HP Operations Smart Plug-in for User Defined Metrics

For HP Operations Manager for Windows®

Software Version: 7.00

PDF version of the online help

This document is a PDF version of the online help that is available in the JMX Metric Builder. It is provided to allow you to print the help, should you want to do so. Note that some interactive topics are not included because they will not print properly, and that this document does not contain hyperlinks.

Document Release Date: December 2009

Software Release Date: December 2009

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

Trademark Notices

Adobe®, Acrobat®, and PostScript® are trademarks of Adobe Systems Incorporated.

AMD is a trademark of Advanced Micro Devices, Inc.

Intel®, Itanium®, and Pentium® are trademarks of Intel Corporation in the U.S. and other countries.

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

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

Windows Vista™ is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

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

TABLE OF CONTENTS

The JMX Metric Builder

The JMX Metric Builder (JMB) is used to edit a UDM file by mapping MBeans to UDMs, validating metric IDs, and creating user defined metrics (UDMs) that conform to the metric definitions DTD. The JMB can also be used to browse MBeans registered in an MBean server.

To use the JMB, it helps to be familiar with the metric definitions DTD .

Information about metrics, metric IDs, the JMB interface, how to use the JMB, and options for the JMB when invoked from the command line are available in the JMB's online help.

For more information about installing the JMB, refer to the *JMX Metric Builder Release Notes* located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\Releasenotes\JMX_Metric_Builder_ReleaseNotes.html`.

Related Topics:

- User defined metrics
- The MBean server environment
- Components

The JMX Metric Builder

Before you can start using the JMB and creating UDMs, the WebSphere SPI, WebLogic SPI, or Oracle AS SPI (for Oracle AS Version 10g R3 only) and other software must be installed and configured.

For more information about installing the JMB, refer to the *JMX Metric Builder Release Notes* located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\Releasenotes\JMX_Metric_Builder_ReleaseNotes.html`.

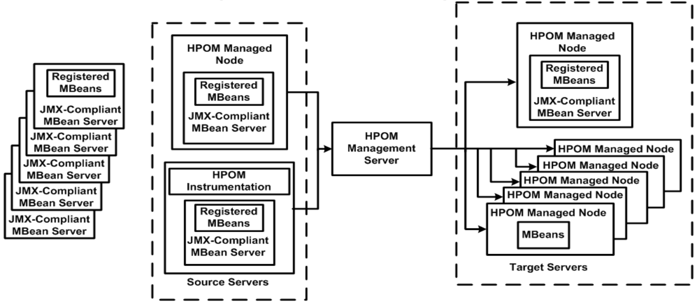
Related Topics:

- Configuring the MBean server environment
- Removing the JMX Metric Builder

The MBean server environment

You can use JMX Metric Builder to monitor MBeans registered in the WebLogic, WebSphere, and Oracle MBean Servers. The HPOM management server must be configured to gather MBean data (also referred to as metadata) from source servers and to monitor target servers using UDMs and policies created for the UDMs.

(Click the image to make it larger or smaller.)



Source servers are systems on which the WebLogic, WebSphere, or Oracle MBean server resides. The HPOM instrumentation must be distributed to the source servers (the HPOM management server may not monitor them). The source servers can be an MBean staging area or development server.

The HPOM management server collects MBean data from the source servers. Select a subset of your MBean servers (those that have a representative set of MBeans registered) to be your source servers.

Target servers are systems that are monitored by the HPOM management server. The target server is usually a production server. Alarms, graphs, and reports are generated by UDMs based on MBeans registered in the WebLogic SPI, WebSphere SPI, or Oracle AS SPI (version 10gR3 only) MBean server.

Related Topics:

- Configuring the MBean server environment
- User defined metrics
- Developing UDMs
- Components

Configuring the MBean server environment

Before you can create UDMs using the JMB , your environment must be configured so that MBean information (metadata) is collected from your MBean server(s). Follow the tasks required for the type of MBean server you are using.

Using the WebLogic, WebSphere, or Oracle MBean server

Both the WebLogic, WebSphere, and Oracle application servers include a built-in MBean server. If you are using the WebLogic, WebSphere, or Oracle MBean server, do the following;

1. **Install the SPI and JMB.**

   Refer to the following documentation for more information:

   - *JMX Metric Builder Release Notes* (located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\Releasenotes\JMX_Metric_Builder_ReleaseNotes.html`)
   - *HP Operations Smart Plug-in for WebSphere Application Server Installation and Configuration Guide* (located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\SPI Guides\WebSphere_AppServer_Install_Config.pdf`)
   - *HP Operations Smart Plug-in for BEA WebLogic Server Installation and Configuration Guide* (located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\SPI Guides\WebLogic_AppServer_Install_Config.pdf`)
   - *HP Operations Smart Plug-in for Oracle Application Server Installation and Configuration Guide* (located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\SPI Guides\Oracle_AppServer_Install_Config.pdf`)

2. **Configure the SPI.**

   Refer to the following documentation for more information:

   - *HP Operations Smart Plug-in for WebSphere Application Server Installation and Configuration Guide* (located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\SPI Guides\WebSphere_AppServer_Install_Config.pdf`)
   - *HP Operations Smart Plug-in for BEA WebLogic Server installation and Configuration Guide* (located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\SPI Guides\WebLogic_AppServer_Install_Config.pdf`)

   - *HP Operations Smart Plug-in for Oracle Application Server Installation and Configuration Guide* (located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\SPI Guides\Oracle_AppServer_Install_Config.pdf`)

3. Configure your MBean server environment .
4. Develop UDMs .

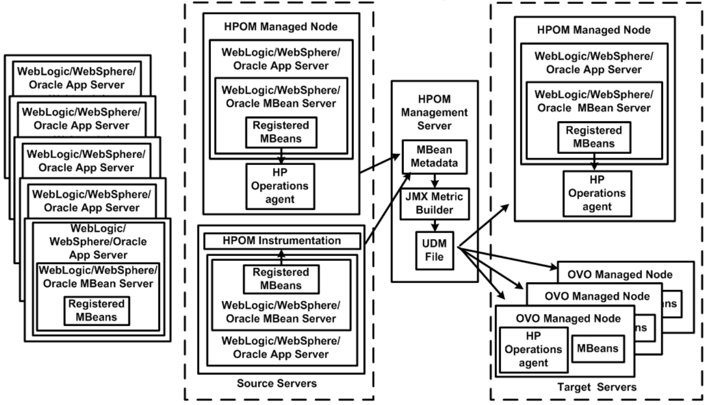
Related Topics:

- MBean server environment

Configuring the WebLogic/WebSphere/Oracle MBean server environment

If you are using the MBean server that comes with the WebLogic, WebSphere, or Oracle application server (version 10gR3 only), your MBean server environment may look similar to the following:

(Click the image to make it larger or smaller.)


WebLogic/WebSphere/ Oracle App Server
WebLogic/WebSphere/ Oracle App Server
WebLogic/WebSphere/ Oracle App Server
WebLogic/WebSphere/ Oracle App Server
WebLogic/ WebSphere/Oracle App Server
WebLogic/WebSphere/ Oracle MBean Server
Registered MBeans
HPOM Managed Node
WebLogic/WebSphere/ Oracle App Server
WebLogic/WebSphere/ Oracle MBean Server
Registered MBeans
HP Operations agent
HPOM Instrumentation
Registered MBeans
WebLogic/WebSphere/ Oracle MBean Server
WebLogic/WebSphere/ Oracle App Server
Source Servers
HPOM Management Server
MBean Metadata
JMX Metric Builder
UDM File
HPOM Managed Node
WebLogic/WebSphere/ Oracle App Server
WebLogic/WebSphere/ Oracle MBean Server
Registered MBeans
HP Operations agent
OVO Managed Node
ns
OVO Managed Node
ns
OVO Managed Node
HP Operations agent
MBeans
Target Servers

To configure this MBean server environment, complete the following steps:

1. **Register custom MBeans(optional)**

   Custom MBeans must be registered in a JMX-compliant MBean server if you want to monitor and collect its data.

   If you are using the WebLogic or Oracle MBean server, the Name attribute is used to identify its MBeans. If your MBean is a multi-instance MBean, then each MBean instance must have a unique value in its Name attribute. For example, WebLogic's ServletRuntime MBeans are multi-instance because a ServletRuntime MBean is instantiated by WebLogic for each deployed servlet. The Name attribute of the MBean identifies the servlet that the MBean is monitoring. If the Name attribute is not provided, then the full ObjectName is used as the instance identifier.

   If you are using the WebSphere MBean server, the mbeanIdentifier ObjectName key property is used to identify its MBeans. If your custom MBean is a multi-instance MBean, then each MBean instance must have a unique value in its mbeanIdentifier ObjectName key property. If the mbeanIdentifier ObjectName key property is not provided, then the full ObjectName is used as the instance identifier.

   For any other JMX-compliant MBean server, the full ObjectName is used as the MBean instance identifier.

Refer to your JMX-compliant server documentation for information about creating and registering MBeans.

JMX specifications are located at:
http://java.sun.com/products/JavaManagement/reference/docs/index.html

2. Configure the WebLogic SPI/WebSphere SPI/Oracle AS SPI (version 10g R3 only) software on the source and target servers (refer to chapter 3 of the SPI Installation and Configuration Guide for more information).

   If you do not want to monitor the source server, do not deploy the SPI policies to the source server during the SPI configuration process.

3. **Set the COLLECT_METADATA server property of the source server to "ON."**

   This example shows the steps using the WebLogic SPI; if you have installed the WebSphere SPI or Oracle AS SPI (version 10gR3 only), change any occurrence of "WLSSPI" to "WBSSPI" or "OASSPI" and "WebLogic Server" to "WebSphere" or "Oracle Server";

   1. From the HPOM console, select **Operations Manager** → **Tools** → **SPI for WebLogic Server** → **WLSSPI** → **WLSSPI - SPI Admin** .
   2. Double-click **Discover or Configure WLSSPI** .
   3. Select the source server.
   4. Click **Launch** .
   5. Select Launch Configure Tool radio button and click **OK** .
   6. Click **Next** in the Introduction window.
   7. From the configuration editor, set the COLLECT_METADATA property to "ON" for the source server.

      Select the help button to display more information about using the editor.
   8. Select **Save** .
   9. Select **Next** .
   10. Select **OK** .

4. Run the Gather Metadata tool.

   To gather MBean information, run the Gather Metadata tool .

   The COLLECT_METADATA property must be set to ON for the managed node on which an

MBean server is running (the source server). MBean information is collected from these managed nodes only. The instructions that follow are for the WebLogic SPI. If you have installed the WebSphere SPI or Oracle AS SPI (version 10gR3 only), replace any occurrence of "WLSSPI" with "WBSSPI" or "OASSPI" and "wls" with "wbs" or "oas".

1. From the HPOM console, select **Tools** → **JMX Metric Builder** → **WLSSPI** .
2. Double-click **Gather Metadata** .
3. Select the managed nodes on which you want to gather MBean information.
4. Select **Launch** .

5. Add a service and associate a message with the service (optional).
   Refer to parts 13 and 14 of the HPOM Basic Training .

Related Topics:

- MBean server environment

Removing the JMX Metric Builder

Complete these tasks only if you no longer want to create nor use UDMs and do not want to use the JMX Metric Builder.

Complete the tasks in the order listed:

1. **Remove the JMB 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** .
   3. In the Program Maintenance window, select **Remove products** .
   4. In the Product Selection window, select the check box next to **JMX Metric Builder** and select **Next** .
   5. Complete the installation by following the instructions in the windows that display.

2. **Uninstall custom policies.**

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

   1. In the console tree, select Policy management → Policy groups.
   2. If your custom policies reside in a custom policy group, right-click the custom policy group and select All Tasks → Uninstall from. A node selection window appears.

      If your custom policies reside in a default policy group (such as SPI for WebSphere, right-click a custom policy and select All Tasks → Uninstall from. A node selection window appears.
   3. Select the nodes on which the policy/policies are installed.
   4. Select **OK** . Repeat steps b - d for each custom policy group or policy.
   5. Verify the policies are uninstalled. Check the status of the job in Deployment jobs under Policy groups. All custom policies must be uninstalled before you start the next task.

3. **Remove JMX Metric Builder tools.**

   1. Right-click **Tools** in the console tree and select **Configure** → **Tools** to display the Configure Tools dialog box.
   2. Right-click the JMX Metric Builder tool group and select **Delete** . A message prompts you to confirm the delete operation.

4. Delete custom services.
5. **Remove custom node groups on the management server.**

   1. In the console tree, select **Nodes** .
   2. Open the Node Configuration editor:
      1. Select the Nodes folder in the console tree.
      2. Click 🖧 on the Configuration toolbar to open the editor.
   3. In the Nodes list, either select the name of the node group you want to delete and press the Delete key or right-click the node group and select Delete.
   4. Click **Yes** to continue the delete operation.

Related Topics:

- Configuring the MBean server environment

User defined metrics

User defined metrics (UDMs) are metrics created by the user to gather data from application MBeans registered in a BEA WebLogic MBean server and Oracle Application Server (version 10gR3 only), and WebSphere application server. You can create UDMs by using the JMX Metric Builder (JMB) or by editing the metrics definition XML file. The metrics definitions XML file is also referred to as the UDM file.

To monitor your applications using HP Operations Manager (HPOM), you configure your MBean server environment and create policies to monitor and collect the data generated by the UDMs.

Related Topics:

- Developing UDMs
- The MBean server environment
- Configuring the MBean server environment
- Components

Components

Additional components are required to create UDMs . Some of these components require additional configuration.

For more information about installing the JMB, refer to the *JMX Metric Builder Release Notes* located on the HP Operations Smart Plug-ins DVD in the file `\Documentation\Releasenotes\JMX_Metric_Builder_ReleaseNotes.html`.

The JMX Metric Builder tool

You can use the JMX Metric Builder (JMB) is used to edit the UDM file by mapping MBeans to UDMs, validating metric IDs, and creating UDMs that conform to the metric definitions DTD. It can also be used to browse MBeans on a configured MBean server.

The JMB is installed with the JMB software.

The Gather Metadata tool

The Gather Metadata tool gathers MBean information from selected managed nodes and allows you to gather the MBean data at any time. The COLLECT_METADATA property for the selected managed nodes must be set to ON to gather MBean information from those selected managed nodes.

The Gather Metadata tool is installed with the JMB software.

WebSphere SPI, WebLogic SPI, and Oracle AS SPI (version 10gR3 only) Installation and Configuration Guides

The latest versions of the WebSphere SPI, WebLogic SPI, Oracle AS SPI (version 10gR3 only) and Installation and Configuration Guides can be found online at: http://h20230.www2.hp.com/selfsolve/manuals .

1. In the select product list, click either Smart Plug-ins for BEA WebLogic server or Smart Plug-ins for WebSphere or Smart Plug-ins for Oracle Application Server.
2. In the select version list, click a version.
3. In the select OS list, click an OS.
4. In the select the product manual title list, click Installation and Configuration Guide.

5. Click **Open** to view the guide.

Related Topics:

- User defined metrics
- Developing UDMs
- The MBean server environment
- Configuring the MBean server environment

Developing UDMs

After you install and configure the software, configured your source and target servers , and MBean data has been collected from your MBean server(s), you are ready to create and monitor UDMs .

To create and monitor UDMs, do the following:

1. **Run the JMX Metric Builder (JMB).**
   View registered MBeans and create UDMs based on these MBeans.

   The instructions that follow are for the WebLogic SPI. If you have installed the WebSphere SPI or Oracle AS SPI (version 10gR3 only), replace any occurrence of "WLSSPI" with "WBSSPI" or "OASSPI" and "wls" with "wbs" or "oas".

   1. From the HPOM console, select **Tools** → **JMX Metric Builder** → **WLSSPI** .
   2. Double-click **JMX Metric Builder** .

   Complete the following tasks to create a UDM using the JMB (refer to the JMB's online help for more information about these tasks):

   - Load metadata/MBean information
   - Organize MBeans (optional)
   - Open the UDM file
   - Add a metric
   - Change metric visibility (optional)
2. **Deploy the UDM file.**
   Deploy the UDM file from the HPOM management server to the selected managed node(s).

   The instructions that follow are for the WebLogic SPI. If you have installed the WebSphere SPI or Oracle AS SPI (version 10gR3 only), replace any occurrence of "WLSSPI" with "WBSSPI" or "OASSPI" and "wls" with "wbs" or "oas".

   1. From the HPOM console, select **Tools** → **JMX Metric Builder** → **WLSSPI** .
   2. Double-click **Deploy UDM** .
   3. Select the managed nodes on which you want to deploy UDMs.

    4. Select **Launch** .

3. **Create a UDM policy group and policies.**
   Create a UDM policy group to simplify and customize policy distribution, create policies that monitor the UDMs created in step 1, and create policies that define the collection interval (how often the metrics are collected and monitored).

JMX Metric Builder tools group

The JMX Metric Builder Admin tools group allows the HPOM administrator to perform routine tasks relating to the JMB.

For a more detailed description of the tools, click the tool name in the table below.

Tool	Description
Deploy UDM	Deploy the UDM file.
Gather Metadata	Gather MBean information.
JMX Metric Builder	Launch the JMB.
UDM Graph Enable/Disable	Start/Stop data collection for UDM graphs.

Deploy UDM

The Deploy UDM tool deploys the UDM file from the management server to the selected managed node(s). UDMs allow you to define your own metrics and monitor applications registered with the WebSphere/WebLogic/Oracle MBean server.

Function

Deploys the UDM file from the management server to the `<OvAgentDir> /wasspi/wbs/conf/wbs_MetricDefinitions.xml` or `<OvAgentDir> /wasspi/wls/conf/wls_MetricDefinitions.xml` or `<AgentDir> /wasspi/oas/conf/oas_MetricDefinitions.xml` file on the selected managed nodes where

- `<OvAgentDir>` is typically `/var/opt/OV/` on HP-UX, Linux, Solaris, and Tru64 managed nodes, `/var/lpp/OV/` on AIX managed nodes, and `\Program Files\HP\HP BTO Software\` on Windows managed nodes.
- *`NAME`* and *`ALIAS`* are the properties set for the managed node and *`ALIAS`* is always used if it is set.

All XML files on the management server in the `<ShareInstallDir> \SPI-Share\wasspi\wbs\conf\workspace\UDMProject\` or `<ShareInstallDir> \SPI-Share\wasspi\wls\conf\workspace\UDMProject\` or `<ShareInstallDir> \SPI-Share\wasspi\oas\conf\workspace\UDMProject\` directory are combined to form a single UDM file where

- `<ShareInstallDir>` is the value of the registry key HKEY_LOCAL_MACHINE\\SOFTWARE\\Hewlett-Packard\\HP OpenView\\OvOWShareInstallDir.
  Where `<ShareInstallDir>` by default is `C :\Documents and Settings\All Users\Application Data\HP\HP BTO Software\shared`.

If duplicate metric IDs are found in the UDM file, the UDM file is not deployed until the duplicate IDs are modified or removed.

To launch the Deploy UDM tool

The instructions that follow are for the WebLogic SPI. If you have installed the WebSphere SPI or Oracle AS SPI (version 10gR3 only), replace any occurrence of "WLSSPI" with "WBSSPI" or "OASSPI" and "wls" with "wbs" or "oas".

1. From the HPOM console, select **Tools** → **JMX Metric Builder** → **WLSSPI** .
2. Double-click **Deploy UDM** .
3. Select the managed nodes on which you want to deploy UDMs.
4. Click **Launch** .

Related Topics:

- Gather Metadata
- JMX Metric Builder
- UDM Graph - Start/Stop

Gather Metadata

The Gather Metadata tool gathers MBean information from all managed nodes whose COLLECT_METADATA property is set to ON. This information is saved in a cache on the HPOM management server.

JMX Metric Builder tool displays MBean information so that you can create UDMs.

Required setup

- The COLLECT_METADATA property for the selected managed nodes must be set to ON to gather MBean information from those selected managed nodes.
- The application server must be running
- The monitoring must be ON

Function

Collects MBean information and saves it to a cache on the HPOM management server.

Initially, the MBean information is saved in an XML file on the managed node in `<`*`OvAgentDir`*`> /wasspi/wbs/tmp/<`*`NAME`*` | `*`ALIAS`*`> .xml` or `<`*`OvAgentDir`*`> /wasspi/wls/tmp/<`*`NAME`*` | `*`ALIAS`*`> .xml` or `<`*`OvAgentDir`*`> /wasspi/oas/tmp/<`*`NAME`*` | `*`ALIAS`*`> .xml`
where

- `<`*`OvAgentDir`*`>` is typically `/var/opt/OV/` on HP-UX, Linux, Solaris, and Tru64 managed nodes, `/usr/lpp/OV/` on AIX managed nodes, and `\Program Files\HP\HP BTO Software\` on Windows managed nodes.
- *`NAME`* and *`ALIAS`* are the properties set for the managed node and *`ALIAS`* is always used if it is set.

After Gather Metadata has collected the MBean information for a managed node, the MBean information is transferred to the HPOM management server and is saved in a cache file named `< > \SPI-Share\wasspi\wbs\metadata\<`*`managed_node`*`> \<`*`NAME`*` | `*`ALIAS`*`> .xml` or `<`*`ShareInstallDir`*`> \SPI-Share\wasspi\wls\metadata\<`*`managed_node`*`> \<`*`NAME`*` | `*`ALIAS`*`> .xml` or `<`*`ShareInstallDir`*`> \SPI-Share\wasspi\oas\metadata\<`*`managed_node`*`> \<`*`NAME`*` | `*`ALIAS`*`> .xml`
where

- `<`*`ShareInstallDir`*`>` is the value of the registry key

HKEY_LOCAL_MACHINE\\SOFTWARE\\Hewlett-Packard\\HP OpenView\\OvOWShareInstallDir.
The registry key could be: `C :\Documents and Settings\All Users\Application Data\HP\HP BTO Software\shared` .

The XML file on the managed node is deleted.

If a cache file on the HPOM management server is no longer collected, it is automatically deleted.

To launch the Gather Metadata tool

The instructions that follow are for the WebLogic SPI. If you have installed the WebSphere SPI or Oracle AS SPI (version 10gR3 only), replace any occurrence of "WLSSPI" with "WBSSPI" or "OASSPI" and "wls" with "wbs" or "oas".

1. From the HPOM console, select **Tools** → **JMX Metric Builder** → **WLSSPI** .
2. Double-click **Gather Metadata** .
3. Select the managed nodes on which you want to gather MBean information.
4. Click **Launch** .

Related Topics:

- Deploy UDM
- JMX Metric Builder
- UDM Graph - Start/Stop

JMX Metric Builder

The JMX Metric Builder tool launches the JMB so that you can edit the UDM file and browse MBeans on an MBean server. You must run only one instance of the JMB at a time.

Required setup

Complete the following tasks before running the JMB:

1. Register custom MBeans (optional)
2. Configure your MBean server environment
3. Run the Gather Metadata tool

Function

JMX Metric Builder allows you do the following:

- Load metadata/MBean information
- Organize MBeans
- Add a metric
- Change metric visibility
- Remove a metric

UDMs for all HPOM managed nodes are maintained in a UDM file on the HPOM management server. Use the Deploy UDM tool to distribute the UDM file from the management server to the managed node(s).

To launch the JMX Metric Builder tool

The instructions that follow are for the WebLogic SPI. If you have installed the WebSphere SPI or Oracle AS SPI (version 10gR3 only), replace any occurrence of "WLSSPI" with "WBSSPI" or "OASSPI" and "wls" with "wbs" or "oas".

1. From the HPOM console, select **Tools** → **JMX Metric Builder** → **WLSSPI** .
2. Double-click **JMX Metric Builder** .

Related Topics:

- Deploy UDM
- Gather Metadata
- UDM Graph - Start/Stop

UDM Graph Enable/UDM Graph Disable

The UDM Graph Enable tool starts stops collection for UDM graphs. The UDM Graph Disable tool stops data collection for UDM graphs.

After you configure UDMs, you can collect data that HP Performance Manager (must be purchased separately) can use.

Function

UDM Graph Enable does the following:

- Starts UDM data collection for graphing.

UDM Graph Disable does the following:

- Stops UDM data collection for graphing and deletes the data source.

To launch the UDM Graph Enable or the UDM Graph Disable tool

To restart data collection for graphing, always run UDM Graph Disable first before running UDM Graph Enable.

The instructions that follow are for the WebLogic SPI. If you have installed the WebSphere SPI, replace any occurrence of "WLSSPI" with "WBSSPI" and "wls" with "wbs". If you have installed the Oracle AS SPI (for Oracle AS Version 10gR3 only), replace any occurrence of "OASSPI" with "WBSSPI" and "oas" with "wbs".

1. From the HPOM console, select **Tools** → **JMX Metric Builder** → **WLSSPI** .
2. Double-click **UDM Graph Enable** or **UDM Graph Disable** .
3. Select the managed nodes on which you want to start or stop data collection for UDM graphs.
4. Click **Launch** .

Allow sufficient collection intervals to occur before attempting to view graphs (using HP Performance Manager).

Related Topics:

- Deploy UDM

- Gather Metadata
- JMX Metric Builder

Metric Definitions DTD

The metric definitions DTD files provide the structure and syntax for the UDM XML file. The WebSphere SPI, WebLogic SPI, and Oracle AS SPI (version 10g R3 only) use these files to parse and validate the UDM file.

On a managed node, the DTD files are located in the following directory:

`<`*`OvAgentDir`*`> /conf/wbsspi/` or
`<`*`AgentDir`*`> /conf/wlsspi/` or
`<`*`AgentDir`*`> /conf/oasspi/`

where `<`*`OvAgentDir`*`>` typically is

- `/var/opt/OV/` for HP-UX, Linux, Solaris, and Tru64
- `/var/lpp/OV/` for AIX
- `\Program Files\HP\HP BTO Software\` for Windows

NOTE:
Because the DTD files are used at runtime, you should not edit, rename, or move them.

NOTE:
Edit the UDM file using the JMX Metric Builder (JMB).

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 collector can perform calculations that combine one or more metrics to define a new metric. The result of the calculation is the metric value.

Syntax

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

Calculations use the following rules:

- Operators supported are +, -, /, *, and unary minus.
- Operator precedence and associativity follow the Java model.
- Parentheses can be used to override the default operator precedence.
- Allowable operands are metric IDs and literal doubles.

A metric ID can refer to an MBean metric and another calculated metric. You can specify Literal doubles 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.
- **prev** returns the previous value of the metric.

Examples

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

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

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

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

FromVersion and ToVersion Elements

The FromVersion and ToVersion elements are used to specify the versions of the application server for which the data source element is valid.

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

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

Syntax

```
<!ELEMENT FromVersion (EMPTY)>
<!ELEMENT ToVersion   (EMPTY)>
<!ATTLIST FromVersion server CDATA #REQUIRED
                      update CDATA "*">
<!ATTLIST ToVersion   server CDATA #REQUIRED
                      update CDATA "*">
```

FromVersion and ToVersion element attributes

Attribute	Type/ Values	Required?	Default Value	Description
server	numeric string	yes	N/A	Specifies a primary server version.
update	numeric string	no	"*"	Specifies a secondary server version, such as "1" for service pack 1. A "*" indicates that no secondary version is specified.

Example

```
<FromVersion server="6.0"/>
<ToVersion server="6.999"/>
```

Get Element

The Get element returns the value of the specified attribute.

The Get element contains the following element:

- **Attribute** - The MBean attribute name.

Syntax

```
<!ELEMENT Get       (Attribute)>
<!ATTLIST Get       id      ID #IMPLIED>
<!ELEMENT Attribute (#PCDATA)>
```

Get element attributes

Attribute	Type/ Values	Required?	Default Value	Description
id	ID	no	N/A	A unique identifier for this element.

Example

```
<Get>
  <Attribute>MessagesMaximum</Attribute>
</Get>
```

Invoke Element

The Invoke element executes an MBean operation with the given parameters.

The Invoke element contains the following elements:

- **Operation** - The MBean operation to be executed.
- **Parameters** - The MBean operation parameter values. You must specify parameters for operations which accept parameters.

Syntax

```
<!ELEMENT Invoke     (Operation, Parameters?)>
<!ATTLIST Invoke     id         ID #IMPLIED>
<!ELEMENT Operation  (#PCDATA)>
<!ELEMENT Parameters (Parameter)+>
<!ELEMENT Parameter  (Numeric | String | BOOLEAN)>
<!ELEMENT Numeric    (Formula)>
<!ATTLIST Numeric    type       (short | int | long | double | float |
                                java.lang.Short | java.lang.Integer |
                                java.lang.Long | java.lang.Double |
                                java.lang.Float) #IMPLIED>
<!ELEMENT String     EMPTY>
<!ATTLIST String     value      CDATA #REQUIRED>
<!ELEMENT Boolean    EMPTY>
<!ATTLIST Boolean    value      (true | false) #REQUIRED>
```

Invoke element attributes

Attribute	Type/ Values	Required?	Default Value	Description
id	ID	no	N/A	A unique identifier for this element.

Example

```
<Invoke>
  <Operation>stagingEnabled</Operation>
  <Parameters>
    <Parameter>
      <String value="examplesServer"/>
    </Parameter>
  </Parameters>
</Invoke>
```

JMXAction Element

The JMXAction element contains one or more JMXCalls elements and all are executed in the order defined. A JMXAction can optionally be associated with specific versions of the application server using the FromVersion and ToVersion elements.

Syntax

```
<!ELEMENT JMXAction (FromVersion?, ToVersion?, JMXCalls+)>
<!ATTLIST JMXAction id      ID #IMPLIED>
```

Metric element attributes

Attribute	Type/ Values	Required?	Default Value	Description
id	ID	no	N/A	A unique identifier for this element.

JMXActions Element

The JMXActions element contains one or more JMXAction elements. All elements matching the server version are executed.

Syntax

```
<!ELEMENT JMXActions (JMXAction+)>
<!ATTLIST JMXActions id    ID #IMPLIED>
```

Metric element attributes

Attribute	Type/ Values	Required?	Default Value	Description
id	ID	no	N/A	A unique identifier for this element.

JMXCalls Element

The JMXCalls element contains one or more JMX calls (invoke, get, or set) that operate on a specific MBean or type of MBean. The MBean instance or type is specified by the ObjectName element.

The JMXCalls element contains the following elements:

- **ObjectName** - The JMX-compliant object name of the MBean. The object name can include JMX-compliant pattern matching.
- **Set** , **Get** , and **Invoke** - JMX calls. Refer to each element's help topic for more information.

Syntax

```
<!ELEMENT JMXCalls   (ObjectName, (Set | Get | Invoke)+)>
<!ATTLIST JMXCalls   id          ID #IMPLIED>
<!ELEMENT ObjectName (#PCDATA)>
```

JMXCalls element attributes

Attribute	Type/ Values	Required?	Default Value	Description
id	ID	no	N/A	A unique identifier for this element.

Example

```
<JMXCalls>
  <ObjectName>*:*,Type=JMSServerConfig</ObjectName>
  <Get>
    <Attribute>MessagesMaximum</Attribute>
  </Get>
</JMXCalls>
```

MBean Element

The MBean element is used when the data source of the metric is an attribute of a JMX MBean. The MBean element contains the following elements:

- **ObjectName** - The JMX-compliant object name of the MBean. The object name can include JMX-compliant pattern matching.
- **Attribute** - The MBean attribute name.
- **AttributeValueMapping** (optional) - Numeric values that should be substituted for the values returned by the MBean attribute. This can be used to convert string attributes to numbers so they can be compared to a threshold. Each AttributeValueMapping contains a number of Map elements. Each Map element specifies one value to be mapped.
- **AttributeFilter** (optional) - Provides basic filtering of MBeans based on an MBean attribute.
- **FromVersion/ToVersion** - The versions of the application server for which the MBean element is valid.
- **InstanceId** (optional) - The unique identifier of a multi-instance MBean.

Syntax

```
<!ELEMENT MBean                 (FromVersion?, ToVersion?, InstanceId?, ObjectName,
                                Attribute, AttributeValueMapping?,
                                AttributeFilter*)>
<!ATTLIST MBean                 instanceType (single | multi) "single"
                                dataType     (numeric | string | BOOLEAN)
                                             "numeric">
<!ELEMENT ObjectName            (#PCDATA)>
<!ELEMENT Attribute             (#PCDATA)>
<!ELEMENT InstanceId            (ObjectNameKey | Attribute)>
<!ELEMENT ObjectNameKey         (#PCDATA)>
<!ELEMENT AttributeValueMapping (Map+)>
<!ELEMENT Map                   EMPTY>
<!ATTLIST Map                   from         CDATA #REQUIRED
                                to           CDATA #REQUIRED >
<!ELEMENT AttributeFilter       EMPTY>
```

```
<!ATTLIST AttributeFilter       type        (include | exclude) "include"
                                name        PCDATA #REQUIRED
                                operator    (initialSubString | finalSubString |
                                            anySubString | match | gt | geq | lt |
                                            leq | eq) #REQUIRED
                                value       CDATA #REQUIRED >
```

MBean element attributes

Attribute	Type/ Values	Required?	Default Value	Description
instanceType	"single" "multi"	no	"single"	Indicates if there could be multiple instances of this MBean.
dataType	"numeric" "string"	no	"numeric"	Indicates if the value returned from the MBean attribute is a string or a numeric value.

Map element attributes

Attribute	Type/ Values	Required?	Default Value	Description
from	text	yes	N/A	The value that is to be mapped.
to	text	yes	N/A	The new metric value to be returned in place of the mapped value.

AttributeFilter element attributes

Attribute	Type/ Values	Required?	Default Value	Description
type	"include" "exclude"	no	"include"	Specifies if an MBean that matches this filter should be included or excluded from consideration by the data collector.
name	text	yes	N/A	Specifies the MBean attribute on which to apply the filter.
operator	"initialSubString" "finalSubString" "anySubString" "match" "gt" "geq" "lt" "leq" "eq"	yes	N/A	Specifies the filter to apply. "initialSubString," "finalSubString," "anySubString," and "match" can be used with MBean attributes that return text values "gt," "geq," "lt," "leq," "eq" can be used for MBean attributes that return numeric values. See the JMX documentation for more information about filtering MBeans.
value	text or number	yes	N/A	Specifies the value to compare. The metric definition creator is responsible for making sure the value data type matches the data type of the corresponding MBean attribute.

InstanceId element subelements

Element	Type/ Values	Required?	Default Value	Description
Attribute	text	no	N/A	Uniquely identifies multi-instance MBeans. Specify this attribute consistently when defining multi-instance metric calculations.
ObjectNameKey	text	no	N/A	Uniquely identifies multi-instance MBeans. Specify this attribute consistently when defining multi-instance metric calculations.

Example

```
<MBean instanceType="multi">
  <FromVersion server="6.0" update="1"/>
  <ObjectName>*:*,Type-ExecuteQueueRuntime</ObjectName>
  <Attribute>PendingRequestCurrentCount</Attribute>
</MBean>
```

MetricDefinitions Element

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

Syntax

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

Metric Element

The Metric element represents one metric. Each metric has a unique ID (for example, WBSSPI_1001 or WLSSPI_1001). If a user defined metric is an alarming, graphing, or reporting metric, the metric ID must be "prefix<*ID_number*> " where prefix is made up of 3-15 letters (case-sensitive), digits, or underscores (“_”) and <*ID_number*> 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, "mbean1").

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

Syntax

```
<!ELEMENT Metric ((MBean+ | Calculation+), JMXActions?)>
<!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 >
```

Metric element attributes

Attribute	Type/ Values	Required?	Default Value	Description
id	ID	yes	N/A	The metric ID.
name	text	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 by using opcmon.
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.

Set Element

The Set element assigns a value to the specified attribute.

The Set element contains the following elements:

- **Attribute** - The MBean attribute name.
- **Value** - The value to assign to the attribute. The value can be a number, string, or BOOLEAN.

Syntax

```
<!ELEMENT Set     (Attribute, Value)>
<!ATTLIST Set     id          ID #IMPLIED>
<!ELEMENT Value   (Numeric | String | BOOLEAN)>
<!ELEMENT Numeric (Formula)>
<!ATTLIST Numeric type        (short | int | long | double | float | java.lang.Short
                              | java.lang.Integer | java.lang.Long |
                              java.lang.Double | java.lang.Float) #IMPLIED>
<!ELEMENT String  EMPTY>
<!ATTLIST String  value       CDATA #REQUIRED>
<!ELEMENT Boolean EMPTY>
<!ATTLIST Boolean value       (true | false) #REQUIRED>
```

Set element attributes

Attribute	Type/ Values	Required?	Default Value	Description
id	ID	no	N/A	A unique identifier for this element.

Example

```
<Set>
  <Attribute>MessagesMaximum</Attribute>
  <Value>
    <Numeric>
```

```
        <Formula>JMSServerConfig_MessagesMaximum + (5-5)</Formula>
      </Numeric>
    </Value>
  </Set>
```

Example 1

Metric 1010 uses metric "mbean1" in its calculation. The MBean for metric "mbean1" was introduced on server version 6.0, service pack 1. In version 6.1, the attribute name changed.

```
<Metric id="mbean1" alarm="no">
  <MBean>
    <FromVersion server="6.0" update="1"/>
    <ToVersion server="6.099"/>
    <ObjectName>*:*,Type=ExecuteQueue</ObjectName>
    <Attribute>ServicedRequestTotalCount</Attribute>
  </MBean>
  <MBean>
    <FromVersion server="6.1"/>
    <ObjectName>*:*,Type=ExecuteQueue</ObjectName>
    <Attribute>ServicedRequestCount</Attribute>
  </MBean>
</Metric>
<Metric id="JMXUDM_1010" alarm="yes">
  <Calculation>
    <Formula>
      (delta(mbean1) / interval(mbean1))*1000)
    </Formula>
  </Calculation>
</Metric>
```

Example 2

Using example 1 above, a decision was made to make metric 1010 a per-minute rate instead of a per-second rate as of server version 7.0. The changes that had to be made to the metric definitions are in bold blue type . Note that the versions supported by the hidden and exposed metrics are not necessarily synchronized.

```
<Metric id="mbean1" alarm="no">
  <MBean>
    <FromVersion server="6.0" update="1"/>
    <ToVersion server="6.099"/>
    <ObjectName>*:*,Type=ExecuteQueue</ObjectName>
    <Attribute>ServicedRequestTotalCount</Attribute>
  </MBean>
  <MBean>
    <FromVersion server="6.1"/>
    <ObjectName>*:*,Type=ExecuteQueue</ObjectName>
    <Attribute>ServicedRequestCount</Attribute>
  </MBean>
</Metric>
```

```
<Metric id="JMXUDM_1010" alarm="yes">
  <Calculation>
    <FromVersion server="6.0"/>
    <ToVersion server="6.999"/>
    <Formula>
      (delta(mbean1) / interval(mbean1))*1000)
    </Formula>
  </Calculation>
  <Calculation>
    <FromVersion server="7.0"/>
    <Formula>
      (delta(mbean1) / interval(mbean1))*1000 * 60)
    </Formula>
  </Calculation>
</Metric>
```

Related Topics:

- Select an element

Error messages

The listed error messages result from conditions detected in the operation of the JMX Metric Builder. Click on the error message number to display detailed information about that error message.

33 - 35	**36 - 38**	**39 - 41**
WASSPI-33	WASSPI-36	WASSPI-39
WASSPI-34	WASSPI-37	WASSPI-40
WASSPI-35	WASSPI-38	WASSPI-41

WASSPI-33

Description	No actions matched server *<server name>* , version *<version>* .
Severity	Warning
Help Text	**Probable Cause** JMXAction element(s) define FromVersion and ToVersion tags which do not match the server version. **Suggested Action** If the action is valid on the server, then either adjust the JMXAction definition's FromVersion/ToVersion elements or the server's VERSION property.

WASSPI-34

Description	Metric *<metric id>* does not define any actions.
Severity	Warning
Help Text	**Probable Cause** The metric ID specified with the action -m option does not define a JMXActions element. **Suggested Action** Correct the action -m option if an incorrect metric ID was specified. Otherwise, add a JMXActions definition to the metric definition.

WASSPI-35

Description	Error executing action *<action command-line>* .
Severity	Serious
Help Text	**Probable Cause** An unexpected error occurred while executing the action. **Suggested Action** View the managed node's error log to determine the root cause which is logged following the error message.

WASSPI-36

Description	MBean <*JMX objectname*> on server <*server name*> , does not expose operation <*operation name*> .
Severity	Warning
Help Text	**Probable Cause** An action's JMXCalls element defines an operation not exposed by the specified MBean. **Suggested Action** Correct the JMXCalls element or remove the operation from the element.

WASSPI-37

Description	MBean <*JMX objectname*> on server <*server name*> , does not expose attribute <*attribute name*> for write.
Severity	Warning
Help Text	**Probable Cause** An action's JMXCalls element defines a write attribute exposed by the specified MBean as read-only. **Suggested Action** If it is a custom MBean, update the MBean's management interface so the attribute is writable. Otherwise, remove the attribute definition from the JMXCalls element.

WASSPI-38

Description	MBean <*JMX objectname*> on server <*server name*> , does not expose attribute <*attribute name*> .
Severity	Warning
Help Text	**Probable Cause** An action's JMXCalls element defines an attribute not exposed by the specified MBean ObjectName. **Suggested Action** Correct the JMXCalls element or remove the attribute from the element.

WASSPI-39

Description	Error invoking operation <*operation name*> on MBean <*JMX objectname*> .
Severity	Serious
Help Text	**Probable Cause** An unexpected error occurred while invoking an operation on the specified MBean. The managed resource may have thrown an exception. **Suggested Action** View the managed node's errorlog to determine the root cause which is logged following the error message.

WASSPI-40

Description	Error setting attribute <*attribute name*> on MBean <*JMX objectname*> .
Severity	Serious
Help Text	**Probable Cause** An unexpected error occurred while setting an attribute on the specified MBean. The managed resource may have thrown an exception. **Suggested Action** View the managed node's errorlog to determine the root cause which is logged following the error message.

WASSPI-41

Description	Error getting attribute <*attribute name*> from MBean <*JMX objectname*> .
Severity	Serious
Help Text	**Probable Cause** An unexpected error occurred while getting an attribute from the specified MBean. The managed resource may have thrown an exception. **Suggested Action** View the managed node's errorlog to determine the root cause which is logged following the error message.

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click on the bookmark “Comments”.

In case you do not have the email client configured, copy the information below to a web mail client, and send this email to **docfeedback@hp.com**

Product name:

Document title:

Version number:

Feedback: