appears in your web browser.

Table 4 Enterprise Java Beans (EJB): 20, 22, 24, 25, 26

Graph Label	Metric Name	Metric Description
EJB Pool Utilization	I020_EJBPoolUtil	Percentage of active beans in the pool.
EJB Method Calls Rate	I022_EJBMethCallsRt	Number of EJB method calls per minute.
EJB Entity Data Load Stored Rate	I024_EJBEntDatLdStRt	Number of times an EJB was written to or loaded from the database per minute.
EJB Pool Missed Percentage	I025_EJBPoolMissPct	Average percentage of time a call to retrieve an EJB from the pool failed.
EJB Connected Lives	I026_EJBConcLives	Average number of bean objects in the pool.

Table 5 JDBC: 61, 62, 65, 66

Graph Label	Metric Name	Metric Description
JDBC Connect Pool Waits	I061_JDBCConPoolWait	Average number of threads waiting for a connection from connections pools.
JDBC Connections Pool Wait Time	I062_JDBCConPoolWtTim	Average time that a client waited for a connections in milliseconds.
JDBC Connection Pool Timeout Rate	I065_JDBCConPoolTimRt	Number of times a client timed out waiting for a connection from the pool per minute.
JDBC Connection Pool Throughput	I066_JDBCConPoolThru	Number of connections allocated and returned by applications per second.

Table 6 Servlet: 40, 41, 42

Graph Label	Metric Name	Metric Description
Servlet Session Average Life	I040_ServSessAveLife	Average lifetime of a servlet session in milliseconds.
Servlet Active Sessions	I041_ServSessActSess	Number of sessions currently being accessed.
Servlet Invalidated Session Rate	I042_ServInvSessRt	Number of sessions being invalidated per second.

Table 7 ThreadPool: 13, 14

Graph Label	Metric Name	Metric Description
Thread Pool Percentage Maximum	I013_ThrdPoolPetMax	Percentage of time number of threads in pool reached configured maximum size.
Thread Pool Create Rate	I014_ThrdPoolCrtRt	Number of threads created per minute.

Table 8 Transaction: 70, 71, 72, 73, 74, 75, 76, 77, 78

Graph Label	Metric Name	Metric Description
Transaction (global) Duration	I070_TranGlobDur	Average duration of global transactions.
Transactions (local) Duration	I071_TranLocDur	Average duration of local transactions.
Transaction (global) Commitment Rate	I072_TranGlobCommDur	Average duration of commits for global transactions.
Transaction (Local) Commitment Duration	I073_TranLocCommDur	Average duration of commits for local transactions.
Transaction Rollback Rate	I074_TranRollbackRt	Number per second of global and local transactions rolled back.
Transaction Timeout Rate	I075_TranTimeoutRt	Number per second of timed out global and local transactions.
Transaction Commitment Rate	I076_TranCommitRt	Number per second of global and local transactions that were committed.
Transaction Throughput Rate	I077_TranThruput	Number per second of global and local transactions that were completed.
Transaction Start Rate	I078_TranStartRt	Number per second of global and local transactions that were begun.

Table 9 Web Application

Graph Label	Metric Name	Metric Description
Web Application Servlet Request Rate	I045_WebAppServReqRt	Number of requests for a servlet per second.
Web Application Servlet Error Rate	I047_WebAppServErrRt	Number of errors in a servlet per second.
Web Application Servlet Load	I048_WebAppServLoad	Number of servlets currently loaded for a web application.
Web Application Server Reload Rate	I049_WebAppServRelRt	Number of servlets reload for a web application per minute.

Removing the WebSphere SPI Grapher Package

To remove the WebSphere SPI Grapher package, follow these steps:

- 1 Insert the HP Operations Smart Plug-ins DVD into the DVD drive of the Windows system running Performance Manager. The HP Operations Manager InstallShield Wizard opens.
- 2 Click **Next**. The Program Maintenance window opens.
- 3 Click **Remove Products**. The Product Selection window opens.
- 4 From the options listed (there are three Product Selection windows), select the **Graphs** option of IBM WebSphere and click **Next** till the Remove the Selected Products window opens.
- 5 Click **Remove**.

6 User Defined Metrics

The WebSphere SPI can collect data on roughly 50 metrics. However, you can expand that number by adding your own. The advantage to defining your own metrics is that you can monitor your own applications.

You can customize what you monitor by creating user-defined metrics (UDMs) that instruct the WBSSPI to gather data from PMI counters.



See the IBM WebSphere documentation for more information about PMI.

You must understand the metric definitions DTD before creating UDMs. The sections that follow assume you are familiar with XML (extensible markup language) and DTDs (Document Type Definitions).

Metric Definitions DTD

The `MetricDefinitions.dtd` file provides the structure and syntax for the XML file that you create. The WebSphere SPI uses this DTD file to parse and validate the XML file you create. Following sections describe the `MetricDefinitions.dtd` file and provide an example XML file.

On a managed node, the `MetricDefinitions.dtd` file is located in the following directory:

Operating System Directory

AIX	<code>/var/lpp/OV/wasspi/wbs/conf/</code>
HP-UX and Solaris	<code>/var/opt/OV/wasspi/wbs/conf/</code>
Windows	<code><AgentDir>\wasspi\wbs\conf\</code>

For HPOM for Windows 8.10 or 8.00, `<AgentDir>` is typically is:

On Windows: `\Program Files\HP OpenView\data\` (for HTTPS managed nodes) or
`C:\Program Files\HP OpenView\Installed Packages\`
`{790C06B4-844E-11D2-972B-080009EF8C2A}` (for DCE managed nodes)

On UNIX: `/var/opt/OV/` or `/var/lpp/OV/`



Because the `MetricDefinitions.dtd` file is used at runtime, you should not edit, rename, or move it.

`MetricDefinitions.dtd` consists of the following elements:

- `MetricDefinitions`
- `Metric`
- `PMICounter`
- `FromVersion/ToVersion`
- `Calculation/Formula`

The MetricDefinitions Element

The `MetricDefinitions` element is the top-level element within the `MetricDefinitions.dtd` file. It contains one collection of metrics, consisting of one or more metric definitions.

```
<!ELEMENT MetricDefinitions (Metrics)>
<!ELEMENT Metrics (Metric+)>
```

Example

```
<MetricDefinitions>
  <Metrics>
    .
    .
    .
  </Metrics>
</MetricDefinitions>
```

The Metric Element

The Metric element represents one metric. Each metric has a unique ID (for example, WBSSPI_1001). If a user-defined metric is an alarming, graphing, or reporting metric, the metric ID must be “WBSSPI_XXXX” where XXXX must be a number from 1000 through 1999. Otherwise, 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 (for example, “counter1”).

A Metric element contains one or more elements that represent the metric data source. Two data sources are supported: PMI counters and calculations. Each metric data source element is scanned for a FromVersion or ToVersion child element to determine which metric data source element to use for the version of the application server being monitored.

```
<!ELEMENT Metric (PMICounter+ | Calculation+)>
<!ATTLIST Metric id          ID          #REQUIRED
                 name        CDATA      ""
                 alarm       (yes | no)  "no"
                 report      (yes | no)  "no"
                 graph       (yes | no)  "no"
                 previous    (yes | no)  "yes"
                 description  CDATA      #IMPLIED >
```

The following table describes Metric element attributes.

Attribute	Type	Required	Default	Description
id	ID	yes	--	The metric ID.
name	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 through <code>opcmon</code> .
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 user-defined metric is graphed.
description	text	no	“”	A description of the metric.

Example

```
<Metric id="WBSSPI_1005" name="B005_JVMMemUtilPct" alarm="yes" graph="no">
.
.
.
</Metric>
```

The PMI Counter Element

The PMICounter element is used when the metric data source is a PMI counter. The PMICounter element contains the following elements:

- **Path** - the location of the counter in the PMI data hierarchy. The content of this element begins with the PMI module name and may contain the wildcard character to specify multiple instances.
- **ID** - the PMI data ID to be retrieved from the counter.
- **Load** (optional) - the data to be retrieved is of type load and thus various time-based values are available for retrieval. Use the data attribute to specify which value to retrieve.
- **Stat** (optional) - the data to be retrieved is of type stat and thus various sample-based values are available for retrieval. Use the data attribute to specify which value to retrieve.

```
<!ELEMENT PMICounter (FromVersion?, ToVersion?, Path, ID,
                      (Load | Stat)?)>
<!ATTLIST PMICounter instanceType (single | multi)      "single"
                      impact (low | medium | high) #REQUIRED>
<!ELEMENT Path (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT Load EMPTY>
<!ATTLIST Load data (mean | weight | sum | current) #REQUIRED>
<!ELEMENT Stat EMPTY>
<!ATTLIST Stat data (mean | count | sumOfSquares | variance |
                    standardDeviation | confidence) #REQUIRED>
```



See the IBM WebSphere documentation for more information about PMI.

The following table lists PMICounter element attributes are described in the following table.

Attribute	Type	Required	Default	Description
instanceType	"single" "multi"	no	single	Indicates if there are multiple instances of this counter.
impact	"low" "medium" "high"	yes	no default	How the metric affects performance.

The following table lists Load element attributes.

Attribute	Type	Required	Default	Description
data	"mean" "sum" "weight" "current"	yes	none	Specifies which time-based data to retrieve.

The following table lists Stat element attributes.

Attribute	Type	Required	Default	Description
data	“mean” “count” “sumOfSquares” “variance” “standardDeviation” “confidence”	yes	no default	Specifies which sample-based data to retrieve.

Example

```
<PMICounter instanceType="multi" impact="high">
  <Path>threadPoolModule/*</Path>
  <ID>3</ID>
  <Load data="weight"/>
</PMICounter>
```

FromVersion and ToVersion Elements

The FromVersion and ToVersion elements are used to determine the version of the application server being monitored.

The following algorithm is used for determining which application server version is supported by each metric source element within the Metric element.

- If a FromVersion element is not present, no lower limit exists to the server versions supported by this metric.
- If a FromVersion element is present, the server attribute indicates the lowest server version supported by this metric. If an update attribute exists, it qualifies the lowest server version supported by specifying the lowest Fix Pack or patch supported for that version.
- If a ToVersion element is not present, no upper limit exists to the server versions supported by this metric.
- If a ToVersion element is present, the server attribute indicates the highest server version supported by this metric. If an update attribute exists, it qualifies the server version supported by specifying the highest service pack or patch supported for that version.

```
<!ELEMENT FromVersion (EMPTY)>
<!ELEMENT ToVersion (EMPTY)>

<!ATTLIST FromVersion    server CDATA #REQUIRED
                        update CDATA  "*">
<!ATTLIST ToVersion     server CDATA #REQUIRED
                        update CDATA  "*">
```

The following table lists FromVersion and ToVersion element attributes.

Attribute	Type	Required	Default	Description
server	string specifying version number	yes	none	Specifies a primary server version; for example, <code><FromVersion server="6.0"/></code>
update	string specifying version number	no	"*"	Specifies a secondary server version, such as "1" for service pack 1. A "*" indicates that a metric is valid for all secondary server versions.

Example

```
<FromVersion server="4.0" update="1"/>  
<ToVersion server="4.0999"/>
```

Calculation and Formula Elements

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 whose content is 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 result of the calculation is the metric value.

```
<!ELEMENT Calculation (FromVersion?, ToVersion?,Formula)>  
<!ELEMENT Formula (#PCDATA)>
```

Syntax

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 see either a PMICounter 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.

Functions

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.
- `sum` returns the summation of the values of all the instances of a multi-instance metric.

- count returns the number of instances of a multi-instance metric.

Examples

In the following example the value of the metric is the ratio (expressed as a percent) of Metric_1 to Metric_3.

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

The following example shows how 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>
```

Sample 1

Metric 710 uses metric “counter1” in its calculation. This calculated metric applies to all WebSphere versions. However, the PMICounter metric on which it is based has changed. Originally the PMICounter for metric 710 was introduced on server version 4.0, update 1. However in version 4.1, the id changed, and this change remains the same up to the latest server version.

```
<Metric id="counter1" alarm="no">
  <PMICounter instanceType="multi" impact="high">
    <FromVersion server="4.0" update="1"/>
    <ToVersion server="4.099"/>
    <Path>threadPoolModule/*</Path>
    <ID>3</ID>
  </PMICounter>
  <PMICounter instanceType="multi" impact="high">
    <FromVersion server="4.1"/>
    <Path>threadPoolModule/*</Path>
    <ID>30</ID>
  </PMICounter>
</Metric>

<Metric id="WBSSPI_1010" alarm="yes">
  <Calculation>
    <Formula>delta(counter1) /interval(counter1) *1000*60</Formula>
  </Calculation>
</Metric>
```

Sample 2

Using the example above, a decision was made to make metric 710 a per-minute rate instead of a per-second rate as of server version 5.0.

```
<Metric id="counter1" alarm="no">
  <PMICounter instanceType="multi" impact="high">
    <FromVersion server="4.0" update="1"/>
    <ToVersion server="4.099"/>
    <Path>threadPoolModule/*</Path>
    <ID>3</ID>
```

```

    </PMICounter>
    <PMICounter instanceType="multi" impact="high">
      <FromVersion server="4.1"/>
      <Path>threadPoolModule/*</Path>
      <ID>30</ID>
    </PMICounter>
  </Metric>
  <Metric id="WBSSPI_1010" alarm="yes">
    <Calculation>
      <FromVersion server="4.0"/>
      <ToVersion server="4.999"/>
      <Formula>delta(counter1)/interval(counter1)*1000*60</Formula>
    </Calculation>
    <Calculation>
      <FromVersion server="5.0"/>
      <Formula>delta(counter1)/interval(counter1)*1000</Formula>
    </Calculation>
  </Metric>

```

Sample 3: Metric Definitions File

The following sample metric definitions file illustrates how you may create your own user-defined metrics. This sample file also contains examples of calculated metrics.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MetricDefinitions SYSTEM "MetricDefinitions.dtd">
<!-- sample UDM metrics configuration File -->
<MetricDefinitions>
  <Metrics>
    <!-- The following metrics illustrate some of the options available when
    creating user-defined metrics.-->
    <!-- The following metric uses a PMICounter that can have multiple
    instances. Note the wildcard (*) character is used to specify multiple
    instances. The path element uses the standard PMI path syntax with the
    exception that it must start at the module level and can include the
    wildcard character at any point in the path.-->
    <Metric id="WBSSPI_1000" name="UDM_700" alarm="yes">
      <PMICounter instanceType="multi" impact="high">
        <FromVersion server="4.0"/>
        <Path>threadPoolModule/*</Path>
        <ID>3</ID>
        <Load data="weight"/>
      </PMICounter>
    </Metric>
    <!-- The following 2 metrics are "base" metrics. They are used in the
    calculation of a "final" metric but are not alarmed, reported, or graphed
    themselves. Base metrics may have an 'id' that begins with a
    letter(case-sensitive) followed by any combination of letters, numbers,
    and underscore. Base metrics normally have alarm="no".-->
    <Metric id="JVMSRuntime_HeapSize" alarm="no" >
      <PMICounter instanceType="single" impact="low">

```



```

        <FromVersion server="4.0"/>
        <Path>jvmRuntimeModule</Path>
        <ID>1</ID>
    </PMICounter>
</Metric>
<Metric id="JVMSRuntime_UsedSpace" alarm="no">
    <PMICounter instanceType="single" impact="low">
        <FromVersion server="4.0"/>
        <Path>jvmRuntimeModule</Path>
        <ID>3</ID>
    </PMICounter>
</Metric>
<!-- The following metric illustrates a calculated metric. The calculation
is based on the previous 2 "base" metrics.-->
<Metric id="WBSSPI_1005" name="B005_JVMMemUtilPct" alarm="yes" graph="no">
    <Calculation>
        <FromVersion server="4.0"/>
        <Formula>((JVMSRuntime_UsedSpace)/JVMSRuntime_HeapSize)*100
        </Formula>
    </Calculation>
</Metric>
<!-- Metric IDs that are referenced from the collector command line must
have a namespace prefix followed by 4 digits. The default namespace prefix
is 'WBSSPI_'. The 'namespace' option must be used on the command line for
the following metric since this metric has a different prefix other than
'WBSSPI_'. Example: wasspi_wbs_ca -x namespace=Testing_ -m 992 ...-->
<Metric id="Testing_0992" name="Testing_Metric">
    <PMICounter instanceType="single" impact="high">
        <FromVersion server="4.0"/>
        <Path>beanModule</Path>
        <ID>9</ID>
        <Load data="sum"/>
    </PMICounter>
</Metric>
</Metrics>
</MetricDefinitions>

```

Creating User-Defined Metrics

To create UDMs, complete the following tasks in the specified order.

Task 1: Disable Graphing (if Enabled)

If graphing is enabled, disable it:

- 1 From the HPOM console, select **Operations Manager** → **Nodes**.
- 2 Right-click the node on which you want to disable UDM graphing and select **All Tasks** → **Launch Tool** → **UDM Graph Disable**.

Task 2: Create a metric definitions file

The metrics definition file you create must be an XML file that follows the format defined by the metric definitions DTD file described in [Metric Definitions DTD](#) on page 66.



Do not edit, rename, or move the MetricDefinitions.dtd file installed with the WebSphere SPI.

The sample metric definitions file is installed on the managed node:

AIX	/var/lpp/OV/wasspi/wbs/conf/UDMMetrics-sample.xml
HP-UX and Solaris	/var/opt/OV/wasspi/wbs/conf/UDMMetrics-sample.xml
Windows	<AgentDir>\wasspi\wbs\conf\UDMMetrics-sample.xml

For HPOM for Windows 8.10 or 8.00, <AgentDir> is typically is:

On Windows: \Program Files\HP OpenView\data\ (for HTTPS managed nodes) or
C:\Program Files\HP OpenView\Installed Packages\
{790C06B4-844E-11D2-972B-080009EF8C2A} (for DCE managed nodes)

On UNIX: /var/opt/OV/ or /var/lpp/OV/

Task 3: Configure the Metric Definitions File Name and Location

For the UDM data collection to occur, the WebSphere SPI configuration must include the name and location of the metric definitions file, preceded by the property name as shown below:

```
UDM_DEFINITIONS_FILE = <full path of metric definitions file>
```

where the path name should use only forward slashes (“/”).

To add the UDM file name and its location to the WebSphere SPI configuration, follow these steps:

- 1 From the HPOM console, select **Operations Manager** → **Tools** → **SPI for WebSphere** → **SPI Admin**.
- 2 Double-click **Configure WBSSPI**.
- 3 Select the managed nodes on which the metrics definition file exists and click **Launch**. The Console Status window opens.

After some time the Configure WBSSPI Tool Introduction window appears. Read the information and click **Next**. The configuration editor opens.

- 4 If the metrics definition file uses the same name and location on all managed nodes, configure the UDM_DEFINITIONS_FILE property at the Defaults (global properties) level. Otherwise, set the property for each managed node selected in step 3:
 - a Click **Default Properties** at the Defaults level or for a node.
 - b Click the **Set Configuration Properties** tab.
 - c From the Select a Property to Add dropdown menu, select **UDM_DEFINITIONS_FILE** and click **Add Property**.
 - d Type the value (metric definitions file name and its absolute path name, using forward slashes in only the path name).
 - e Click **Save** to save the changes.
 - f Click **Next**. The Confirm Operation window opens.
 - g Click **OK** to save changes and exit the configuration editor.

Changes you made to managed nodes that were not selected are saved to the configuration on the management server. However, to configure those managed nodes, you must deploy the WBSSPI Service Discovery policy to these nodes.

Task 4: Create a UDM Policy Group and Policies

To run the UDM data collection and establish thresholds for alarming, create a UDM policy group and policies:

- 1 Copy an existing WebSphere SPI policy group.
 - a From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere** → *<impact>*- **Impact**.
 - b Right-click the policy group you want to use as a starting point, and select **Copy**.
 - c Right-click *<impact>*- **Impact** and select **Paste**.
- 2 Rename the new policy group according to how you plan to identify the new metric and collector policies. For example, you might include UDM in the name to clearly indicate that the group consists of custom metric monitors.
 - a Right-click the copy of the original policy group and select **Rename**.
 - b Type in the new name.
- 3 Edit and rename each policy in the new group:
 - a Double-click the policy you plan to use in the new group.
 - b Configure the collector policy command line (in the Command box) to include the UDM metric number. For more information, see [Changing the Collection Interval for Selected Metrics](#) on page 39.
 - c Configure thresholds in the policy, as appropriate. [Changing the Collection Interval for Selected Metrics](#) on page 39.
 - d Select **File** → **Save As**. Rename the policy according to your naming scheme.
 - e The name you give the new metric policy in the group would contain each new UDM number. For example, a copy of WBSSPI_0001 could be called WBSSPI_1001.
The name you give the new collector policy must also contain the identifying name.
- 4 Right-click the original policy and select **Delete**, to delete the original policy. You must delete all the original policies from the new group.

Task 5: Deploy the policy group

- 1 Right-click the new policy group and select **All Tasks → Deploy on**.
- 2 Select the nodes on which to deploy the policy group.
- 3 Click **OK**.

Task 6: Enable graphing

If you are using graphing (HP Performance Manager must be purchased and installed), enable data collecting for UDM graphing:

- 1 From the HPOM console, select **Operations Manager → Nodes**.
- 2 Right-click the node on which you want to enable UDM graphing and select **All Tasks → Launch Tool → UDM Graph Enable**.

Allow sufficient collection intervals to occur before attempting to view graphs.

7 Troubleshooting the WebSphere SPI

This chapter covers basic troubleshooting for the WebSphere SPI. The Error messages section of the WebSphere SPI online help lists error messages by number.

The Self-Healing Info Tool

The Self-Healing Info tool gathers troubleshooting information about the SPI and stores it in a file that you can submit to HP support for assistance. For more information about this tool, see the WBSSPI Admin tools section under Tools in the WebSphere SPI online help.



The file created by the Self-Healing Info tool may be hidden on some Windows managed nodes. If you do not see the file, open Windows Explorer and from the **Tools** menu, select **Folder Options**. Click the **View** tab. Under Hidden files and folders, select **Show hidden files and folders**.

Log and Trace Files

Log and trace files are maintained on the managed nodes. You can gather troubleshooting information about the WebSphere SPI from the data logged in these log and trace files.

UNIX Managed Nodes

The following log and trace files are found on the managed nodes running on UNIX (typically, *<AgentDir>* is `/var/opt/OV/` or `/var/lpp/OV/`):

File Type	Log
Directory	<code>/<AgentDir>/wasspi/wbs/log/config.log</code>
Description	Output from the WebSphere SPI configuration scripts is recorded in this log file.

File Type	Log
Directory	<code>/<AgentDir>/wasspi/wbs/log/errorlog</code>
Description	WebSphere SPI logs the error messages in this file. This log file is monitored by WebSphere SPI policies.

File Type	Log
Filename	<code>/<AgentDir>/wasspi/wbs/log/discovery.log</code>
Description	Records output from the WebSphere SPI discovery process.

File Type	Trace
Filename	<code>/<AgentDir>/wasspi/wbs/log/discovery.trace</code> (archived files have a three digit number appended to the filename)
Description	Discovery binary trace file used by your HP support representative. By default, tracing to this file is enabled. To disable tracing, in <code><AgentDir>/bin/instrumentation/wasspi_wbs_discovery.pl</code> , set the <code>\$trace_on</code> variable to 0. To enable this trace, set the <code>\$trace_on</code> to 1. When instrumentation is deployed, the <code>wasspi_wbs_discovery.pl</code> file is overwritten (therefore, if you disable tracing, it becomes enabled when instrumentation is deployed). Five archived versions are kept. A new trace file is created when the discovery policy is run.

File Type	Trace
Directory	/<AgentDir>/wasspi/wbs/log/trace.log (archived files have a three digit number appended to the filename)
Description	The HP support representative uses this trace file. This file gives information about the CollectorServer, regardless of whether the Collector is set to PERSISTANT or TRANSIENT mode in the SPIConfig file. The default value of the Collector Mode is 'PERSISTENT'. By default, tracing to this file is disabled. To enable this tracing, run the Start Tracing tool.

File Type	Trace
Directory	/<AgentDir>/wasspi/wbs/log/traceCollectorClient.log (archived files have a three digit number appended to the filename)
Description'	Trace file used by your HP support representative. This file gives information about the CollectorClient when the Collector is set to 'PERSISTENT mode in the SPIConfig file. The default value of the Collector Mode is 'PERSISTENT'.

Windows Managed Nodes

The following log and trace files are found on the managed nodes running on Windows.

<AgentDir> typically is \Program Files\HP OpenView\Data on HTTPS managed nodes for HPOM for Windows 8.00 and 8.10.

<AgentDir> typically is C:\Program Files\HP OpenView\Installed Packages\{790C06B4-844E-11D2-972B-080009EF8C2A} on DCE managed nodes for HPOM for Windows 8.00 and 8.10.

File Type	Log
Filename	\<AgentDir>\log\system.txt
Description	This is the HPOM discovery agent log file containing the status of the HPOM discovery agent. By default, logging to this file is enabled at LOG_LEVEL 3. Set the LOG_LEVEL variable in <InstallDir>\conf\svcDisc\OvJavaAgent.cfg to 6 or higher (up to 9) to capture troubleshooting information (the higher the number, the more information is collected). To disable this log, set the LOG_LEVEL to 0. You can configure additional information to define log file size and the number of files kept in archive. By default, the log file size is 1MB and five versions of the log file are archived.

File Type	Log
Directory	\<AgentDir> \wasspi\wbs\log\config.log
Description	Records output from configuration scripts.
File Type	Log
Directory	\<AgentDir> \wasspi\wbs\log\errorlog
Description	Records WebSphere SPI error messages. This log file is monitored by WebSphere SPI policies.
File Type	Log
Filename	\<AgentDir> \wasspi\wbs\log\discovery.log
Description	Records output from the WebSphere SPI discovery process.
File Type	Trace
Directory	\<AgentDir>\wasspi\wbs\log\discovery.trace (archived files have a three digit number appended to the filename)
Description	Discovery binary trace file used by your HP support representative. By default, tracing to this file is enabled. To disable tracing, in <AgentDir>\bin\instrumentation\wasspi_wbs_discovery.pl, set the \$trace_on variable to 0. To enable this trace, set the \$trace_on to 1. When instrumentation is deployed, the wasspi_wbs_discovery.pl file is overwritten (therefore, if you disable tracing, it becomes enabled when instrumentation is deployed). Five archived versions are kept. A new trace file is created when the discovery policy is run.
File Type	Trace
Directory	\<AgentDir>\wasspi\wbs\log\traceCollectorClient.log (archived files have a three digit number appended to the filename)
Description	Trace file used by your HP support representative. This file gives information about the CollectorClient when the Collector is set to 'PERSISTENT' mode in the SPIConfig file. The default value of the Collector Mode is 'PERSISTENT'. By default, tracing to this file is disabled. To enable this tracing, use the Start Tracing application.

Troubleshooting the Discovery Process

Problem

The WBSSPI Discovery policies do not automatically discover and update the WebSphere SPI configuration.

Solutions

To troubleshoot the discovery process, do one or more of the following (as applicable):

- Check if the WBSSPI Discovery policies are still being deployed:

From the HPOM console, select **Operations Manager** → **Policy management** → **Deployment jobs**.

- If the state of a WBSSPI Discovery policy is *Active*, then the policy is still being deployed. Wait for the deployment of the policy to complete.
- If the state of a WBSSPI Discovery policy is *Suspended* or *Error*, then check for any error messages in the message browser and continue to troubleshoot the problem by reading the rest of this section.
- If the WBSSPI Discovery policies are not listed, check the message browser for the following message:

```
WASSPI-502: INFO - Updating the WBSSPI configuration data with
discovered information
```

If this message is present and the letter “S” (for successful) appears in the **A** column, the WBSSPI Discovery policies are successfully deployed. If this message is not present or the letter “F” appears in the **A** column, the WBSSPI Discovery policies are not successfully deployed.

Continue to troubleshoot the problem by reading the rest of this section.

- Verify the WebSphere Application Server status. The application server must be working. For more information, see [Task 2: Verify the Application Server Status](#) on page 15.
- If the WebSphere application server is not installed in the default path configure the HOME property using the non-default installation path (for more information about setting the properties, see the WebSphere SPI online help).

The default installation path for the WebSphere application server is:

Operating System	Default Installation Path
AIX	/usr/WebSphere/AppServer
HP-UX, Linux, Solaris	/opt/WebSphere/AppServer
Windows	C:\Program Files\WebSphere\AppServer

- Verify that the Configure WBSSPI tool is not running or a configuration is not open in an editor. Only one process can access a configuration at a time. If a configuration is open, other processes that must access that file (like the discovery policy) hang until the file becomes available.
- If the service map is not updated and the Medium-Impact policy group is not deployed, do the following:
 - Verify the WebSphere application server status. The application server must be running. See [Task 2: Verify the Application Server Status](#) on page 15 for more information.

- Verify that the WebSphere Admin Server is running on the managed node. If you can start the WebSphere Admin Console, the WebSphere Admin Server is running.

Manually Deploying the Discovery Policies

If the WBSSPI Discovery policies are not deployed successfully when you run the Discover WebSphere tool, you can manually deploy the policies on the managed nodes on which the WebSphere Admin Servers are running (you *must* deploy the policies in the given order only):

- 1 From the HPOM console, select **Operations Manager** → **Policy management** → **Policy groups** → **SPI for WebSphere** → **WBSSPI Discovery**.
- 2 Right-click **WBSSPI-Messages** and select **All Tasks** → **Deploy on**. The Deploy Policies on... window opens.
- 3 Select the nodes on which to deploy the auto-discovery policies and click **OK**.
- 4 Right-click **WBSSPI Service Discovery** and select **All Tasks** → **Deploy on**. The Deploy Policies On... window opens.
- 5 Select the nodes on which you want to deploy the auto-discovery policies and click **OK**.

Verifying the Node Name

Verify that the node name specified in a node or group block matches the primary node name configured in HPOM. To display the primary node name, follow these steps:

- a From the HPOM console, select **Operations Manager** → **Nodes**.
- b Right-click the node and select **Properties**.
- c Select the **Network** tab.

Troubleshooting the Tools

Message Configuration variable SERVER<n>_START_CMD missing for server "Default Server"

Solution To successfully run the Start WebSphere tool, you must set the START_CMD and USER properties. Set these properties using the Configure WBSSPI tool. For more information about this tool, see the WebSphere SPI online help.

Message Configuration variable SERVER<n>_STOP_CMD missing for server "Default Server"

Solution To successfully run the Stop WebSphere tool, you must set the STOP_CMD and USER properties. Set these properties using the Configure WBSSPI tool. For more information about this tool, see the WebSphere SPI online help.

Problem When launching the tools, the tools hang or there is no output.

Solution The tools will not work if the memory is low. Check the performance of the node and the management server. The physical memory available must be more than 500 MB.

Problem Verify tool lists files and directories related to the management server as missing. For example:

```
/MGMT_SERVER/SPI-Share/wasspi/wbs/bin/parseDefs.pl  
/MGMT_SERVER/SPI-Share/wasspi/wbs/bin/  
processWASSPIDiscovMsg.pl  
  
/MGMT_SERVER/SPI-Share/wasspi/wbs/conf
```

Solution This is a known problem. The verify tool lists management server related files if you install the WebSphere Application Server on the management server itself. This problem occurs if both the managed node and the management server are the same.

Problem Check WebSphere tool does not give any output.
Solution Ensure that the Collector is running for the WebSphere Application Server instance on that node.

Problem When launched, the Verify tool gives improper output.
Solution Before you launch the Verify tool ensure that you installed the latest version of Self-Healing Service (SHS) component from the SPI DVD.

Problem When launched, the Self-Healing Info tool gives improper output.
Solution Ensure that you installed the latest version of Self-Healing Service (SHS) component from the SPI DVD.

Glossary

agent

A program or process running on a remote device or computer system that responds to management requests, performs management operations, or sends performance and event notification. An agent can provide access to managed objects and MIB variables, interpret policy for resources and configure resources.

application

Packaged software that provides functionality designed to accomplish a set of related tasks. An application is generally more complex than a tool.

ASCII

American Standard Code for Information Interchange.

assigned policy

A policy assigned to one or more resources in the computing environment but not yet deployed or installed on those resources.

automatic action

A pre-configured program or script executed in response to an event, message, or a change in information in the management database. without operator intervention.

client

When the context is network systems, a computer system on a network that accesses a service from another computer (server). When the context is software, a program or executable process that requests a service from a server.

client console

An instance of the user interface that appears on the client system while the application runs on a server.

command

An instruction to a computer program that causes a specified operation to be carried out. Commands are typically typed by users on a command line.

configuration

In a network context, the complete set of inter-related systems, devices and programs that make up the network. For example the components of a network may include computer systems, routers, switches, hubs, operating systems and network software. The configuration of the network determines the way that it works and the way that it is used. In a software context, the combination of settings of software parameters and attributes that determine the way the software works, the way it is used, and how it appears.

configuration file

A file that contains specifications or information that can be used for determining how a software program should look and operate.

connection

A representation of a logical or physical relationship between objects.

console

An instance of the user interface from which the user can control an application or set of applications.

customization

The process of designing, constructing or modifying software to meet the needs and preferences of a particular customer or user.

data type

A particular kind of data; for example database A repository of data that is electronically stored. Typically databases are organized so that data can be retrieved and updated.

deploy

To install and start software, hardware, capabilities, or services so that they work in the business environment.

deployed application

An application and its components that have been installed and started to work in the business environment.

deployed policy

A policy that is deployed on one or more resources in the computing environment.

deployment

The process of installing and activating software, hardware, capabilities or services so that they work in the business environment.

deployment package

A software package that can be deployed automatically and installed on a managed node.

error log

An output file containing error messages.

event

An unsolicited notification such as an SNMP trap or WMI notification generated by an agent or process in a managed object or by a user action. An event usually indicates a change in the state of a managed object or cause an action to occur.

Hypertext Transfer Protocol (HTTP).

The protocol that World Wide Web clients and servers use to communicate.

HTTPS

Hypertext Transfer Protocol Secure.

icon

An on-screen image that represents objects that can be monitored or manipulated by the user or actions that can be executed by the user.

managed object

A network, system, software or service object that is both monitored for performance, status and messages and is manipulated by means of actions in the management software.

management console

An instance of the user interface from which the user can control the management application or set of management applications. The console may be on the system that contains the management software or it may be on another system in the management domain.

management server

A server that provides management services, processes, or a management user interface to clients. A management server is a type of management station.

message

A structured, readable notification that is generated as a result of an event, the evaluation of one or more events relative to specified conditions, or a change in application, system, network, or service status.

message browser

A graphical user interface that presents notifications that are generated as a result of an event, the evaluation of one or more events relative to specified conditions or a change in application, system, network, or service status.

message description

Detailed information about an event or message.

message key

A message attribute that is a string used to identify messages that were triggered from particular events. The string summarizes the important characteristics of the event. Message keys can be used to allow messages to acknowledge other messages, and allows for the identification of duplicate messages.

message severity level

A property of a message indicating the level of impact of the event or notification that initiated the message. See also severity level.

metadata

Data that defines data.

metric

A measurement that defines a specific operational or performance characteristic.

module

A self-contained software component that performs a specific type of task or provides for the presentation of a specific type of data. Modules can interact with one another and with other software.

node

When the context is network, a computer system or device (for example, printer, router, bridge) in a network. When the context is a graphical point to point layout, a graphical element in a drawing that acts as a junction or connection point for other graphical elements.

parameter

A variable or attribute that may be given an arbitrary value for use during an execution of either a computer program or a procedure within a program.

parameter type

An abstraction or categorization of a parameter that determines the particular kind of data that is valid for the parameter. For example a parameter type could be IP Address which indicates that parameter values must have four numbers separated by decimals with the value for each number being in the range of 0 to 255.

parameter value

A value given to a variable.

policy

A set of one or more specifications rules and other information that help automate network, system, service, and process management. Policies can be deployed to various targets (for example, managed systems, devices, network interfaces) providing consistent, automated administration across the network.

policy management

The process of controlling policies (for example, creating, editing, tracking, deploying, deleting) for the purposes of network, system or service management.

policy type

An abstraction or categorization of policies based on the function of the policy or the services that the policy supports.

port

If the context is hardware, a location for passing information into and out of a network device. If the context is ECS, a location for passing information into and out of a correlation node.

server

If the context is hardware plus software, a computer system that provides a service (for example, management capabilities, file storage capabilities) to other computer systems (clients) on the network. If the context is a software component, a program or executable process that responds to and services requests issued by clients.

severity level

A property of an object indicating the status of the object. Severity level is based on the impact of events or messages associated with the object.

Smart Plug-in (SPI)

Prepackaged software that installs into a management console and provides management capabilities specific to a given type of business application, database, operating system, or service.

trace log

An output file containing records of the execution of application software.

Index

A

- attributes, 31
 - Reset, 31
 - Short-term peaks, 31
 - Threshold limit, 31
- Automatic Command reports, 42

B

- basic policy customizations, 30

C

- change collection interval, 39
 - metrics, 39
- CODA, 49
- collection intervals
 - changing for all servers, 39
 - changing for selected metrics, 39
- collector/analyzer
 - what it does, 28
- collector policy
 - description of, 28
- create new policy group, 34
- create the new policy group, 41
- customizations
 - creating new policies, 41

D

- defaults, restoring policy, 42
- deinstallation
 - removing WebSphere SPI, 13
- directories
 - locations for trace file/error logs, 79

F

- files, locations on management server/managed nodes, 78, 79

G

- graphs
 - for UDMs, 76
 - generating with Reporter's graphing capabilities, 59
 - instructions for manually generating, 60
 - list of metrics for server status graph, 61
 - OVPM, 49
 - policies available for, 61
 - showing alarm conditions, 59

H

- HP Performance Manager, using WebSphere SPI with, 49
- HP Reporter, integrating WebSphere SPI with, 49

L

- Linux, monitoring WebSphere installed on, 44
- Log, 78
- Log and Trace files, 78

M

- managed nodes, 9
- Manually Generated Reports, 43
- messages
 - policy configuration for, 33
- Message Source policy groups, description of WebSphere SPI groups, 27
- metric element attributes for creating user defined metrics (UDMs), 67
- metric policies
 - description of, 28
- metrics, 39
 - modifying collections in the collector policy, 35

O

- operator actions
 - graphs generated from, 59

P

- performance impact rating, metrics in tools, 43
- Performance Manager, using WebSphere SPI with, 49
- policies
 - changing, 30
 - customizing message displayed for alerts, 33
 - customizing message text, 33
 - customizing thresholds, 31
 - customizing with the tag option, 41
 - deploying, 19
 - description of, 28
 - modifying, 30
 - re-installing defaults, 42
- policy groups
 - changing collection intervals, 39
 - creating custom with the tag parameter, 41
 - description, 27
- Polling Interval, 39
- proxy configured monitoring, 44

R

- remote monitoring
 - requirements for, 45
- remote systems
 - setup procedure for monitoring, 46
- remote systems, monitoring, 44
- remove WebSphere SPI Grapher package, 63
- remove WebSphere SPI Reporter Package, 59
- removing WebSphere SPI software, 13
- Reporter
 - integrating WebSphere SPI to work with, 49
 - setting up WebSphere SPI to work with, 52
- reports
 - Automatic Command, 42
 - generated from HP Reporter, 55
 - OVPI, 49
 - Reporter, 49
 - using HP Reporter to generate, 49

S

- scheduled metrics, 39
- Self-Healing Info tool, 77
- servers
 - setting thresholds for different, 40

T

- tag option
 - creating custom policy groups with, 41
- Text-Based Reports, 42
- thresholds
 - customizing, 31
 - exceeded
 - viewing graphs resulting from, 59
 - settings for different servers, 40

U

- UDMs, *please see user defined metrics*
- unsupported platforms, monitoring WebSphere on, 45
- upgrading WebSphere SPI, 11
- user defined metrics
 - graphing, 76
 - metric definitions element, description of, 66
 - metric element, description of, 67
 - metric element attributes, description of, 67
 - PMI counter element, description of, 68
 - sample XML file for, 72

V

- verify discovery process, 21

W

- wasspi_wbs_ca command, 35
- WebSphere SPI
 - removing, 13
 - upgrading, 11