HP Interconnect

Software Version: 3.50

User 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

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

HP Interconnect uses software from the Apache Jakarta Project including:

- Logging Services (log4j)

Additional third party software used by HP Interconnect includes:

- Xalan from the Apache XML Project
- Xerces from the Apache XML Project
- Mortbay Jetty from Apache

Trademark Notices

Intel and Pentium are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

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

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

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

UNIX® is a registered trademark of The Open Group.

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 Overview

HP Interconnect Tools for Developers (OVI) provides an easy-to-use, consistent, and stable gateway for integrating with HP Software applications. Because OVI is based on industry-standard technologies like Java, XML, HTTP, SOAP, JMS, and TCP sockets, it enables HP Software data to be obtained in a simplified, platform-independent fashion. All data is exchanged in standard XML document format. OVI gives systems integrators, enterprise application integrators, and independent software vendors a single, consistent way to exchange data with HP Software.

Overview

HP Interconnect (OVI) provides access to selected event, performance, service and inventory data from:

- HP Operations Manager for Unix (formerly OVO for UNIX, 7.1 and 8.x) and HP Operations Manager for Windows (formerly OVOW, 7.x)
- HP Network Node Manager version 6.4.1 and 7.x
- HP Performance Manager (OVPM) 4.x for Windows and HP Performance Manager (OVPM) 5.x, 6.x, and 8.0 for Windows or UNIX (with HP Performance Agent, HP Operations Performance Agent, HP Internet Services, and HP Reporter data sources)

  ➤ OVI only works with the standalone OVPM product that has been purchased alone or in a product bundle. It does not work with the OVPM HP Graphing Component (HPGC) that is embedded in products like OVO, OVOW, HP Internet Services (OVIS), and Reporter.

- HP Service Desk 4.5 (with Service Pack 5 or greater. SP 10 or greater is recommended) and 5.x.
- HP Service Navigator 7.1 and 8.1 (UNIX) and 7.x (Windows)
- HP Transaction Analyzer (OVTA) 2.x

In addition, OVI can interact with an OVO agent to allow messages to be sent from the agent to an OVO management station (on Windows or UNIX operating systems). OVI must be running on the same system as the OVO agent.

➤ Only OVO 7.25-8.x agents are supported. An `opcmsg` template must be deployed to the agent.

Also, OVI enables users to create HP Operations Manager events and to easily transform XML messages from one form to another. An OVI toolkit that contains aids for writing custom OVI components is also included.

HP Interconnect supports:

- HP-UX 11.0, 11.11, 11.23 and 11.31

- Solaris 2.8, 2.9 and 2.10
- Microsoft® Windows® 2000 SP3 or greater and Windows® 2003
- Red Hat Linux Advanced Server 2.1

OVI delivers HP Software data over the following message transports:

- HTTP(S), including SOAP over HTTP(S)
- TCP sockets
- Java Messaging Service (JMS). To view a list of message buses verified to work with OVI, see Install a JMS-Compliant Message Bus section in the Installation Guide.

Understanding Pluglets

OVI consists of multiple software components called pluglets, which operate within a common framework, the Core Pluglet Framework (CPF). Pluglets can be linked together to obtain data from an HP Software application (for example, HP Operations Manager) and to send data via a message transport to any external application. OVI pluglets can be used in their default configuration based on parameters you supply during installation. Changes in the environment or desired customizations may require minor reconfiguration.

The OVI pluglets support two data exchange paradigms:

- Publish/Subscribe
- Request/Reply (synchronous)

OVI Components

OVI consists of:

- Infrastructure Components
- HP Software application Pluglets
- Toolkit

Infrastructure Components

Infrastructure components provide the supporting infrastructure for OVI, including the Core Pluglet Framework (environment in which OVI pluglets run) and the `Log4j` logging library. The OVI infrastructure also includes test and core OVI pluglets. Test pluglets help you determine if OVI is set up properly on your system. Core pluglets create HP Software data as messages on a transport (message bus, HTTP(S), or TCP socket), and read messages from the transport. To learn more, see the Using Pluglets section.

Additionally, the OVI infrastructure contains three Special-purpose Pluglets.

HP Software application Pluglets

HP Software application pluglets obtain data from and return messages to HP Software applications. They are an optional part of the OVI installation. For more information see:

- Integrating with HP Operations Manager
- Integrating with HP Service Desk
- Integrating with HP Operations Manager for Unix and Service Desk
- Integrating with HP Service Navigator
- Integrating with HP Performance Manager
- Integrating with HP Transaction Analyzer
- Integrating with HP Network Node Manager

HP Interconnect Toolkit

The toolkit is an optional component of the OVI installation. When it is selected for installation, toolkit components are available in: `OVI_BASE_DIR/paperdocs/C/OVI/toolkit` and `OVI_BASE_DIR/examples/OVI/toolkit`.

The OVI toolkit contains aids for OVI custom pluglet writers. It includes a guidelines manual, API documentation, and additional sample code. Material in the toolkit is intended to reduce the complexity and time required to write a pluglet from scratch. Be sure to read the *Pluglet Developer's Guide*, `OVI_BASE_DIR/paperdocs/C/OVI/toolkit/pluglet_developer_s_guide.htm`.

Environment Variables

Throughout the OVI documentation, references are made to `OVI_BASE_DIR` and `OVI_DATA_DIR` variables.

Variable	UNIX Path	Windows Path
`OVI_BASE_DIR`	`/opt/OV`	OVI base directory, by default: `C:\Program Files\HP OpenView`
`OVI_DATA_DIR`	`/var/opt/OV`	OVI data directory defined during installation, by default: `C:\Program Files\HP OpenView\data`

Throughout the OVI documentation, UNIX notation is used when defining directory paths.

OVI Product Download

You can visit the HP software support web site at **http://support.openview.hp.com/support.jsp** to download the latest HP Interconnect product:

1. Click **Download** from the list on the left of the page. The Downloads page displays.
2. In **Evaluation software** section, click **Learn more**. The Evaluation software page displays.
3. Locate **HP Interconnect** in the Evaluation software column and click it.
4. Review HP Interconnect evaluation software terms and click **I Agree** to proceed. Follow screen instructions to download HP Interconnect.

2 Configuring OVI

Pluglets must be combined in the proper order and deployed into their runtime environment, the Core Pluglet Framework (CPF). There are two kinds of configuration files, one for CPF and one for each pluglet:

- Deploy files have the `.deploy` suffix. These contain pluglet deployment instructions for CPF. A `.deploy` file specifies a collection of pluglets grouped to perform a task, and includes references to associated pluglet configuration (`.config`) files. OVI uses one `.deploy` file during startup to define what that instance actually does.
- Pluglet configuration files have the `.config` suffix. Each pluglet has an associated `.config` file, which contains specific attribute information for that pluglet.

When starting OVI, you can provide the name of the `.deploy` file to be used when initiating the CPF. This starts the pluglets specified in the `.deploy` file. For example, you could start OVI and instruct it to use the `PubSubExample.<selectedTransport>.deploy` file by executing the following command:

UNIX:
`OVI_BASE_DIR/bin/OVI.sh -d OVI_BASE_DIR/examples/OVI/configs/PublishSubscribeTest/PubSubExample.<selectedTransport>.deploy`

Windows:
`OVI_BASE_DIR\bin\OVI.wsf -d OVI_BASE_DIR\examples\OVI\configs\PublishSubscribeTest\PubSubExample.<selectedTransport>.deploy`

If you do not specify a `.deploy` file, the default `OVI_DATA_DIR/conf/OVI/OVI.deploy` is used. This is true unless a different deploy file is configured in the OVI environment file, `OVI_DATA_DIR/conf/OVI/OVI.env`. See Deploy Files for more information about the default `.deploy` file.

OVI `.config` and `.deploy` files exist in these locations:

- `OVI_BASE_DIR/examples/OVI/configs/<subdirectories>`
- `OVI_BASE_DIR/examples/OVI/sampleCode/<subdirectories>`
- `OVI_BASE_DIR/newconfig/OVI`
- `OVI_DATA_DIR/conf/OVI`
- `OVI_BASE_DIR/examples/OVI/toolkit/configs/<subdirectories>`

Do not modify the `.config` and `.deploy` files located in `OVI_BASE_DIR/newconfig/OVI`. They are an archive of the original contents of OVI `.config` files installed on the system.

The files in `OVI_BASE_DIR/examples/OVI/configs` directory are provided as examples. If you want to change them, copy the files you need to *`OVI_DATA_DIR`*`/conf/OVI` directory, then make any necessary changes.

Deploy Files

The `.deploy` files installed with OVI can be used as-is if you want to perform one of the common tasks discussed in Common Pluglet Usage Scenarios. Otherwise, you can create your own.

`OVI.deploy` in *`OVI_DATA_DIR`*`/conf/OVI` is the default OVI deploy file. It is used to start OVI when a deploy file name is not supplied with `OVI.sh` on UNIX or `OVI.wsf` on Windows. (This is true unless a different deploy file is configured in your OVI environment file, *`OVI_DATA_DIR`*`/conf/OVI/OVI.env`.) You can edit the `OVI.deploy` file so that it contains the deploy information with which you typically want to run. This lets you start OVI with a simple command:

UNIX:
`/opt/OV/bin/OVI.sh`

Windows:
`"OV_BASE_DIR\bin\OVI.wsf"`

OVI `.deploy` files are in XML format. All `.deploy` files must contain the following lines as the first elements after the root element:

```
<!--
    This is the JMX Agent configuration for OVI. You should not have to
change this configuration!  CHANGE AT YOUR OWN RISK!!!
-->
   <Agent>
      <Configuration url="file:///OVI_BASE_DIR/conf/OVI/OviAgent.config"/>
   </Agent>
```

Remember that the provided OVI `.deploy` files depict example scenarios. This means that they typically contain references to a set of four pluglets, either:

- `TestSourcePluglet` or an application pluglet
- message transport-specific `PublisherPluglet`
- transport-specific `SubscriberPluglet`
- `TestSinkPluglet`

OR

- `TestRequesterPluglet`
- message transport-specific `RequesterProxyPluglet`
- transport-specific `ResponderProxyPluglet`
- `TestResponderPluglet` or an application pluglet

When all utilized pluglets are on the same system, a message transport is unnecessary, so the OVI `.deploy` file needs to contain only two pluglets: a `SourcePluglet` (or `RequesterPluglet`) and a `SinkPluglet` (or `ResponderPluglet`). The `<targetPluglet>` element in the `SourcePluglet`'s config file must reference the `SinkPluglet` (or the `RequesterPluglet`'s `<targetPluglet>` must reference the `ResponderPluglet`) in order for this to work correctly.

When a custom application uses OVI to allow it to receive data from an HP Software Product (for example, HP Operations Manager), the OVI `.deploy` file needs to contain only two pluglets: a transport-specific `SubscriberPluglet` (or `ResponderProxyPluglet`) and a `SinkPluglet` (or `ResponderPluglet`).

Pluglet Configuration Files

The OVI installer program prompts you for important information about the OVI environment. If you provide the information at the time of installation, the example `.config` files are automatically customized to work in your environment. If you do *not* supply the environment information during the installation or if the environment changes, you must manually edit the OVI example `.config` files to match your environment according to the instructions that follow. After you settle on a usage scenario, copy the necessary files from `OVI_BASE_DIR/examples/OVI/configs` to `OVI_DATA_DIR/conf/OVI`. For more information, see the examples in the Common Pluglet Usage Scenarios section.

The `.config` files are in XML format, and must include a reference to the associated schema (see examples). OVI cannot read a `.config` file without a reference to the associated schema.

XML elements in the `.config` files must be modified to contain the correct message transport information for your environment.

Test Pluglet Configuration Elements

Some XML elements are unique to OVI test pluglet's `.config` files, and are described in the following table. Not all of these elements are present in every test pluglet's `.config` file.

Element Name	Value

`<inputFileName>`
`  /opt/OV/examples/OVI/configs/`
`  <example directory>/<name>.in`
`</inputFileName>`
	Input file containing XML used to generate a test message.

`<outputFileName>`
`  /var/opt/OV/tmp/OVI/<name>.out`
`</outputFileName>`
	Output file that contains a test message or reply; it is in XML format and is UTF-8-encoded.

`<sendDelay>milliseconds</sendDelay>`	The time in milliseconds that the source pluglet waits before it begins sending information to its target pluglet. (Default: `4000`)
`<repeatCount>integer</repeatCount>`	The number of times a message is sent. (Default: `1`)

Transformer Pluglet Configuration

There are two OVI pluglets that perform XSL transformations:

- `PubSubXslTransformerPluglet` is an `OviMessageListener` for the publish/subscribe paradigm. It talks to another `OviMessageListener` target pluglet and does one XSL transformation per incoming message.

- `ReqRespXslTransformerPluglet` is a `SyncRequestListener` that communicates with another `SyncRequestListener` target pluglet for the request/response paradigm. It does two XSL transformations: a transformation of the incoming request and a transformation of the response on the way back.

Config files

The configuration of these two pluglets is very similar. For each transformation, there must be an XSL stylesheet file specified in a stylesheet element of the pluglet's configuration file. `PubSubXslTransformerPluglet`'s stylesheet element is `<stylesheet>`. `ReqRespXslTransformerPluglet` has two stylesheet elements, `<requestStylesheet>` and `<responseStylesheet>`.

Deploy files

References to Transformer pluglets can be inserted within deploy files anywhere you can insert a `PublisherPluglet/SubscriberPluglet` or a `RequesterProxyPluglet/ResponderProxyPluglet` pair. It doesn't matter which transport pluglets, if any, are used.

Possible Transformer arrangements for the publish/subscribe paradigm include:

- SourcePluglet → **PubSubXslTransformer** → SinkPluglet
- SourcePluglet > [PublisherPluglet > SubscriberPluglet] > **PubSubXslTransformer** > SinkPluglet
- Source Pluglet > **PubSubXslTransformer** > [PublisherPluglet > SubscriberPluglet] > SinkPluglet
- Source Pluglet > [PublisherPluglet > SubscriberPluglet] > **PubSubXslTransformer** > [PublisherPluglet > SubscriberPluglet] > SinkPluglet

For the request/response paradigm, possible Transformer arrangements include:

- RequesterPluglet > **ReqRespXslTransformer** > ResponderPluglet
- RequesterPluglet > [RequesterProxyPluglet > ResponderProxyPluglet] > **ReqRespXslTransformer** > ResponderPluglet
- RequesterPluglet > **ReqRespXslTransformer** > [RequesterProxyPluglet > ResponderProxyPluglet] > Responder Pluglet
- RequesterPluglet > [RequesterProxyPluglet > ResponderProxyPluglet] > **ReqRespXslTransformer** > [RequesterProxyPluglet > ResponderProxyPluglet] > ResponderPluglet

Example Transformer `.config` and `.deploy` files are in `the OVI_BASE_DIR/examples/OVI/configs/Transformer PublishSubscribe` and `RequestResponse` subdirectories. Some basic stylesheets are provided in `OVI_BASE_DIR/examples/OVI/configs/Transformer/xsl`. For more information about XSL stylesheets, refer to **http://www.zvon.org/** or **http://www.w3.org/Style/XSL/**.

Special Requirements for JMS-Compliant Message Buses

If you are using a JMS-compliant message bus, the `.deploy` files and their associated `.config` files make use of four special topics in your environment. The following are the four topics under which messages are sent and received on the JMS bus:

- `ovitest`
- `ov.event.notify` (for OVO pluglet examples)
- `ov.event.enrich` (for OVO and Service Desk pluglet examples)
- `snquery` (for Service Navigator pluglet example)

Some message buses (for example, SonicMQ and OpenJMS) automatically create these topics on demand. Others (for example, TIBCO and BEA WebLogic JMS) do not. So, the topics must be created manually using the message bus administration utility. You can use alternate topic names by changing the element values in the provided `.config` files.

JMS-compliant message bus users should also ensure that their `CLASSPATH` environment variable or the `CLASSPATH` variable in the `OVI.env` file (located in `OVI_DATA_DIR/conf/OVI`) includes the path to their message bus provider's client-appropriate `.jar` file(s). See the Prerequisites section in the Installation Guide for more details.

To support the Tibco JMS server failover capability, you must specify the following value for the `<providerUrl>` element in the configuration files for pluglets that use the JMS transport:

```
<providerUrl>tibjmsnaming://host1:7222, tibjmsnaming://host2:7222</
providerUrl>
```

where:

- `tibjmsnaming` is a specific Tibco transport protocol.
- `host1` and `host2` represent the hostnames or IP addresses of the Tibco servers.
- `7222` is the port on which the Tibco servers listen. You do not have to use this port number and the port numbers can be different for each server.

You can find examples of this notation in the sample configuration files for the JMS transport that are provided with this product in the `examples` folder.

Changing Transports

If you wish to switch transports, you must make two changes for each transport pluglet in the `.deploy` file:

1. Change the class parameter to match the transport and pluglet type you wish to use. See the following table for class parameter information.

2 Change the configuration URL value to reflect the new transport type.

Transport	Pluglet Type	Class Parameter
JMS	PublisherPluglet	com.hp.ov.ovi.inf.jms.PublisherPluglet
	SubscriberPluglet	com.hp.ov.ovi.inf.jms.SubscriberPluglet
	RequesterProxy-Pluglet	com.hp.ov.ovi.inf.jms.RequesterProxy-Pluglet
	ResponderProxy-Pluglet	com.hp.ov.ovi.inf.jms.ResponderProxy-Pluglet
HTTP(S)	PublisherPluglet	com.hp.ov.ovi.inf.http.PublisherPluglet
	SubscriberPluglet	com.hp.ov.ovi.inf.http.Subscriber-Pluglet
	RequesterProxy-Pluglet	com.hp.ov.ovi.inf.http.RequesterProxy-Pluglet
	ResponderProxy-Pluglet	com.hp.ov.ovi.inf.http.ResponderProxy-Pluglet
Socket	PublisherPluglet	com.hp.ov.ovi.inf.socket.Publisher-Pluglet
	SubscriberPluglet	com.hp.ov.ovi.inf.socket.Subscriber-Pluglet
	RequesterProxy-Pluglet	com.hp.ov.ovi.inf.socket.Requester-ProxyPluglet
	ResponderProxy-Pluglet	com.hp.ov.ovi.inf.socket.Responder-ProxyPluglet

For example, to switch the `RequestRespondExample.deploy` file in `OVI_BASE_DIR/examples/OVI/configs/RequestResponseTest` to use HTTP instead of JMS, you would change these four lines: (changes are shown in bold type)

```
 <Service name="ResponderProxyPluglet"
   class="com.hp.ov.ovi.inf.jms.ResponderProxyPluglet">
   <Configuration url="ResponderProxyPlugletExample.jms.config"/>
<Service name="RequesterProxyPluglet"
   class="com.hp.ov.ovi.inf.jms.RequesterProxyPluglet">
   <Configuration url="RequesterProxyPlugletExample.jms.config"/>
```

to:

```
<Service name="ResponderProxyPluglet"
  class="com.hp.ov.ovi.inf.http.ResponderProxyPluglet">
   <Configuration url="ResponderProxyPlugletExample.http.config"/>
<Service name="RequesterProxyPluglet"
  class="com.hp.ov.ovi.inf.http.RequesterProxyPluglet">
   <Configuration url="RequesterProxyPlugletExample.http.config"/>
```

JMS-Compliant Message Bus Pluglet Configuration File Example

Here is an example pluglet `.config` file set up for SonicMQ® using Java's reflection capability (taken from `PublisherPlugletExample.jms.config`):

```
<?xml version="1.0" encoding="UTF-8"?>

<publisherPluglet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="/opt/OV/schema/OVI/jms/
PublisherPlugletExample.xsd">

    <topicConnection>
        <!--
            Get the JMS topic connection factory using reflection
            given the JMS TopicConnectionFactory class
         -->
        <connectionFactory>
            <reflection>
                <className>
                    progress.message.jclient.TopicConnectionFactory
                </className>

                <parameter>
                    <name>broker</name>
                    <type>java.lang.String</type>
                    <value>jmsserver.mycompany.com:2506</value>
                </parameter>
            </reflection>

            <!--
                Get the JMS topic connection factory using JNDI
            -->

   <!--
           Currently using reflection and not JNDI so this part
           is commented out!
           Use the following value for providerURL if you want to use
           the Tibco JMS server failover capability. Replace host1 and
           host2 with the hostnames or IP addresses of your Tibco servers.
          If 7222 is not the port number of your Tibco servers, replace it
           with the port numbers on which your Tibco servers listen.

           <providerUrl>tibjmsnaming://host1:7222, tibjmsnaming://
host2:7222</providerUrl>
           -->
            <!--
            <jndi>
                <initialContextFactory>
                   com.sun.jndi.rmi.registry.RegistryContextFactory
                </initialContextFactory>
                <providerUrl>rmi://localhost</providerUrl>
                <jndiName>JMSTopicConnectionFactory</jndiName>
            </jndi>
            -->
        </connectionFactory>

        <!--
            Specify the JMS connection information
        -->
        <connection>
            <userName>user</userName>
            <!--
                 If you do not wish to use encrypted passwords, use
                 the following XML tag.
```

```
             <password>pass</password>
            -->

            <!--
                 Use the OviEncryptUtil script to encrypt this
                 password string.
            -->
            <encryptedPassword>EA0KFlw</encryptedPassword>
            <!--
                 If the <clientID> tag is present, then a clientID will
                 be set for the connection. Otherwise, no clientID will
                 be set. Some JMS implementations (e.g., TIBCO) need a
                 clientID when dealing with durable subscriptions.
                 These clientIDs should be unique for each JMS connection.
             -->
             <!--Currently commented out - no clientID will be set
                 <clientID>client0</clientID>
             -->
        </connection>
    </topicConnection>

    <!--
        Specify the publisher JMS connection properties
    -->
    <jmsSettings>
        <!--
            Specify the JMS Topic to publish to
        -->
        <topic>jmstopic</topic>
        <persistent>true</persistent>
        <priority>6</priority>
        <timeToLive>10000</timeToLive>
    </jmsSettings>

    <!--
        Switch message trace logging on/off
    -->
    <traceMessages>true</traceMessages>

    <!--
        Switch message route logging on/off
     -->
    <traceRoute>true</traceRoute>
</publisherPluglet>
```

JMS Pluglet Configuration File Changes

1 Modify the values for the <className>, <name>, <type>, and <value> elements according to your JMS-compliant message bus. OVI supports Java reflection and JNDI to establish a connection to the JMS messaging server. The PublisherPlugletExample.jms.config example file is configured for SonicMQ® using Java reflection. Here are the relevant XML lines that illustrate this configuration:

```
<reflection>
    <className>


        progress.message.jclient.TopicConnectionFactory
    </className>
    <parameter>
        <name>broker</name>
        <type>java.lang.String</type>
        <value>jmsserver.mycompany.com:2506</value>
    </parameter>
</reflection>
```

2 To use JNDI, comment out the <reflection> section and remove the comments from the <jndi> section. You must also correct the <jndi> configuration for your JMS-compliant message bus. The following example illustrates how to enter configuration data for JNDI in the XML file:

```
<!--
    Get the JMS topic connection factory using reflection
    given the JMS TopicConnectionFactory class
-->
<connectionFactory>
    <!--
         Currently using JNDI and not reflection so this part is
         commented out!
    -->
       <!--
     <reflection>
        <className>
            progress.message.jclient.TopicConnectionFactory
        </className>
        <parameter>
            <name>broker</name>
            <type>java.lang.String</type>
            <value>jmsserver.mycompany.com:2506</value>
        </parameter>
    </reflection>

    -->
    <!--
        Get the JMS topic connection factory using JNDI
    -->
    <jndi>
        <initialContextFactory>
           com.sun.jndi.rmi.registry.RegistryContextFactory
        </initialContextFactory>
        <providerUrl>rmi://localhost</providerUrl>
        <jndiName>JMSTopicConnectionFactory</jndiName>
        <!--
          Currently not using SSL so this part is commented out!
        -->
```

```
        <!--
           Specify the authentication and SSL parameters. The JMS servers's
           SSL settings are quite diverse; You can set two types of
           parameters: contextProperties (set into
            javax.naming.InitialContext) and system properties (set through
            java.lang.System.setProperties()). OVI will call JMS vendor's
            method with these parameters. OVI does no validation check for
           the tag names and values under contextProperties and
            systemProperties. Please follow the JMS vendor's user guide to
            define the parameters . Make sure you fill in values for each
           property that you want to set.
        -->
        <!--
        <contextProperties>
            <!-- authentication setting -->
            <java.naming.security.principal>
                user
            </java.naming.security.principal>
            <java.naming.security.credentials>
                password
            </java.naming.security.credentials>
            <!-- ssl setting -->
            <com.tibco.tibjms.naming.security_protocol>
                ssl
            </com.tibco.tibjms.naming.security_protocol>
            <com.tibco.tibjms.naming.ssl_trusted_certs>
                C:/tibco/ems/bin/certs/server_root.cert.pem
            </com.tibco.tibjms.naming.ssl_trusted_certs>
            <com.tibco.tibjms.naming.ssl_enable_verify_hostname>
                false
            </com.tibco.tibjms.naming.ssl_enable_verify_hostname>
            <com.tibco.tibjms.naming.ssl_enable_verify_host>
                false
            </com.tibco.tibjms.naming.ssl_enable_verify_host>
            <com.tibco.tibjms.naming.ssl_identity>
                C:/tibco/ems/bin/certs/client_identity.p12
            </com.tibco.tibjms.naming.ssl_identity>
            <com.tibco.tibjms.naming.ssl_password>
                tibco_pwd
            </com.tibco.tibjms.naming.ssl_password>
        </contextProperties>
        <systemProperties>
        </systemProperties>
        -->
    </jndi>
</connectionFactory>
```

3 Configure the `<jmsSettings>` elements, which are used for message publishing:

```
<!--
    Specify the JMS connection information
-->
<connection>
          <userName>user</userName>
    <!--
         If you do not wish to use encrypted passwords,
         use the following XML tag.
                        <password>pass</password>
    -->

    <!--
          Use the OviEncryptUtil script to encrypt
          this password string.
    -->
    <encryptedPassword>EA0KFlw</encryptedPassword>

     <!--
          If the <clientID> tag is present, then a clientID
          will be set for the connection. Otherwise, no clientID
          will be set. Some JMS implementations (for example,
          TIBCO) need a clientID when dealing with durable
          subscriptions. These clientIDs should be unique for
          each JMS connection.
     -->
     <!--Currently commented out - no clientID will be set
         <clientID>client0</clientID>
     -->
</connection>
</topicConnection>

<!--
    Specify the publisher JMS connection properties
-->
<jmsSettings>
<!--
    Specify the JMS Topic to publish to
-->
  <topic>jmstopic</topic>


    <persistent>true</persistent>


    <priority>6</priority>


    <timeToLive>10000</timeToLive>
</jmsSettings>

<!--
    Switch message trace logging on/off
-->
<traceMessages>true</traceMessages>

<!--
    Switch message route logging on/off
-->
<traceRoute>true</traceRoute>
```

This table describes elements that commonly exist in pluglet `.jms.config` files:

Element Name	Value

`<clientID>`
`    clientID`
`</clientID>`
	Unique identifier for this pluglet's connection to the message bus. This is used by some JMS-enabled message buses to distinguish among multiple connections of the same user. (Default: `client0`)

`<userName>user</userName>`
`<password>pass</password>`
 	
`or`
 	
`<encryptedPassword>`
`    pass`
`</encryptedPassword>`
	User name and password for the JMS-compliant message bus connection.
	Use the `<password>` element to specify an unencrypted password.
	To specify an encrypted password, use the `OviEncryptUtil` script to create an encrypted password string, and then place that string in the `<encryptedPassword>` element.

`<topic>`
`   jmsTopicName`
`</topic>`
	JMS virtual channel (topic) to be used for messages.

`<persistent>`
`   boolean`
`</persistent>`
	Used to determine the delivery mode for the publish call. If `persistent` is set to `true`, messages are saved to a persistent store as soon as they are received by the JMS vendor. Messages are not lost if the message bus fails. If `persistent` is set to `false`, messages could be lost if the message bus fails. This is only used in the `PublisherPlugletExample.jms.config` and the `RequesterProxyPlugletExample.jms.config` files. (Default: `true`)

`<priority>`
`    integer`
`</priority>`
	An integer between `0` and `9` that is used in the publish call. (Default: `6`)
	Levels `0 – 4` are gradations of normal priority.
	Levels `5 – 9` are gradations of expedited priority.
	This is only used in the example `.config` files `PublisherPlugletExample.jms.config` and `RequesterProxyPlugletExample.http.config.jms.config`.

`<timeToLive>`
`    milliseconds`
`</timeToLive>`
	The lifetime of a message (in milliseconds). (Default: `10000`)

`<selector></selector>`	A JMS subscriber parameter used in the `SubscriberPlugletExample.jms.config` and `ResponderProxyPlugletExample.jms.config` example `.config` files. Only messages with properties matching the message selector expression are delivered. This value is optional.

`<traceMessages>`
`   boolean`
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are: `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)

Element Name	Value
<traceRoute> *boolean* </traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and eventually logged. Valid values are: `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)
<targetPluglet> *targetPlugletName* </targetPluglet>	Name of the target pluglet for this pluglet. This pluglet passes messages to the target pluglet. The target pluglet's name must match the name of a service specified in the `.deploy` file being used.
<targetPlugletReqTimeOut> *milliseconds* </targetPlugletReqTimeOut>	Specifies the time in milliseconds before a timeout occurs if a response is not received. (Default: `5000`)

TCP Socket Pluglet Configuration File Example

Client-side

The following example shows a configuration file utilizing TCP sockets (taken from `SubscriberPlugletExample.socket.config`). This is an example of the client-side configuration of the TCP socket transport. You must specify the server name (or IP address) and port number for the socket connection.

```
<?Xml version="1.0" encoding="UTF-8"?>
    <subscriberPluglet xmlns:xsi=
        "http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/socket/SubscriberPluglet.xsd">

    <!--
        Specify the Pluglet that the SubscriberPluglet should
        acquire
    -->
        <targetPluglet>TestSinkPluglet</targetPluglet>
        <socketSettings>

    <!--
        The name of the server to connect to.
    -->
            <serverName>localhost</serverName>

    <!--
        The TCP port on the server to connect to.
    -->
            <serverPort>12440</serverPort>
        </socketSettings>

    <!--
        Switch on message trace logging
    -->
        <traceMessages>true</traceMessages>

    <!--
        Switch on message route logging
    -->
        <traceRoute>true</traceRoute>

    <!--
        Number of times to try publisher socket
        connection on startup
    -->
        <pollLimit>5</pollLimit>

    </subscriberPluglet>
```

Server-side

The following configuration file utilizes TCP sockets (taken from `PublisherPlugletExample.socket.config`). This is an example of the server-side configuration of the TCP socket transport. You must specify the port number on which you will listen to accept socket connections and the maximum number of threads the pluglet accepting socket connections may spawn at one time.

```
<?xml version="1.0" encoding="UTF-8"?>
<publisherPluglet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="/opt/OV/schema/OVI/socket/
PublisherPluglet.xsd">
    <socketSettings>
        <!--
            The TCP port this Publisher listens on.
        -->
        <serverPort>12440</serverPort>

        <!--
            Uncomment this section to restrict access to the
            Publisher. Default is to allow access to all
            systems. If entries are specified within the allow
            section, only systems matching those entries will be
            allowed. All hostname entries must precede any
            ipAddress entries.

            hostname can be a simple name or use the "*" as a wildcard.
           Valid examples are "*.xyz.mycompany.com" and "*.mycompany.com".

            ipAddress can be a simple IP Address or use the "*" as a
            wildcard.

            Valid examples are "192.66.33.*" and "192.2.*".

            See documentation for further information.

            <allow>
               <hostname>server1.mycompany.com</hostname>
               <hostname>*.mycompany.com</hostname>
               <ipAddress>192.66.33.4</ipAddress>
               <ipAddress>192.2.*</ipAddress>
            </allow>
        -->

    </socketSettings>

    <!--
        Switch on message trace logging
    -->
    <traceMessages>true</traceMessages>
    <!--
        Switch on message route logging
    -->
    <traceRoute>true</traceRoute>

</publisherPluglet>
```

TCP Socket Pluglet Configuration File Changes

OVI supports TCP sockets as a transport mechanism for sending and retrieving data between pluglets. TCP socket example `.config` files use port `12440` on the local system. Values for the `<serverName>` and `<serverPort>` elements must be modified to reflect the hostname or IP address and TCP port you want to use.

```
<socketSettings>
   <serverName>localhost<serverName>
   <serverPort>12440<serverPort>
</socketSettings>
```

Element Name	Value
<serverName>*name*</serverName>	The name of the server (or IP address) to which a socket connection is made.
<serverPort>*port*</serverPort>	The port number for the socket connection.
<maxThreads>*numThreads*</maxThreads>	The maximum number of threads the pluglet accepting socket connections should spawn at one time.

You can limit the systems permitted to connect to the TCP socket `PublisherPluglet` and `ResponderProxyPluglet`. By default, connections to these pluglets are allowed from any system. To restrict access to them, modifications must be made to each pluglet's configuration file under the `<socketSettings>` element. The optional `<allow>` block specifies which systems are permitted to connect. Example `.config` files have this section commented out:

```
<allow>
    <hostname>server1.mycompany.com</hostname>
    <hostname>*.mycompany.com</hostname>
    <ipAddress>192.66.33.4</ipAddress>
    <ipAddress>192.2.*</ipAddress>
</allow>
```

Within the `<allow>` block, `hostname` elements, if any, must come before `<ipAddress>` elements. The hostname value specified must be the name of the host as recognized from the system running OVI. Depending on how your system is configured for hostname resolution, this may be the fully qualified name or not. If a TCP socket `PublisherPluglet` or `ResponderProxyPluglet` pluglet does not allow a system to connect, it issues a warning message with the `OVI-6755` identifier; this message lists the hostname and IP address of the system that attempted a connection. That information can then be added to the `<allow>` block to permit the specified system to successfully connect in the future.

A simple wildcarding scheme can be used to grant access to multiple systems; the wildcard character is the asterisk (`*`). For hostnames, the wildcard must be the first character, and it must be followed by a dot. This allows access setting by subdomains. Examples are `*.xyz.mycompany.com` and `*.mycompany.com`. For IP addresses, the wildcard must be the last character, and it must be preceded by a dot. This allows setting by subnets. Examples are `192.66.33.*` and `192.2.*`

If you are using the special alias `localhost` as a <hostname> value, there are situations in which your system may not be permitted to connect. If you encounter this problem, specify your system's IP address in an <ipAddress> element. Check the contents of the `OVI-6755` warning message to determine how OVI sees the hostname and IP address of `localhost`, and change the <allow> section of the pluglet configuration file to match.

HTTP Pluglet Configuration File Example

Client-side

The following configuration file (taken from `RequesterProxyPlugletExample.http.config.http.config`) demonstrates required client-side configuration for the OVI HTTP transport. In the `<url>` element of this file, you must specify a server hostname or IP address, port number, and target pluglet for the HTTP connection. This `<url>` element requires further modification to support SOAP message exchange using HTTP. The `<url>` must also be modified if encrypted (HTTPS) communication is required, as demonstrated below.

If HTTP basic authentication is enabled on the OVI HTTP(S) server with which the client communicates, the `<authentication>` element must contain appropriate user and password information in the format shown below. If the target server does not support authentication, the `<authentication>  </authentication>` tags must still be present but the element should be empty.

```
<?Xml version="1.0" encoding="UTF-8"?>

<requesterProxyPluglet xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
   xsi:noNamespaceSchemaLocation=
   "/opt/OV/schema/OVI/http/RequesterProxyPluglet.xsd">

   <!--
       The URL for the request. It must start with "http://" or "https://".
       The structure is:
           http[s]://<server-hostname>:<server-port>/ovi/<targetPluglet>

       Example:
           http://localhost:8080/ovi/TestResponderPluglet

      NOTE: To send SOAP messages to OVI over HTTP, use this URL structure:
             http[s]://<server-hostname>:<server-port>/ovi-soap/
<targetPluglet>
             Example:
                 https://localhost:8080/ovi-soap/TestResponderPluglet
   -->
   <url>http://localhost:8080/ovi/TestResponderPluglet</url>

   <!--
       Client HTTP basic authentication credential.
    -->
   <authentication>
   <!--
     NOTE: If HTTP basic authentication is enabled on the OVI HTTP(S)
       server, the elements below (starting with <basic>) must be present.
       If morethan one <user> block is provided, only the first one is used.
   -->
      <basic>
         <user>
            <name>Bob</name>
            <!--
                Use the OviEncryptUtil script to manually encrypt
                this password string
             -->
            <encryptedPassword>f1a1s0f</encryptedPassword>
         </user>
      </basic>
```

```
   </authentication>
   <!--
      Switch on message trace logging
   -->
   <traceMessages>true</traceMessages>
   <!--
      Switch on message route logging
   -->
   <traceRoute>true</traceRoute>
</requesterProxyPluglet>
```

Server-side

The following configuration file (taken from `ResponderProxyPlugletExample.http.config`) demonstrates required server-side configuration for the OVI HTTP transport. In the `<serverPort>` element, the port number on which to accept HTTP(S) connections must be specified. You may also configure the maximum number of threads the pluglet accepting HTTP(S) connections may spawn at one time (in the `<maxThreads>` element) and whether or not encrypted (HTTPS) communication should be used (in the `<useSSL>` element). If you want to use HTTP basic authentication, the `<authentication>` element must contain appropriate user and password information in the format shown below. If not, the `<authentication>  </authentication>` tags must still be present but the element should be empty.

```
<?xml version="1.0" encoding="UTF-8"?>

<responderProxyPluglet xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
   xsi:noNamespaceSchemaLocation=
   "/opt/OV/schema/OVI/http/ResponderProxyPluglet.xsd">

   <!--
       Specify the Pluglet that the ResponderProxyPluget should acquire
       and the timeout (in milliseconds) used for the synchronous
       connection.
   -->
   <targetPluglet>TestResponderPluglet</targetPluglet>
   <targetPlugletReqTimeOut>5000</targetPlugletReqTimeOut>
   <serverSettings>
      <!--
          The port the HTTP(S) server listens on.
      -->
      <serverPort>8080</serverPort>
      <!--
          The max number of threads the pluglet may spawn at one time.
      -->
      <maxThreads>10</maxThreads>
      <!--
          Server authentication control list.
      -->
      <authentication>
         <!--
             NOTE: If you want to enable HTTP basic authentication for OVI,
                the elements below (starting with <basic>) must be present.
                 A <user> block containing a <name> and <encryptedPassword>
               must be supplied for each approved user.
         -->
         <basic>
            <user>
```

```
                <name>Bob</name>
                <!--
                    Use the OviEncryptUtil script to manually encrypt
                     this password string
                -->
                <encryptedPassword>f1als0f</encryptedPassword>
             </user>
          </basic>

      </authentication>
      <!--
          Flag indicating whether the server will use encrypted (HTTPS)
          communication using SSL.
      -->
      <useSSL>false</useSSL>
   </serverSettings>
   <!--
       Switch on message trace logging
   -->
   <traceMessages>true</traceMessages>
   <!--
       Switch on message route logging
   -->
   <traceRoute>true</traceRoute>
</responderProxyPluglet>
```

HTTP Pluglet Configuration File Changes

OVI supports HTTP as a transport mechanism for sending and retrieving data between pluglets. The HTTP example `.config` files use port `8080` on the local system. You must modify `<url>` and `<serverPort>` to contain the hostname (or IP address) and HTTP(S) port you want to use.

Remember:

- The server port number specified in the `RequesterProxyPlugletExample.http.config.http.config` and `ResponderProxyPlugletExample.http.config` files must be the same.
- If you would like OVI to use encrypted communication (SSL) for exchanging messages, the `<url>` element in the `RequesterProxyPlugletExample.http.config.http.config` file must be begin with `https://` and the `<useSSL>` element in the `ResponderProxyPlugletExample.http.config` file must be set to `true`.
- If you want to enable HTTP(S) basic authentication for OVI message exchange, the `<authentication>` element in both client (`RequesterProxyPlugletExample.http.config.http.config`) and server (`RequesterProxyPlugletExample.http.config`) side `.config` files must contain username and password information for authorized users. The correct format for this information is given in the examples in this section.
- If you would like to send SOAP messages to OVI, you must modify the `<url>` element to reflect this, as shown below.

RequesterProxyPlugletExample.http.config

```
<!-- For HTTP communication -->
<url>http://localhost:8080/ovi/TestResponderPluglet</url>

<!-- For HTTPS communication -->
<url>https://localhost:8080/ovi/TestResponderPluglet</url>

<!-- For SOAP over HTTP communication -->
<url>http://localhost:8080/ovi-soap/TestResponderPluglet</url>

<!-- For SOAP over HTTPS communication -->
<url>https://localhost:8080/ovi-soap/TestResponderPluglet</url>
```

ResponderProxyPlugletExample.config

```
<serverPort>8080</serverPort>
<useSSL>false</useSSL>
```

This table describes the HTTP pluglet configuration elements:

Element Name	Value
<url> http[s]://<serverHostname>:<serverPort>/ ovi/<targetPluglet> </url> or <url> http[s]://<serverHostname>:<serverPort>/ ovi-soap/<targetPluglet> </url>	The URL for the HTTP(S) request. The hostname of the server (or IP address) to which a connection is made, the port number for the HTTP(S) connection, and the name of the target pluglet to be acquired. If SOAP messages are being sent to OVI over HTTP(S), the `ovi` portion of the URL must be changed to be `ovi-soap`.
<serverPort>*port*</serverPort>	The server's port number on which to accept HTTP(S) connections.

Element Name	Value
<maxThreads>*numThreads*</maxThreads>	The maximum number of threads the pluglet accepting HTTP(S) server's connections may spawn at one time to service requests. If numThreads requests are being serviced and a new request arrives, it waits until one of the old requests completes before being serviced.
<useSSL>*boolean*</useSSL>	The flag indicating whether the server provides encrypted (HTTPS) communication.
<authentication> <basic> <user> <name>Bob</name> <!-- Use the OviEncryptUtil script to manually encrypt this password string --> <encryptedPassword>f1als0f</encryptedPassword> </user> </basic> </authentication>	HTTP basic authentication credentials. The <authentication> element is present but empty if the use of HTTP basic authentication for OVI message exchange is not desired.

HTTP Basic Authentication Support

OVI provides the option of enabling HTTP basic authentication for message exchange. This feature may be utilized when messages are transported via HTTP(S). The OVI installer allows you to enter username and password information for all users that you wish to authorize. After encrypting the supplied passwords, this information is stored in the HTTP-related pluglet example configuration files. This causes authentication processing to occur prior to messages being exchanged between pluglets. See HTTP Pluglet Configuration File Example for more information.

3 Starting and Stopping OVI

This section describes the methods for starting and stopping OVI, depending on your platform and preferences. Be patient as OVI completes its startup activities; it may take a minute or more.

Starting OVI for UNIX

To start OVI for UNIX, enter **`/opt/OV/bin/OVI.sh`**. The resulting OVI process ID is stored in the file `OVI_DATA_DIR/tmp/OVI/ovi.pid`.

This table describes startup options:

Option	Description
`-b`	Starts OVI in the background.
`-d <deploy file name>`	Starts OVI using the specified deploy file. If a deploy file is not specified, OVI's default deploy file (`OVI_DATA_DIR/conf/OVI/OVI.deploy`) is used unless another deploy file is configured in the environment file.
`-env <environment file name>`	Starts OVI using the specified environment file to configure its runtime environment. If an environment file is not specified, OVI's default environment file (`OVI_DATA_DIR/conf/OVI/OVI.env`) is used. See Starting OVI for UNIX Using a Custom Environment File.
`-h`	Displays command usage help for `OVI.sh`.
`-l`	Starts the OVI License Manager. You must be user `root`.
`-ssl`	*For repair purposes only, not for normal operation.* Creates the OVI SSL keystore required when using HTTPS as a message transport. You must be user `root` to perform this operation.
`-stopOnFail`	Stops OVI as quickly as possible when a pluglet specified in the current deploy file fails.

Option	Description
-t	Tests OVI to determine if it has been correctly installed and configured to work with the specified message transport. Restart OVI after performing this test.
-v	Displays version information for important OVI components.
-x	Specifies non-default settings for the Java Virtual Machine in which OVI will run (for example, -x "-Xms128m" to set initial heap size). **Note**: If you use HTTP as the OVI message transport, you should set the JVM initial heap size to at least 128 Mb when starting OVI. This happens automatically when OVI.sh is used to start OVI. If you want to permanently specify OVI JVM settings, edit the OVI_DATA_DIR/conf/OVI/OVI.env file to include a JAVA_ARGS="-Xms128" entry. The arguments on the command line have precedence over the settings from the configuration file.

Starting OVI for UNIX Using a Custom Environment File

By default, OVI uses OVI_DATA_DIR/conf/OVI/OVI.env as the file used to configure its runtime environment. If you want OVI to use a custom environment file, you may specify it (using the -env parameter) when starting OVI. The custom environment file must contain a definition for each of the variables shown in the following table:

Required Definition	Description	Example
OVI_BASE	OVI base installation directory	OVI_BASE="/opt/OV"
OVI_TMP	Directory into which temporary files are written	OVI_TMP="/var/opt/OV/tmp/OVI"
OVI_DEPLOY	Path to the .deploy file used when starting OVI	OVI_DEPLOY="/var/opt/OV/conf/ OVI/OVI.deploy"
OVI_PROPERTIES	Path to the .properties file used when running OVI	OVI_PROPERTIES="/var/opt/OV/conf/OVI/User.properties"
CLASSPATH	Java classpath used for OVI processes. The value specified here is prepended to the value of the system %CLASSPATH% variable	CLASSPATH=""

Required Definition	Description	Example
JVM_ARGS	Arguments passed to the Java Virtual Machine (JVM) started by OVI	JVM_ARGS="-Xmx128m"
JAVA	Path to the Java executable used for running OVI	JAVA="/opt/j2sdk1.4.1_02/bin/java"
OVI_PID_FILE	Path to the file which stores the process ID of the last OVI instance that was run. It is used when stopping OVI with the **OVI.sh -stop** command	OVI_PID_FILE="/var/opt/OV/tmp/OVI/ovi.pid"

Starting OVI for Windows Using the Graphical User Interface

To start OVI for Windows using the Graphical User Interface:

1. Select **Start** → **Programs** → **HP** → **HP Interconnect** → **Start HP Interconnect**. The OVI GUI opens.
2. Specify the following options:
   a. Deploy File. The default deploy file is OVI_BASE_DIR\examples\OVI\configs\RequestResponseTest\RequestRespondExample.deploy. Click **Browse** to specify an alternate deploy file.
   b. JVM Options. Specify non-default settings for the JVM in which OVI runs (for example, -Xms128m to set initial heap size).
3. Select the OVI Version tab to display version information for important OVI components.
4. Click **Run** to start OVI.

If you want to permanently specify OVI JVM settings, edit the OVI_BASE_DIR\OVI\OVI.env file to include a JAVA_ARGS="-Xms128" entry. If JAVA_ARGS is present in the OVI.env file and OVI is started with an entry in the JVM options field, the options field overrides the JAVA_ARGS entry.

Starting OVI for Windows from a Command Prompt Window

To start OVI for Windows from a Command Prompt window, enter `OVI_BASE_DIR\bin\OVI.wsf`. You can supply additional startup options with this command:

Option	Description
`-d <deploy file name>`	Starts OVI using the specified deploy file. If a deploy file is not specified, OVI's default deploy file (`OVI_BASE_DIR\conf\OVI\OVI.deploy`) is used unless another deploy file is configured in the environment file.
`-env <environment file name>`	Starts OVI using the specified environment file to configure its runtime environment. If an environment file is not specified, OVI's default environment file (`OVI_DATA_DIR/conf/OVI/OVI.env`) is used. See Starting OVI for Windows Using a Custom Environment File.
`-h`	Displays command usage help for `OVI.wsf`.
`-l`	Starts the OVI License Manager. You must be user `Administrator`.
`-ssl`	*For repair purposes only, not for normal operation.* Creates the OVI SSL keystore required when using HTTPS as a message transport. You must be user `Administrator`.
`-stopOnFail`	Stops OVI as quickly as possible when a pluglet specified in the current deploy file fails.
`-t`	Tests OVI to determine if it has been correctly installed and configured to work with the specified message transport.
`-v`	Displays version information for important OVI components.
`-x`	Specifies non-default settings for the Java Virtual Machine in which OVI will run (for example, `-x "-Xms128m"` to set initial heap size). **Note**: If you use HTTP as the OVI message transport, you should set the JVM initial heap size to at least 128 Mb when starting OVI. This happens automatically when `OVI.wsf` is used to start OVI. If you want to permanently specify OVI JVM settings, edit the `OVI_DATA_DIR/conf/OVI/OVI.env` file to include a `JAVA_ARGS="-Xms128"` entry. If `JAVA_ARGS` is present in the `OVI.env` file and OVI is started with the `-x` option, then the option overrides the `JAVA_ARGS` entry.

Starting OVI for Windows Using a Custom Environment File

By default, OVI uses `OVI_DATA_DIR\conf\OVI\OVI.env` as the file used to configure its runtime environment. If you want OVI to use a custom environment file, you may specify it (using the `-env` parameter) when starting OVI. The custom environment file must contain a definition for each of the variables shown in the following table:

Required Definition	Description	Example
OVI_BASE	OVI base installation directory	OVI_BASE="C:/Program Files/HP OpenView";
OVI_DATA	OVI data installation directory	OVI_DATA="C:\Program Files\HP OpenView\data";
OVI_TMP	Directory into which temporary files are written	OVI_TMP="C:\Program Files\HP OpenView\data\tmp\OVI";
OVI_DEPLOY	Path to the .deploy file used when starting OVI	OVI_DEPLOY="C:\Program Files\HP OpenView\data\conf\OVI\OVI.deploy";
OVI_PROPERTIES	Path to the .properties file used when running OVI	OVI_PROPERTIES="C:\Program Files\HP OpenView\data\conf\OVI\User.properties";
CLASSPATH	Java classpath used for OVI processes. The value specified here is prepended to the value of the system %CLASSPATH% variable	CLASSPATH="";
JVM_ARGS	Arguments passed to the Java Virtual Machine (JVM) started by OVI	JVM_ARGS="-Xmx128m";
JAVA	Path to the Java executable used for running OVI	JAVA="C:\j2sdk1.4.1_02\bin\java.exe";
JVM_DLL	Path to the Java jvm.dll used for running OVI	JVM_DLL="C:\j2sdk1.4.1_02\jre\bin\server\jvm.dll";

Stopping OVI for UNIX

To stop OVI on UNIX, enter: **/opt/OV/bin/OVI.sh -stop** or enter **Control-C** in the window from which OVI was started. This shuts down the OVI process whose ID was stored in /var/opt/OV/tmp/OVI/ovi.pid. It does not stop any other running instances of OVI. (See *Shutdown* of *Known Problems, Limitations and Workarounds* section of the *OVI Release Notes*.)

Stopping OVI for Windows

To stop OVI, close the window in which OVI was started. If startup was done via a Command Prompt window, stop OVI by pressing Control-C.

Verifying OVI Functionality

For UNIX operating systems, to verify that OVI and your selected OVI transport (JMS-compliant bus, sockets, or HTTP) are working correctly:

1. Verify `RequestRespondExample.deploy`.
   a. Start OVI by entering:
   `/opt/OV/bin/OVI.sh -t`
   b. Examine the output from this test to determine if OVI is correctly installed and configured to run with your message transport.
2. Verify `PubSubExample.deploy`.
   a. Start OVI by entering (from `/opt/OV/bin`):
   `OVI.sh -d ../examples/OVI/configs/ PublishSubscribeTest/ PubSubExample.<selectedTransport>.deploy`
   b. Verify
   `/var/opt/OV/tmp/OVI/TestSinkPlugletExample.out`
   contains the same information in the same format as `/opt/OV/examples/OVI/configs/PublishSubscribeTest/TestSourcePlugletExample.in`

For Windows operating systems, to verify that OVI and your selected OVI transport (JMS-compliant bus, sockets, or HTTP) are working correctly:

1. Verify `RequestRespondExample.deploy`:
   a. Start OVI in one of the following ways:
   From a command Prompt Window, enter:
   `cscript OVI_BASE_DIR\bin\OVI.wsf -t`

   or, select **Start** → **Programs** → **HP** → **HP Interconnect** → **Test Installation**
   b. Examine the output from this test to determine if OVI is correctly installed and configured to run with your message transport.
2. Verify `PubSubExample.deploy`:
   a. Start OVI by entering (from `OVI_BASE_DIR\bin`):
   `OVI.wsf -d ..\examples\OVI\configs\ PublishSubscribeTest\PubSubExample.<selectedTransport>.deploy`
   b. Verify that the file
   `OVI_DATA_DIR\tmp\TestSinkPlugletExample.out`
   contains the same information in the same format as `OVI_BASE_DIR\examples\OVI\configs\PublishSubscribeTest\TestSourcePlugletExample.in`
   or examine the test output in the Command Prompt window to determine if OVI is correctly installed and configured to run with your selected message transport.

Server `host` and `port` values entered during the installation are used to verify that OVI is functioning when using the socket or HTTP transport.

4 Using Pluglets

OVI provides several pluglets for you to use. Some are test pluglets whose purpose is to help you assess whether OVI is set up properly on your system. Others are core pluglets that send HP Software data as messages on a message transport and read messages from the transport. Still others are application pluglets that are useful for obtaining data from HP Software applications. In addition, there are some special-purpose pluglets.

For testing or real application use, pluglets must be combined in a proper order (for example, start pluglets that receive messages before you start pluglets that send messages) and then deployed into the runtime environment (the Core Pluglet Framework).

The `.config` file for each pluglet contains pluglet-specific attribute information, as described in Pluglet Configuration Files. A `.deploy` file holds pluglet deployment instructions for the CPF. It specifies collections of pluglets grouped to perform a task, and includes references to associated pluglet `.config` files. When you start OVI, you can provide the name of a `.deploy` file to be used when initiating the CPF. This results in the startup of those pluglets specified in the `.deploy` file.

Pluglets can be combined in a variety of ways, depending on your goal. Several common and useful assembly patterns are explained and illustrated in Figure 1, Figure 2, and Figure 3. OVI provides `.config` and `.deploy` files for each of these common scenarios in the subdirectories of `OVI_BASE_DIR/examples/OVI/configs`. You may need to copy these files to the directory tree of `OVI_DATA_DIR/conf/OVI` and edit the contents of some elements in the `.config` files to match your message transport environment. (See Pluglet Configuration Files and Changing Transports for more details.) The supplied `.deploy` files can be used as-is if you want to perform one of the common tasks described in Common Pluglet Usage Scenarios. Otherwise, you can create your own.

Pluglet Capabilities

OVI provides several application and utility pluglets that enable you to easily interact with HP Software data. Their names and capabilities are summarized here:

Pluglet	HP Software application with Which it Interacts	Capabilities
OvoInventoryPluglet	HP Operations Manager for Unix	Create or delete nodes within OVO.
OvoEventPluglet		Receive notification of newly created messages in OVO.
OvoEventReadPluglet		Retrieve OVO messages satisfying specified criteria.
OvoEventChangePluglet		Receive OVO message change events.
OvoEventModifyPluglet		Modify existing OVO messages.
OvoEventEnrichmentPluglet		Enrich OVO messages using your own custom application OR, in combination with SdServiceInfo EnrichmentPluglet, enrich OVO messages using Service Desk data.
OvoEventActionPluglet		Execute automatic and operator-initiated actions defined on OVO messages. Also, query the status of those actions.
OvoEventCreatePluglet	HP Operations Manager for Unix and HP Operations Manager for Windows with OVO agents	Create messages within OVO/U or OVO/W.
OvoWEventPluglet	HP Operations Manager for Windows	Receive notification of messages created in OVO/W.
OvoWEventChangePluglet		Receive message change events from OVO/W.
OvoWEventModifyPluglet		Modify existing OVO/W messages.
OvoWEventActionPluglet	HP Operations Manager for Windows	Execute automatic and operator-initiated actions defined on OVO/W messages. Also, query the status of those actions.

Pluglet	HP Software application with Which it Interacts	Capabilities
NnmInventoryPluglet	HP Network Node Manager	Retrieve network inventory information from NNM.
NnmSnmpCollectPluglet		Retrieve performance data directly from NNM SNMP performance metric files.
NnmInventorySnmpEventPluglet		Receive raw or correlated SNMP inventory events from NNM.
NnmEventReadPluglet	HP Network Node Manager version 7.5x	Retrieve events from NNM.
SdInventoryPluglet	HP Service Desk	Perform create, read, update and delete operations on specified configuration items, services, persons, organizations, incidents or service calls.
SdServiceInfoEnrichmentPluglet		When used with OvoEvent EnrichmentPluglet, allow event data from OVO to be enhanced with service-related data from Service Desk.
SNQueryPluglet	HP Service Navigator for UNIX	Obtain service problem root cause and impacted services information from Service Navigator. Alternatively, retrieve service hierarchy data representing the tree of children and dependencies of a given service. Returned service information includes the status of each listed service.
SNWQueryPluglet	HP Service Navigator for Windows	
OvpmQueryPluglet	HP Performance Manager.	Query OVPM for performance metric data and performance reports from specified systems within a given time frame. Also, query OVPM for a list of data sources and entities (i.e., systems or probes) for which it is collecting data and a list of the metrics being collected for a specific data source and entity pair.

Pluglet	HP Software application with Which it Interacts	Capabilities
`OvtaQueryPluglet`	HP Transaction Analyzer	Retrieve transaction data from OVTA that meets specified criteria.
`CommandResponderPluglet`	–	Receive a request containing a command to execute on the local system, execute the command, and return its result.
`FileReaderPluglet`	–	Periodically monitor the files written to a specified directory, read the new files that are there, and publish the files' content.
`TestDirectorySinkPluglet`	–	Receive incoming messages and write each message to a new file in the configured directory.
`PubSubXslTransformerPluglet` & `ReqRespXslTransformerPluglet`	–	Perform one XSL transformation per incoming message (for publish/subscribe scenarios) *or* two XSL transformations (for request/response situations), one on the incoming request and another on the response before it is returned.

Test Pluglets

These pluglets are provided for demonstration and testing purposes. They are not intended for production environment use.

Pluglet	Description
`TestSourcePluglet`	Reads test message from specified input file and sends message to `PublisherPluglet`, which is `TestSourcePluglet`'s target pluglet.

TestSinkPluglet	Receives test message from SubscriberPluglet and writes it to specified output file.
TestRequesterPluglet	Reads test message from specified input file, sends message to RequesterProxyPluglet (which is TestRequesterPluglet's target pluglet), waits for a response message, and when response is received, writes it to specified output file.
TestResponderPluglet	Receives message from ResponderProxyPluglet, writes message to specified output file, reads a response message from specified input file, and sends that message to ResponderProxyPluglet.

Core Pluglets

Pluglet	Description
PublisherPluglet	Sends received messages onto message transport. Messages are received from pluglets such as OvoEventPluglet or TestSourcePluglet.
SubscriberPluglet	Reads message from message transport and sends it to a pluglet such as TestSinkPluglet, which is SubscriberPluglet's target pluglet.
RequesterProxyPluglet	Receives message from pluglets such as OvoEventEnrichmentPluglet,OVOEventCreatePluglet or TestRequesterPluglet, sends message to message transport, waits for response message, and when it is received, sends response to a pluglet such as OvoEventEnrichmentPluglet,OvoEventCreatePluglet or TestRequesterPluglet, as appropriate.
ResponderProxyPluglet	Reads message from message transport, sends message to its target pluglet (e.g., TestResponderPluglet or SdServiceInfoEnrichmentPluglet), waits for a response message, and, when response is received, sends it to message transport.

Pluglet	Description
PubSubXslTransformerPluglet	Receives message from a pluglet such as TestSourcePluglet or SubscriberPluglet, transforms message according to instructions in the designated XSL stylesheet, and passes transformed message to its target pluglet (e.g., TestSinkPluglet or PublisherPluglet).
ReqRespXslTransformerPluglet	Receives message from a pluglet such as TestRequesterPluglet or RequesterProxyPluglet, transforms incoming message according to instructions in the designated XSL stylesheet, passes transformed message to its target pluglet (e.g., TestResponderPluglet or ResponderProxyPluglet), and waits for a response message. When response is received, it transforms response message according to instructions in the designated XSL stylesheet and returns the transformed response message to the originating pluglet such as TestRequesterPluglet or RequesterProxyPluglet.
Sub2ReqBrokerPluglet	Receives message from a pluglet such as SubscriberPluglet, send incoming message to its target pluglet (e.g., ReqRespXslTransformerPluglet or RequesterProxyPluglet), and waits for a response message. When response is received, it either sends response message to other Sub2ReqBrokerPluglet or writes the response to a new file in the configured directory.

Special-purpose Pluglets

OVI provides three special-purpose pluglets for your use:

- FileReaderPluglet

  The FileReaderPluglet periodically monitors files written to a specified directory, reads new files that are there, and sends the files' content to a pluglet that implements the OviMessageListener interface. The file that has been read is deleted from the directory. Directory name and time between scans are configurable.

  FileReaderPluglet expects all files that it reads to contain valid XML. If it attempts to read a file containing non-valid XML, the pluglet moves that file to a badFiles subdirectory. Errors are logged when a file is moved to the badFiles subdirectory or when this attempted file move fails. When a file cannot be read or sent to the targetPluglet, an error is also logged. If a file cannot be moved to the badFiles subdirectory because of a permissions problem, the file remains in the original directory and an error is logged each time the FileReaderPluglet tries to read and move it.

- TestDirectorySinkPluglet

  The TestDirectorySinkPluglet receives incoming messages and writes each message to a new file in the configured directory. Files are named file#.xml, where # is a number that is incremented for each new file created.

- CommandResponderPluglet

The `CommandResponderPluglet` receives a request containing a command to execute on the local system and then executes the command. It returns the command's result to the requesting pluglet. `CommandResponderPluglet` allows only pre-approved commands to be executed. These commands (called `validCommands`) are listed in the `CommandResponderPlugletExample.config` file.

The format of a command's result is not guaranteed to be consistent between different operating systems, application versions, etc. OVI causes the specified command to be executed on the local system and returns the result, as is, in an XML wrapper. The result's format is not manipulated.

`CommandResponderPluglet` may be configured (via its `allowInput` flag) to permit input from `STDIN` to be accepted as part of the command. This option may create a security hole in your environment, so it should only be used after fully understanding its implications!

The command that is executed by `CommandResponderPluglet` should carefully validate its input parameters to ensure they match what is expected. This is necessary since special characters in the input could be used by an attacker to run rogue commands. Perl scripts that run commands on the system are especially vulnerable to this type of attack. Testing should be done on the command to ensure it has no security holes. In addition, use of a secure transport, such as HTTPS with authentication, is recommended in order to reduce the chances of unauthorized users accessing this pluglet.

There are many resources available to help you write secure commands. A good one can be found at **http://www.dwheeler.com/secure-programs/ w.dwheeler.com/secure-program**.

FileReaderPluglet Configuration

Example `FileReaderPluglet .deploy` and `.config` files have been provided in the `OVI_BASE_DIR/examples/OVI/configs/FileReader` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory tree before editing. The `FileReaderPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<fileReaderPluglet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="/opt/OV/schema/OVI/
FileReaderPluglet.xsd">
    <!--
        The target pluglet that read in files will be sent to.
    -->
    <targetPluglet>TestDirectorySinkPluglet</targetPluglet>
    <!--
        The directory from which to read files.
    -->
    <fileReadDir>/opt/OV/data/tmp/OVI/ExampleReadDir</fileReadDir>
    <!--
       The delay time (in milliseconds) between successive directory reads.
    -->
    <fileReadDelay>5000</fileReadDelay>
    <!--
        Message trace logging switch.
    -->
    <traceMessages>true</traceMessages>
    <!--
        Message route logging switch.
    -->
    <traceRoute>true</traceRoute>
</fileReaderPluglet>
```

The following table describes elements in the `FileReaderPlugletExample.config` file:

Element Name	Value

`<targetPluglet>`
`TestDirectorySinkPluglet`
`</targetPluglet>`
	Name of this pluglet's target pluglet. The contents of files that have been read are passed to this pluglet.

`<fileReadDir>`
`/opt/OV/data/tmp/OVI/ExampleReadDir`
`</fileReadDir>`
	Directory that is periodically scanned. New files are read from this directory.

`<fileReadDelay>`
`5000`
`</fileReadDelay>`
	Time, in milliseconds, between successive reads of the `fileReadDir`. (Default: `5000`, or 5 seconds)

`<traceMessages>`
`boolean`
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
`boolean`
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and, eventually, displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

TestDirectorySinkPluglet Configuration

An example `TestDirectorySinkPluglet.config` file has been provided in the `OVI_BASE_DIR/examples/OVI/configs/FileReader` directory. If you need to make changes, copy it to the `OVI_DATA_DIR/conf/OVI` directory tree before editing. The `TestDirectorySinkPluglet.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<testDirectorySinkPluglet xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/TestDirectorySinkPluglet.xsd">
    <!--
        Directory into which files containing incoming OVI messages will be
        written.
    -->
    <outputDirectory>/opt/OV/data/tmp/OVI/ExampleWriteDir</outputDirectory>
    <!--
        Message trace logging switch.
    -->
    <traceMessages>true</traceMessages>
    <!--
        Message route logging switch.
    -->
    <traceRoute>true</traceRoute>
</testDirectorySinkPluglet>
```

The following table describes elements in the `TestDirectorySinkPluglet.config` file:

Element Name	**Value**

`<outputDirectory>`
`  /opt/OV/data/tmp/OVI/ExampleWriteDir`
`</outputDirectory>`
	Directory into which new files are written. Each file contains the contents of an incoming OVI message.

`<traceMessages>`
`  boolean`
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
`  boolean`
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and, eventually, displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

CommandResponderPluglet Configuration

Example `CommandResponderPluglet .deploy`, `.config` and `.in` files have been provided in the `OVI_BASE_DIR/examples/OVI/configs/CommandResponder` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory tree before editing. The UNIX system version of the `CommandResponderPlugletExample.config` file is shown below. A slightly different version of this file is installed on Windows systems.

```
<?xml version="1.0" encoding="UTF-8"?>
<commandResponderPluglet xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation=
   "opt/OV/schema/OVI/CommandResponderPluglet.xsd">
   <!--
      The list of approved commands that the
      CommandResponderPluglet is allowed to execute.
   -->
   <validCommands>
      <!--
         This example is a command with no STDIN.
      -->
      <command>
         <!--
            The literal string of the command that will be
            given to the OS command interpreter. If it has
            any special XML characters (e.g. &gt;, &lt;),
            they must be escaped.
         -->
         <commandString>ls</commandString>
         <!--
            If true, XML  output of command will be embedded
            in the response message, so it must be well
            formed XML. If false, all special XML characters
            will be escaped. (false is the safest option,
            but output may require further manipulation to
            convert it back to XML.)
         -->
         <commandReturnsXmlString>
            false
         </commandReturnsXmlString>
         <!--
            If true, input via STDIN is allowed to be passed to
            the command within the "input" tag in the request.
            If false, no STDIN will be allowed, and if there is
            an "input" tag, an OperationException XML will be
            returned.
         -->
         <allowInput>false</allowInput>
      </command>

      <!--
         This example is a command that reads arguments
         from STDIN.
      -->
      <command>
         <!--
            The literal string of the command that will be
            given to the OS command interpreter. If it has
            any special XML characters (e.g. &gt;, &lt;),
            they must be escaped.
         -->
         -->
         <commandString>
```

```
          /opt/OV/examples/OVI/configs/CommandResponder/add.sh
        </commandString>
        <!--
           If true, XML  output of command will be embedded
           in the response message, so it must be well
           formed XML. If false, all special XML characters
           will be escaped. (false is the safest option,
           but output may require further manipulation to
           convert it back to XML.)
        -->
        <commandReturnsXmlString>
           false
        </commandReturnsXmlString>
        <!--
           If true, input via STDIN is allowed to be passed to
           the command within the "input" tag in the request.
           If false, no STDIN will be allowed, and if there is
           an "input" tag, an OperationException XML will be
           returned.
        -->
        <allowInput>true</allowInput>
      </command>
   </validCommands>
   <!--
      Message trace logging switch.
   -->
   <traceMessages>true</traceMessages>
   <!--
      Message route logging switch.
   -->
   <traceRoute>true</traceRoute>
</commandResponderPluglet>
```

The following table describes elements in the `CommandResponderPluglet.config` file:

Element Name	**Value**

`<validCommands>`
`  <command></command>`
`</validCommands>`
	List of approved commands that the `CommandResponderPluglet` is allowed to execute.

`<commandString>`
`  ls`
`</commandString>`
	Command string that will be given to the OS command interpreter. If it has any special XML characters, they must be escaped. (e.g. `>`, `<`).

`<commandReturnsXmlString>`
`  boolean`
`</commandReturnsXmlString`
	If `true`, indicates that the output from the command will be embedded in the response message. This means that it must be well-formed XML. If the value of this tag is `false`, all special XML characters are escaped. (Default: `false`)

Element Name	Value
<allowInput> *boolean* </allowInput>	If `true`, input via `STDIN` is allowed to be passed to the command within the `input` tag in the request. If `false`, no `STDIN` is allowed, and if there is an `input` tag, an `OperationException` is returned. This is an optional element. (Default: `false`)
<traceMessages> *boolean* </traceMessages>	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See traceMessages Settings section for more details. (Default: `true`)
<traceRoute> *boolean* </traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and, eventually, displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

If you use `TestRequesterPluglet` (via `TestRequesterPlugletExampleNoInput.config` or `TestRequesterPlugletExampleWithInput.config`) to generate a test request for `CommandResponderPluglet`, you can modify `commandRequestNoInput.in` or `commandRequestWithInput.in`, respectively, to contain the command you want to execute. The request message in the `.in` file you select will be sent to `CommandResponderPluglet` by the `TestRequesterPluglet`. The `commandRequestWithInput.in` file's contents appear below, and elements that you should change are indicated. `commandRequestNoInput.in` is a subset of `commandRequestWithInput.in`.

Here is the `commandRequestWithInput.in` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<commandRequest
    xmlns="http://openview.hp.com/xmlns/ico/command"
    version="1.0">
    <!-- You should change this -->
    <command>sh ../config/app/command/reverse.sh</command>
    <!-- You should change this -->
    <input>
        line 1
        line 2
        line 3
    </input>
</commandRequest>
```

The following table describes elements in the `commandRequestWithInput.in` file:

Element Name	Value

`<command>`
`  sh ../config/app/command/reverse.sh`
`</command>`
	Command that the `CommandResponderPluglet` should execute.

`<input>`
`  line 1`
`  line 2`
`  line 3`
`</input>`
	Values to provide when the command prompts for input. This is an optional element and is used only if the supplied command normally accepts input from `STDIN`.

Namespaces

The namespace used in the command request/response XML messages are `xmlns=http://openview.hp.com/xmlns/ico/command`. This namespace is required for `<commandRequest>`, `<commandResponse>`, and `<commandException>` messages.

The contents of the command request XML message is only partially validated. Full validation is not performed for performance reasons. Instead, specific parts of each command request are checked.

Command Request/Response Message Format

The command request and response message schemas and example XML can be found in `OVI_BASE_DIR/examples/OVI/schemas/command`. (This directory is referred to as `SCHEMAS_DIR` below.

XML Message Description	Location
Command request message	`SCHEMAS_DIR/commandRequest.xml` and `SCHEMAS_DIR/commandRequestWithInput.xml`
Command request schema	`SCHEMAS_DIR/commandRequest.xsd`
Command response message	`SCHEMAS_DIR/commandResponse.xml` and `SCHEMAS_DIR/commandResponseWithInput.xml`
Command response schema	`SCHEMAS_DIR/commandResponse.xsd`

Command exception response message	`SCHEMAS_DIR/commandOperationException.xml`
	`SCHEMAS_DIR/commandInvalidXmlException.xml`
	`SCHEMAS_DIR/commandSecurityException.xml`

Command exception response schema	`SCHEMAS_DIR/commandException.xsd`

Application Pluglets

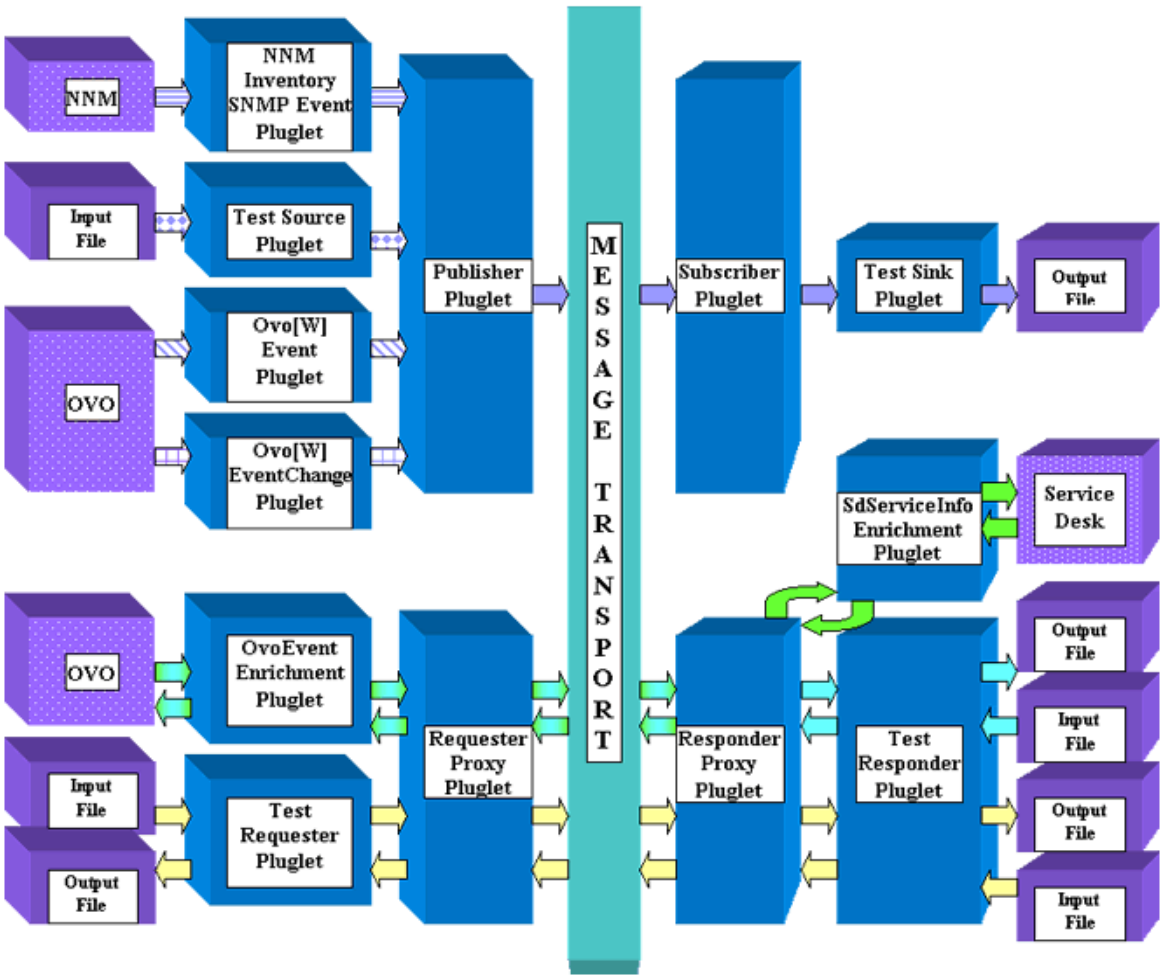
Pluglet	Description
NnmEventReadPluglet	Receives a message read request from RequesterProxyPluglet, obtains NNM messages meeting request criteria, and returns a message read response to the requesting pluglet via ResponderProxyPluglet.
NnmInventoryPluglet	Receives a network inventory request from RequesterProxyPluglet, queries NNM for inventory information, and returns a network inventory response to the requesting pluglet via ResponderProxyPluglet.
NNMInventorySnmpEventPluglet	Receives SNMP inventory event from HP Network Node Manager (NNM) and passes event information in an XML-formatted message to its targetPluglet.
NnmSnmpCollectPluglet	Receives a performance metric request from RequesterProxyPluglet, reads NNM SNMP collect data files, and returns a performance metric response to the requesting pluglet via ResponderProxyPluglet.
OvoEventPluglet or OvoWEventPluglet	Receives event from HP Operations Manager for Unix or HP Operations Manager for Windows and passes event information as XML-formatted message to PublisherPluglet.
OvoEventActionPluglet or OvoWEventActionPluglet	Receives message action request from RequesterProxyPluglet to execute an automatic or operator-initiated action defined on an OVO[W] message or to query the status of an action. The pluglet initiates the requested operation in OVO/U or OVO/W and returns the status of the action in a message action response via ResponderProxyPluglet.
OvoEventChangePluglet or OvoWEventChangePluglet	Receives filtered message change events from HP Operations Manager or HP Operations Manager for Windows and passes event information as XML-formatted message to PublisherPluglet.
OvoEventCreatePluglet	Receives request to create an OVO message (via OVO agent) and returns the OVO message ID of the newly created message.
OvoEventEnrichmentPluglet	Receives event from OVO, passes event information as XML-formatted message to RequesterProxyPluglet, waits for a response message, and when it is received, returns it to OVO.

Pluglet	Description
OvoEventModifyPluglet or OvoWEventModifyPluglet	Receives message modify request from RequesterProxyPluglet to modify an existing OVO/W message, initiates the requested operation and returns the ID of the message that has been modified in a message modify response via ResponderProxyPluglet.
OvoEventReadPluglet	Receives message read request from RequesterProxyPluglet to retrieve specified OVO event(s), initiates the requested operation and returns the event(s) in an message read response via ResponderProxyPluglet.
OvoInventoryPluglet	Receives inventory request from RequesterProxyPluglet to perform a create or delete operation, initiates the requested OVO operation on the specified node, and returns the result in an inventory response via ResponderProxyPluglet.
OvpmQueryPluglet	Receives a performance metric, performance report, performance metric info or performance source info request from RequesterProxyPluglet, sends request to Performance Manager, and returns the response to the requesting pluglet via ResponderProxyPluglet.
OvtaQueryPluglet	Receives transaction information request from RequesterProxyPluglet, sends request to Transaction Analyzer, and returns the response to the requesting pluglet via ResponderProxyPluglet.
SdInventoryPluglet	Receives inventory request to perform a create, read, update or delete operation from RequesterProxyPluglet, initiates the requested Service Desk operation on the specified configuration item, service, person, organization, incident or service call, and returns the result in an inventory response via ResponderProxyPluglet.
SdServiceInfoEnrichmentPluglet	Receives OVO message from RequesterProxyPluglet, obtains service-related data from Service Desk, adds this data to the original message, and returns it as a response to OVO via ResponderProxyPluglet.
SNQueryPluglet or SNWQueryPluglet	Receives service status information request or service hierarchy request from RequesterProxyPluglet, sends request to Service Navigator, and returns the response to the requesting pluglet via ResponderProxyPluglet.

Common Pluglet Usage Scenarios

The following figures explain how to assemble OVI pluglets to support several common usage scenarios. The role of each pluglet is described in each scenario. Click the Scenario links in each table for more detailed information.

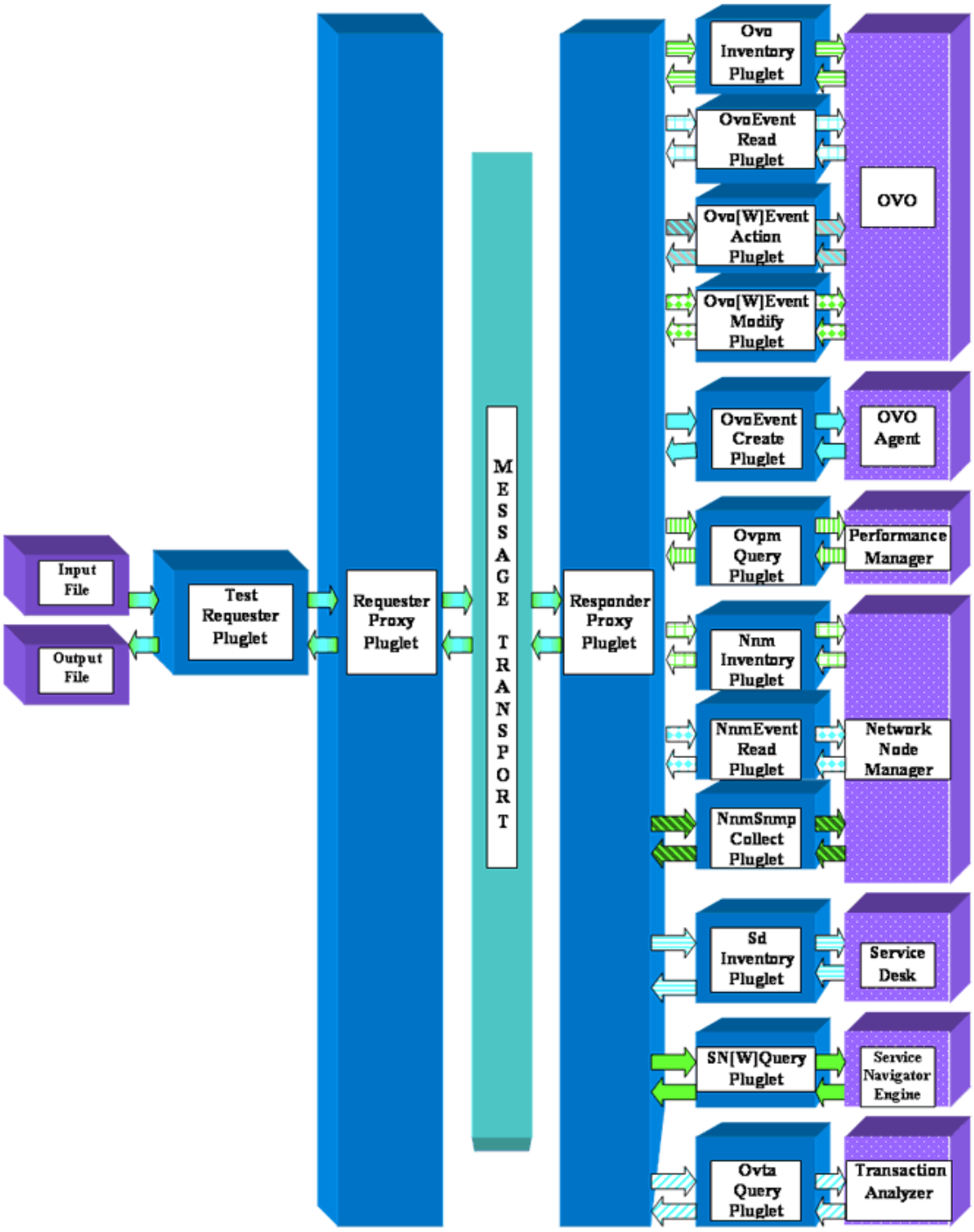
Figure 1



Scenario Description	Detailed Information
Send test message onto message transport and read it back. See Figure 1 objects connected with purple diamond arrows.	Scenario 1
Send test request onto message transport and wait for response. Also, read request from transport and send response. See Figure 1 objects connected with yellow arrows.	Scenario 2
(UNIX operating systems) Get message from OVO Message Stream Interface (MSI), send filtered message onto transport, read message from transport, and store in output file. See Figure 1 objects connected with diagonally striped purple arrows.	Scenario 3

Scenario Description	Detailed Information
(Windows operating systems) Get message from OVO/W Windows Management Instrumentation (WMI) Provider, send filtered message onto transport, read message from transport, and store in output file. See Figure 1 objects connected with diagonally striped purple arrows.	Scenario 4
Get message from OVO MSI, turn it into an enrichment request, send request onto transport, read request from transport, and store in output file. Then, read response from input file and return it to OVO. See Figure 1 objects connected with light blue arrows.	Scenario 5
Get message from OVO MSI, turn it into an enrichment request, send request onto transport, and read message from transport. Then, get related service information from Service Desk and use it to enrich message, which is returned to OVO. See Figure 1 objects connected with green arrows.	Scenario 6
Get SNMP inventory event from Network Node Manager, package event in `OviMessage` format, send onto transport, read message from transport, and store in output file. See Figure 1 objects connected with horizontally striped purple arrows.	Scenario 7
Get message change events from the OVO Message Event Interface (MEI), filter then package events in OviMessage format, send message onto transport, read message from transport, and store in output file. See Figure 1 objects connected with grid-patterned purple arrows.	Scenario 8

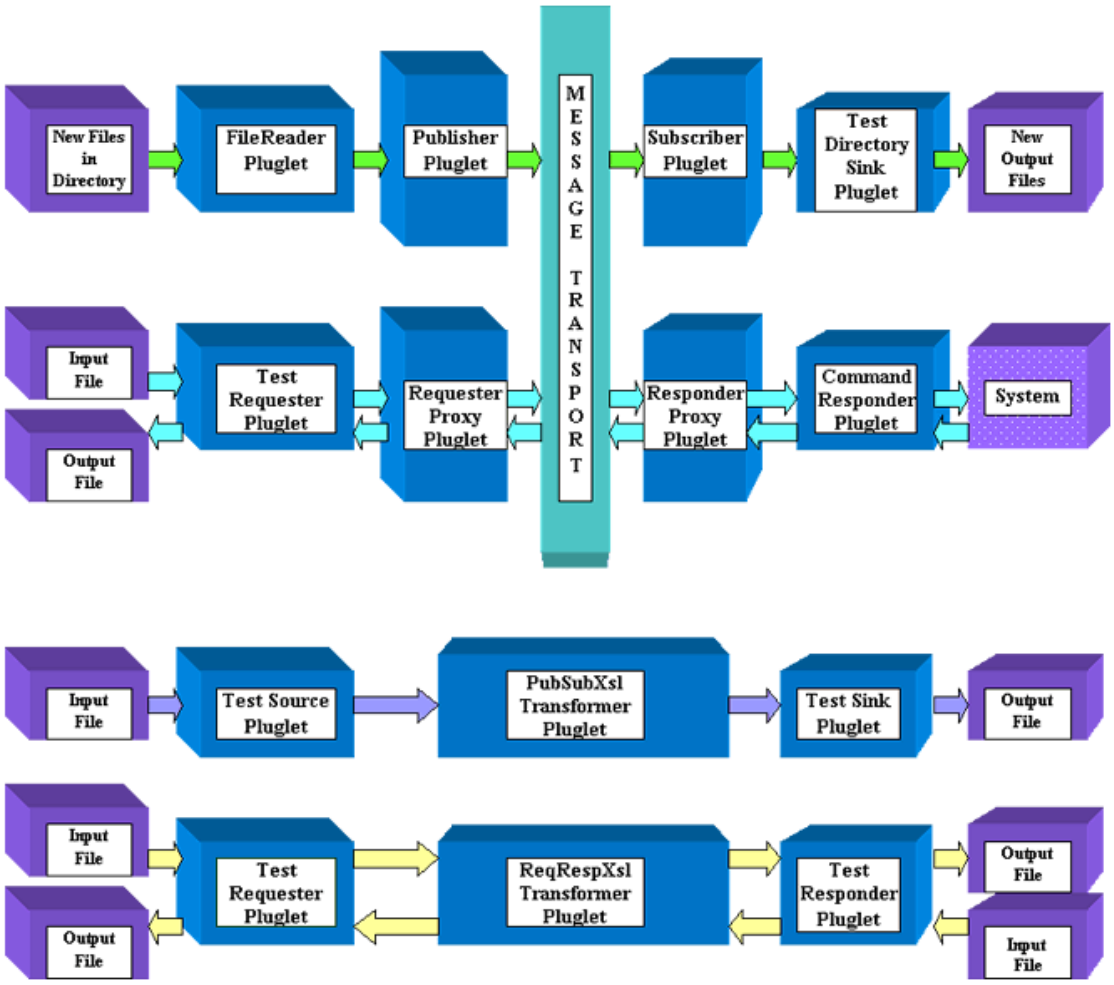
Figure 2


Input File
Output File
Test Requester Pluglet
Requester Proxy Pluglet
MESSAGE TRANSPORT
Responder Proxy Pluglet
Ovo Inventory Pluglet
OvoEvent Read Pluglet
Ovo[W]Event Action Pluglet
Ovo[W]Event Modify Pluglet
OVO
OvoEvent Create Pluglet
OVO Agent
Ovpm Query Pluglet
Performance Manager
Nnm Inventory Pluglet
NnmEvent Read Pluglet
NnmSnmp Collect Pluglet
Network Node Manager
Sd Inventory Pluglet
Service Desk
SN[W]Query Pluglet
Service Navigator Engine
Ovta Query Pluglet
Transaction Analyzer

Scenario Description	Detailed Information
Send request for creation of an OVO message to OVO agent and return ID of created OVO message to requesting pluglet. See Figure 2 objects connected with solid blue arrows.	Scenario 9
Read request for a service status from input file, send the request to transport, read request from transport, query Service Navigator for status of service, receive response from Service Navigator, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with solid green arrows.	Scenario 10
Read a performance metric request from input file, send the request to transport, read request from transport, query HP Performance Manager for performance metrics, receive response from Performance Manager, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with green vertically striped arrows.	Scenario 11
Read a network inventory request from input file, send the request to transport, read request from transport, query HP Network Node Manager for inventory data, receive response from Network Node Manager, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with green grid-patterned arrows.	Scenario 12
Read a performance metric request from input file, send the request to transport, read request from transport, gather required performance data from HP Network Node Manager SNMP data files, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with dark green diagonally striped arrows.)	Scenario 13
Read request for NNM events from input file, send the request to transport, read request from transport, gather events matching specified criteria from HP Network Node Manager, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with blue diamond arrows.)	Scenario 14
Read a transaction metric request from input file, send the request to transport, read request from transport, gather required transaction data from HP Transaction Analyzer, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with blue/white diagonally striped arrows.	Scenario 15
Read request for a create, read, update or delete operation from input file; send request to transport; read request from transport; initiate requested operation (on specified configuration item, service, person or organization) via HP Service Desk and receive results; send response containing results to transport; read response from transport; and store in output file. See Figure 2 objects connected with blue horizontally striped arrows.	Scenario 16

Scenario Description	Detailed Information
Read request (from input file) to modify an OVO/U or OVO/W message; send request to transport; read request from transport; initiate requested operation (on specified message UUID) and receive results; send response containing modified message's UUID to transport; read response from transport; and store in output file. See Figure 2 objects connected with green diamond arrows.	Scenario 17
Read request to create or delete OVO node (from input file); send request to transport; read request from transport; initiate requested OVO operation and receive results; send response containing operation's result to transport; read response from transport; and store in output file. See Figure 2 objects connected with green horizontally striped arrows.	Scenario 18
Read request (from input file) to retrieve OVO event identified by the supplied UUID; send request to transport; read request from transport; initiate OVO request and receive results; send response containing the requested event to transport; read response from transport; and store in output file. See Figure 2 objects connected with blue grid-patterned arrows.	Scenario 19
Read request (from input file) to retrieve OVO events in the supplied time range; send request to transport; read request from transport; initiate OVO request and receive results; send response containing the requested events to transport; read response from transport; and store in output file. See Figure 2 objects connected with blue grid-patterned arrows.	Scenario 20
Read request (from input file) to execute an automatic or operator-initiated action defined on an OVO/U or OVO/W message; send request to transport; read request from transport; initiate requested action and receive results of its initiation; send response containing initiation results to transport; read response from transport; and store in output file. See Figure 2 objects connected with blue/gray diagonally striped arrows.	Scenario 21

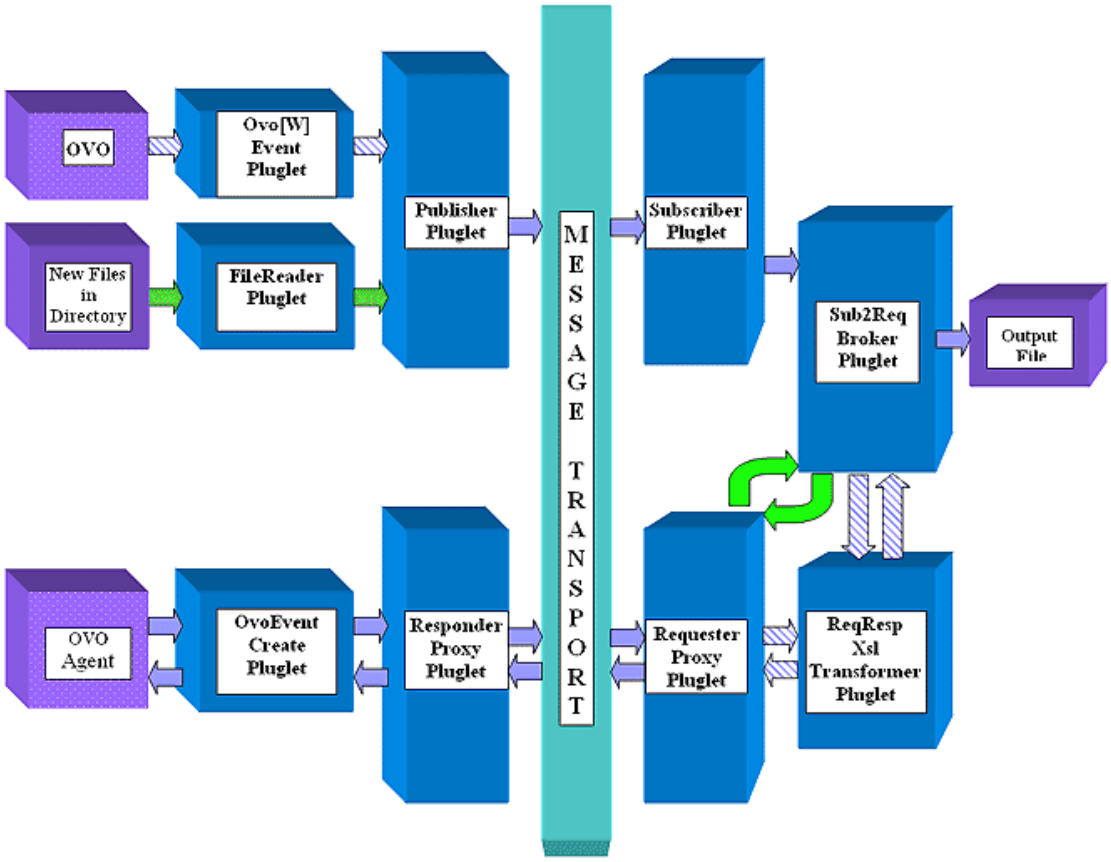
Figure 3



Scenario Description	Detailed Information
A process or application writes new files into a specified directory. `FileReaderPluglet` periodically monitors that directory, reads each new file that is there, and sends a message containing the file's content to `TestDirectorySinkPluglet`, which writes each message to a new file in another (configured) directory. The original file is then deleted from the monitored directory. See Figure 3 objects connected with green arrows.	Scenario 22

Scenario Description	Detailed Information
Send message and transform it. See Figure 3 objects connected with purple arrows.	Scenario 23
Send request message, transform request message, receive response message, and transform response message. See Figure 3 objects connected with the yellow arrows.	Scenario 24
Read request to execute command (from input file); send request to transport; read request from transport; initiate requested command on system and receive results; send response containing command's result to transport; read response from transport; and store in output file. See _Figure 3 objects connected with blue arrows.	Scenario 25

Figure 4



Scenario Description	Detailed Information
Get message from OVO MSI, send filtered message onto transport, and read message from transport. Then `Sub2ReqBrokerPluglet` forwards it and `PubSubXslTransformerPluglet` turns it into an event creation request and sends it to an OVO agent. See Figure 4 objects connected with diagonally striped purple arrows.	Scenario 26

Scenario 1

Send test message onto message transport and read it back. See Figure 1 objects connected with purple diamond arrows.

Deploy File	`PubSubExample.deploy`

Modify (to match your environment)	`PublisherPlugletExample.<selectedTransport>.config`
	`SubscriberPlugletExample.<selectedTransport>.config`
	Optionally:
	`TestSourcePluglet.config`
	`TestSinkPluglet.config`

Pluglets Involved	`TestSourcePluglet`
	`PublisherPluglet`
	`SubscriberPluglet`
	`TestSinkPluglet`

Default Values of Interest	Test message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/PublishSubscribeTest/TestSourcePlugletExample.in`
	Output file default:
	`OVI_DATA_DIR/tmp/OVI/TestSinkPlugletExample.out`

Scenario 2

Send test request onto message transport and wait for response. Also, read request from transport and send response. See Figure 1 objects connected with yellow arrows.

Deploy File	`RequestRespondExample.deploy`

Modify (to match your environment)	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`ResponderProxyPlugletExample.<selectedTransport>.config`
	Optionally:
	`TestRequesterPluglet.config`
	`TestResponderPluglet.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`TestResponderPluglet`

Default Values of Interest	Test message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/RequestResponseTest/TestRequesterPlugletExample.in`
	Output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`
	Message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/RequestResponseTest/TestResponderPlugletExample.in`
	Message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestResponderPlugletExample.out`

Scenario 3

(UNIX operating systems) Get message from OVO Message Stream Interface (MSI), send filtered message onto transport, read message from transport, and store in output file. See Figure 1 objects connected with diagonally striped purple arrows.

Deploy File	`OvoEventPlugletExample.deploy`

Modify (to match your environment)	`PublisherPlugletExample.<selectedTransport>.config`
	`OvoEventPlugletExample.config`
	Optionally:
	`SubscriberPlugletExample.<selectedTransport>.config`
	`TestSinkPluglet.config`

Pluglets Involved	`OvoEventPluglet`
	`PublisherPluglet`
	`SubscriberPluglet`
	`TestSinkPluglet`

Default Values of Interest	Output file default: `OVI_DATA_DIR/tmp/OVI/TestSinkPlugletExample.out`

Scenario 4

(Windows operating systems) Get message from OVO/W Windows Management Instrumentation (WMI) Provider, send filtered message onto transport, read message from transport, and store in output file. See Figure 1 objects connected with diagonally striped purple arrows.

Deploy File	`OvoWEventPlugletExample.deploy`

Modify (to match your environment)	`PublisherPlugletExample.<selectedTransport>.config`
	`OvoWEventPlugletExample.config`
	Optionally:
	`SubscriberPlugletExample.<selectedTransport>.config`
	`TestSinkPluglet.config`

Pluglets Involved	`OvoWEventPluglet`
	`PublisherPluglet`
	`SubscriberPluglet`
	`TestSinkPluglet`

Default Values of Interest	Output file default: `OVI_DATA_DIR/tmp/OVI/TestSinkPlugletExample.out`

Scenario 5

Get message from OVO MSI, turn it into an enrichment request, send request onto transport, read request from transport, and store in output file. Then, read response from input file and return it to OVO. See Figure 1 objects connected with light blue arrows.

Deploy File	`OvoEventEnrichmentExample.deploy`

Modify (to match your environment)	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`OvoEventEnrichmentPlugletExample.config`
	Optionally:
	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`TestResponderPluglet.config`

Pluglets Involved	`OvoEventEnrichmentPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`TestResponderPluglet`

Default Values of Interest	OVO message output file default: `OVI_DATA_DIR/tmp/OVI/TestResponderPlugletExample.out` Input file default: `OVI_BASE_DIR/examples/OVI/configs/OVO/EventEnrichment/TestResponderPlugletExample.in`

Scenario 6

Get message from OVO MSI, turn it into an enrichment request, send request onto transport, and read message from transport. Then, get related service information from Service Desk and use it to enrich message, which is returned to OVO. See Figure 1 objects connected with green arrows.

Deploy File	`SdServiceInfoEnrichmentPlugletExample.deploy`
	`OvoEventEnrichmentPlugletExample.deploy`
	Note: Start two OVI instances to perform this scenario, with each instance using one of the specified `.deploy` files.

Modify (to match your environment)	`OvoEventEnrichmentPlugletExample.config`
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`SdServiceInfoEnrichmentPlugletExample.config`

Pluglets Involved	`OvoEventEnrichmentPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`SdServiceInfoEnrichmentPluglet`

Default Values of Interest	(none)

Scenario 7

Get SNMP inventory event from Network Node Manager, package event in `OviMessage` format, send onto transport, read message from transport, and store in output file. See Figure 1 objects connected with horizontally striped purple arrows.

Deploy File	`NnmInventorySnmpEventPlugletExample.<selectedTransport>.deploy`

Modify (to match your environment)	`PublisherPlugletExample.<selectedTransport>.config`
	`NnmInventorySnmpEventPlugletExample.config`
	Optionally:
	`SubscriberPlugletExample.<selectedTransport>.config`
	`TestSinkPlugletExample.config`

Pluglets Involved	`NnmInventorySnmpEventPluglet`
	`PublisherPluglet`
	`SubscriberPluglet`
	`TestSinkPluglet`

Default Values of Interest	Output file default: `OVI_DATA_DIR/tmp/OVI/TestSinkPlugletExample.out`

Scenario 8

Get message change events from the OVO Message Event Interface (MEI), filter then package events in OviMessage format, send message onto transport, read message from transport, and store in output file. See Figure 1 objects connected with grid-patterned purple arrows.

Deploy File	`OvoEventChangePlugletExample.deploy` (UNIX) or `OvoWEventChangePlugletExample.deploy` (Windows)

Modify (to match your environment)	`PublisherPlugletExample.<selectedTransport>.config`
	`OvoEventChangePlugletExample.config` (UNIX) or
	`OvoWEventChangePlugletExample.config` (Windows)
	Optionally:
	`SubscriberPlugletExample.<selectedTransport>.config`
	`TestSinkPlugletExample.config`

Pluglets Involved	`OvoEventChangePluglet` (UNIX) or
	`OvoWEventChangePluglet` (Windows)
	`PublisherPluglet`
	`SubscriberPluglet`
	`TestSinkPluglet`

Default Values of Interest	Output file default: `OVI_DATA_DIR/tmp/OVI/TestSinkPlugletExample.out`

Scenario 9

Send request for creation of an OVO message to OVO agent and return ID of created OVO message to requesting pluglet. See Figure 2 objects connected with solid blue arrows.

Deploy File	`OvoEventCreationPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`OvoEventCreationPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`OvoEventCreatePluglet`

Default Values of Interest	Test message input file default: `OVI_BASE_DIR/examples/OVI/configs/OVO/EventCreation/TestRequesterPlugletExample.in`

Scenario 10

Read request for a service status from input file, send the request to transport, read request from transport, query Service Navigator for status of service, receive response from Service Navigator, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with solid green arrows.

Deploy File	`SNQueryPlugletExample.deploy` (UNIX) or `SNWQueryPlugletExample.deploy` (Windows)

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`SNQueryPlugletExample.config` or
	`SNWQueryPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`SNQueryPluglet`

Default Values of Interest	Service Navigator message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`
	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/SN/Query/TestRequesterPlugletExample.in`

Scenario 11

Read a performance metric request from input file, send the request to transport, read request from transport, query HP Performance Manager for performance metrics, receive response from Performance Manager, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with green vertically striped arrows.

Deploy File	`OvpmQueryPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`OvpmQueryPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`OvpmQueryPluglet`

Default Values of Interest	HP Performance Manager message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`
	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/OVPM/Query/TestRequesterPlugletExample.in`

Scenario 12

Read a network inventory request from input file, send the request to transport, read request from transport, query HP Network Node Manager for inventory data, receive response from Network Node Manager, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with green grid-patterned arrows.

Deploy File	`NnmInventoryPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`NnmInventoryPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`NnmInventoryPluglet`

Default Values of Interest	HP Network Node Manager message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`
	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/NNM/Inventory/TestRequesterPlugletExample.in`

Scenario 13

Read a performance metric request from input file, send the request to transport, read request from transport, gather required performance data from HP Network Node Manager SNMP data files, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with dark green diagonally striped arrows.

Deploy File	`NnmSnmpCollectPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`NnmSnmpCollectPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`NnmSnmpCollectPluglet`

Default Values of Interest	HP Network Node Manager message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`
	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/NNM/SnmpCollect/TestRequesterPlugletExample.in`

Scenario 14

Read request for NNM events from input file, send the request to transport, read request from transport, gather events matching specified criteria from HP Network Node Manager, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with blue diamond arrows.

Deploy File	`NnmEventReadPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`NnmEventReadPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`NnmEventReadPluglet`

Default Values of Interest	HP Network Node Manager message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`
	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/NNM/EventRead/NnmEventReadPlugletExampleRequest.in`

Scenario 15

Read a transaction metric request from input file, send the request to transport, read request from transport, gather required transaction data from HP Transaction Analyzer, send response to transport, read response from transport, and store in output file. See Figure 2 objects connected with blue diagonally striped arrows.

Deploy File	`OvtaQueryPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`OvtaQueryPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`OvtaQueryPluglet`

Default Values of Interest	HP Transaction Analyzer message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`
	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/OVTA/Query/TestRequesterPlugletExample.in`

Scenario 16

Read request for a create, read, update or delete operation from input file; send request to transport; read request from transport; initiate requested operation (on specified configuration item, service, person or organization) via HP Service Desk and receive results; send response containing results to transport; read response from transport; and store in output file. See Figure 2 objects connected with blue horizontally striped arrows.

Deploy File	`SdInventoryPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`SdInventoryPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`SdInventoryPluglet`

Default Values of Interest	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/SD/Inventory/TestRequesterPlugletExample_createCI.in`
	Response message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`

Scenario 17

Read request (from input file) to modify an OVO/U or OVO/W message; send request to transport; read request from transport; initiate requested operation (on specified message UUID) and receive results; send response containing modified message's UUID to transport; read response from transport; and store in output file. See Figure 2 objects connected with green diamond arrows.

Deploy File	`Ovo[W]EventModifyPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`Ovo[W]EventModifyPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`Ovo[W}EventModifyPluglet`

Default Values of Interest	Request message input file default: `OVI_BASE_DIR/examples/OVI/configs/OVO/EventModify/ovMessageModifyAcknowledgeRequest.in` Response message output file default: `OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`

Scenario 18

Read request to create or delete OVO node (from input file); send request to transport; read request from transport; initiate requested OVO operation and receive results; send response containing operation's result to transport; read response from transport; and store in output file. See Figure 2 objects connected with green horizontally striped arrows.

Deploy File	`OvoInventoryPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`OvoInventoryPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`OvoInventoryPluglet`

Default Values of Interest	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/OVO/Inventory/TestRequesterPlugletExample_createNode.in`
	Response message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`

Scenario 19

Read request (from input file) to retrieve OVO event identified by the supplied UUID; send request to transport; read request from transport; initiate OVO request and receive results; send response containing the requested event to transport; read response from transport; and store in output file. See Figure 2 objects connected with blue grid-patterned arrows.

Deploy File	`OvoEventReadPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`OvoEventReadPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`OvoEventReadPluglet`

Default Values of Interest	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/OVO/EventRead/OvMessageUUIDReadRequestExample.in`
	Response message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`

Scenario 20

Read request (from input file) to retrieve OVO events in the supplied time range; send request to transport; read request from transport; initiate OVO request and receive results; send response containing the requested events to transport; read response from transport; and store in output file. See Figure 2 objects connected with blue grid-patterned arrows.

Deploy File	`OvoEventReadPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`OvoEventReadPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`OvoEventReadPluglet`

Default Values of Interest	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/OVO/EventRead/OvMessageTimeRangeReadRequestExample.in`
	Response message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`

Scenario 21

Read request (from input file) to execute an automatic or operator-initiated action defined on an OVO/U or OVO/W message; send request to transport; read request from transport; initiate requested action and receive results of its initiation; send response containing initiation results to transport; read response from transport; and store in output file. See Figure 2 objects connected with blue/gray diagonally striped arrows.

Deploy File	`Ovo[W]EventActionPlugletExample.deploy`

Modify (to match your environment)	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`Ovo[W]EventActionPlugletExample.config`
	Optionally:
	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`TestRequesterPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`Ovo[W}EventActionPluglet`

Values of Interest	Request message input file options:
	`OVI_BASE_DIR/examples/OVI/configs/OVO/EventAction/ovMessageActionExecuteAutomaticActionRequest.in` or `ovMessageActionExecuteOperatorActionRequest.in`
	Response message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`

Scenario 22

A process or application writes new files into a specified directory. `FileReaderPluglet` periodically monitors that directory, reads each new file that is there, and sends a message containing the file's content to `TestDirectorySinkPluglet`, which writes each message to a new file in another (configured) directory. The original file is then deleted from the monitored directory. See Figure 3 objects connected with green arrows.

Deploy File	`FileReaderPlugletExample.deploy`

Modify (to match your environment)	`PublisherPlugletExample.<selectedTransport>.config`
	`FileReaderPlugletExample.config`
	Optionally:
	`SubscriberPlugletExample.<selectedTransport>.config`
	`TestDirectorySinkPlugletExample.config`

Pluglets Involved	`FileReaderPluglet`
	`PublisherPluglet`
	`SubscriberPluglet`
	`TestDirectorySinkPluglet`

Default Values of Interest	Input file directory default:
	`OVI_DATA_DIR/tmp/OVI/ExampleReadDir`
	Output file directory default:
	`OVI_DATA_DIR/tmp/OVI/ExampleWriteDir`

Scenario 23

Send message and transform it. See Figure 3 objects connected with purple arrows.

Deploy File	`PubSubXslTransformerExample.deploy`

Modify (to match your environment)	`PubSubXslTransformerPlugletExample.config`
	Optionally:
	`TestSourcePlugletExample.config`
	`TestSinkPlugletExample.config`

Pluglets Involved	`TestSourcePluglet`
	`PubSubXslTransformerPluglet`
	`TestSinkPluglet`

Default Values of Interest	Test message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/Transformer/PublishSubscribe/TestSourcePlugletExample.in`
	Stylesheet default for XSL transformation:
	`OVI_BASE_DIR/examples/OVI/configs/Transformer/xsl/ChangeElement_nodeName2hostName.xsl`
	Output file default:
	`OVI_DATA_DIR/tmp/OVI/TestSinkPlugletExample.out`

Scenario 24

Send request message, transform request message, receive response message, and transform response message. See Figure 3 objects connected with the yellow arrows.

Deploy File	`ReqRespXslTransformerExample.deploy`

Modify (to match your environment)	`ReqRespXslTransformerPlugletExample.config`
	Optionally:
	`TestRequesterPlugletExample.config`
	`TestResponderPlugletExample.config`

Pluglets Involved	`TestRequesterPluglet`
	`ReqRespXslTransformerPluglet`
	`TestResponderPluglet`

Default Values of Interest	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/Transformer/RequestResponse/TestRequesterPlugletExample.in`
	Stylesheet default for XSL transformation for request message:
	`OVI_BASE_DIR/examples/OVI/configs/Transformer/xsl/ChangeElement_nodeName2hostName.xsl`
	Response message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/Transformer/RequestResponse/TestResponderPlugletExample.in`
	Stylesheet default for XSL transformation for response message:
	`OVI_BASE_DIR/examples/OVI/configs/Transformer/xsl/ChangeElement_nodeName2hostName.xsl`

Scenario 25

Read request to execute command (from input file); send request to transport; read request from transport; initiate requested command on system and receive results; send response containing command's result to transport; read response from transport; and store in output file. See Figure 3 objects connected with blue arrows.

Deploy File	`CommandResponderPlugletExample.deploy`

Modify (to match your environment)	`RequesterProxyPlugletExample.<selectedTransport>.config`
	`ResponderProxyPlugletExample.<selectedTransport>.config`
	`CommandResponderPlugletExample.config`
	Optionally:
	`TestRequesterPlugletExampleNoInput.config`

Pluglets Involved	`TestRequesterPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`CommandResponderPluglet`

Default Values of Interest	Request message input file default:
	`OVI_BASE_DIR/examples/OVI/configs/OVO/CommandResponder/commandRequestNoInput.in`
	Response message output file default:
	`OVI_DATA_DIR/tmp/OVI/TestRequesterPlugletExample.out`

Scenario 26

Get message from OVO MSI, send filtered message onto transport, and read message from transport. Then `Sub2ReqBrokerPluglet` forwards it and `PubSubXslTransformerPluglet` turns it into an event creation request and sends it to an OVO agent. See Figure 4 objects connected with diagonally striped purple arrows.

Deploy File	`Sub2ReqBrokerPlugletExample.deploy`

Modify (to match your environment)	`PublisherPlugletExample.<selectedTransport>.config`
	`OvoEventPlugletExample.config`
	`Sub2ReqBrokerPluglet.config`
	`PubSubXslTransformerPluglet.config`
	`RequesterProxyPluglet.<selectedTransport>.config`
	`ResponderProxyPluglet.<selectedTransport>.config`
	`OvoEventCreatePluglet.config`
	Optionally:
	`SubscriberPlugletExample.<selectedTransport>.config`
	`TestSinkPluglet.config`

Pluglets Involved	`OvoEventPluglet`
	`PublisherPluglet`
	`SubscriberPluglet`
	`Sub2ReqBrokerPluglet`
	`PubSubXslTransformerPluglet`
	`RequesterProxyPluglet`
	`ResponderProxyPluglet`
	`OvoEventCreatePluglet`

Default Values of Interest	Output file default: `OVI_DATA_DIR/tmp/OVI/TestSinkPlugletExample.out`

Custom Pluglet Groupings

In addition to assembling pluglets for common usage scenarios, you can create custom pluglet arrangements for specialized needs. To do this successfully, you must remember:

A pluglet can interact only with other pluglets that implement the interface it requires.

This means, for example, that pluglet A's target pluglet, as specified in pluglet A's `.config` file, must be another pluglet that implements the interface required by pluglet A. Pluglets that do not implement the specified interface cannot be pluglet-A's target pluglet. This table gives pluglet interface details:

Pluglet	Interfaces Implemented	Its Target Pluglet Must Implement
CommandResponderPluglet	SyncRequestListener	--
FileReaderPluglet	--	OviMessageListener
NnmEventReadPluglet	SyncRequestListener	--
NnmInventoryPluglet	SyncRequestListener	--
NnmInventorySnmpEventPluglet	--	OviMessageListener
NnmSnmpCollectPluglet	SyncRequestListener	--
OvoEventPluglet	--	OviMessageListener
OvoWEventPluglet	--	OviMessageListener
OvoEventActionPluglet	SyncRequestListener	--
OvoWEventActionPluglet	SyncRequestListener	--
OvoEventChangePluglet	--	OviMessageListener
OvoWEventChangePluglet	--	OviMessageListener
OvoEventCreatePluglet	SyncRequestListener	--
OvoEventEnrichmentPluglet	--	SyncRequestListener
OvoEventModifyPluglet	SyncRequestListener	--
OvoWEventModifyPluglet	SyncRequestListener	--
OvoEventReadPluglet	SyncRequestListener	--
OvoInventoryPluglet	SyncRequestListener	--
OvpmQueryPluglet	SyncRequestListener	--
OvtaQueryPluglet	SyncRequestListener	--
PublisherPluglet	OviMessageListener	--
PubSubXslTransformerPluglet	OviMessageListener	OviMessageListener
ReqRespXslTransformerPluglet	SyncRequestListener	SyncRequestListener

Pluglet	Interfaces Implemented	Its Target Pluglet Must Implement
RequesterProxyPluglet	SyncRequestListener	--
ResponderProxyPluglet	--	SyncRequestListener
SdInventoryPluglet	SyncRequestListener	--
SdServiceInfoEnrichmentPluglet	SyncRequestListener	--
SNQueryPluglet	SyncRequestListener	--
SNWQueryPluglet	SyncRequestListener	--
SubscriberPluglet	--	OviMessageListener
TestDirectorySinkPluglet	OviMessageListener	--
TestRequesterPluglet	--	SyncRequestListener
TestResponderPluglet	SyncRequestListener	--
TestSinkPluglet	OviMessageListener	--
TestSourcePluglet	--	OviMessageListener

SOAP Message Support

OVI provides support for receiving SOAP-formatted requests and returning SOAP-formatted responses over HTTP(S). This enables external applications (for example, .NET clients) to easily interact with OVI. Web Services Description Language (WSDL) files, which define the format of SOAP OVI request and response messages, are included in the `OVI_BASE_DIR/examples/OVI/wsdl` subdirectories. They are useful for creating client applications that can communicate with OVI. (Currently, .NET 2002 and Apache Axis 1.1rc2 clients have been verified to work with OVI.) Once created, a client's SOAP-formatted requests can be sent to a special OVI URL, `http://<server-hostname>:<server-port>/ovi-soap/<targetPluglet>`, as described in HTTP Pluglet Configuration File Example.

For your convenience, simple Java and .NET C# clients are provided in the `OVI_BASE_DIR/examples/OVI/sampleCode/soapClients` subdirectories. `.bat` files are included in the `csharp` subdirectory, for generating .NET client executables. The .NET Framework and the .NET SDK are required for the proper execution of these `.bat` files, which rely on `wsdl.exe`, and the C# compiler `csc.exe`. Script files are included in the `java` subdirectory for generating Java for Apache Axis clients. The environment variable `AXIS_HOME` must be set prior to executing these scripts.

Two primary scenarios exist for interacting with OVI via SOAP over HTTP(S):

- OVI WSDL is used to develop an external application (e.g., .NET or Axis client). The application sends SOAP-formatted requests to an OVI system at `http://<server-hostname>:<server-port>/ovi-soap/<targetPluglet>`. The OVI system that receives these requests has been started with a `.deploy` file containing only a `ResponderProxyPluget` (using HTTP as its transport) and an application pluglet (e.g., `SNQueryPluglet`). OVI processes each SOAP request and returns to the client a SOAP-formatted response containing the requested data or a SOAP Fault if an error occurs during processing.

- A custom OVI pluglet that creates SOAP-formatted requests is developed. It passes these requests to a `RequesterProxyPluglet` that is configured as described in HTTP Pluglet Configuration File Example. The `RequesterProxyPluglet` passes requests to a `ResponderProxyPluglet`, which, in turn, sends them to an application pluglet (e.g., `SNQueryPluglet`). A SOAP-formatted response containing the requested data or a SOAP Fault is returned to the custom pluglet.

5 Integrating with HP Operations Manager

OVO pluglets enable interaction with HP Operations Manager for Unix and HP Operations Manager for Windows.

On Windows® operating systems, pluglets work with HP Operations Manager for Windows to:

- receive event and event change data from OVO/W
- modify existing messages in OVO/W
- create events in OVO/W
- execute the automatic or operator-initiated action defined on an OVO/W message and return the status of this action when queried

On UNIX operating systems, pluglets interact with HP Operations Manager for Unix to:

- receive event and event change data from OVO
- receive an event and return an enhanced message to OVO
- retrieve existing events that meet user-specified criteria
- modify existing messages in OVO
- create events in OVO
- execute the automatic or operator-initiated action defined on an OVO message and return the status of this action when queried
- create and delete existing OVO nodes

There are two categories of OVO pluglets:

- Pluglets that interact with the OVO management server, including the `OvoEventEnrichmentPluglet`, `OvoEventPluglet`, `OvoEventReadPluglet`, `OvoEventChangePluglet`, `OvoEventModifyPluglet`, `OvoEventActionPluglet` and `OvoInventoryPluglet` pluglets. They must be installed on an OVO (UNIX) server system. On Windows operating systems, the pluglets do not have this restriction and can be installed on a system with network access to the OVO/W management server.
- Pluglets that interact with an OVO agent (including the `OvoEventCreatePluglet` pluglet), which must be installed on an OVO agent system.

Configuration Overview

To enable data from OVOU or OVO/W to flow through OVI, the following configuration tasks must be performed:

1 For UNIX operating systems only: Configure OVO to put messages on the OVO Message Stream Interface (MSI). See OVO Configuration (UNIX only) section.

2 For Windows operating systems only: Run OVI under an account that belongs to the `OV_Operators` group of the OVO/W management server.

3 Configure the appropriate application pluglets to achieve your intent:

Pluglet Name	Purpose	Detailed Information In
`OvoInventoryPluglet`	To create or delete nodes within OVO. UNIX only.	OvoInventoryPluglet Configuration
`OvoEventPluglet`	To receive notification of messages created in OVO (i.e., to pass OVO messages through OVI). To use this pluglet, you must understand how to configure OVO and how to filter and make use of the resulting OVO messages that OVI delivers. UNIX only.	OvoEventActionPluglet Configuration
`OvoEventChangePluglet`	To receive OVO message change events. UNIX only.	OvoEventChangePluglet Configuration
`OvoEventModifyPluglet`	To modify existing OVO messages. UNIX only.	OvoEventModifyPluglet Configuration
`OvoEventReadPluglet`	To retrieve specified OVO messages by specifying a UUID or time range. UNIX only.	OvoEventReadPluglet Configuration
`OvoEventEnrichmentPluglet`	To enrich OVO messages using your own custom application. To use this pluglet, you must understand how to configure OVO and how to filter and make use of the resulting OVO messages that OVI delivers. UNIX only.	OvoEventEnrichmentPluglet Configuration
`OvoEventActionPluglet`	To execute automatic and operator-initiated actions defined on OVO messages. It also allows you to query the status of those actions. UNIX only.	OvoEventActionPluglet Configuration

Pluglet Name	Purpose	Detailed Information In
`OvoEventCreatePluglet`	To create messages within OVOU or OVO/W. UNIX and Windows.	OvoEventChangePluglet Configuration
`OvoWEventPluglet`	To receive notification of messages created in OVO/W (i.e., to pass OVO/W messages through OVI). To use this pluglet, you must understand how to filter and make use of the resulting OVO/W messages that OVI delivers. Windows only.	OvoWEventPluglet Configuration
`OvoWEventChangePluglet`	To receive message change events from OVO/W. Windows only.	OvoWEventChangePluglet Configuration
`OvoWEventModifyPluglet`	To modify existing OVO/W messages. Windows only.	OvoWEventModifyPluglet Configuration
`OvoWEventActionPluglet`	To execute automatic and operator-initiated actions defined on OVO/W messages. It also allows you to query the status of those actions. Windows only.	OvoWEventActionPluglet Configuration

Use of the `OvoWEventPluglet`, `OvoWEventChangePluglet`, `OvoWEventActionPluglet` or `OvoWEventModifyPluglet` causes a WMI connection to be made to OVO/W. The number and configuration of these WMI connections are limited and affect how an OVI `.deploy` file containing these pluglets may be constructed. Each `.deploy` file's associated pluglet `.config` files must observe the following limitations:

- a single remote connection to WMI, *or*
- multiple local connections to WMI, *or*
- multiple remote connections to WMI if all pluglet `.config` files associated with the `.deploy` use the same OVO `wmiMachineName`, `wmiUserName` and password.

OVO Configuration (UNIX only)

OVO does not, by default, put its new message events onto the server Message Stream Interface (MSI) where an OVI pluglet can read them. To configure OVO to place events onto the MSI, perform one of the following actions on the OVO server:

Special configuration activites are NOT required to receive message change events from the OVO Message Event Interface (MEI).

- If you want a copy of *all* OVO messages to go onto the server MSI: From the GUI, click **Actions:Server → Configure**. In the MSI configuration area, select **Enable Output, Send All Messages to Server MSI**, and **Copy Messages**.
- If you want to copy or divert only *selected* events onto the server MSI: From the GUI, click **Actions:Server → Configure**. In the MSI configuration area, select **Enable Output**. Then, configure each OVO event template condition. This approach greatly reduces the number of events placed onto the MSI, and it can also prevent sensitive events from being sent to the MSI where any integrating product can gain access.

  If you choose this method, you must change all related preexisting template conditions to copy or divert events to the MSI. Also, you must turn off the OVO server configuration. If there are many templates with many conditions in each template, this approach is labor-intensive.

  If you are using OVI's `OvoEventPluglet`, you should *copy* events onto the MSI unless you have another MSI connection that requires events to be diverted. If you are using the `OvoEventEnrichmentPluglet`, you should *divert* events to the MSI; this ensures that the OVO operators receive only one copy of each event.

For more details on configuring OVO, please consult the *HP OpenView Operations for UNIX Concepts Guide*.

OVO and OVO/W Pluglet Configuration

OVI provides four application pluglets that receive events from the OVO MSI/MEI (UNIX operating systems):

- `OvoEventPluglet`: When this pluglet receives an OVO event from the MSI, it places the event data into an XML-formatted notification message, which it passes to its target pluglet.
- `OvoEventChangePluglet`: When this pluglet receives a message change event from the OVO MEI, it places the event information in an XML-formatted change event message, which it passes to its target pluglet. Change events handled by this pluglet include own/disown, acknowledge/unacknowledge, change severity, change duplicate, modify message text and add/delete/modify annotation text.\
- `OvoEventEnrichmentPluglet`: When this pluglet receives an OVO event from the MSI, the event data is placed into an enriched XML message and passed to its target pluglet. `OvoEventEnrichmentPluglet` can also receive a response to the enriched message, which it returns to OVO.

In addition to receiving events, the OvoEventEnrichmentPluglet can also return a message to OVO containing updated, or enriched, event data. See the SdServiceInfoEnrichmentPluglet Configuration section for more information.

- OvoEventReadPluglet: This pluglet retrieves OVO events meeting user-specified criteria, places the event data into an XML-formatted response message, and passes the response to its target pluglet.

OVI also supplies two pluglets, OvoWEventPluglet and OvoWEventChangePluglet, that receive events from the OVO/W WMI provider (on Windows operating systems). When one of these pluglets receives an OVO/W event, it places the event data into an XML-formatted message, which it passes to its target pluglet. Change events handled by the OvoWEventChangePluglet pluglet include own/disown, acknowledge, change severity, modify message text and add/delete annotation text.

OVI's OvoEventModifyPluglet and OvoWEventModifyPluglet provide the ability to modify messages in OVO and OVO/W, respectively. They allow messages to be acknowledged, unacknowledged, owned or disowned. In addition, the message severity may be changed, message text may be modified or an annotation may be added.

Further, the OvoInventoryPluglet allows nodes to be created or deleted in OVO. It receives an ovInventoryRequest message and returns the result of the requested operation in an ovInventoryResponse message. For created nodes, OvoInventoryPluglet sets the name, label, node group and machine configuration type attributes. The pluglet can set both traditional RPC-based machine configuration types or new HTTPS-based types. Machine configuration type determines which agent software bundle that OVO will distribute to the node.

OVI's OvoEventCreatePluglet provides the ability to create an OVO/U or OVO/W message and returns the OVO/U or OVO/W message ID of the newly created message.

The OvoEventActionPluglet and OvoWEventActionPluglet allow automatic or operator-initiated actions that are defined for OVO [W] messages to be executed on HP Operations Manager for Unix or HP Operations Manager for Windows, respectively. These pluglets also return the status of these actions when queried.

To make this functionality work in your environment, you may need to make simple configuration changes to the example .config files provided for these pluglets. Elements that are good candidates for modification are marked with a comment that reads: **You may want to modify this.**

The OVI instance that runs the OvoEventPluglet, OvoEventChangePluglet or OvoEventEnrichmentPluglet must be run as user root to access the OVO MSI/MEI (UNIX operating systems).

OvoInventoryPluglet Configuration

Example OvoInventoryPluglet .deploy, .config and .in files are provided in the OVI_BASE_DIR/examples/OVI/configs/OVO/Inventory directory. If you need to make changes, copy them to the OVI_DATA_DIR/conf/OVI directory before editing.

The `OvoInventoryPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <ovoInventoryPluglet
    xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/OvoInventoryPluglet.xsd">

    <!--
        This is the OVO operator username.
        You may want to modify this.
    -->
    <ovoOperator>
        operatorName
    </ovoOperator>

    <!--
        This is the OVO encrypted password for the operator.
        Use the OviEncryptUtil script to encrypt the password
        string.
        You may want to modify this.
    -->
   <ovoOperatorEncryptedPassword>
        operatorEncryptPassword
   </ovoOperatorEncryptedPassword>

   <!--
        This is the OVO unencrypted password for the operator.
        If you want to use an unencrypted password, fill in
        and uncomment the following XML element AND comment
        out the ovoOperatorEncryptedPassword element above.
        You may want to modify this.
    <ovoOperatorPassword></ovoOperatorPassword>
    -->

    <!--
        Switch message trace logging on/off
        You may want to modify this.
    -->
    <traceMessages>true</traceMessages>

    <!--
        Switch message route logging on/off
        You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoInventoryPluglet>
```

This table describes elements in the `OvoInventoryPlugletExample.config` file:

Element Name	Value

`<ovoOperator>`
operator
`</ovoOperator>`
	Name of the OVO operator that will be used to connect to OVO.

`<ovoOperatorEncryptedPassword>`
encryptedPassword
`</ovoOperatorEncryptedPassword>`
 	
or
 	
`<ovoOperatorPassword>`
password
`</ovoOperatorPassword>`
	OVO operator password, in encrypted or unencrypted form, that will be used to connect to OVO.

`<traceMessages>`*boolean*`</traceMessages>`	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are: `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)
`<traceRoute>boolean</traceRoute>`	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

OvoEventActionPluglet Configuration

Example `OvoEventActionPluglet .deploy`, `.config` and **`.in`** files are provided in the `OVI_BASE_DIR/examples/OVI/configs/OVO/EventAction` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory before editing.

The `OvoEventActionPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <OvoEventActionPluglet
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/OvoEventActionPluglet.xsd">
   <!--
        The OVO operator and password are optional. If they are not
        specified here, they must be specified in the
        ovMessageActionRequest that is sent to OvoEventActionPluglet.
    -->
   <!--
        This is the OVO operator username.
        You may want to modify this.
    -->
    <ovoOperator>
        operatorName
    </ovoOperator>

    <!--
        This is the OVO encrypted password for the operator.
        Use the OviEncryptUtil script to encrypt the
        password string.
        You may want to modify this.
    -->
    <ovoOperatorEncryptedPassword>
        operatorEncryptPassword
    </ovoOperatorEncryptedPassword>

    <!--
        This is the OVO unencrypted password for the operator.
        If you wish to use an unencrypted password, fill in
        and uncomment the following XML element AND comment
        out the ovoOperatorEncryptedPassword element above.
        You may want to modify this.
    <ovoOperatorPassword></ovoOperatorPassword>
    -->

    <!--
         Switch message trace logging on/off
         You may want to modify this.
    -->
    <traceMessages>true</traceMessages>

    <!--
         Switch message route logging on/off
         You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</OvoEventActionPluglet>
```

This table describes elements in the `OvoEventActionPlugletExample.config` file:

Element Name	Value

`<ovoOperator>`
`  operator`
`</ovoOperator>`
	Name of the OVO operator that will be used to connect to OVO. If it is not `opc_adm`, you may see warnings that some messages cannot be read.

`<ovoOperatorEncryptedPassword>`
`  encryptedPassword`
`</ovoOperatorEncryptedPassword>`
 	
`or`
 	
`<ovoOperatorPassword>`
`  password`
`</ovoOperatorPassword>`
	OVO operator password, in encrypted or unencrypted form, that will be used to connect to OVO.

`<traceMessages>boolean</traceMessages>`	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are: `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)
`<traceRoute>boolean</traceRoute>`	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

OvoEventChangePluglet Configuration

Example `OvoEventChangePluglet .deploy`, `.config` and `.in` files are provided in the `OVI_BASE_DIR/examples/OVI/configs/OVO/EventChange` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory before editing.

The `OvoEventChangePlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <ovoEventChangePluglet
    xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/OvoEventChangePluglet.xsd">

    <!--
        Specify the Pluglet that the OvoEventChangePluglet
        should acquire
    -->
    <targetPluglet>
        PublisherPluglet
    </targetPluglet>

    <!--
        This is the OVO operator username.
        You may want to modify this.
    -->
    <ovoOperator>
        operatorName
    </ovoOperator>

    <!--
        This is the OVO encrypted password for the operator.
        Use the OviEncryptUtil script to encrypt the password
        string.
        You may want to modify this.
    -->
    <ovoOperatorEncryptedPassword>
        operatorEncryptPassword
    </ovoOperatorEncryptedPassword>

    <!--
        This is the OVO unencrypted password for the operator.
        If you wish to use an unencrypted password, fill in
        and uncomment the following XML element AND comment
        out the ovoOperatorEncryptedPassword element above.
        You may want to modify this.
    <ovoOperatorPassword></ovoOperatorPassword>
    -->

    <!--
        This is the MEI instance name that will be used to
        make the connection to OVO server MEI. The name
        must consist of 1 to 13 alphanumeric characters.
        You may want to modify this.
    -->
    <meiInstanceName>OviInstance2</meiInstanceName>

    <!--
        This filter will accept all OVO messages on the OVO
        Server MEI.
        You may want to modify this.
    -->
    <filter>
        <all/>
    </filter>

    <!--
        Switch message trace logging on/off
        You may want to modify this.
    -->
```

```
    <traceMessages>true</traceMessages>

    <!--
        Switch message route logging on/off
        You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoEventChangePluglet>
```

This table describes elements in the `OvoEventChangePlugletExample.config` file:

Element Name	Value

`<targetPluglet>`
targetPlugletName
`</targetPluglet>`
	Name of this pluglet's target pluglet. This is the pluglet to which messages will be passed.

`<ovoOperator>`
operator
`</ovoOperator>`
	Name of the OVO operator that will be used to connect to OVO.

`<ovoOperatorEncryptedPassword>`
encryptedPassword
`</ovoOperatorEncryptedPassword>`
 	
or
 	
`<ovoOperatorPassword>`
password
`</ovoOperatorPassword>`
	OVO operator password, in encrypted or unencrypted form, that will be used to connect to OVO.

`<meiInstanceName>`
meiInstanceName
`</meiInstanceName>`
	Specifies the MEI instance name that is used to make the connection to OVO server MEI. It must consist of 1 to 13 alphanumeric characters.

`<filter>`*filterDefinition*`</filter>`	See OVO MSI/MEI and OVO/W WMI Event Filters for a list of supported filter types.
`<traceMessages>`*boolean*`</traceMessages>`	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are: `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)
`<traceRoute>`*boolean*`</traceRoute>`	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

OvoEventModifyPluglet Configuration

Example `OvoEventModifyPluglet .deploy`, `.config` and `.in` files are provided in the `OVI_BASE_DIR/examples/OVI/configs/OVO/EventModify` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory before editing.

The `OvoEventModifyPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <ovoEventModifyPluglet
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/OvoEventModifyPluglet.xsd">
   <!--
        The OVO operator and password are now optional. If they are
        not specified here, they must be specified in the
        ovMessageModifyRequest that is sent to OvoEventModifyPluglet.
    -->
   <!--
        This is the OVO operator username.
        You may want to modify this.
    -->
    <ovoOperator>
        operatorName
    </ovoOperator>

    <!--
        This is the OVO encrypted password for the operator.
        Use the OviEncryptUtil script to encrypt the
        password string.
        You may want to modify this.
    -->
    <ovoOperatorEncryptedPassword>
        operatorEncryptPassword
    </ovoOperatorEncryptedPassword>

    <!--
        This is the OVO unencrypted password for the operator.
        If you wish to use an unencrypted password, fill in
        and uncomment the following XML element AND comment
        out the ovoOperatorEncryptedPassword element above.
        You may want to modify this.
    <ovoOperatorPassword></ovoOperatorPassword>
    -->

    <!--
         Switch message trace logging on/off
         You may want to modify this.
    -->
    <traceMessages>true</traceMessages>

    <!--
         Switch message route logging on/off
         You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoEventModifyPluglet>
```

This table describes elements in the `OvoEventModifyPlugletExample.config` file:

Element Name	Value

`<ovoOperator>`
`  operator`
`</ovoOperator>`
	Name of the OVO operator that will be used to connect to OVO. If it is not `opc_adm`, you may see warnings that some messages cannot be read.

`<ovoOperatorEncryptedPassword>`
`  encryptedPassword`
`</ovoOperatorEncryptedPassword>`
 	
`or`
 	
`<ovoOperatorPassword>`
`  password`
`</ovoOperatorPassword>`
	OVO operator password, in encrypted or unencrypted form, that will be used to connect to OVO.

`<traceMessages>boolean</traceMessages>`	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are: `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)
`<traceRoute>boolean</traceRoute>`	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

OvoEventPluglet Configuration

Example `OvoEventPluglet .deploy`, `.config` and `.in` files are provided in the `OVI_BASE_DIR/examples/OVI/configs/OVO/PublishEvent` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory before editing.

The `OvoEventPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <ovoEventPluglet xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/OvoEventPluglet.xsd">

    <!--
         Specify the Pluglet to which the OvoEventPluglet
         should send messages.
    -->
    <targetPluglet>
         PublisherPluglet
    </targetPluglet>

    <!--
         Specify the name of the OVI MSI instance to use. The
         name must consist of 1 to 13 alphanumeric characters.
         You may want to modify this.
    -->
    <msiInstanceName>
         Instance1
    </msiInstanceName>

   <!--
         This filter will accept all OVO messages on the OVO
         Server MSI.
         You may want to modify this.
    -->
    <filter>
         <all/>
    </filter>

    <!--
         Switch message trace logging on/off
         You may want to modify this.
    -->
    <traceMessages>true</traceMessages>
   <!--
         Switch message route logging on/off
         You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoEventPluglet>
```

This table describes elements in the `OvoEventPlugletExample.config` file:

Element Name	Value

`<targetPluglet>`
targetPlugletName
`</targetPluglet>`
	Name of this pluglet's target pluglet. This is the pluglet to which messages will be passed.

`<msiInstanceName>`
msiInstanceName
`</msiInstanceName>`
	Specifies the MSI instance name that is used to make the connection to OVO server MSI. It consists of 1 to 13 alphanumeric characters.

Element Name	Value
<filter>*filterDefinition*</filter>	See OVO MSI/MEI and OVO/W WMI Event Filters for a list of supported filter types.
<traceMessages>*boolean*</traceMessages>	Controls whether an INFO level log message is created when a message arrives at this pluglet. Valid values are: true or false. See the traceMessages Settings section for more details. (Default: true)
<traceRoute>*boolean*</traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: true or false. See the traceRoute Settings section for more details. (Default: true)

OvoEventReadPluglet Configuration

Example OvoEventReadPluglet .deploy, .config and .in files are provided in the OVI_BASE_DIR/examples/OVI/configs/OVO/EventRead directory. If you need to make changes, copy them to the OVI_DATA_DIR/conf/OVI directory before editing.

The OvoEventReadPlugletExample.config file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <ovoEventReadPluglet xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/OvoEventReadPluglet.xsd">

    <!--
         OVO operator username.
         You may want to modify this.
    -->
    <ovoOperator>opc_adm</ovoOperator>

    <!--
         OVO encrypted password for the operator. Use the
         OviEncryptUtil script to encrypt the password string.
         You will want to modify this.
    -->
    <ovoOperatorEncryptedPassword>xyz123abc
    </ovoOperatorEncryptedPassword>

    <!--
         OVO unencrypted password for the operator. If you wish
         to use an unencrypted password, fill in and uncomment
         the following XML element. Also, the ovoOperatorEncryptedPassword
         element above must be commented out.
         You may want to modify this.
   <ovoOperatorPassword></ovoOperatorPassword>
    -->

    <!--
         The maximum number of messages to return in a response.
         The default value is 150; it should suffice for most
         cases. The maximum heap size of the JVM may need to be
         increased to accommodate the specified number of
         maxMessagesToReturn. It can be set in the JVM_ARGS
         variable in the OVI.env file.
```

```
          You may want to modify this.
     -->
    <maxMessagesToReturn>150</maxMessagesToReturn>

    <!--
          This filter will accept all OVO messages on the OVO
          Server MSI.
          You may want to modify this.
     -->
     <filter>
          <all/>
     </filter>

     <!--
          Switch message trace logging on/off
          You may want to modify this.
     -->
     <traceMessages>true</traceMessages>

     <!--
          Switch message route logging on/off
          You may want to modify this.
     -->
     <traceRoute>true</traceRoute>

</ovoEventReadPluglet>
```

This table describes elements in the `OvoEventReadPlugletExample.config` file:

Element Name	Value

`<ovoOperator>`
opc_adm
`</ovoOperator>`
	Name of the OVO operator that will be used to connect to OVO.

`<ovoOperatorEncryptedPassword>`
encryptedPassword
`</ovoOperatorEncryptedPassword>`
 	
or
 	
`<ovoOperatorPassword>`
password
`</ovoOperatorPassword>`
	OVO operator password, in encrypted or unencrypted form, that will be used to connect to OVO.

`<maxMessagesToReturn>`
150
`</maxMessagesToReturn>`
	Optional element containing the maximum number of messages to return in an `ovMessageReadResponse`. The heap size of the JVM may need to be increased to accommodate larger `maxMessagesToReturn` values. (Default: `150`)

Element Name	Value
<filter>*filterDefinition*</filter>	See OVO MSI/MEI and OVO/W WMI Event Filters for a list of supported filter types.
<traceMessages>*boolean*</traceMessages>	Controls whether an INFO level log message is created when a message arrives at this pluglet. Valid values are: true or false. See the traceMessages Settings section for more details. (Default: true)
<traceRoute>*boolean*</traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: true or false. See the traceRoute Settings section for more details. (Default: true)

OvoEventEnrichmentPluglet Configuration

Example OvoEventEnrichmentPluglet .deploy, .config, and .in files are provided in the OVI_BASE_DIR/examples/OVI/configs/OVO/EventEnrichment directory. If you need to make changes, copy them to the OVI_DATA_DIR/conf/OVI directory tree before editing.

The TestResponderPlugletExample.in file contains the default value for eventSource of system.xyz.com. You need to edit this file to change the value of eventSource to the name of your managed system.

The OvoEventEnrichmentPlugletExample.config file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <ovoEventEnrichmentPluglet xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/OvoEventEnrichmentPluglet.xsd">

    <!--
        Specify the Pluglet that the ovoEventEnrichmentPluglet
        should acquire and the timeout used for receiving a
        response to its request.
        You may want to modify the timeout value.
    -->
    <targetPluglet>RequesterProxyPluglet</targetPluglet>

    <targetPlugletReqTimeOut>30000</targetPlugletReqTimeOut>

    <!--
        This is the MSI instance name that will be used to make
        the connection to OVO server MSI. The name must consist
        of 1 to 13 alphanumeric characters.
        You may want to modify this.
    -->
    <msiInstanceName>OviInstance2</msiInstanceName>

    <!--
        Specify the enrichmentThreshold value in milliseconds.
        If the difference between the OVO receive time and the
        current time is greater than this value, then the OVO
```

```
        message is not enriched but simply written back to the MSI.
        You may want to modify this.
    -->
    <enrichmentThreshold>60000</enrichmentThreshold>

    <!--
        This filter will accept all OVO messages on the OVO Server MSI.
        You may want to modify this.
    -->
    <filter>
        <all/>
    </filter>

    <!--
        Switch message trace logging on/off.
        You may want to modify this.
    -->
    <traceMessages>true</traceMessages>

    <!--
        Switch message route logging on/off.
        You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoEventEnrichmentPluglet>
```

The following table describes elements in the `OvoEventEnrichmentPlugletExample.config` file:

Element Name	Value
`<targetPluglet>` *targetPlugletName* `</targetPluglet>`	Name of this pluglet's target pluglet. This is the pluglet to which messages are passed.
`<targetPlugletReqTimeOut>` *milliseconds* `</targetPlugletReqTimeOut>`	Timeout value, in milliseconds, used for receiving a response to the pluglet's request. (Default: `30000`)
`<msiInstanceName>` *msiInstanceName* `</msiInstanceName>`	Specifies the MSI instance name that is used to make the connection to OVO server MSI. It must consist of 1 to 13 alphanumeric characters.
`<enrichmentThreshold>` *milliseconds* `</enrichmentThreshold>`	Specifies the `enrichmentThreshold` value in milliseconds. If the difference between the OVO receive time and the current time is greater than this value, then the OVO message is not enriched but simply written back to the MSI. This is used to prevent the `OvoEventEnrichmentPluglet` from delaying OVO messages too much. (Default: `60000`)

Element Name	Value
<filter> *filterDefinition* </filter>	See OVO MSI/MEI and OVO/W WMI Event Filters definitions for a list of the supported filter types.
<traceMessages> *boolean* </traceMessages>	Controls whether an INFO level log message is created when a message arrives at this pluglet. Valid values are: true or false. See the traceMessages Settings section for more details. (Default: true)
<traceRoute> *boolean* </traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are true or false. See the traceRoute Settings section for more details. (Default: true)

OvoEventCreatePluglet Configuration

Example OvoEventCreationPluglet.deploy and .config files are provided in the OVI_BASE_DIR/examples/OVI/configs/OVO/EventCreation directory. If you need to make changes, copy them to the OVI_DATA_DIR/conf/OVI directory before editing.

The OvoEventCreationPlugletExample.config file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <ovoEventCreatePluglet
    xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/OvoEventCreatePluglet.xsd">

    <!--
         Switch message trace logging on/off
         You may want to modify this.
    -->
    <traceMessages>true</traceMessages>

    <!--
         Switch message route logging on/off
         You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoEventCreatePluglet>
```

This table describes elements in the `OvoEventCreationPlugletExample.config` file:

Element Name	Value
`<traceMessages>`*boolean*`</traceMessages>`	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are: `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)
`<traceRoute>`*boolean*`</traceRoute>`	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

OvoWEventActionPluglet Configuration

Example `OvoWEventActionPluglet .deploy`, `.config` and `.in` files are provided in the `OVI_BASE_DIR/examples/OVI/configs/OVO/EventAction` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory before editing.

The `OvoWEventActionPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <OvoWEventActionPluglet
    xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation= "C:\Program Files\HP OpenView\schema\
    OVI\OvoWEventActionPluglet.xsd">

   <!--
       Specify the name of the machine used to connect to WMI.
       For a local connection, localhost or the name of the
       local system may be entered and the wmiUserName and
       wmiUserPassword (or wmiUserEncryptedPassword)
       must be empty. If a value is supplied for either of
       them for a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
   <wmiMachineName>machineName</wmiMachineName>

   <!--
        The WMI username and password are now optional. If they are
        not specified here, they must be specified in the
        ovMessageActionRequest that is sent to OvoWEventActionPluglet.
-->

   <!--
       Specify the name of the user used to connect to WMI.
       For a local connection, this should be left blank to
       indicate that the username is that of the currently
       logged on user. If a value is supplied for wmiUserName
       in the case of a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
    <wmiUserName>userName</wmiUserName>

    <!--
```

```
      Specify the unencrypted user password <wmiUserPassword>
      or the encrypted password <wmiUserEncryptedPassword>
      used to connect to WMI. For a local connection, this
      should be left blank. If a value is supplied for either
      password type in the case of a local connection, a
      WBEM_E_LOCAL_CREDENTIALS error will result.
      You may want to modify this.
   -->
   <wmiUserPassword>     OR    <wmiUserEncryptedPassword>
 userPassword                 userEncryptedPassword
   </wmiUserPassword>           </wmiUserEncryptedPassword>

   <!--
        Switch message trace logging on/off
        You may want to modify this.
   -->
   <traceMessages>true</traceMessages>

   <!--
        Switch message route logging on/off
        You may want to modify this.
   -->
   <traceRoute>true</traceRoute>

</OvoWEventActionPluglet>
```

This table describes elements in the `OvoWEventActionPlugletExample.config` file:

Element Name	Value

`<wmiMachineName>`
machineName
`</wmiMachineName>`
	Specify the name of the machine used to connect to WMI. For local connections, leave this blank. Optionally, `localhost` or the name of the local system can be entered. In the case of a local connection, the `wmiUserName` and `wmiUserPassword` (or `wmiUserEncryptedPassword`) should be empty. If a value is supplied for either of them, a `WBEM_E_LOCAL_CREDENTIALS` error occurs.

`<wmiUserName>`
userName
`</wmiUserName>`
	Specify the name of the user used to connect to WMI. It should be empty when the connection to WMI is local, indicating the currently logged on user. If a value is supplied for `wmiUserName` in the case of a local connection, a `WBEM_E_LOCAL_CREDENTIALS` error occurs.

Element Name	Value
<wmiUserPassword> *userPassword* </wmiUserPassword> or <wmiUserEncryptedPassword> *userEncryptedPassword* </wmiUserEncryptedPassword>	Specify the unencrypted user password <wmiUserPassword> or the encrypted password <wmiUserEncryptedPassword> used to connect to WMI. Should be empty when the connection to WMI is local. If a value is supplied for either password type in the case of a local connection, a WBEM_E_LOCAL_CREDENTIALS error occurs.
<traceMessages>*boolean*</traceMessages>	Controls whether an INFO level log message is created when a message arrives at this pluglet. Valid values are: true or false. See the traceMessages Settings section for more details. (Default: true)
<traceRoute>*boolean*</traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: true or false. See the traceRoute Settings section for more details. (Default: true)

OvoWEventChangePluglet Configuration

Example OvoWEventChangePluglet .deploy and .config files are provided in the OVI_BASE_DIR/examples/OVI/configs/OVO/EventChange directory. If you need to make changes, copy them to the OVI_DATA_DIR/conf/OVI directory tree before editing.

The OvoWEventChangePlugletExample.config file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<ovoWEventChangePluglet xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation= "C:\Program Files\HP OpenView\
   schema\OVI\OvoWEventChangePluglet.xsd">

   <!--
       Specify the Pluglet(s) that the OvoWEventChangePluglet
       should acquire. More than one <targetPluglet> tag can
       be listed for specifying multiple targets.
   -->
   <targetPluglet>PublisherPluglet1</targetPluglet>
   <targetPluglet>PublisherPluglet2</targetPluglet>

   <!--
       Specify the name of the machine used to connect to WMI.
       For a local connection, this may be left blank.
       Optionally, for a local connection, localhost or the
       name of the local system can be entered, but, in any
       of these cases, the wmiUserName and wmiUserPassword
       (or wmiUserEncryptedPassword) must be empty. If a
       value is supplied for either of them, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
   <wmiMachineName>machineName</wmiMachineName>
```

```
   <!--
       Specify the name of the user used to connect to WMI.
       For a local connection, this should be left blank to
       indicate that the username is that of the currently
       logged on user. If a value is supplied for wmiUserName
       in the case of a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
   <wmiUserName>userName</wmiUserName>
    <!--
       Specify the unencrypted user password <wmiUserPassword>
       or the encrypted password <wmiUserEncryptedPassword>
       used to connect to WMI. For a local connection, this
       must be left blank. If a value is supplied for either
       password type in the case of a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
    <wmiUserPassword>    OR  <wmiUserEncryptedPassword>
     userPassword              userEncryptedPassword
    </wmiUserPassword>       </wmiUserEncryptedPassword>

    <!--
        This filter will accept all OVO/W message change events
        for the WMI provider.
        You may want to modify this.
    -->
    <filter>
       <all/>
    </filter>

    <!--
        Switch message trace logging on/off
        You may want to modify this.
    -->
    <traceMessages>true</traceMessages>

    <!--
        Switch message route logging on/off
        You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoWEventChangePluglet>
```

The following table describes elements in the `OvoWEventChangePlugletExample.config` file:

Element Name	Value

`<targetPluglet>`
`  targetPlugletName`
`</targetPluglet>`
	Name of this pluglet's target pluglet. This is the pluglet to which messages are passed. More than one `<targetPluglet>` tag can be listed for specifying multiple targets.

`<wmiMachineName>`
`  machineName`
`</wmiMachineName>`
	Specify the name of the machine used to connect to WMI. For local connections, leave this blank. Optionally, `localhost` or the name of the local system can be entered here, but, in any of these cases of a local connection, the `wmiUserName` and `wmiUserPassword` (or `wmiUserEncryptedPassword`) should be empty. If a value is supplied for either of them, a `WBEM_E_LOCAL_CREDENTIALS` error occurs.

`<wmiUserName>`
`  userName`
`</wmiUserName>`
	Specify the name of the user used to connect to WMI. It must be empty when the connection to WMI is local, indicating the currently logged on user. If a value is supplied for `wmiUserName` in the case of a local connection, a `WBEM_E_LOCAL_CREDENTIALS` error occurs.

`<wmiUserPassword>`
`  userPassword`
`</wmiUserPassword>`
 	
`or`
 	
`<wmiUserEncryptedPassword>`
`  userEncryptedPassword`
`</wmiUserEncryptedPassword>`
	Specify the unencrypted user password `<wmiUserPassword>` or the encrypted password `<wmiUserEncryptedPassword>` used to connect to WMI. It must be empty when the connection to WMI is local. If a value is supplied for either password type in the case of a local connection, a `WBEM_E_LOCAL_CREDENTIALS` error occurs.

`<filter>`
`  filterDefinition`
`</filter>`
	See OVO MSI/MEI and OVO/W WMI Event Filters definitions for a list of the supported filter types.

`<traceMessages>`
`  boolean`
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are: `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
`  boolean`
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

OvoWEventModifyPluglet Configuration

Example `OvoWEventModifyPluglet .deploy`, `.config` and `.in` files are provided in the `OVI_BASE_DIR/examples/OVI/configs/OVO/EventModify` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory before editing.

The `OvoWEventModifyPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

 <ovoWEventModifyPluglet
    xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation= "C:\Program Files\HP OpenView\
    schema\OVI\OvoWEventModifyPluglet.xsd">
    <!--
       Specify the name of the machine used to connect to WMI.
       For a local connection, localhost or the name of the
       local system may be entered and the wmiUserName and
       wmiUserPassword (or wmiUserEncryptedPassword)
       must be empty. If a value is supplied for either of
       them for a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
  <wmiMachineName>machineName</wmiMachineName>

   <!--
        The WMI username and password are now optional. If they
        are not specified here, they must be specified in the
        ovMessageModifyRequest that is sent to
        OvoWEventModifyPluglet.
    -->

   <!--
       Specify the name of the user used to connect to WMI.
       For a local connection, this should be left blank to
       indicate that the username is that of the currently
       logged on user. If a value is supplied for wmiUserName
       in the case of a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
    <wmiUserName>userName</wmiUserName>

    <!--
       Specify the unencrypted user password <wmiUserPassword>
       or the encrypted password <wmiUserEncryptedPassword>
       used to connect to WMI. For a local connection, this
       should be left blank. If a value is supplied for either
       password type in the case of a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
    <wmiUserPassword>     OR    <wmiUserEncryptedPassword>
        userPassword                userEncryptedPassword
    </wmiUserPassword>          </wmiUserEncryptedPassword>

    <!--
         Switch message trace logging on/off
         You may want to modify this.
    -->
    <traceMessages>true</traceMessages>

    <!--
         Switch message route logging on/off
         You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoWEventModifyPluglet>
```

This table describes elements in the OvoWEventModifyPlugletExample.config file:

Element Name	Value
<wmiMachineName> *machineName* </wmiMachineName>	Specify the name of the machine used to connect to WMI. For local connections, leave this blank. Optionally, localhost or the name of the local system can be entered. In the case of a local connection, the wmiUserName and wmiUserPassword (or wmiUserEncryptedPassword) should be empty. If a value is supplied for either of them, a WBEM_E_LOCAL_CREDENTIALS error occurs.
<wmiUserName> *userName* </wmiUserName>	Specify the name of the user used to connect to WMI. It should be empty when the connection to WMI is local, indicating the currently logged on user. If a value is supplied for wmiUserName in the case of a local connection, a WBEM_E_LOCAL_CREDENTIALS error occurs.
<wmiUserPassword> *userPassword* </wmiUserPassword> or <wmiUserEncryptedPassword> *userEncryptedPassword* </wmiUserEncryptedPassword>	Specify the unencrypted user password <wmiUserPassword> or the encrypted password <wmiUserEncryptedPassword> used to connect to WMI. Should be empty when the connection to WMI is local. If a value is supplied for either password type in the case of a local connection, a WBEM_E_LOCAL_CREDENTIALS error occurs.
<traceMessages>*boolean*</traceMessages>	Controls whether an INFO level log message is created when a message arrives at this pluglet. Valid values are: true or false. See the traceMessages Settings section for more details. (Default: true)
<traceRoute>*boolean*</traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are: true or false. See the traceRoute Settings section for more details. (Default: true)

OvoWEventPluglet Configuration

Example OvoWEventPluglet .deploy and .config files are provided in the OVI_BASE_DIR/examples/OVI/configs/OVO/PublishEvent directory. If you need to make changes, copy them to the OVI_DATA_DIR/conf/OVI directory tree before editing.

The `OvoWEventPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<ovoWEventPluglet
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="C:\Program Files\HP OpenView\
   schema\OVI\OvoWEventPluglet.xsd">
   <!--
       Specify the Pluglet(s) that the OvoWEventPluglet
       should acquire. More than one <targetPluglet> tag
       can be listed for specifying multiple targets.
   -->
   <targetPluglet>PublisherPluglet1</targetPluglet>
   <targetPluglet>PublisherPluglet2</targetPluglet>

   <!--
       Specify the name of the machine used to connect to
       WMI. For a local connection, this should be left
       blank. Optionally, for a local connection, localhost
       or the name of the local system can be entered, but,
       in any of these cases, the wmiUserName and
       wmiUserPassword (or wmiUserEncryptedPassword) should
       be empty. If a value is supplied for either of them,
       a WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
   <wmiMachineName>machineName</wmiMachineName>

   <!--
       Specify the name of the user used to connect to WMI.
       For a local connection, this must be left blank to
       indicate that the username is that of the currently
       logged on user. If a value is supplied for wmiUserName
       in the case of a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
    <wmiUserName>userName</wmiUserName>

    <!--
       Specify the unencrypted user password <wmiUserPassword>
       or the encrypted password <wmiUserEncryptedPassword>
       used to connect to WMI. For a local connection, this must
       be left blank. If a value is supplied for either password
       type in the case of a local connection, a
       WBEM_E_LOCAL_CREDENTIALS error will result.
       You may want to modify this.
    -->
    <wmiUserPassword>      OR      <wmiUserEncryptedPassword>
     userPassword                   userEncryptedPassword
    </wmiUserPassword>             </wmiUserEncryptedPassword>

    <!--
       This filter will accept all OVO/W messages for the
       WMI provider.
       You may want to modify this.
    -->
    <filter>
       <all/>
    </filter>

    <!--
       Switch message trace logging on/off
       You may want to modify this.
```

```
    -->
    <traceMessages>true</traceMessages>

    <!--
       Switch message route logging on/off
       You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</ovoWEventPluglet>
```

The following table describes elements in the `OvoWEventPlugletExample.config` file:

Element Name	Value

`<targetPluglet>`
targetPlugletName
`</targetPluglet>`
	Name of this pluglet's target pluglet. This is the pluglet to which messages are passed. More than one `<targetPluglet>` tag can be listed for specifying multiple targets.

`<wmiMachineName>`
machineName
`</wmiMachineName>`
	Specify the name of the machine used to connect to WMI. For local connections, leave this blank. Optionally, `localhost` or the name of the local system can be entered here, but, in any of these cases of a local connection, the `wmiUserName` and `wmiUserPassword` (or `wmiUserEncryptedPassword`) should be empty. If a value is supplied for either of them, a `WBEM_E_LOCAL_CREDENTIALS` error occurs.

`<wmiUserName>`
userName
`</wmiUserName>`
	Specify the name of the user used to connect to WMI. Should be empty when the connection to WMI is local, indicating the currently logged on user. If a value is supplied for `wmiUserName` in the case of a local connection, a `WBEM_E_LOCAL_CREDENTIALS` error occurs.

`<wmiUserPassword>`
userPassword
`</wmiUserPassword>`
 	
or
 	
`<wmiUserEncryptedPassword>`
userEncryptedPassword
`</wmiUserEncryptedPassword>`
	Specify the unencrypted user password `<wmiUserPassword>` or the encrypted password `<wmiUserEncryptedPassword>` used to connect to WMI. Should be empty when the connection to WMI is local. If a value is supplied for either password type in the case of a local connection, a `WBEM_E_LOCAL_CREDENTIALS` error occurs.

Element Name	Value

`<filter>`
`  filterDefinition`
`</filter>`
	See OVO MSI/MEI and OVO/W WMI Event Filters definitions for a list of the supported filter types.

`<traceMessages>`
`  boolean`
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are: `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
`  boolean`
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

Namespaces

The namespace used in XML messages for OVO and OVO/W is `xmlns=http://openview.hp.com/xmlns/ico/message`. This is required for `<ovMessage>`, `<ovMessageChangeEvent>`, `<ovMessageEnrichmentRequest>`, and `<ovMessageEnrichmentResponse>` messages. The same namespace is used for `<ovMessageCreateRequest>`, `<ovMessageCreateResponse>`, `<ovMessageModifyRequest>`, `<ovMessageModifyResponse>`, `<ovMessageActionRequest>`, `<ovMessageActionResponse>`, `<ovMessageReadRequest>` and `<ovMessageReadResponse>`. The namespace used in `<ovInventoryRequest>` and `<ovInventoryResponse>` messages is `xmlns=http://openview.hp.com/xmlns/ico/entity`.

Message Formats

This section contains the format of the request/response and exception messages used by each of the OVO and OVO/W pluglets. It also contains additional information about how each pluglet works:

- OvoInventoryPluglet
- OvoEventPluglet and OvoWEventPluglet
- OvoEventChangePluglet and OvoWEventChangePluglet
- OvoEventModifyPluglet and OvoWEventModifyPluglet
- OvoEventActionPluglet and OvoWEventActionPluglet
- OvoEventReadPluglet
- OvoEventEnrichmentPluglet
- OvoEventCreatePluglet

The contents of XML request messages are only partially validated. A full validation is not performed for performance reasons. Only specific parts of each request message are checked.

OvoInventoryPluglet

OVO nodes can be created or deleted by sending requests to *OvoInventoryPluglet*. The pluglet responds with either the name of the node that was created/deleted or an exception response, if the action was not completed successfully.

Description	Location
Request schema	OVI_BASE_DIR/examples/OVI/schemas/entity/ovInventoryRequest.xsd
Response schema	OVI_BASE_DIR/examples/OVI/schemas/entity/ovInventoryResponse.xsd
Request example (ovInventoryRequest)	OVI_BASE_DIR/examples/OVI/schemas/entity/ovInventoryRequest_createOvoNode.xml or OVI_BASE_DIR/examples/OVI/schemas/entity/ovInventoryRequest_deleteOvoNode.xml
Response example (ovInventoryResponse)	OVI_BASE_DIR/examples/OVI/schemas/entity/ovInventoryResponse_createOvoNode.xml or OVI_BASE_DIR/examples/OVI/schemas/entity/ovInventoryResponse_deleteOvoNode.xml
Exception response (ovInventoryException)	Example: OVI_BASE_DIR/examples/OVI/schemas/entity/ovInventoryException.xml Schema: OVI_BASE_DIR/examples/OVI/schemas/entity/ovInventoryException.xsd

Here is an example of a node create request. The `name` element must contain a DNS-resolvable node; it can be specified as a hostname or IP address. The `nodeGroup` element must already exist within OVO. `agentType`, `machType`, `osName` and `osVersion` element values must be one of the sets of values shown in the table annotation below; if they are not, the node will be added to OVO with an unknown machine type. If `agentType` is not specified, RPC will be used.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovInventoryRequest xmlns="http://openview.hp.com/xmlns/ico/entity"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <requestDetails>
      <!-- A timeStamp value must be included, but currently
         it is NOT USED in request processing -->
      <timeStamp>2003-09-30T13:30:47-05:00</timeStamp>
      <action>create</action>
         <entityType>Node</entityType>
   </requestDetails>
   <entityDetails>
      <inventoryItem>
         <ovit:data>
            <!ºTM- Node name can be specified as a hostname or
               IP address. -->
            <ovit:name>name</ovit:name>
            <ovit:value type="string">systemA.company.com</ovit:value>
         </ovit:data>
         <ovit:data>
            <!ºTM- Node label must be 32 characters or less. -->
           <ovit:name>label</ovit:name>
            <ovit:value type="string">systemA</ovit:value>
         </ovit:data>
         <ovit:data>
            <ovit:name>nodeGroup</ovit:name>
            <ovit:value type="string">net_devices</ovit:value>
         </ovit:data>
         <ovit:structuredData>
            <ovit:id>Node</ovit:id>
            <ovit:type>machineConfig</ovit:type>
            <!-- Allowable values are:
      agentType machType  osName               osVersion
      ========  ========  ======               =========
     RPC       PA-RISC   HPUX                10, 11, 11.11, 11.20, or 11.22
      HTTPS     PA-RISC   HPUX                   11, 11.11, or 11.23
      RPC       PowerPC   AIX
      HTTPS     PowerPC   AIX
      RPC       Intel     Linux                  2.2 or 2.4
      HTTPS     Intel     Linux                  2.4
      RPC       Intel     Netware
      RPC       Intel     Windows NT             4.0
      HTTPS     Intel     Windows NT             4.0
      RPC       Intel     Windows 2000           5.0
      HTTPS     Intel     Windows 2000           5.0
      RPC       Intel     Windows XP             5.1
      HTTPS     Intel     Windows XP             5.1
      RPC       Intel     Windows Server 2003    5.2
      HTTPS     Intel     Windows Server 2003    5.2
      RPC       SPARC     Solaris
      HTTPS     SPARC     Solaris
      HTTPS     Alpha     Tru64

      Where osVersions are not specified, any version is allowed.
      For combinations other than the ones specified, the node will
      be added to OVO as an Unknown machine type.

      If agentType is not specified, RPC will be used. The HTTPS agentType
      is only supported with OVO 8.0 and above.
             -->
            <ovit:data>
               <ovit:name>agentType</ovit:name>
               <ovit:value type="string">RPC</ovit:value>
```

```
            </ovit:data>
            <ovit:data>
               <ovit:name>machType</ovit:name>
               <ovit:value type="string">Intel</ovit:value>
            </ovit:data>
            <ovit:data>
               <ovit:name>osName</ovit:name>
               <ovit:value type="string">Windows 2000</ovit:value>
            </ovit:data>
            <ovit:data>
               <ovit:name>osVersion</ovit:name>
               <ovit:value type="string">5.0</ovit:value>
            </ovit:data>
         </ovit:structuredData>
      </inventoryItem>
   </entityDetails>
</ovInventoryRequest>
```

The example inventory request message above adds the specified node with the hostname `systemA.company.com` to OVO. In OVO, the node will have the label `systemA` and be part of the `net_devices` node group; it will be displayed with a Windows 2000 icon. If there are no problems executing the preceding create request, the `OvoInventoryPluglet` returns a response message similar to this:

```
<?xml version="1.0" encoding="UTF-8"?>
<ovInventoryResponse xmlns="http://openview.hp.com/xmlns/ico/entity"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">
   <entityDetails>
      <inventoryItem>
         <ovit:data>
            <ovit:name>name</ovit:name>
            <ovit:value type="string">systemA.company.com</ovit:value>
         </ovit:data>
      </inventoryItem>
   </entityDetails>
</ovInventoryResponse>
```

Here is an example of a node delete request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovInventoryRequest xmlns="http://openview.hp.com/xmlns/ico/entity"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <requestDetails>
      <!-- A timeStamp value must be included, but currently
         it is NOT USED in request processing -->
      <timeStamp>2003-09-30T13:30:47-05:00</timeStamp>
      <action>delete</action>
      <entityType>Node</entityType>
   </requestDetails>
   <entityDetails>
      <inventoryItem>
         <ovit:data>
            <!-- Node name can be specified as a hostname
                 or IP address. -->
            <ovit:name>name</ovit:name>
            <ovit:value type="string">systemA.company.com</ovit:value>
         </ovit:data>
      </inventoryItem>
```

```
   </entityDetails>
</ovInventoryRequest>
```

The example inventory request message above deletes the specified node with the hostname `systemA.company.com` from OVO. If there are no problems executing the delete request, the `OvoInventoryPluglet` returns a response message similar to this:

```
<?xml version="1.0" encoding="UTF-8"?>
<ovInventoryResponse xmlns="http://openview.hp.com/xmlns/ico/entity"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <entityDetails>
      <inventoryItem>
         <ovit:data>
            <ovit:name>name</ovit:name>
            <ovit:value type="string">systemA.company.com</ovit:value>
         </ovit:data>
      </inventoryItem>
   </entityDetails>
</ovInventoryResponse>
```

OvoEventActionPluglet and OvoWEventActionPluglet

Existing OVO/U or OVO/W automatic and operator-initiated message actions can be executed by sending requests to `OvoEventActionPluglet` or `OvoWEventActionPluglet`, respectively. Message/action relationships must be predefined in an OVO message template or an OVO/W policy. `OvoEventActionPluglet` and `OvoWEventActionPluglet` respond to action requests with either the initiation status of the request action or an exception response, if the action does not start successfully. These pluglets may also be queried for the completion status of a requested message action.

It is important to understand that when a requested action **starts** successfully (i.e., does not generate an exception), its action status will indicate success. To determine whether the action completed successfully, a QueryStatus request must be issued.

You may specify OVO[W] username and password values in action request messages. If you do this, use of a secure transport (e.g., HTTPS or JMS) is highly recommended. Alternatively, username and password may be supplied in the pluglet's `.config` file. If this information is specified in both places, request message values are used. Note that if username and password values are not specified in the `Ovo[W]EventActionPluglet.config` file, they must be supplied in request messages.

You can make these types of requests. Click a link to see an example:

- execute automatic message action
- query automatic message action status
- execute operator-initiated message action
- query operator-initiated message action status

Only one message action can be specified per request; multiple actions cannot be accomplished by a single request.

Description	Location
Request schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageActionRequest.xsd`
Response schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageActionResponse.xsd`
Exception schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageActionException.xsd`
Request example (`ovMessageActionRequest`)	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageActionRequest.xml`
Response example (`ovMessageActionResponse`)	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageActionResponse.xml`

Exception response (`ovMessageActionException`)
 	
Note: Must be of type `invalidXmlException` or `operationException`
	Examples:
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageActionInvalidXmlException.xml`
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageActionOperationException.xml`

If processing is successful, `Ovo[W]EventActionPluglet` returns a response similar to this.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageActionResponse xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <ovit:actionStatus>Succeeded</ovit:actionStatus>
</ovMessageActionResponse>
```

Execute Automatic Message Action Request

Here is an example of a request to execute an automatic message action. The request must contain the OVO[W] UUID of the message whose automatic action is to be executed. Both the message UUID and its associated automatic action must already be defined on the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageActionRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data sequenceNumber="1">
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data sequenceNumber="2">
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message whose action will be executed. -->
      You may want to modify this.
   <ovit:messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </ovit:messageUUID>
   <!-- OVO message action source.
        Valid sources are OperatorAction or AutomaticAction -->
   <ovit:actionSource>AutomaticAction</ovit:actionSource>
   <!-- The action command to perform.
        Valid action commands are: Execute or QueryStatus -->
   <ovit:actionCommand>Execute</ovit:actionCommand>
</ovMessageActionRequest>
```

Execute Operator-Initiated Message Action Request

Here is an example of a request to execute an operator-initiated message action. The request must contain the OVO[W] UUID of the message whose operator-initiated action is to be executed. Both the message UUID and its associated operator-initiated action must already be defined on the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageActionRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data sequenceNumber="1">
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data sequenceNumber="2">
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message whose action will be executed. -->
      You may want to modify this.
   <ovit:messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </ovit:messageUUID>
   <!-- OVO message action source.
        Valid sources are OperatorAction or AutomaticAction -->
   <ovit:actionSource>OperatorAction</ovit:actionSource>
   <!-- The action command to perform.
        Valid action commands are: Execute or QueryStatus -->
   <ovit:actionCommand>Execute</ovit:actionCommand>
</ovMessageActionRequest>
```

Query Automatic Message Action Status Request

Here is an example of a request to query the status of an automatic message action. The request must contain the OVO[W] UUID of the message whose automatic action is to be queried. Both the message UUID and its associated automatic action must already be defined on the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageActionRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data sequenceNumber="1">
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data sequenceNumber="2">
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message whose message action status
        will be queried. -->
      You may want to modify this.
   <ovit:messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </ovit:messageUUID>
   <!-- OVO message action source.
        Valid sources are OperatorAction or AutomaticAction -->
   <ovit:actionSource>AutomaticAction</ovit:actionSource>
   <!-- The action command to perform.
        Valid action commands are: Execute or QueryStatus -->
   <ovit:actionCommand>QueryStatus</ovit:actionCommand>
</ovMessageActionRequest>
```

Query Operator-Initiated Message Action Status Request

Here is an example of a request to query the status of an operator-initiated message action. The request must contain the OVO[W] UUID of the message whose operator-initiated action is to be queried. Both the message UUID and its associated operator-initiated action must already be defined on the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageActionRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data sequenceNumber="1">
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data sequenceNumber="2">
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message whose message action status
        will be queried. -->
      You may want to modify this.
   <ovit:messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </ovit:messageUUID>
   <!-- OVO message action source.
        Valid sources are OperatorAction or AutomaticAction -->
   <ovit:actionSource>OperatorAction</ovit:actionSource>
   <!-- The action command to perform.
        Valid action commands are: Execute or QueryStatus -->
   <ovit:actionCommand>QueryStatus</ovit:actionCommand>
</ovMessageActionRequest>
```

OvoEventChangePluglet and OvoWEventChangePluglet

When message change events are received from OVO MEI (or OVO/W WMI), `OvoEventChangePluglet` (or `OvoWEventChangePluglet`) reformats and publishes change event messages onto the selected OVI message transport.

Description	Example

Event change message (in `OVI_BASE_DIR/examples/OVI/schemas/message`)	`ovMessageChangeEvent-Acknowledge.xml`
	`ovMessageChangeEvent-AnnotationChange.xml`
	`ovMessageChangeEvent-Disown.xml`
	`ovMessageChangeEvent-Own.xml`
	`ovMessageChangeEvent-SeverityChange.xml`
	`ovMessageChangeEvent-TextChange.xml`
	`ovMessageChangeEvent-Unacknowledge.xml`
	`ovMessageChangeEvent-DuplicateChange.xml`

Event change message schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageChangeEvent.xsd`

OvoEventModifyPluglet and OvoWEventModifyPluglet

Existing OVO/U or OVO/W messages can be modified by sending requests to `OvoEventModifyPluglet` or `OvoWEventModifyPluglet`, respectively. These pluglets respond with either the message UUID of the modified message or an exception response if the modification was not completed successfully.

You may specify OVO[W] username and password values in request messages. If you do this, use of a secure transport (e.g., HTTPS or JMS) is highly recommended. Alternatively, username and password may be supplied in the pluglet's `.config` file. If this information is specified in both places, request message values are used. Note that if username and password values are not specified in the `Ovo[W]EventModifyPluglet.config` file, they must be supplied in request messages.

You can make these types of message modifications. Click a link to see an example:

- acknowledge
- unacknowledge
- own
- disown
- modify severity
- modify message text
- new annotation text

Only one modification action can be specified per request; multiple actions cannot be accomplished by a single request.

Description	Location
Request schema	OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageModifyRequest.xsd
Response schema	OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageModifyResponse.xsd
Exception schema	OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageModifyException.xsd
Request example (ovMessageModifyRequest)	OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageModifyRequest.xml
Response example (ovMessageModifyResponse)	OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageModifyResponse.xml
Exception response (ovMessageModifyException) **Note**: Must be of type invalidXmlException, operationException or OviUnsupported OperationException	Examples: OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageModifyInvalidXmlException.xml OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageModifyOperationException.xml OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageModifyOviUnsupported OperationException.xml

If processing is successful, the example modify request messages modify a message in OVO/U or OVO/W and the OvoEventModifyPluglet returns a response similar to this. The messageUUID element contains the ID of the modified message.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageModifyResponse xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <messageUUID>75ba8cf4-9a9b-71d6-1350-0f0278270000
   </messageUUID>
</ovMessageModifyResponse>
```

Message Acknowledge Request

Here is an example of a message acknowledge request. The request must contain an OVO/U or OVO/W message UUID that exists in the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageModifyRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data>
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message to modify -->
      You may want to modify this.
   <messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </messageUUID>
   <!-- The modify action to perform on the OVO message.
      Only one action can be specified per request! -->
   <action>
      <modifyState>Acknowledge</modifyState>
   </action>
</ovMessageModifyRequest >
```

Message Unacknowledge Request

Here is an example of a message unacknowledge request. The request must contain an OVO/U or OVO/W message UUID that exists in the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageModifyRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data>
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message to modify -->
      You may want to modify this.
   <messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </messageUUID>
   <!-- The modify action to perform on the OVO message.
      Only one action can be specified per request! -->
   <action>
      <modifyState>Unacknowledge</modifyState>
   </action>
</ovMessageModifyRequest>
```

Message Own Request

Here is an example of a message own request. The request must contain an OVO/U or OVO/W message UUID that exists in the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageModifyRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data>
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message to modify -->
      You may want to modify this.
   <messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </messageUUID>
   <!-- The modify action to perform on the OVO message.
      Only one action can be specified per request! -->
   <action>
      <modifyState>Own</modifyState>
   </action>
</ovMessageModifyRequest >
```

Message Disown Request

Here is an example of a message disown request. The request must contain an OVO/U or OVO/W message UUID that exists in the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageModifyRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data>
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message to modify -->
      You may want to modify this.
   <messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </messageUUID>
   <!-- The modify action to perform on the OVO message.
      Only one action can be specified per request! -->
   <action>
      <modifyState>Disown</modifyState>
   </action>
</ovMessageModifyRequest>
```

Modify Message Severity Request

Here is an example of a request to modify message severity. The request must contain an OVO/U or OVO/W message UUID that exists in the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageModifyRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data>
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message to modify -->
      You may want to modify this.
   <messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </messageUUID>
   <!-- The modify action to perform on the OVO message.
      Only one action can be specified per request! -->
   <action>
      <!-- Change the severity of the OVO message. Valid severities are:
         Critical, Major, Minor, Warning and Normal -->
      <severity>Critical</severity>
   </action>
</ovMessageModifyRequest>
```

Modify Message Text Request

Here is an example of a request to modify message text. The request must contain an OVO/U or OVO/W message UUID that exists in the OVO/U or OVO/W server to which the request is sent. Note that when the message text is modified, OVO/U or OVO/W also changes message ownership to the user who requested that modification.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageModifyRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data>
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message to modify -->
      You may want to modify this.
   <messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </messageUUID>
   <!-- The modify action to perform on the OVO message.
           Only one action can be specified per request! -->
   <action>
      <!-- Change the OVO message text -->
      <data>
         <name>text</name>
         <value type="string">New message text</value>
      </data>
   </action>
</ovMessageModifyRequest>
```

Add Annotation Text Request

Here is an example of a request to add annotation text to a message. The request must contain an OVO/U or OVO/W message UUID that exists in the OVO/U or OVO/W server to which the request is sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageModifyRequest  xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!-- The OVO[W] user and password can optionally be
      specified within this request. If these credentials
      are specified here, they will override any user and
      password credentials specified in the pluglet
      configuration file (for this request only).
      When specifying the OVO[W]user and password in
      this request, be sure to use a secure transport
      such as HTTPS to ensure that this sensitive information
      is as secure as possible.
   <ovit:auth>
      <ovit:data>
         <ovit:name>user</ovit:name>
         <ovit:value type="string">user</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>password</ovit:name>
         <ovit:value type="string">password</ovit:value>
      </ovit:data>
   </ovit:auth>
   -->

   <!-- The UUID of the OVO message to modify -->
      You may want to modify this.
   <messageUUID>
      75ba8cf4-9a9b-71d6-1350-0f0278270000
   </messageUUID>
   <!-- The modify action to perform on the OVO message.
      Only one action can be specified per request! -->
   <action>
      <!-- Add annotation text to the OVO message -->
      <data>
         <name>annotation</name>
         <value type="string">New annotation text</value>
      </data>
   </action>
</ovMessageModifyRequest>
```

OvoEventPluglet and OvoWEventPluglet

When event notification messages are received from OVO MSI (or OVO/W WMI), `OvoEventPluglet` or `OvoWEventPluglet` reformats and publishes the messages onto the selected OVI message transport.

Description	Example
Event message	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessage.xml`
Event message schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessage.xsd`

OvoEventReadPluglet

OVO events can be retrieved by sending a message read request to `OvoEventReadPluglet`. The request can contain either the UUID of a specific event or a time range for which events are to be returned. (Time range refers to the period during which messages were received on the OVO management server rather than their creation times.) The pluglet responds with either the desired event(s) or an exception, if a problem occurs during event retrieval. If no events match the criteria specified in the request, a response that does not contain any events is returned.

The response contains an `allRequestedMessagesReturned` flag, which indicates whether all available event data meeting the specified criteria has been returned. If the flag is set to `false`, the number of available events exceeded the maximum number of messages that `OvoEventReadPluglet` is permitted to return. (The maximum messages setting can be adjusted in the `OvoEventReadPluglet.config` file.) When this happens, you can receive the remaining available events by noting the time of the last message returned and then making another request with this time as the request's start time.

In order to use the `OvoEventReadPluglet`, you must increase your JVM's thread stack size to at least 1 Mb. This is necessary in order to accommodate OVO processing requirements on UNIX systems. If you do not modify the JVM's thread stack size, a segmentation violation or stack overflow will occur.

To increase thread stack size, set the `JVM_ARGS` variable in the `/var/opt/OV/conf/OVI/OVI.env` file (`JVM_ARGS="-Xss1m"`) or use the `-x "-Xss1m"` parameter when starting OVI from the command line (e.g., `/opt/OV/bin/OVI.sh -x "-Xss1m" -d OvoEventReadPlugletExample.deploy`).

Description	**Location**
Request schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadRequest.xsd`
Response schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadResponse.xsd`
Exception schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadException.xsd`
Request examples	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadRequest-OVO-GUID.xml` `OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadRequest-OVO-timeRange.xml`
Response examples	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadResponse-OVO.xml`

Exception response (`ovMessageReadException`)
 	
Note: Must be of type `invalidXmlException`, `operationException` or `timeOutException`
	`Examples:`
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadInvalidXmlException.xml`
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadOperationException.xml`
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageReadTimeOutException.xml`

Here is an example of a message read request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageReadRequest xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types"
   version="1.0">

   <messageReadCriteria>
      <-- The UUID of the message to read and return -->
      <ovit:messageUUID>
         75ba8cf4-9a9b-71d6-1350-0f0278270000
      </ovit:messageUUID>

      <-- Time range used to determine which messages to return.
          Only messages that arrived on the OV Management Server
          after startTime and before endTime will be returned.
          Both startTime and endTime are in ISO 8601 format.
      <timeRange>
         <ovit:startTime>2004-05-10T09:10:00-05:00
         </ovit:startTime>
         <ovit:endTime>2004-05-10T10:30:00-05:00
         </ovit:endTime>
      </timeRange>
      -->
   </messageReadCriteria>

</ovMessageReadRequest>
```

Elements in the message read request whose values you must change are explained here:

Element Name	**Value**

`<messageUUID>`
`   75ba8cf4-9a9b-71d6-1350-0f0278270000`
`</messageUUID>`
	UUID of the message to return.

`<timeRange>`
`   <ovit:startTime>`
`      2004-05-10T09:10:00-05:00`
`   <ovit:startTime>`
`   <ovit:endTime>`
`      2004-05-10T10:30:00-05:00`
`   <ovit:endTime>`
`</timeRange>`
	Time range, in ISO 8601 format, used to determine which messages to return. Only messages that arrived on the OV management server after `startTime` and before `endTime` will be returned.

Here is an example of a message read response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageReadResponse xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types"
   version="1.0">

   <allRequestedMessagesReturned>true
   </allRequestedMessagesReturned>
   <ovMessageList>
      <ovMessage version="1.0">
         <messageEnvelope>
            <messageUUID>75ba8cf4-9a9b-71d6-1350-0f0278270000
            </messageUUID>
            <timeStamp>
```

```
            <seconds>1095800501</seconds>
            <dateTime>2004-05-10T09:01:41-06:00</dateTime>
         </timeStamp>
         <severity>Normal</severity>
         <messageSource>system.company.com</messageSource>
         <eventSource>system.company.com</eventSource>
         <primaryMessageRepository>system.company.com
         </primaryMessageRepository>
         <messageMetadataIdentifier/>
         <managedEntityReference/>
      </messageEnvelope>
      <messageBody>
         <owner/>
         <numberOfDuplicates>0</numberOfDuplicates>
         <state>PENDING</state>
         <creatingCondition/>
         <messageData application="OVO">
            <data>
               <name>application</name>
               <value type="string">app A</value>
            </data>
            <data>
               <name>cmanormalprop</name>
               <value type="string">NORMAL</value>
            </data>
            <data>
               <name>cmaprop</name>
               <value type="string">OVI</value>
            </data>
            <data>
               <name>group</name>
               <value type="string">OVO_Event_Publisher
               </value>
            </data>
            <data>
               <name>object</name>
               <value type="string">obj A</value>
            </data>
            <data>
               <name>originalText</name>
               <value type="string">
                  Node: system.company.com
                  Message group: MSG_GROUP
                  Application: app A Object: obj A
                  Severity: Normal
                  Text: testing text
               </value>
            </data>
            <data>
               <name>receiveTimeSeconds</name>
               <value type="long">1095800506</value>
            </data>
            <data>
               <name>serviceName</name>
               <value type="string">oviService</value>
            </data>
            <data>
               <name>source</name>
               <value type="string">opcmsg(1|3)</value>
            </data>
            <data>
               <name>text</name>
               <value type="string">testing text</value>
```

```
                </data>
                <data>
                   <name>type</name>
                   <value type="string">A_msg</value>
                </data>
             </messageData>
             <actionResult/>
             <messageMetadataAnnotation/>
          </messageBody>
          </ovMessage>
   </ovMessageList>
</ovMessageReadResponse>
```

OvoEventEnrichmentPluglet

`OvoEventEnrichmentPluglet` enriches OVO messages with data from other applications or a JMS client. It receives event notification messages from the OVO MSI, enriches them, and publishes them onto the message transport. OVI messages are enriched in a request/response model, where the responder has the opportunity to enrich the request. Enrichment of messages published by OVO/W is currently not supported on Windows operating systems.

Description	**Example**
Request message	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentRequest.xml`
Request schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentRequest.xsd`
Modify response message. The modified response is placed onto the OVO MSI.	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentResponseModify.xml`
No modify response message. The original message is placed onto the OVO MSI.	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentResponseNoModify.xml`
Delete response message. When the request message is marked for deletion, the message is not placed onto the OVO MSI and does not show up in the OVO operator's console.	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentResponseDelete.xml`
Response message schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentResponse.xsd`

OvoEventCreatePluglet

`OvoEventCreatePluglet` works exclusively with version 7.25-8.x OVO agents.

OVO and OVO/W messages can be created by sending requests to `OvoEventCreatePluglet`. This pluglet responds with either the message ID of the newly created message or an exception response.

Description	Location
Request schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageCreateRequest.xsd`
Response schema	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageCreateResponse.xsd`
Request example (`ovMessageCreateRequest`)	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageCreateRequest.xml`
Response example (`ovMessageCreateResponse`)	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageCreateResponse.xml`

Exception response (`ovMessageCreateException`)
 	
Note: Must be of type `invalidXmlException` or `operationException`
	Examples:
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageCreateInvalidXmlException.xml`
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageCreateOperationException.xml`
	Schema:
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageCreateException.xsd`

Any extra data or name elements (other than `application`, `object`, `text`, `group`, or `serviceName`) in a request sent to `OvoEventCreatePluglet` are treated as option variables to the OVO/U or OVO/W `opcmsg` API. For more information on this functionality, refer to the command line parameter `-option` in the man page for `opcmsg`.

Here is an example of a create request message.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageCreateRequest xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <ovit:severity>Critical</ovit:severity>
   <ovit:eventSource/>
   <messageData>
      <ovit:data>
         <ovit:name>application</ovit:name>
         <ovit:value type="string">appA</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>object</ovit:name>
         <ovit:value type="string">objA</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>text</ovit:name>
         <ovit:value type="string">the message text</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>serviceName</ovit:name>
         <ovit:value type="string">serviceA</ovit:value>
      </ovit:data>
      <ovit:data>
         <ovit:name>group</ovit:name>
```

```
        <ovit:value type="string">groupA</ovit:value>
      </ovit:data>
   </messageData>
</ovMessageCreateRequest>
```

If processing is successful, the example create request message above creates a message in OVO/U or OVO/W and the `OvoEventCreatePluglet` returns a response similar to this. The `messageUUID` element contains the ID of the newly created message.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageCreateResponse xmlns=
   "http://openview.hp.com/xmlns/ico/message"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <ovit:messageUUID>f78c16a8-0caf-71d7-188c-0f02787d0000
   </ovit:messageUUID>
</ovMessageCreateResponse>
```

OVO MSI/MEI and OVO/W WMI Event Filters

The OVO pluglets can be configured to use event filters. This is a way to restrict the OVO and OVO/W events that are translated into OVI messages by the pluglet. See OVO and OVO/W Pluglet Configuration for information on including an event filter in an OVO pluglet's `.config` file.

There are several types of event filters:

- `ALL`
- `Simple`
- `AND`
- `Simple OR`
- `Complex OR`

An example of each filter type is shown in the table. You can also find these examples in the `OVI_BASE_DIR/examples/OVI/OVO/FilterExamples` directory. All event fields that may be included in filters appear in the fifth example, simple OR filter with all possible condition elements.

OVO/W WMI event filters do not support fully-qualified node names.

Filter Description	XML Code
This ALL filter passes all events.	<filter> <all/> </filter>
This simple filter passes only events of Critical severity.	<filter> <severity>Critical</severity> </filter>
This AND filter passes only events whose severity is Critical *and* whose application is Oracle.	<filter> <and> <severity>Critical</severity> <application>Oracle</application> </and> </filter>
This simple OR filter passes only events whose severity is Critical *or* whose severity is Normal *or* whose application is Oracle.	<filter> <or> <severity>Critical</severity> <severity>Normal</severity> <application>Oracle</application> </or> </filter>

Filter Description	XML Code

This is a simple `OR` filter with all possible condition elements.
This filter passes events that meet one of the following conditions: Severity is `Critical` *or* application is `Oracle` *or* type is `MessageTypeA` *or* nodeName is `imail` *or* group is `MessageGroupA` *or* object is `Object A`.
	`<filter>`
	`  <or>`
	`  <severity>Critical</severity>`
	`  <application>Oracle</application>`
	`  <type>MessageTypeA</type>`
	`  <nodeName>imail</nodeName>`
	`  <group>MessageGroupA</group>`
	`  <object>Object A</object>`
	`  </or>`
	`</filter>`

This simple `OR` filter with all possible severity values passes only events whose severity is `Critical` *or* `Major` *or* `Minor` *or* `Warning` *or* `Normal` *or* `Unchanged` *or* `Unknown`.	`<filter>`
	`  <or>`
	`  <!-- All possible severity values -->`
	`  <severity>Critical</severity>`
	`  <severity>Major</severity>`
	`  <severity>Minor</severity>`
	`  <severity>Warning</severity>`
	`  <severity>Normal</severity>`
	`  <severity>Unchanged</severity>`
	`  <severity>Unknown</severity>`
	`  </or>`
	`<filter>`

This complex `OR` filter passes events that meet one of the following conditions:
[severity is `Critical` *and* application is `Oracle`] *or*
[severity is `Major` *and* application is `Exchange`] *or*
object is `Object A`.
	`<filter>`
	`  <or>`
	`  <and>`
	`  <severity>Critical</severity>`
	`  <application>Oracle</application>`
	`  </and>`
	`  <and>`
	`  <severity>Major</severity>`
	`  <application>Exchange</application>`
	`  </and>`
	`  <object>Object A</object>`
	`  </or>`
	`</filter>`

6 Integrating with HP Service Desk

The Integrating with HP Operations Manager for Unix and Service Desk chap, discusses how to use the SdServiceInfoEnrichmentPluglet (ses SdServiceInfoEnrichmentPluglet Configuration section) to enrich OVO messages with Service Desk data. OVI also provides the `SdInventoryPluglet`, which is useful when a direct integration with Service Desk is required. This pluglet allows create, read, update, and delete (CRUD) operations to be performed on selected types of Service Desk inventory entities including configuration items (CIs), services, persons, organizations, incidents and service calls.

The `SdInventoryPluglet` receives an inventory request (`ovInventoryRequest`), which indicates the CRUD operation to perform on a specified Service Desk inventory entity. The pluglet invokes this operation and then returns its result in an inventory response (`ovInventoryResponse`) message. If an error occurs, an `ovInventoryException` is returned in the response message.

Operations performed via OVI must follow relevant Service Desk rules. For example, child organizations can have only one parent organization. Refer to Service Desk documentation for a complete set of rules.

OVI application pluglets may be used to interact with HP Service Desk must be installed; the use of Service Pack 10 or greater is strongly encouraged.

All examples provided in this guide are specific to Service Desk 5.x. Please note that in some cases, there are schema and message format differences between Service Desk 5.x and 4.5. Refer to the `OVI_BASE_DIR/examples/OVI/configs/SD/Inventory` directory for examples that are specific to each Service Desk release. Also, schema files for both releases are provided in the `OVI_BASE_DIR/examples/OVI/schema/entity` folder. The example and schema files that are release-specific have 50 or 45 embedded in their names to make it obvious.

Service Desk Configuration

OVI requires that certain custom fields be activated for Service Desk configuration item, service, person, organization, incident and service call inventory entities. These custom fields are used internally by OVI to track the time it last modified each entity and to support the ability to reference an inventory entity by a user-specified alternate identifier. To activate custom fields in your Service Desk installation:

1. In the GUI, click **Tools** → **System**.
2. In the Administrator Console configuration area, expand **Data**, and then select **Custom Fields**.

3 Configure each inventory entity (configuration item, service, person, organization, incident and service call) by double-clicking on the entity and selecting the **Field** dropdown to look for `Date 5` and `Shorttext 5` respectively. For each field, select **Activate** and click **OK**.

4 An inventory entity may expose one or more of the following data items: `ci_1`, `workgroup_1`, `person_1`, and `organization_1`. The sample XML discussed in SdInventoryPluglet Configuration gives details about which inventory entities support which data items. If any of these data items are required for your usage of that SD inventory entity, then the corresponding fields (`CI 1`, `Workgroup 1`, `Person 1`, and `Organization 1`) also need to be activated for that entity.

See the *Service Desk Integration* section of the OVI Release Notes for a list of OVI problems that may be caused by improper Service Desk configurations.

SdInventoryPluglet Configuration

If you supplied Service Desk environment information during OVI installation, required pluglet configuration changes have been made for you. However, if you did not enter environment information during installation or if your Service Desk environment has changed, you must make manual configuration changes to the `SdInventoryPlugletExample.config` file, as shown below. Elements that are good candidates for modification are marked with a comment that reads: **`You will/may want to modify this.`**

Example `SdInventoryPluglet .deploy` and `.config` files have been provided in the `OVI_BASE_DIR/examples/OVI/configs/SD/Inventory` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory tree before editing.

The `SdInventoryPluglet.config` file looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is the SdInventoryPluglet configuration file. -->
<SdInventoryPluglet xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation=
   "/opt/OV/schema/OVI/SdInventoryPluglet.xsd">

   <!--
      Service Desk server and login account info.
   You will want to modify these.
   -->
   <ovsdServer>mysdserver.mycompany.com</ovsdServer>
   <ovsdUser>sdUser</ovsdUser>
   <ovsdUserPassword>sdUserPassword</ovsdUserPassword>
   <!--
      If you want an encrypted password, then use
      the following tag. Uncomment the
      <ovsdEncryptedUserPassword> element and
      comment out the <ovsdUserPassword> element above.

      You can use the OviEncryptUtil.sh script to encrypt
      a password string. Use the script's output as the
      value of the element below.
      You may want to modify this.
      <ovsdEncryptedUserPassword>sdUserEncryptedPassword
```

```
      </ovsdEncryptedUserPassword>
   -->

   <!--
      Switch message trace logging on/off.
          You may want to modify this.
   -->
   <traceMessages>true</traceMessages>

   <!--
      Switch message route logging on/off.
          You may want to modify this.
   -->
   <traceRoute>true</traceRoute>

</SdInventoryPluglet>
```

This table describes elements in the `SdInventoryPluglet.config` file:

Element Name	Value

`<ovsdServer>`
fully-qualified hostname
`</ovsdServer>`
	Service Desk server on which OVI performs CRUD operations.

`<ovsdUser>`*sdUser*`</ovsdUser>`	Service Desk user name that OVI uses to acquire service-oriented information.

`<ovsdUserPassword>`
sdUserPassword
`</ovsdUserPassword>`
	Password of the Service Desk user that is passed to SD as clear (unencrypted) text.

`<ovsdEncryptedUserPassword>`
sdUserEncryptedPassword
`</ovsdEncryptedUserPassword>`
	Encrypted password of the Service Desk user. It is decrypted before being used by Service Desk.
	OVI provides a utility for performing very simple password encryption. To use it, type this command at the command prompt:
	UNIX:
	`/opt/OV/bin/OviEncryptUtil.sh -e <password>`
	Windows:
	`<OVI_BASE_DIR>\bin\OviEncryptUtil.wsf -e <password>`
	The `-d <encryptedPasswd>` option decrypts a password previously encrypted by this utility.

`<traceMessages>`
boolean
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See the traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
boolean
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet will be collected and, eventually, displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

Namespaces

The namespaces used in inventory request and response (XML) messages for Service Desk are `xmlns=http://openview.hp.com/xmlns/ico/entity` and `xmlns=http://openview.hp.com/xmlns/ico/types`. These namespace declarations are required for `<ovInventoryRequest>`, `<ovInventoryResponse>`, and `<ovInventoryException>` messages.

Message Format

Requests received by `SdInventoryPluglet` are reformatted, and the response is returned via the configured transport mechanism. The inventory request and response message schemas and example XML can be found in `OVI_BASE_DIR/examples/OVI/schemas/entity`. (This directory is referred to as `SCHEMAS_DIR` in the table below.)

Schema Description	**Location**
Inventory request message	`SCHEMAS_DIR/ovInventoryRequest.xsd`
Inventory response message	`SCHEMAS_DIR/ovInventoryResponse.xsd`
Inventory exception response message	`SCHEMAS_DIR/ovInventoryException.xsd`

XML messages sent to Service Desk are only partially validated. Full validation is not performed for performance reasons. Instead, specific parts of each message are checked.

See the topics in this section for sample XML messages.

In some cases, there are schema and message format differences between Service Desk 5.x and 4.5. Refer to the `OVI_BASE_DIR/examples/OVI/schema/entity` directory for sample XML messages that are specific to each Service Desk release. The sample XML message files that are release-specific have 50 or 45 embedded in their names to make it obvious.

Configuration Item Entity Sample XML Messages

XML Message Description	Location
Inventory request message for a create operation	SCHEMAS_DIR/ovInventoryRequest_createCi50.xml SCHEMAS_DIR/ovInventoryRequest_createCi45.xml
Inventory response message for a create operation	SCHEMAS_DIR/ovInventoryResponse_createCi.xml
Inventory request message for a read operation	SCHEMAS_DIR/ovInventoryRequest_readCi.xml
Inventory response message for a read operation	SCHEMAS_DIR/ovInventoryResponse_readCi50.xml SCHEMAS_DIR/ovInventoryResponse_readCi45.xml
Inventory request message for an update operation	SCHEMAS_DIR/ovInventoryRequest_updateCi50.xml SCHEMAS_DIR/ovInventoryRequest_updateCi45.xml
Inventory response message for an update operation	SCHEMAS_DIR/ovInventoryResponse_updateCi.xml
Inventory request message for a delete operation	SCHEMAS_DIR/ovInventoryRequest_deleteCi.xml
Inventory response message for a delete operation	SCHEMAS_DIR/ovInventoryResponse_deleteCi.xml

Incident Entity Sample XML Messages

XML Message Description	Location
Inventory request message for a create operation	SCHEMAS_DIR/ovInventoryRequest_createIncident50.xml SCHEMAS_DIR/ovInventoryRequest_createIncident45.xml
Inventory response message for a create operation	SCHEMAS_DIR/ovInventoryResponse_createIncident.xml
Inventory request message for a read operation	SCHEMAS_DIR/ovInventoryRequest_readIncident.xml
Inventory response message for a read operation	SCHEMAS_DIR/ovInventoryResponse_readIncident50.xml SCHEMAS_DIR/ovInventoryResponse_readIncident45.xml
Inventory request message for an update operation	SCHEMAS_DIR/ovInventoryRequest_updateIncident50.xml SCHEMAS_DIR/ovInventoryRequest_updateIncident45.xml

Inventory response message for an update operation	SCHEMAS_DIR/ovInventoryResponse_updateIncident.xml
Inventory request message for a delete operation	SCHEMAS_DIR/ovInventoryRequest_deleteIncident.xml
Inventory response message for a delete operation	SCHEMAS_DIR/ovInventoryResponse_deleteIncident.xml

Organization Entity Sample XML Messages

XML Message Description	Location
Inventory request message for a create operation	SCHEMAS_DIR/ovInventoryRequest_createOrganization50.xml SCHEMAS_DIR/ovInventoryRequest_createOrganization45.xml
Inventory response message for a create operation	SCHEMAS_DIR/ovInventoryResponse_createOrganization.xml
Inventory request message for a read operation	SCHEMAS_DIR/ovInventoryRequest_readOrganization.xml
Inventory response message for a read operation	SCHEMAS_DIR/ovInventoryResponse_readOrganization50.xml SCHEMAS_DIR/ovInventoryResponse_readOrganization45.xml
Inventory request message for an update operation	SCHEMAS_DIR/ovInventoryRequest_updateOrganization50.xml SCHEMAS_DIR/ovInventoryRequest_updateOrganization45.xml
Inventory response message for an update operation	SCHEMAS_DIR/ovInventoryResponse_updateOrganization.xml
Inventory request message for a delete operation	SCHEMAS_DIR/ovInventoryRequest_deleteOrganization.xml
Inventory response message for a delete operation	SCHEMAS_DIR/ovInventoryResponse_deleteOrganization.xml

Person Entity Sample XML Messages

XML Message Description	Location
Inventory request message for a create operation	SCHEMAS_DIR/ovInventoryRequest_createPerson50.xml SCHEMAS_DIR/ovInventoryRequest_createPerson45.xml
Inventory response message for a create operation	SCHEMAS_DIR/ovInventoryResponse_createPerson.xml
Inventory request message for a read operation	SCHEMAS_DIR/ovInventoryRequest_readPerson.xml
Inventory response message for a read operation	SCHEMAS_DIR/ovInventoryResponse_readPerson50.xml SCHEMAS_DIR/ovInventoryResponse_readPerson45.xml
Inventory request message for an update operation	SCHEMAS_DIR/ovInventoryRequest_updatePerson50.xml SCHEMAS_DIR/ovInventoryRequest_updatePerson45.xml
Inventory response message for an update operation	SCHEMAS_DIR/ovInventoryResponse_updatePerson.xml
Inventory request message for a delete operation	SCHEMAS_DIR/ovInventoryRequest_deletePerson.xml
Inventory response message for a delete operation	SCHEMAS_DIR/ovInventoryResponse_deletePerson.xml

Service Call Entity Sample XML Messages

XML Message Description	Location
Inventory request message for a create operation	SCHEMAS_DIR/ovInventoryRequest_createServiceCall50.xml SCHEMAS_DIR/ovInventoryRequest_createServiceCall45.xml
Inventory response message for a create operation	SCHEMAS_DIR/ovInventoryResponse_createServiceCall.xml
Inventory request message for a read operation	SCHEMAS_DIR/ovInventoryRequest_readServiceCall.xml
Inventory response message for a read operation	SCHEMAS_DIR/ovInventoryResponse_readServiceCall50.xml SCHEMAS_DIR/ovInventoryResponse_readServiceCall45.xml

Inventory request message for an update operation	SCHEMAS_DIR/ovInventoryRequest_updateServiceCall50.xml SCHEMAS_DIR/ovInventoryRequest_updateServiceCall45.xml
Inventory response message for an update operation	SCHEMAS_DIR/ovInventoryResponse_updateServiceCall.xml
Inventory request message for a delete operation	SCHEMAS_DIR/ovInventoryRequest_deleteServiceCall.xml
Inventory response message for a delete operation	SCHEMAS_DIR/ovInventoryResponse_deleteServiceCall.xml

Service Entity Sample XML Messages

XML Message Description	Location
Inventory request message for a create operation	SCHEMAS_DIR/ovInventoryRequest_createService50.xml SCHEMAS_DIR/ovInventoryRequest_createService45.xml
Inventory response message for a create operation	SCHEMAS_DIR/ovInventoryResponse_createService.xml
Inventory request message for a read operation	SCHEMAS_DIR/ovInventoryRequest_readService.xml
Inventory response message for a read operation	SCHEMAS_DIR/ovInventoryResponse_readService50.xml SCHEMAS_DIR/ovInventoryResponse_readService45.xml
Inventory request message for an update operation	SCHEMAS_DIR/ovInventoryRequest_updateService50.xml SCHEMAS_DIR/ovInventoryRequest_updateService45.xml
Inventory response message for an update operation	SCHEMAS_DIR/ovInventoryResponse_updateService.xml
Inventory request message for a delete operation	SCHEMAS_DIR/ovInventoryRequest_deleteService.xml
Inventory response message for a delete operation	SCHEMAS_DIR/ovInventoryResponse_deleteService.xml

Request/Response Examples

Sample request and response messages sent for different types of operations in the lifecycle of a Service Desk configuration item (CI) are provided in:

- Create Request/Response
- Delete Request/Response
- Read Request/Response
- Update Request/Response

These are simplified examples showing the types of data that can be created, updated, read, and deleted from a SD 5.x installation using OVI.

All examples provided in this guide are specific to Service Desk 5.x. Please note that in some cases, there are schema and message format differences between Service Desk 5.x and 4.5. Refer to the `OVI_BASE_DIR/examples/OVI/configs/SD/Inventory` directory for examples that are specific to each Service Desk release. The example files that are release-specific have 50 or 45 embedded in their names to make it obvious.

Although the examples deal exclusively with CI inventory entities, SD person records, services, organizations, incidents and service calls may be used as well. There are many CI data fields that can be set and read but are not shown in these examples. For additional information about setting and reading CI data fields, refer to the sample `ovInventoryRequest` and `ovInventoryResponse` XML files installed with OVI in `OVI_BASE_DIR/examples/OVI/schemas/entity`.

In request and response messages, four distinct fields may be used to identify a specific Service Desk inventory entity:

- Service Desk `oid`
- Service Desk `id`
- Service Desk `searchcode`
- user-specified `alternateId`

`oid`, `id`, and `searchcode` are assigned by Service Desk and, in some cases, are not user-modifiable. `alternateId` is user-specified; it must be enabled for certain inventory entities by activating the `Shorttext 5` field, as discussed in Service Desk Configuration. Not all inventory entities can be referenced by each of these identifiers. Person and organization entities cannot be referred to by a Service Desk `id`; Service, incident and service call entities cannot be referenced by a Service Desk `searchcode`. All inventory entities can be referenced by Service Desk `oid` and (if the field is activated for the entity) by the user-specified `alternateId`.

If more than one identifier is provided in an `ovInventoryRequest`, OVI searches for the specified entity according to the following precedence: Service Desk `oid`, user-specified `alternateId`, Service Desk `id`, and Service Desk `searchcode`. Upon completion of the requested operation on the inventory entity, OVI returns an `ovInventoryResponse` containing all relevant identifiers for that entity. For subsequent operations on that inventory entity, any of the returned identifiers may be used.

Create Request/Response

Here is an example of a create operation:

```
<?xml version="1.0" encoding="UTF-8"?>
<ovInventoryRequest xmlns="http://openview.hp.com/xmlns/ico/entity"
    xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

    <requestDetails>
        <timeStamp>2003-09-30T13:30:47-05:00</timeStamp>
        <action>create</action>
        <entityType>ConfigurationItem</entityType>
    </requestDetails>

    <entityDetails>
        <inventoryItem>
            <ovit:data>
                <ovit:name>searchcode</ovit:name>
                <ovit:value type="string">TEST-CI</ovit:value>
            </ovit:data>

            <ovit:data>
               <ovit:name>name</ovit:name>
               <ovit:value type="string">test-ci-name</ovit:value>
            </ovit:data>

            <ovit:data>
               <ovit:name>status</ovit:name>
               <ovit:value type="string">Test</ovit:value>
            </ovit:data>

            <ovit:data>
               <ovit:name>maxInstallations</ovit:name>
               <ovit:value type="integer">5</ovit:value>
            </ovit:data>
        </inventoryItem>
    </entityDetails>
</ovInventoryRequest>
```

The example inventory request message above creates a simple test configuration item with a search code of `TEST-CI`, a name of `test-ci-name`, a status of **`Test`**, and a max installations value of `5`. The `searchcode`, `name`, `status`, and `maxInstallations` fields are the minimum fields required for the creation of a configuration item with the default Service Desk 4.5 CI template. The status value of `Test` assumes that that status value has been previously defined in Service Desk. If there are no problems executing the preceding create request, the `SdInventoryPluglet` returns a response message similar to this:

```
<?xml version="1.0" encoding="UTF-8"?>

<ovInventoryResponse xmlns="http://openview.hp.com/xmlns/ico/entity"
    version="1.0" xmlns:ovit="http://openview.hp.com/xmlns/ico/types">

    <entityDetails>
        <inventoryItem>
            <ovit:data>
                <ovit:name>id</ovit:name>
                <ovit:value type="long">1654</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>oid</ovit:name>
                <ovit:value type="long">281478317867908</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>searchcode</ovit:name>
                <ovit:value type="string">TEST-CI</ovit:value>
            </ovit:data>

        </inventoryItem>
    </entityDetails>
</ovInventoryResponse>
```

The only values returned in this create operation response are the identifiers that can be used to read, update, or delete the newly created configuration item. In this case, the newly created CI has a search code of `TEST-CI`, and Service Desk assigned the new CI an ID of `1654` and an OID (object id) of `281478317867908`. The difference between the `id` and `oid` fields is that the `id` field is visible through the Service Desk graphical user interface and is not always unique; the `oid` is always unique but is not viewable via the GUI.

Update Request/Response

Once a new configuration item has been created, it can be updated by sending the following request to the `SdInventoryPluglet`:

```
<?xml version="1.0" encoding="UTF-8"?>

<ovInventoryRequest xmlns="http://openview.hp.com/xmlns/ico/entity"
    xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

    <requestDetails>
        <timeStamp>2003-09-30T13:30:47-05:00</timeStamp>
        <action>update</action>
        <entityType>ConfigurationItem</entityType>
    </requestDetails>

    <entityDetails>
        <inventoryItem>
            <ovit:data>
                <ovit:name>oid</ovit:name>
                <ovit:value type="long">281478317867908</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>alternateId</ovit:name>
                <ovit:value type="string">GRE39BA5493</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>maxInstallations</ovit:name>
                <ovit:value type="integer">10</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>name</ovit:name>
                <ovit:value type="string">name value</ovit:value>
            </ovit:data>

            <ovit:arrayData>
                <ovit:name>usedByOrganizations</ovit:name>
                <ovit:value type="string">EGOTRADING</ovit:value>
                <ovit:value type="string">DEVELOPMENT</ovit:value>
            </ovit:arrayData>
        </inventoryItem>
    </entityDetails>
</ovInventoryRequest>
```

The preceding inventory request message directs `SdInventoryPluglet` to find the configuration item with an oid equal to `281478317867908` and to update its `alternateId`, `maxInstallations`, `name`, and `Used By Organizations` fields. Since `maxInstallations` is the only field that was previously set, its value is updated to `10`. The remaining fields are set for the first time. If there are no problems with the update request, the `SdInventoryPluglet` returns an inventory response message similar to this:

```
<?xml version="1.0" encoding="UTF-8"?>

<ovInventoryResponse xmlns="http://openview.hp.com/xmlns/ico/entity"
    version="1.0" xmlns:ovit="http://openview.hp.com/xmlns/ico/types">

    <entityDetails>
        <inventoryItem>
            <ovit:data>
                <ovit:name>alternateId</ovit:name>
                <ovit:value type="string">GRE39BA5493</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>id</ovit:name>
                <ovit:value type="long">1654</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>oid</ovit:name>
                <ovit:value type="long">281478317867908</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>searchcode</ovit:name>
                <ovit:value type="string">TEST-CI</ovit:value>
            </ovit:data>
        </inventoryItem>
    </entityDetails>
</ovInventoryResponse>
```

As in the response to the create operation, the update operation response contains the identifier fields of the CI that were updated. Since the `alternateId` field now contains a value, it is returned along with the other identifier fields. Any of the four identifiers (`oid`, `id`, `searchcode`, or `alternateId`) can be used to find our test CI for a read, update, or delete operation.

Read Request/Response

An example request that can be sent to the `SdInventoryPluglet` to read the test CI (identified by the alternate id field) appears below:

```
<?xml version="1.0" encoding="UTF-8"?>

<ovInventoryRequest xmlns="http://openview.hp.com/xmlns/ico/entity"
    xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

    <requestDetails>
        <timeStamp>2003-09-30T13:30:47-05:00</timeStamp>
        <action>read</action>
        <entityType>ConfigurationItem</entityType>
    </requestDetails>

    <entityDetails>
        <inventoryItem>
            <ovit:data>
                <ovit:name>alternateId</ovit:name>
                <ovit:value type="string">GRE39BA5493</ovit:value>
            </ovit:data>
        </inventoryItem>
    </entityDetails>
</ovInventoryRequest>
```

The preceding request directs `SdInventoryPluglet` to find the CI that has an `alternateId` field value of `GRE39BA5493` and returns all information that Service Desk knows about that CI. If there is no problem finding the CI, the `SdInventoryPluglet` returns the following response:

```
<?xml version="1.0" encoding="UTF-8"?>

<ovInventoryResponse xmlns="http://openview.hp.com/xmlns/ico/entity"
    version="1.0" xmlns:ovit="http://openview.hp.com/xmlns/ico/types">

    <entityDetails>
        <inventoryItem>
            <ovit:data>
                <ovit:name>alternateId</ovit:name>
                <ovit:value type="string">GRE39BA5493</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>blocked</ovit:name>
                <ovit:value type="boolean">false</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>id</ovit:name>
                <ovit:value type="long">1654</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>lastModifiedByOvi</ovit:name>
                <ovit:value type="dateTime">
                2006-05-19T11:45:59+08:00</ovit:value>
            </ovit:data>

           <ovit:data>
                <ovit:name>maxInstallations</ovit:name>
                <ovit:value type="integer">10</ovit:value>
```

```
            </ovit:data>

            <ovit:data>
                <ovit:name>name</ovit:name>
                <ovit:value type="string">name value</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>oid</ovit:name>
                <ovit:value type="long">281478317867908</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>searchcode</ovit:name>
                <ovit:value type="string">TEST-CI</ovit:value>
            </ovit:data>

            <ovit:data>
               </ovit:name>serviceNavigatorStatus</ovit:name>
                <ovit:value type="string">Retrieving...</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>status</ovit:name>
                <ovit:value type="string">Test</ovit:value>
            </ovit:data>

            <ovit:data>
                </ovit:name>timezone</ovit:name>
                <ovit:value type="string">(UTC-07:00) - Phoenix Standard
Time - PNT</ovit:value>
            </ovit:data>

            <ovit:data>
                </ovit:name>unique</ovit:name>
                <ovit:value type="boolean">false</ovit:value>
            </ovit:data>

            <ovit:arrayData>
                <ovit:name>usedByOrganizations</ovit:name>
                <ovit:value type="string">EGOTRADING</ovit:value>
                <ovit:value type="string">DEVELOPMENT</ovit:value>
            </ovit:arrayData>
        </inventoryItem>
    </entityDetails>
</ovInventoryResponse>
```

The read response from `SdInventoryPluglet` contains just those fields that have been assigned values. The only returned field that was not assigned a value by the user (other than the `oid` and `id` identifier fields, which are assigned values by Service Desk itself) is the `lastModifiedByOvi` field. For each of the inventory items (CIs, services, persons, organizations, incidents and service calls), the `SdInventoryPluglet` automatically fills in this field with a value representing the last time that the item was modified by OVI.

Delete Request/Response

A delete request for our test CI sent to `SdInventoryPluglet` looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>

<ovInventoryRequest xmlns="http://openview.hp.com/xmlns/ico/entity"
    xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

    <requestDetails>
        <timeStamp>2003-09-30T13:30:47-05:00</timeStamp>
        <action>delete</action>
        <entityType>ConfigurationItem</entityType>
    </requestDetails>

    <entityDetails>
        <inventoryItem>
            <ovit:data>
                <ovit:name>id</ovit:name>
                <ovit:value type="long">1654</ovit:value>
            </ovit:data>
        </inventoryItem>
    </entityDetails>
</ovInventoryRequest>
```

The preceding request causes `SdInventoryPluglet` to find and delete the CI whose id field is set to `1654`. The test CI could have just as easily been identified by its `oid`, `searchcode`, or `alternateId`. If the `SdInventoryPluglet` successfully locates and deletes the CI, the pluglet returns the following response. The identifier fields of the inventory item that was deleted are returned in the response.

```
<?xml version="1.0" encoding="UTF-8"?>

<ovInventoryResponse xmlns="http://openview.hp.com/xmlns/ico/entity"
    version="1.0" xmlns:ovit="http://openview.hp.com/xmlns/ico/types">

    <entityDetails>
        <inventoryItem>
            <ovit:data>
                <ovit:name>alternateId</ovit:name>
                <ovit:value type="string">GRE39BA5493</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>id</ovit:name>
                <ovit:value type="long">1654</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>oid</ovit:name>
                <ovit:value type="long">281478317867908</ovit:value>
            </ovit:data>

            <ovit:data>
                <ovit:name>searchcode</ovit:name>
                <ovit:value type="string">TEST-CI</ovit:value>
            </ovit:data>
        </inventoryItem>
    </entityDetails>
</ovInventoryResponse>
```

7 Integrating with HP Operations Manager for Unix and Service Desk

OVI includes the `SdServiceInfoEnrichmentPluglet`, which receives an OVO message, obtains corresponding Service Desk data, and adds selected pieces of it to the received message. Service Desk data is added to the OVO custom message attribute fields and is tagged with the `<cmaFieldName>` descriptor provided in the `.config` file for `SdServiceInfoEnrichmentPluglet`. This enriched response message of Service Desk is returned to OVO via the response created by `SdServiceInfoEnrichmentPluglet`.

If you want to have OVO messages enriched with service-related data, start one instance of OVI using the `SdServiceInfoEnrichmentPlugletExample.deploy` file and another OVI instance using the `OvoEventEnrichmentPlugletExample.deploy` file.

Log information for each OVI instance appears in its own file. Information from the first OVI instance is initially written to `ovi.log`, but when the second OVI instance starts, log data from the first instance is transferred to `ovi.log.1` and the second instance uses `ovi.log`. If a problem occurs, you must examine *both* log files for errors.

HP Operations Manager messages may be enriched with data from HP Service Desk 4.5 with Service Pack 5 or greater. SP 10 or greater is recommended.

Only OVO/U messages can be enriched.

Configuration Overview

To enrich OVO messages with Service Desk data, the following configuration tasks must be performed:

1 Configure OVO to put messages on the OVO Message Stream Interface (MSI). See the OVO Configuration (UNIX only) section.

2 Configure the appropriate application pluglets to achieve your intent:

Pluglet Name	Purpose	Detailed Information In
`OvoEventEnrichment Pluglet` and `SdServiceInfoEnrichment Pluglet`	To enrich OVO messages using Service Desk (i.e., to get Service Desk-enriched OVO messages). To use `OvoEventEnrichment Pluglet`, you must understand how to configure OVO and how to filter and make use of the resulting OVO messages that OVI delivers. UNIX only.	OvoEventEnrichmentPluglet Configuration and SdServiceInfoEnrichmentPluglet Configuration

3 Ensure that your system's `CLASSPATH` environment variable (or its setting in the OVI environment file) contains entries for the `.zip/.jar` files required to connect to your Service Desk installation. See the *Prerequisites* section in the Installation Guide for details.

Message Format and Namespaces

The contents of XML messages returned to OVO are only partially validated. A full validation is not performed for performance reasons. Only specific parts of each enriched message are checked.

The namespace used in OVO enrichment XML messages is `xmlns=http://openview.hp.com/xmlns/ico/message`. This is required for `<ovMessageEnrichmentRequest>` and `<ovMessageEnrichmentResponse>` messages.

This table describes OVI event enrichment request and valid response messages:

Description	Example

When `OvoEventEnrichmentPluglet` is used to receive event notification messages from the OVO MSI, enriched messages (data added by HP Service Desk) are published onto the message transport. OVI messages are enriched in a request/response model, where the responder has the opportunity to enrich the request. Enrichment of messages published by OVO/W is currently not supported on Windows operating systems.	Message:
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentRequest.xml`
	Schema:
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentRequest.xsd`
	Schema for enrichment response:
	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentResponse.xsd`

Modify the message. The modified response is placed onto the OVO MSI.	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentResponseModify.xml`
Do not modify the message. The original message is placed onto the OVO MSI.	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentResponseNoModify.xml`
Delete the message. When the request message is marked for deletion, the message is not placed onto the OVO MSI and does not show up in the OVO operator's console.	`OVI_BASE_DIR/examples/OVI/schemas/message/ovMessageEnrichmentResponseDelete.xml`

SdServiceInfoEnrichmentPluglet Configuration

To enable SD message enrichment in your environment, a few configuration changes must be made to the `SdServiceInfoEnrichmentPlugletExample.config` file. If you supplied Service Desk environment information during installation, the configuration changes have been made for you. However, if you did not enter environment information during installation (or if your Service Desk environment has changed), you must make manual configuration changes to the file.

Copy the example `.config` and `.deploy` files to the `OVI_DATA_DIR/conf/OVI` directory tree before editing.

Example `SdServiceInfoEnrichmentPluglet .deploy` and `.config` files are provided in the `OVI_BASE_DIR/examples/OVI/SD/ServiceInfoEnrichment` directory. The `SdServiceInfoEnrichmentPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is the configuration file for OVO
     SdServiceInfoEnrichment pluglet. -->
<SdServiceInfoEnrichmentPluglet xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation=
   "/opt/OV/schema/OVI/SdServiceInfoEnrichmentPluglet.xsd">

    <!--
       Service Desk server and login account.
              You will want to modify these.
    -->
    <ovsdServer>mysdserver.mycompany.com</ovsdServer>
    <ovsdUser>sdUser</ovsdUser>
    <ovsdUserPassword>sdUserPassword</ovsdUserPassword>

    <!--
       If you want an encrypted password, then use the
       following tag. Uncomment the <ovsdEncryptedUserPassword>
       element and comment out the <ovsdUserPassword>
       element above.

       You can use the OviEncryptUtil.sh script to encrypt
       a password string. Use the script's output as the value
       of the element below.
       You may want to modify this.

       <ovsdEncryptedUserPassword>sdUserEncryptedPassword
       </ovsdEncryptedUserPassword>
    -->

    <!--
       Field name that describes the Service Desk information
       that will be added to incoming OVO messages. It will
       be included with the information passed back to OVO.
       You may want to modify this.
    -->
    <cmaFieldName>Service</cmaFieldName>

    <!--
       Switch message trace logging on/off.
       You may want to modify this.
    -->
    <traceMessages>true</traceMessages>

    <!--
       Switch message route logging on/off.
       You may want to modify this.
    -->
    <traceRoute>true</traceRoute>

</SdServiceInfoEnrichmentPluglet>
```

The following table describes elements in the `SdServiceInfoEnrichmentPlugletExample.config` file:

Element Name	Value

`<ovsdServer>`
`   fully-qualified hostname`
`</ovsdServer>`
	Service Desk server from which OVI receives service information.

`<ovsdUser>sdUser</ovsdUser>`	Service Desk user name that OVI uses to acquire service-oriented information.

`<ovsdUserPassword>`
`   sdUserPassword`
`</ovsdUserPassword>`
	Password of the Service Desk user that is passed to Service Desk as clear (unencrypted) text.

`<ovsdEncryptedUserPassword>`
`   sdUserEncryptedPassword`
`</ovsdEncryptedUserPassword>`
	Encrypted password of the Service Desk user. It is decrypted before being used by Service Desk.
	
	OVI provides a utility for performing very simple password encryption. To use it, type this command at the command prompt:
	UNIX: `/opt/OV/bin/OviEncryptUtil.sh -e <password>`
	Windows: `<OVI_BASE_DIR>\bin\OviEncryptUtil.wsf -e <password>`
	The `-d <encryptedPasswd>` option decrypts a password previously encrypted by this utility.

`<cmaFieldName>`
`  Service`
`</cmaFieldName>`
	Field name that describes the Service Desk information that is added to incoming OVO messages. It is included with the information passed back to OVO in the message's CMA fields.

`<traceMessages>`
`   boolean`
`</traceMessages>`
	Sets message trace logging on or off for this pluglet. See the traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
`   boolean`
`</traceRoute>`
	Sets message route logging on or off for this pluglet. See the traceRoute Settings section for more details. (Default: `true`)

8 Integrating with HP Service Navigator

Navigator pluglets allow interaction with HP Service Navigator (SN). OVI and SN must be installed on the same system.

OVI provides two application pluglets, `SNQueryPluglet` (UNIX) and `SNWQueryPluglet` (Windows), that obtain service problem root cause and impacted services information from Service Navigator. Alternatively, they retrieve service hierarchy data representing the tree of children and dependencies of a given service. In either case, returned service information includes the status of each listed service. The maximum number of concurrent connections supported by Service Navigator on UNIX is 8. If an attempt is made to exceed this number, an exception is generated, indicating that OVI is unable to connect to Service Navigator.

`SNQueryPluglet` (for UNIX) allows you to request data from HP Service Navigator, which is shipped with OVO. `SNWQueryPluglet` (for Windows) provides access to data from HP Service Navigator 7.x, which is shipped with OVO/W. On Windows systems, OVI must run under an account that is a member of the `OVO Operators` group on the OVO/W management server.

Service Navigator Query Pluglets

When `SNQueryPluglet` or `SNWQueryPluglet` receives a request, it places data from the request in a Service Navigator XML-formatted service request that is used to query the Service Navigator service engine. The response from Service Navigator is put into an XML-formatted message, which is returned to the requesting pluglet.

To make this functionality work in your environment, you may need to make simple configuration changes to the example `.config` and `.in` files provided for these pluglets. Elements that are good candidates for modification are marked with a comment that reads: **`You may want to modify this.`**

See:

- SNQueryPluglet Configuration
- SNWQueryPluglet Configuration

SNQueryPluglet Configuration

Example `SNQueryPluglet .deploy`, `.config` and `.in` files have been provided in the `OVI_BASE_DIR/examples/OVI/configs/SN/QueryStatus and QueryHierarchy` directories. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory tree before editing.

The SNQueryPlugletExample.config file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<snQueryPluglet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="/opt/OV/schema/OVI/SNQueryPluglet.xsd">

   <!--
      Switch on message trace logging
   -->
   <traceMessages>true</traceMessages>
   <!--
      Switch on message route logging
   -->
   <traceRoute>true</traceRoute>
</snQueryPluglet>
```

The following table describes elements in the SNQueryPlugletExample.config file:

Element Name	Value
<traceMessages> *boolean* </traceMessages>	Controls whether an INFO level log message is created when a message arrives at this pluglet. Valid values are true or false. See the traceMessages Settings section for more details. (Default: true)
<traceRoute> *boolean* </traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet will be collected and, eventually, displayed. Valid values are true or false. See the traceRoute Settings section for more details. (Default: true)

If you use TestRequesterPluglet to generate a test request for SNQueryPluglet, you can modify its associated TestRequesterPlugletExample.in file to contain your desired request values. For example, if you use the TestRequesterPluglet in the QueryStatus subdirectory, you can edit the QueryStatus/TestRequesterPlugletExample.in file so that it contains the identifier of the entity whose status information you want to receive. The QueryStatus/TestRequesterPlugletExample.in file's contents appear below, and the element you may want to change is indicated.

```
<?xml version="1.0" encoding="UTF-8"?>

<entityStatusRequest
   xmlns="http://openview.hp.com/xmlns/ico/entity"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types"
   version="1.0">

   <!--
      System for which to acquire status information.
   -->
            You may want to modify this:
    <entity>
        <ovit:reference>email</ovit:reference>
    </entity>
 </entityStatusRequest>
```

The following table describes elements in the QueryStatus/ TestRequesterPlugletExample.in file:

Element Name	Value
<ovit:reference> *email* </ovit:reference>	Service Navigator name attribute of the service whose status is to be obtained.

SNWQueryPluglet Configuration

Example SNWQueryPluglet .deploy, .config and .in files have been provided in the OVI_BASE_DIR/examples/OVI/configs/SN/QueryStatus and QueryHierarchy directories. If you need to make changes, copy them to the OVI_DATA_DIR/conf/OVI directory tree before editing.

The SNWQueryPlugletExample.config file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<snWQueryPluglet
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation=
   "C:\Program Files\HP OpenView\schema\OVI\SNWQueryPluglet.xsd">

   <!--
      Switch on message trace logging
   -->
   <traceMessages>true</traceMessages>
   <!--
      Switch on message route logging
   -->
   <traceRoute>true</traceRoute>
</snWQueryPluglet>
```

The following table describes elements in the SNWQueryPlugletExample.config file:

Element Name	Value
<traceMessages> *boolean* </traceMessages>	Controls whether an INFO level log message is created when a message arrives at this pluglet. Valid values are true or false. See the traceMessages Settings section for more details. (Default: true)
<traceRoute> *boolean* </traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet will be collected and, eventually, displayed. Valid values are true or false. See the traceRoute Settings section for more details. (Default: true)

If you use TestRequesterPluglet to generate a test request for SNWQueryPluglet, you can modify its associated TestRequesterPlugletExample.in file to contain your desired request values. For example, if you use the TestRequesterPluglet in the QueryHierarchy subdirectory, you can edit the QueryHierarchy/TestRequesterPlugletExample.in file so that it contains the identifier of the entity

whose service hierarchy information you want to receive. The `QueryHierarchy/TestRequesterPlugletExample.in` file's contents appear below, and the element you may want to change is indicated.

```
<?xml version="1.0" encoding="UTF-8"?>

<entityHierarchyRequest
   xmlns="http://openview.hp.com/xmlns/ico/entity"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types"
   version="1.0">

   <!--
      System for which to acquire service hierarchy information.
   -->
      You may want to modify this:
   <entity>
      <ovit:reference>WINOSSPI:App</ovit:reference>
   </entity>
</entityHierarchyRequest>
```

The following table describes elements in the `QueryHierarchy/TestRequesterPlugletExample.in` file:

Element Name	Value

`<ovit:reference>`
`WINOSSPI:App`
`</ovit:reference>`
	Service Navigator name attribute of the service whose hierarchy information is to be obtained.

Namespaces and Message Format

The namespaces used in the XML messages for Service Navigator are `xmlns=http://openview.hp.com/xmlns/ico/entity` and `xmlns=http://openview.hp.com/xmlns/ico/types`.

These namespaces are required for `<entityStatusRequest>` and `<entityStatusResponse>`, which are used to retrieve service problem root cause and impacted services data. They are also required for `<entityHierarchyRequest>` and `<entityHierarchyResponse>` messages, which are used when obtaining service hierarchy information.

The contents of XML messages sent to Service Navigator are only partially validated. Full validation is not performed for performance reasons. Instead, only specific parts of each message are checked.

Message formats are described for:

- entityStatus Messages
- entityHierarchy Messages

entityStatus Messages

`entityStatus` requests received by `SN[W]QueryPluglet` are reformatted, and the response is returned via the configured transport mechanism. There are two valid responses to this request message:

- entity status response
- entity status exception (error message)

Message/Schema Description	Location
Request message	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityStatusRequest.xml`
Request schema	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityStatusRequest.xsd`
Entity status response message	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityStatusResponse.xml`
Entity status response schema	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityStatusResponse.xsd`

Entity status exception response message	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityStatusEntityNotFoundException.xml`
	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityStatusInvalidXmlException.xml`
	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityStatusOperationException.xml`

Entity status exception response schema	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityStatusException.xsd`

entityHierarchy Messages

`entityHierarchy` requests received by `SNQueryPluglet` or `SNWQueryPluglet` are reformatted, and the response is returned via the configured transport mechanism. There are two valid responses to this request message:

- entity hierarchy response
- entity hierarchy exception (error message)

Message/Schema Description	Location
Request message	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityHierarchyRequest.xml`
Request schema	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityHierarchyRequest.xsd`
Entity hierarchy response message	`OVI_BASE_DIR/examples/OVI/schemas/entity/entityHierarchyResponse.xml`

Message/Schema Description	Location
Entity hierarchy response schema	OVI_BASE_DIR/examples/OVI/schemas/entity/entityHierarchyResponse.xsd
Entity hierarchy exception response message	OVI_BASE_DIR/examples/OVI/schemas/entity/entityHierarchyEntityNotFoundException.xml OVI_BASE_DIR/examples/OVI/schemas/entity/entityHierarchyInvalidXmlException.xml OVI_BASE_DIR/examples/OVI/schemas/entity/entityHierarchyOperationException.xml
Entity hierarchy exception response schema	OVI_BASE_DIR/examples/OVI/schemas/entity/entityHierarchyException.xsd

9 Integrating with HP Performance Manager

`OvpmQueryPluglet` enables interaction with HP Performance Manager (OVPM). This pluglet queries OVPM for:

- Metric data from specified systems within a given time frame
- Report data from specified systems within a given time frame
- Metric information, i.e., metrics being collected for a specific data source and entity pair
- Source information, i.e., data sources and entities (systems or probes) for which a system is collecting data

Performance metric data can be retrieved by sending a request for specific performance metric data to `OvpmQueryPluglet`. The request must contain the name of the system(s)/probe types for which performance data is to be returned, a date/time range, and an OV Performance Manager data source name. Optionally, the request may contain a directive for OVPM-based data summarization to occur. This is an alternative to the data source-based summarization that occurs by default. The OVPM-based option may be useful for summarizing multi-instance data from external sources such as the HP WebLogic SPI. Most OVI users can ignore this option and use default summarization.

When a performance metric request (`performanceMetricRequest`) arrives, `OvpmQueryPluglet` gathers the designated metrics from the specified systems for the given time frame and returns them to the requester in the form of a `performanceMetricResponse`. If an error occurs, a `performanceMetricException` is returned either by itself or as part of the performance metric response.

Report data can be obtained by sending a `performanceReportRequest` to `OvpmQueryPluglet`. OVI currently supports the retrieval of only baseline report data. `OvpmQueryPluglet` obtains the requested report and returns it in the form of a `performanceReportResponse`. If an error occurs, a `performanceReportException` is returned.

Performance metric information, consisting of the metrics being collected for one specific data source and entity pair, can also be obtained by sending a request to `OvpmQueryPluglet`. The request must contain the name of a valid OV Performance Manager data source. If the data source is OVPA or CODA, the request must also specify a system/probe name; other data sources do not require specification of an entity because OVPM provides the same information for all entities. When a `performanceMetricInfoRequest` arrives, the `OvpmQueryPluglet` gathers metric information and returns it to the requester in a `performanceMetricInfoResponse`. If an error occurs, a `performanceMetricException` is returned as part of the performance metric information response.

Performance source information, which contains a list of the data sources and entities (i.e., systems or probes) for which the OVPM server is collecting data, can be retrieved by sending a `performanceSourceInfoRequest` to `OvpmQueryPluglet`. `OvpmQueryPluglet` returns the desired information to the requester in a `performanceSourceInfoResponse`. If an error occurs, a `performanceMetricException` is returned as part of the response.

`OvpmQueryPluglet` retrieves performance information from HP Performance Manager via HTTP. Thus, the `OvpmQueryPluglet` and OV Performance Manager do not have to run on the same machine.

HP Performance Manager must be installed before using `OvpmQueryPluglet`. The full OVPM product, whether acquired alone or in a product bundle, is required rather than just the HPGC (HP Graphing Component) portion embedded in products like OVO/W, OVIS, and Reporter.

Supported OVPM data sources include:

- HP Performance Agent (OVPA)
- HP Operations Performance Agent (CODA)
- HP Internet Services (OVIS)
- HP Reporter (REPORTER)
- Generic Database (DB)

The DB data source is available only with OVPM 5.0 or greater.

HP Performance Manager System Configuration

When you use `OvpmQueryPluglet` to gather a large amount of performance metric data, system heap size requirements increase. You can easily run out of memory when requesting a large number of metrics. The table below provides estimates of the heap size required when gathering 30 metrics at 5 minute granularity from 1 system (entity) during various time periods.

Recommended Heap Size				
	Data for 3 days	**Data for 4 days**	**Data for 5 days**	**Data for 7 days**
HP-UX	128 Mb	128 Mb	256 Mb	256 Mb
Solaris	128 Mb	128 Mb	128 Mb	256 Mb
Linux	128 Mb	128 Mb	128 Mb	256 Mb
Windows	128 Mb	128 Mb	128 Mb	256 Mb

OvpmQueryPluglet Configuration

When `OvpmQueryPluglet` receives a performance metric request, it extracts the needed information and then queries HP Performance Manager for the requested performance metrics. The returned performance metric data is packaged in a performance metric response, which is returned to the requester. Alternatively, if the pluglet receives a performance report, performance metric information or performance source information request,

OvpmQueryPluglet obtains the specified information from OVPM and returns it to the requester in a performance report, performance metric information or performance source information response, respectively.

To make this functionality work in your environment, you may need to make simple configuration changes to the example .config and .infiles provided for this pluglet, as shown below.

Example OvpmQueryPluglet .deploy, .config, and .in files have been provided in OVI_BASE_DIR/examples/OVI/configs/OVPM subdirectories. If you need to make changes, copy the files to the OVI_DATA_DIR/conf/OVI directory tree before editing.

The OvpmQueryPlugetExample.config file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<ovpmQueryPluglet xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation=
   "/opt/OV/schema/OVI/OvpmQueryPluglet.xsd">
      <!--
          The name of the OVPM server.
                    You may want to modify this:
      -->
      <ovpmServer>ovpmServer.x.y.z.com</ovpmServer>

      <!--
          The OVPM server HTTP port.
                    You may want to modify this:
      -->
      <ovpmServerPort>80</ovpmServerPort>

      <!--
          Switch on message trace logging
      -->
      <traceMessages>true</traceMessages>
      <!--
          Switch on message route logging
       -->
      <traceRoute>true</traceRoute>
</ovpmQueryPluglet>
```

The following table describes elements in the OvpmQueryPlugletExample.config file:

Element Name	Value
<ovpmServer> *OVPM server* </ovpmServer>	The name of the HP Performance Manager server. (Default: localhost)

Element Name	Value
<ovpmServerPort> *OVPM server HTTP port* </ovpmServerPort>	The HP Performance Manager server HTTP port. (Default: 8080) Note that by default, OVPM 4.04 for Windows uses port 80, and OVPM 5.x for Windows and UNIX uses 8080.
<traceMessages> *boolean* </traceMessages>	Controls whether an INFO level log message is created when a message arrives at this pluglet. Valid values are true or false. See traceMessages Settings for more details. (Default: true)
<traceRoute> *boolean* </traceRoute>	Controls whether route-tracing information containing the time of message arrival at this pluglet will be collected and, eventually, displayed. Valid values are true or false. See traceRoute Settings for more details. (Default: true)

If you use TestRequesterPluglet to generate a test request for OvpmQueryPluglet, you can modify these files as appropriate, to contain your desired message parameters:

- PerformanceMetricRequest.in: Contains a performance metric request message
- PerformanceReportRequest.in: Contains a report request message
- PerformanceMetricInfoRequest.in: Contains a performance metric information request message
- PerformanceSourceInfoRequest.in: Contains a performance source information request message

The appropriate request message is sent to OvpmQueryPluglet by OVPM's TestRequesterPluglet.

Sample Request Message Files

If you use TestRequesterPluglet to generate a test request for OvpmQueryPluglet, you can modify the sample request message files in this section to contain your desired message parameters.

Performance Metric Request

Here is the `PerformanceMetricRequest.in` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<performanceMetricRequest xmlns=
   "http://openview.hp.com/xmlns/ico/performance"
   xmlns:ovit=
   "http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!--
       System or probe type from which to acquire performance
       metrics.
    -->
      You may want to modify this:
   <entity>system1.xyz.com</entity>
   <entity>system2.xyz.com</entity>

   <!--
       Performance metric desired
    -->
       You may want to modify this:
   <metric>
      <group>GLOBAL</group>
      <name>GBL_CPU_TOTAL_UTIL</name>
   </metric>
   <metric>
      <group>GLOBAL</group>
      <name>GBL_CPU_TOTAL_TIME</name>
      <!-- Optional Filter
      <filter>GBL_CPU_TOTAL_TIME&gt;25</filter>
   -->
   </metric>

   <!--
       Data source (agent) that OV Performance Monitor should use
       to acquire performance metrics
   -->
     You may want to modify this:
  <dataSource>OVPA</dataSource>

  <!--
       Time frame (in ISO 8601 format) within which performance
       metrics are gathered
  -->
    You may want to modify this:
  <timeRange>
     <ovit:startTime>2003-02-25T01:00:00</ovit:startTime>
     <ovit:endTime>2003-02-25T01:10:59</ovit:endTime>
  </timeRange>

  <!--
      PM logon customer name and password
  -->
  You may want to modify and uncomment this:
  <!-- Optional customer name and password
  <user>
     <ovit:userName>SMITHW</ovit:userName>
     <ovit:password>WHTIMS</ovit:password>
  </user>
  -->  <!--
      Metric granularity is set to 600 seconds (10 minutes)
  -->
  You may want to modify and uncomment this:
```

```
  <!-- Optional summarization
     <summarization>600</summarization>
     <appSummarizationParameter>SUMFROMRAW
     </appSummarizationParameter>
  --></performanceMetricRequest>
```

The following table describes elements in `OvpmQueryPluglet`'s `PerformanceMetricRequest.in` file.

Element Name	Value
`<entity>system2.`*`xyz`*`.com</entity>`	Name of the system or probe type from which performance metric data will be requested. There can be multiple `<entity>` elements, each containing a single system name. This element is required only if the `<dataSource>` element (see `dataSource` element in table) is `OVPA`, `CODA`, or `DB`. In the case of `OVPA` and `CODA`, the entity is the system name; in the case of the `DB` data source, the entity is the database name as it appears in OVPM.

`<metric>`
`  <group>`*`GLOBAL`*`</group>`
`  <name>`*`GBL_CPU_TOTAL_UTIL`*`</name>`
`  <filter>`*`GBL_TOTALUTIL`*`>50</filter>`
`</metric>`
	Specification of performance metric data to retrieve. Multiple `<metric>`elements may exist in this file.
	There are *many* valid values for `<metric>`. To see a list of them, go to **http://ovweb.external.hp.com/lpe/doc_serv/** and select **operations for UNIX**, **7.x**, and **HP-UX**, then **Search**. Then from the list displayed, select **performance agent metrics** or **embedded performance component metric comparison** and click. This list contains default metrics that are supported. If you have installed one or more Smart Plug-Ins (SPIs), additional metrics are available. See SPI documentation for more details. Refer to the OVPM documentation for available OVIS metrics and filtering capabilities.

`<dataSource>`*`OVPA`*`</dataSource>`	HP Performance Manager data source to use for gathering performance metrics. Valid values are `OVPA`, `CODA`, `OVIS`, `REPORTER`, and `DB`. `OVPA` is the HP Performance Agent; `CODA` is the HP Operations Performance Agent; `OVIS` is HP Internet Services; `REPORTERis` the HP Reporter; and `DB` is the generic data source available in OVPM. Only one data source is supported per request.

`<timeRange>`
`  <ovit:startTime>`
`  `*`2003-02-25T01:00:00`*
`  </ovit:startTime>`
`  <ovit:endTime>`
`  `*`2003-02-25T01:10:59`*
`  </ovit:endTime>`
`</timeRange>`
	ISO 8601-formatted starting and ending times, which define the time frame for which performance metrics will be retrieved.

Element Name	Value
<user> <ovit:userName> *SMITHW* </ovit:userName> <ovit:password> *WHTIMS* </ovit:password></user>	Optional logon customer name and password. See HP Performance Manager documentation for additional details about logon customer.
<summarization>*600*</summarization>	The desired metric granularity in seconds. This element is optional. See HP Performance Manager documentation for additional details about summarization.
<appSummarizationParameter>*SUMFROMRAW*</appSummarizationParameter>	Optional element that contains a directive for OVPM-based data summarization to occur. It is an alternative to the data source-based summarization that occurs by default. The OVPM-based option may be useful for summarizing multi-instance data from external sources such as the HP WebLogic SPI. Most OVI users can ignore this option and use default summarization.

Performance Report Request

Here is the `PerformanceReportRequest.in` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<performanceReportRequest xmlns="http://openview.hp.com/xmlns/ico/
performance"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!--
      Name of the file containing the report.
      For example, for HP Performance Manager
      reports within the
      OV_INSTALLDIR/newconfig/VPI_GraphsInternet_Services.txt
      template file, you specify: Internet_Services.
           You may want to modify this:
   -->
   <reportTemplate>Internet_Services</reportTemplate>

   <!--
      Name of the report within the template file. For HP
      Performance Manager, the report name is specified within the
      template file after "GRAPH:", and only reports of
      "GRAPHTYPE: baseline" are currently supported.
      For example, for a report template of Internet_Services, you
      could specify one of the following:
      Trend Availability, Trend Response, or Trend Service Level.

      If multiple metrics and/or system names are
      specified in either the reportName template or in the
      performanceReportRequest, only the first baseline for a given
      metric/system pairing is returned from OVI.
           You may want to modify this:
   -->
   <reportName>Trend Response</reportName>
```

```
   <!--
      Application-specific name that will restrict data in the report to
      that which belongs to the specified group. For HP
      Performance Manager, this corresponds to the probe type.
      This element is optional.
           You may want to modify this:
   -->
   <group>DNS</group>
      You may want to modify and uncomment this:
   <!--
      Application specific string specifying a data filter that is applied
      to the report data. This usually results in restricting the values
      for the report to only those that meet the filter specification. An
      HP Performance Manager example would be: CUSTOMER_NAME=HP.

   <filter>CUSTOMER_NAME=HP</filter>
   -->   <!--
       PM logon customer name and password
   -->
   You may want to modify and uncomment this:
   <!-- Optional customer name and password
   <user>
      <ovit:userName>SMITHW</ovit:userName>
      <ovit:password>WHTIMS</ovit:password>
   </user>
  -->
</performanceReportRequest>
```

The following table describes elements in `OvpmQueryPluglet`'s `PerformanceReportRequest.in` file.

Element Name	Value

`<reportTemplate>`
Internet_Services
`</reportTemplate>`
	Name of the file containing the specification for the desired report. For example, for HP Performance Manager reports within the `OV_INSTALLDIR/newconfig/VPI_GraphsInternet_Services.txt`template file, you would specify: `Internet_Services`.

`<reportName>`
Trend Response
`</reportName>`
	Name of the desired report within the template file. For HP Performance Manager, the report name is specified within the template file after `GRAPH:`. (Only reports of `GRAPHTYPE: baseline`are currently supported by OVI.) For example, for a report template of `Internet_Services`, you would specify one of the following:
	`Trend Availability`, `Trend Response`, or `Trend Service Level`.
	If multiple `<metric>` and/or system name `<entity>` elements are supplied in a `reportName` template or `performanceReportRequest`, OVI returns only the first baseline for a given metric/entity pairing.

`<group>`*DNS*`</group>`	Application-specific name that restricts data in the report to what belongs to the specified group. For HP Performance Manager's OVIS-related templates, this corresponds to the probe type. This element is optional.

`<filter>`
`  `*CUSTOMER_NAME=HP*
`</filter>`
	Application-specific string specifying a data filter that is applied to the report data. This usually results in restricting the values for the report to only those that meet the filter specification. This element is optional.

`<user>`
`  <ovit:userName>`
`  `*SMITHW*
`  </ovit:userName>`
`  <ovit:password>`
`  `*WHTIMS*
`  </ovit:password>`
`</user>`
	Optional logon customer name and password. See HP Performance Manager documentation for additional details about logon customer.

Performance Metric Info Request

Here is the `PerformanceMetricInfoRequest.in` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<performanceMetricInfoRequest xmlns="http://openview.hp.com/xmlns/ico/
performance"
   xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

   <!--
      Data source (agent) that OV Performance Monitor should use to acquire
      performance metric information.
   -->
      You may want to modify this:
   <dataSource>OVPA</dataSource>

   <!--
      System/probe that OV Performance Monitor should use to acquire
      performance metric information. This is required for OVPA and CODA
      data sources, but is optional for all others.
   -->
     You may want to modify this:
   <ovit:entity>
      <ovit:reference>system1.xyz.com</ovit:reference>
   </ovit:entity>
   <!--
      PM logon customer name and password
   -->
   You may want to modify and uncomment this:
   <!-- Optional customer name and password
   <user>
      <ovit:userName>SMITHW</ovit:userName>
      <ovit:password>WHTIMS</ovit:password>
   </user>
   -->
</performanceMetricInfoRequest>
```

The following table describes elements in OvpmQueryPluglet's PerformanceMetricInfoRequest.in file.

Element Name	Value
<dataSource>*OVPA*</dataSource>	HP Performance Manager data source to use for gathering performance metrics. Valid values are OVPA, CODA, OVIS, REPORTER, and DB. OVPA is the HP Performance Agent; CODA is the HP Operations Performance Agent; OVIS is HP Internet Services; REPORTER is the HP Reporter; and DB is the generic data source available in OVPM. Only one data source is supported per request.
<ovit:entity> <ovit:reference> *system1.xyz.com* </ovit:reference> </ovit:entity>	Name of the system or probe that HP Performance Manager will use for gathering performance metric information. This element is required only if <dataSource> is OVPA, CODA, or DB. In the case of OVPA and **CODA**, the entity is the system name; in the case of the DB data source, the entity is the database name as it appears in OVPM.
<user> <ovit:userName> *SMITHW* </ovit:userName> <ovit:password> *WHTIMS* </ovit:password> </user>	Optional logon customer name and password. See HP Performance Manager documentation for additional details about logon customer.

Performance Source Info Request

Here is the PerformanceSourceInfoRequest.in file:

```
<?xml version="1.0" encoding="UTF-8"?>
<performanceSourceInfoRequest xmlns="http://openview.hp.com/xmlns/ico/
performance"
    xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">
</performanceSourceInfoRequest>
```

The following table describes elements in OvpmQueryPluglet's PerformanceSourceInfoRequest.in file.

Namespaces and Message Formats

The namespaces used in the performance metric, report, metric information and source information request/response XML messages for HP Performance Manager are xmlns=http://openview.hp.com/xmlns/ico/performance and xmlns=http://openview.hp.com/xmlns/ico/types.

These namespaces are required for <performanceMetricRequest>, <performanceMetricResponse>, and <performanceMetricException> messages (for processing of performance metrics). They are also mandatory for <performanceReportRequest>, <performanceReportResponse>, and <performanceReportException> messages (for processing of performance reports).

Similarly, they are required for `<performanceMetricInfoRequest>`, `<performanceMetricInfoResponse>`, `<performanceSourceInfoRequest>`, and `<performanceSourceInfoResponse>` messages.

The contents of performance (metric, report, metric information and source information) request XML messages are only partially validated. A full validation is not performed for performance reasons. Instead, specific parts of each performance request are checked.

Performance Metric Messages

Requests received by `OvpmQueryPluglet` are reformatted, and the response is returned via the configured transport mechanism. The performance metric request and response message schemas and example XML can be found in OVI_BASE_DIR/examples/OVI/schemas/performance. (This directory is referred to as SCHEMAS_DIR in the table.)

There are two types of valid responses to this performance metric request message:

- performance metric response
- performance metric exception (error message). See Performance Metric Exception Handling for details.

XML Message Description	**Location**

Performance metric request message	`SCHEMAS_DIR/performanceMetricRequest.xml`
	`SCHEMAS_DIR/performanceMetricRequestWithFilter.xml`
	`SCHEMAS_DIR/performanceMetricRequestWithSummarization.xml`

Performance metric request schema	`SCHEMAS_DIR/performanceMetricRequest.xsd`
Performance metric response message	`SCHEMAS_DIR/performanceMetricResponse.xml`
Performance metric response schema	`SCHEMAS_DIR/performanceMetricResponse.xsd`

Performance metric exception response messages	`SCHEMAS_DIR/performanceMetricOperationException.xml`
	`SCHEMAS_DIR/performanceMetricInvalidXmlException.xml`
	`SCHEMAS_DIR/performanceMetricTimeOutException.xml`

Performance metric exceptions embedded within a performance metric response	`SCHEMAS_DIR/performanceMetricDataSourceNotAvailableException.xml` `SCHEMAS_DIR/performanceMetricMetricDoesNotExistException.xml`
Performance metric exception response schema	`SCHEMAS_DIR/performanceMetricException.xsd`

Performance Report Messages

Requests received by `OvpmQueryPluglet` are reformatted, and the response is returned via the configured transport mechanism. The performance report request and response message schemas and example XML can be found in `OVI_BASE_DIR/examples/OVI/schemas/performance`. (This directory is referred to as `SCHEMAS_DIR` in the table.)

There are two types of valid responses to this performance report request message:

- performance report response
- performance report exception (error message). See Performance Report Exception Handling for details.

XML Message Description	Location
Performance report request message	`SCHEMAS_DIR/performanceReportRequest.xml`
Performance report request schema	`SCHEMAS_DIR/performanceReportRequest.xsd`
Performance report response message	`SCHEMAS_DIR/performanceReportResponse.xml`
Performance report response schema	`SCHEMAS_DIR/performanceReportResponse.xsd`
Performance report exception response messages	`SCHEMAS_DIR/performanceReportException.xml` `SCHEMAS_DIR/performanceReportUnsupportedReportTypeException.xml`
Performance report exception response schema	`SCHEMAS_DIR/performanceReportException.xsd`

Performance Metric Information Messages

Requests received by `OvpmQueryPluglet` are reformatted, and the response is returned via the configured transport mechanism. The performance metric information request and response message schemas and example XML can be found in `OVI_BASE_DIR/examples/OVI/schemas/performance`. (This directory is referred to as `SCHEMAS_DIR` in the table.)

There are two types of valid responses to this performance metric information request message:

- performance metric information response

- performance metric information exception (error message). See Performance Metric Information Exception Handling for details.

XML Message Description	Location
Performance metric information request message	SCHEMAS_DIR/performanceMetricInfoRequest.xml
Performance metric information request schema	SCHEMAS_DIR/performanceMetricInfoRequest.xsd
Performance metric information response message	SCHEMAS_DIR/performanceMetricInfoResponse.xml
Performance metric information response schema	SCHEMAS_DIR/performanceMetricInfoResponse.xsd
Performance metric information exception response messages	SCHEMAS_DIR/ performanceMetricInfoDataSourceNotAvailableException.xml SCHEMAS_DIR/performanceMetricInfoInvalidXmlException.xml SCHEMAS_DIR/ performanceMetricInfoNoDataAvailableException.xml SCHEMAS_DIR/performanceMetricInfoOperationException.xml SCHEMAS_DIR/performanceMetricInfoTimeOutException.xml
Performance metric information exception response schema	SCHEMAS_DIR/performanceMetricInfoException.xsd

Performance Source Information Messages

Requests received by OvpmQueryPluglet are reformatted, and the response is returned via the configured transport mechanism. The performance source information request and response message schemas and example XML can be found in OVI_BASE_DIR/examples/OVI/schemas/performance. (This directory is referred to as SCHEMAS_DIR in the table.)

There are two types of valid responses to this performance source information request message:

- performance source information response
- performance source information exception (error message). See Performance Source Information Exception Handling for details.

XML Message Description	Location
Performance source information request message	SCHEMAS_DIR/performanceSourceInfoRequest.xml
Performance source information request schema	SCHEMAS_DIR/performanceSourceInfoRequest.xsd
Performance source information response message	SCHEMAS_DIR/performanceSourceInfoResponse.xml
Performance source information response schema	SCHEMAS_DIR/performanceSourceInfoResponse.xsd
Performance source information exception response messages	SCHEMAS_DIR/ performanceSourceInfoInvalidXmlException.xml SCHEMAS_DIR/ performanceSourceInfoNoDataAvailableException.xml SCHEMAS_DIR/ performanceSourceInfoOperationException.xml SCHEMAS_DIR/ performanceSourceInfoTimeOutException.xml
Performance source information exception response schema	SCHEMAS_DIR/performanceSourceInfoException.xsd

Exception Handling

The topics in this section describe the exceptions that are generated when problems occur and give examples of how to handle these exceptions.

Performance Metric Exception Handling

Exceptions are generated when a problem occurs while processing a performance metric request. These exceptions occur when the `OvpmQueryPlugletis` unable to return all the performance metric data requested. These exceptions can be categorized into the two groups shown in the table.

Exceptions with the Ability to Return Partial Valid Metric Data

Exception Description	Example

Metric Does Not Exist
 	
This is generated when all or some metrics requested do not exist for all or some data sources. In the event that all metrics do not exist for all sources, then no data is provided in the performance metric response.
	`<?xml version="1.0" encoding="UTF-8"?>`
	
	`<performanceMetricResponse xmlns=`
	`  "http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <exceptions>`
	`  <performanceMetricException version="1.0">`
	`  <metricDoesNotExistException>`
	`  <ovit:message>`
	`  The metric does not exist.`
	`  </ovit:message>`
	`  <entity>{entity with missing metric}</entity>`
	`  <group>{metric group}</group>`
	`  <metric>{metric name}</metric>`
	`  </metricDoesNotExistException>`
	`  </performanceMetricException>`
	`  </exceptions>`
	
	`</performanceMetricResponse>`

Data Source Not Available
 	
This occurs when the data source (for example, HP Reporter database or HP Performance Agent) cannot be examined for metric values. There will be no corrective action taken, such as trying to start the database process.
	`<?xml version="1.0" encoding="UTF-8"?>`
	
	`<performanceMetricResponse`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <exceptions>`
	`  <performanceMetricException version="1.0">`
	`  <dataSourceNotAvailableException>`
	`  <ovit:message>`
	`  The data source is not available or unreachable.`
	`  </ovit:message>`
	`  <dataSource>{data source name}</dataSource>`
	`  <entity>`
	`  {entity where data source was unavailable}`
	`  </entity>`
	`  </dataSourceNotAvailableException>`
	`  </performanceMetricException>`
	`  </exceptions>`
	
	`</performanceMetricResponse>`

Exceptions Without the Ability to Return Valid Metric Data

Exception Description	Example

Time Out
 	
This is generated when the connection between `OvpmQueryPluglet`and HP Performance Manager times out.
	`<?xml version="1.0" encoding="UTF-8"?>`
	
	`<performanceMetricException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <ovit:timeOutException>`
	`  <ovit:message>`
	`  The response did not return within the specified timeout limit.`
	`  </ovit:message>`
	`  </ovit:timeOutException>`
	`</performanceMetricException>`

Invalid XML
 	
This occurs when the performance metric request did not contain all required request elements or contained invalid request values.
	`<?xml version="1.0" encoding="UTF-8"?>`
	
	`<performanceMetricException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <ovit:invalidXmlException>`
	`  <ovit:message>`
	`  The dataSource provided is invalid.`
	`  </ovit:message>`
	`  </ovit:invalidXmlException>`
	`</performanceMetricException>`

Operation Exception
 	
This is returned when something goes wrong while trying to communicate with HP Performance Manager. The cause could be a failure in the connection to HP Performance Manager.
	`<?xml version="1.0" encoding="UTF-8"?>`
	
	`<performanceMetricException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <ovit:operationException>`
	`  <ovit:message>`
	`  The operation encountered an unrecoverable error.`
	`  </ovit:message>`
	`  </ovit:operationException>`
	`</performanceMetricException>`

Performance Report Exception Handling

Exceptions are generated when a problem occurs while processing a performance report request.

Exception Description	Example

Unsupported Report Type
 	
This is generated when the `reportName` and `reportTemplatepairing` specified in the performance report request generates a report response that is not supported by OVI. Currently OVI only supports baseline reports.
	`<?xml version="1.0" encoding="UTF-8"?>`
	`  <performanceReportException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  version="1.0"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types">`
	
	`  <unsupportedReportTypeException>`
	`  <ovit:message>`
	`  The reportTemplate/reportName pairing specified in the request translated to a report type that was not supported by OVI. OVI currently supports only the baseline report type.`
	`  </ovit:message>`
	`  </unsupportedReportTypeException>`
	`</performanceReportException>`

Performance Metric Information Exception Handling

Exceptions are generated when a problem occurs while processing a performance metric information request. This means that `OvpmQueryPluglet` is unable to return the requested information.

Exception Description	Example

Data Source Not Available
 	
This occurs when the data source (for example, HP Performance Agent) cannot be examined for metric information. There will be no corrective action taken, such as trying to start the database process.
	`<?xml version="1.0" encoding="UTF-8"?>`
	
	`<performanceMetricInfoException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <dataSourceNotAvailableException>`
	`  <ovit:message>`
	`  The designated data source is unavailable.`
	`  </ovit:message>`
	`  <dataSource>OVPA</dataSource>`
	`  </dataSourceNotAvailableException>`
	`</performanceMetricInfoException>`

Invalid XML
 	
This occurs when the performance metric information request did not contain all required elements or contained invalid request values.
	`<?xml version="1.0" encoding="UTF-8"?>`
	
	`<performanceMetricInfoException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <ovit:invalidXmlException>`
	`  <ovit:message>`
	`  Exception text.`
	`  </ovit:message>`
	`  </ovit:invalidXmlException>`
	`</performanceMetricInfoException>`

Exception Description	Example
No Data Available This is generated when data is not available from HP Performance Manager for the designated data source and entity.	<?xml version="1.0" encoding="UTF-8"?> <performanceMetricInfoException xmlns="http://openview.hp.com/xmlns/ico/performance" xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0"> <ovit:noDataAvailableException> <ovit:message> No data is available for the designated data source and entity. </ovit:message> </ovit:noDataAvailableException </performanceMetricInfoException>
Operation Exception This is returned when something goes wrong while trying to communicate with HP Performance Manager. The cause could be a failure in the connection to OVPM.	<?xml version="1.0" encoding="UTF-8"?> <performanceMetricInfoException xmlns="http://openview.hp.com/xmlns/ico/performance" xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0"> <ovit:operationException> <ovit:message> An unrecoverable error has occurred. </ovit:message> </ovit:operationException> </performanceMetricInfoException>
Time Out This is generated when the connection between `OvpmQueryPluglet` and HP Performance Manager times out.	<?xml version="1.0" encoding="UTF-8"?> <performanceMetricInfoException xmlns="http://openview.hp.com/xmlns/ico/performance" xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0"> <ovit:timeOutException> <ovit:message> A response did not arrive within the time limit. </ovit:message> </ovit:timeOutException> </performanceMetricInfoException>

Performance Source Information Exception Handling

Exceptions are generated when a problem occurs while processing a performance source information request, meaning that `OvpmQueryPluglet` cannot return the requested information.

Exception Description	Example

Invalid XML
 	
This occurs when the performance source information request contained invalid XML.
	`<?xml version="1.0" encoding="UTF-8"?>`
	
	`<performanceSourceInfoException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <ovit:invalidXmlException>`
	`  <ovit:message>`
	`  XML in the request was invalid.`
	`  </ovit:message>`
	`  </ovit:invalidXmlException>`
	`</performanceSourceInfoException>`

Exception Description	Example
No Data Available This is generated when data is not available from HP Performance Manager to satisfy the request.	<?xml version="1.0" encoding="UTF-8"?> <performanceSourceInfoException xmlns="http://openview.hp.com/xmlns/ico/performance" xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0"> <ovit:noDataAvailableException> <ovit:message> No data was returned from the request. </ovit:message> </ovit:noDataAvailableException </performanceSourceInfoException>
Operation Exception This is returned when something goes wrong while trying to communicate with HP Performance Manager. The cause could be a failure in the connection to OVPM.	<?xml version="1.0" encoding="UTF-8"?> <performanceSourceInfoException xmlns="http://openview.hp.com/xmlns/ico/performance" xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0"> <ovit:operationException> <ovit:message> An unrecoverable error occurred. </ovit:message> </ovit:operationException> </performanceSourceInfoException>
Time Out This is generated when the connection between `OvpmQueryPluglet` and HP Performance Manager times out.	<?xml version="1.0" encoding="UTF-8"?> <performanceSourceInfoException xmlns="http://openview.hp.com/xmlns/ico/performance" xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0"> <ovit:timeOutException> <ovit:message> The response did not arrive within the specified time limit </ovit:message> </ovit:timeOutException> </performanceSourceInfoException>

10 Integrating with HP Transaction Analyzer

The OvtaQueryPluglet allows you to retrieve data from HP Transaction Analyzer. This can be done by sending a request for transaction data matching designated criteria to OvtaQueryPluglet. When a transaction metric request (transactionMetricRequest) arrives, the OvtaQueryPluglet queries the OVTA Measurement Server for transactions that meet the specified criteria and returns them to the requester in the form of a transactionMetricResponse. If an error occurs, a transactionMetricException is returned. Because OvtaQueryPluglet retrieves transaction metric data from HP Transaction Analyzer via HTTP, OvtaQueryPluglet and OVTA do not have to run on the same system.

HP Transaction Analyzer must be installed before using OVI OvtaQueryPluglet.

OvtaQueryPluglet Configuration

To make this functionality work in your environment, you may need to make simple configuration changes to the example .config and .in files provided for this pluglet, as shown below.

Example OvtaQueryPluglet .deploy, .config, and .in files have been provided in the OVI_BASE_DIR/examples/OVI/configs/OVTA/Query directory. If you need to make changes, copy them to the OVI_DATA_DIR/conf/OVI directory tree before editing.

The OvtaQueryPlugetExample.config file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<ovtaQueryPluglet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="/opt/OV/schema/OVI/OvtaQueryPluglet.xsd">
      <!--
          The name of the OVTA server.
                    You may want to modify this:
      -->
      <ovtaServer>ovtaServer.x.y.z.com</ovtaServer>

      <!--
          The OVTA server HTTP port.
                   You may want to modify this:
     -->
      <ovtaServerPort>7555</ovtaServerPort>

      <!--
          Switch on message trace logging.
       -->
      <traceMessages>true</traceMessages>
      <!--
          Switch on message route logging.
      -->
      <traceRoute>true</traceRoute>
</ovtaQueryPluglet>
```

The following table describes elements in the `OvtaQueryPlugletExample.config` file:

Element Name	Value

`<ovtaServer>`
OVTA server
`</ovtaServer>`
	Name of the HP Transaction Analyzer Measurement Server. (Default: `localhost`)

`<ovtaServerPort>`
OVTA server HTTP port
`</ovtaServerPort>`
	HP Transaction Analyzer Measurement Server HTTP port. (Default: `7555`)

`<traceMessages>`
boolean
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
boolean
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet will be collected and, eventually, displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

If you use `TestRequesterPluglet` to generate a test request for `OvtaQueryPluglet`, you can modify `TestRequesterPlugetExample.in` to contain your desired message parameters. The `TestRequesterPlugetExample.in` file contains the transaction metric request message that is sent to `OvtaQueryPluglet` by OVTA's `TestRequesterPluglet`. This file's contents appear below, and elements that you may want to change are indicated.

```
<?xml version="1.0" encoding="UTF-8"?>
<transactionMetricRequest xmlns="http://openview.hp.com/xmlns/ico/
performance"
 xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

     <!--
        Transaction metric filtering criteria
     -->
          You will want to modify this:
     <transactionMetric>
        <type>HTTP</type>
        <group>systemx.com:8080</group>
        <name>/test/url.html</name>
     </transactionMetric>

     <!--
        Time frame (in ISO 8601 format) within which transaction metrics
        are gathered
     -->
          You will want to modify this:
     <timeRange>
         <ovit:startTime>2003-12-1T09:30:47-05:00</ovit:startTime>
         <ovit:endTime>2003-12-2T09:30:47-05:00</ovit:endTime>
     </timeRange>
</transactionMetricRequest>
```

The following table describes elements in the `TestRequesterPlugletExample.in` file.

Element Name	Value

`<transactionMetric>`
`  <type>`*HTTP*`</type>`
`  <group>systemx.com:8080</group>`
`  <name>/test/url.html</name>`
`</transactionMetric>`
	Specification of transaction metric data to retrieve. (This is the same data that can be seen in an OVTA Console navigation tree.)
	`<type>` contains the transaction kind.
	`<group>` contains the transaction group.
	`<name>` contains the transaction name.

`<timeRange>`
`  <ovit:startTime>`
`  2003-12-1T09:30:47-05:00`
`  </ovit:startTime>`
`  <ovit:endTime>`
`  2003-12-2T09:30:47-05:00`
`  </ovit:endTime>`
`</timeRange>`
	ISO 8601-formatted starting and ending times, which define the time frame for which transaction metrics are retrieved.

Namespaces

The namespaces used in the transaction metric request/response XML messages for HP Transaction Analyzer are `xmlns=http://openview.hp.com/xmlns/ico/performance` and `xmlns=http://openview.hp.com/xmlns/ico/types`.

These namespaces are required for `<transactionMetricRequest>`, `<transactionMetricResponse>`, and `<transactionMetricException>` messages.

The transaction metric request XML message is only partially validated. A full validation is not performed for performance reasons. Only specific parts of each transaction metric request are checked.

Transaction Metric Request/Response Message Format

Requests received by `OvtaQueryPluglet` are reformatted, and the response is returned via the configured transport mechanism. The transaction metric request and response message schemas and example XML can be found in `the OVI_BASE_DIR/examples/OVI/schemas/performance` directory, which is referred to as `SCHEMAS_DIR`.

There are two possible types of responses to this transaction metric request message:

- transaction metric response

- transaction metric exception (error message). See Transaction Metric Exception Handling for details.

XML Message Description	Location
Transaction metric request message	SCHEMAS_DIR/transactionMetricRequest.xml
Transaction metric request schema	SCHEMAS_DIR/transactionMetricRequest.xsd
Transaction metric response message	SCHEMAS_DIR/transactionMetricResponse.xml
Transaction metric response message schema	SCHEMAS_DIR/transactionMetricResponse.xsd
Transaction metric exception response messages	SCHEMAS_DIR/ transactionDataNotAvailableException.xml SCHEMAS_DIR/transactionDoesNotExistException.xml
Transaction metric exception response schema	SCHEMAS_DIR/transactionMetricException.xsd

Transaction Metric Exception Handling

Exceptions are generated when a problem occurs while processing a transaction metric request and `OvtaQueryPluglet` is unable to return the requested data. This table gives some examples.

Exception Description	Example
Transaction Does Not Exist This is generated when the requested metric does not exist.	<?xml version="1.0" encoding="UTF-8"?> <transactionMetricException xmlns= "http://openview.hp.com/xmlns/ico/ performance" xmlns:ovit="http://openview.hp.com/xmlns/ico/ types" version="1.0"> <transactionDoesNotExistException> <ovit:message> No transaction matched the specified criteria. </ovit:message> <type>BOZO</type> <group>mysystem.mycompany.com:82</group> <name>/petstore/ enter_order_information.screen</name> </transactionDoesNotExistException> </transactionMetricException>
Transaction Data Not Available This occurs when data is not available for the requested metric in the requested time range.	<?xml version="1.0" encoding="UTF-8"?> <transactionMetricException xmlns="http://openview.hp.com/xmlns/ico/ performance" xmlns:ovit="http://openview.hp.com/xmlns/ico/ types" version="1.0"> <transactionDataNotAvailableException> <ovit:message> Transaction data was not available for the specified criteria. </ovit:message> <type>HTTP</type> <group>mysystem.mycompany.com:82</group> <name> /petstore/enter_order_information.screen </name> </transactionDataNotAvailableException> </transactionMetricException>

Exception Description	Example

Time Out
 	
This is generated when the connection between `OvtaQueryPluglet` and HP Transaction Analyzer times out.
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<transactionMetricException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <ovit:timeOutException>`
	`  <ovit:message>`
	`  The response did not return within the specified`
	`  timeout limit.`
	`  </ovit:message>`
	`  </ovit:timeOutException>`
	`</transactionMetricException>`

Invalid XML
 	
This occurs when the transaction metric request does not contain all required request elements or contains invalid XML.
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<transactionMetricException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <ovit:invalidXmlException>`
	`  <ovit:message>`
	`  The dataSource provided is invalid.`
	`  </ovit:message>`
	`  </ovit:invalidXmlException>`
	`  </transactionMetricException>`

Operation Exception
 	
This is returned when something goes wrong while trying to communicate with HP Transaction Analyzer. The cause could be a failure in OVI's connection to Transaction Analyzer.
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<transactionMetricException`
	`  xmlns="http://openview.hp.com/xmlns/ico/performance"`
	`  xmlns:ovit="http://openview.hp.com/xmlns/ico/types"`
	`  version="1.0">`
	
	`  <ovit:operationException>`
	`  <ovit:message>`
	`  The operation encountered an unrecoverable error.`
	`  </ovit:message>`
	`  </ovit:operationException></transactionMetricException>`

11 Integrating with HP Network Node Manager

The Network Node Manager (NNM) pluglets enable interaction with HP Network Node Manager. OVI must be installed on the same system as NNM.

OVI supplies four pluglets that provide an integration with HP Network Node Manager:

- NnmInventoryPluglet
- NnmSnmpCollectPluglet
- NnmInventorySnmpEventPluglet
- NnmEventReadPluglet

HP Network Node Manager must be installed on the same system as `NnmInventoryPluglet`, `NnmSnmpCollectPluglet` or `NnmInventorySnmpEventPluglet`. The `NnmEventReadPluglet` requires that HP Network Node Manager 7.5 be present on the same system.

All required NNM patches must be loaded. If this is not done, erratic behavior may occur when using OVI pluglets to retrieve Network Node Manager data. When running NNM 6.4.1, installation of `PHSS_29795` (or its successor) for HP-UX, `PSOV_03292` (or its successor) for Solaris, or `NNM_01008` (or its successor) for Windows 2000 systems is vital. This NNM patch fixes a problem that causes the JVM in which OVI is running to abort when an attempt is made to retrieve NNM node interface information. Patches may be obtained from **http://support.openview.hp.com/selfsolve/patches**.

NnmInventoryPluglet

The `NnmInventoryPluglet` allows you to retrieve network inventory information from NNM. This can be done by sending an inventory request to the `NnmInventoryPluglet`. Due to the potentially large volume of inventory data that may be returned, the request can optionally contain a starting index (`startIndex`) and a request size (`requestSize`) that define inventory data response chunks.

When an inventory request (`networkInventoryRequest`) arrives, the `NnmInventoryPluglet` gathers the inventory information according to the designated `startIndex` and `requestSize` and returns appropriate data to the requester in the form of a `networkInventoryResponse`. In addition to inventory data, two additional pieces of information are returned in the response: total entities and response size. The total entities element holds the total number of entities available within the NNM inventory database; it enables clients to know when they have received all available data chunks. The response size element contains the number of inventory entities that are included within the current response; it can also help to indicate the last bunch of data, which might be smaller than the request size. If an error occurs, a `networkInventoryException` is returned as the response.

If the number of entities you want to retrieve is more than `100`, multiple `networkInventoryRequests` must be used to retrieve all desired information from NNM. Since NNM's inventory database can be updated while these multiple requests are being processed, an inconsistent data snapshot could be obtained and returned by OVI. To prevent this from happening, stop all NNM discovery processes prior to sending the first `networkInventoryRequest` in a series. Restart these processes after OVI returns NNM's response to the last request.

Data filtering criteria can be specified in inventory requests. Filtering may be based on overlapping address domain, NNM ID, IP address, or DNS name attributes. When filtering is used, the total entities element in the `networkInventoryResponse` contains the number of nodes that matched the specified filter.

See NnmInventoryPluglet Configuration_ for more information.

NnmInventoryPluglet Configuration

Example `NnmInventoryPluglet .deploy`, `.config` and `.in` files have been provided in the `OVI_BASE_DIR/examples/OVI/configs/NNM/Inventory` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory tree before editing. The `NnmInventoryPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<nnmInventoryPluglet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="/opt/OV/schema/OVI/
NnmInventoryPluglet.xsd">

    <!--
        Switch on message trace logging
    -->
    <traceMessages>true</traceMessages>
    <!--
        Switch on message route logging
    -->
    <traceRoute>true</traceRoute>
</nnmInventoryPluglet>
```

The following table describes elements in the `NnmInventoryPluglet.config` file:

Element Name	Value

`<traceMessages>`
boolean
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
boolean
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet will be collected and, eventually, displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

If you use TestRequesterPluglet to generate a test request for NnmInventoryPluglet, you can modify TestRequesterPlugetExample.in to contain your desired starting index, request size, and filtering criteria. The TestRequesterPlugetExample.in file contains the network inventory request message that is sent to the NnmInventoryPluglet. This file's contents appear below, and elements that you may want to change are indicated.

NNM ET inventory data can be quite large, so requesting many inventory entities may cause you to run out of memory. If this problem occurs, reduce the number of inventory entities requested by changing the request size parameter.

```
<?xml version="1.0" encoding="UTF-8"?>
<networkInventoryRequest xmlns="http://openview.hp.com/xmlns/ico/entity"
    xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">

        You may want to modify this:
    <startIndex>1</startIndex>

        You may want to modify this:
    <requestSize>15</requestSize>

        You may want to modify this:
    <!--
       Use the following syntax to filter the inventory request:
          filterName=filterValue.
          Only one filter name can be specified per request. Valid
          NMN filter names are:
             overlappingAddrDomainId - overlapping address domain ID
             reference - unique reference ID of the entity as stored
                         by the application
             entityDnsName - DNS name of the entity to be filtered
             address - IP address of the entity to be filtered
    -->
    <!--
       <inventoryFilter>overlappingAddrDomainId=3</inventoryFilter>
    -->


</networkInventoryRequest>
```

The following table describes elements in NnmInventoryPluglet's TestRequesterPlugletExample.in file:

Element Name	Value
<startIndex>1</startIndex>	The starting index for the network inventory request. By advancing the start index, a series of requests can progress through the entire data set. (Default = 1).
<requestSize>15</requestSize>	The number of network inventory entities requested. Both the default and maximum request size are 100.
<inventoryFilter> overlappingAddrDomainId=3 </inventoryFilter>	The network inventory request filter. Its format is: FilterName=FilterValue. Valid FilterNames are: overlappingAddrDomainId reference entityDnsName address

Network Inventory Namespaces

The namespaces used in the network inventory request/response XML messages for HP Network Node Manager are `xmlns=http://openview.hp.com/xmlns/ico/entity` and `xmlns=http://openview.hp.com/xmlns/ico/types`. These namespaces are required for `<networkInventoryRequest>`, `<networkInventoryResponse>`, and `<networkInventoryException>` messages.

The contents of the network inventory request XML message is only partially validated. A full validation is not performed for performance reasons. Instead, specific parts of each network inventory request are checked.

Network Inventory Request/Response Message Format

The network inventory request and response message schemas and example XML can be found in `OVI_BASE_DIR/examples/OVI/schemas/entity`. (This directory is referred to as `SCHEMAS_DIR` below.)

XML Message Description	**Location**
Network inventory request message	`SCHEMAS_DIR/networkInventoryRequest.xml` `SCHEMAS_DIR/filteredNetworkInventoryRequest.xml`
Network inventory request schema	`SCHEMAS_DIR/networkInventoryRequest.xsd`
Network inventory response message	`SCHEMAS_DIR/networkInventoryResponse.xml`
Network inventory response schema	`SCHEMAS_DIR/networkInventoryResponse.xsd`

Network inventory exception response message	`SCHEMAS_DIR/operationException.xml`
	`SCHEMAS_DIR/invalidXmlException.xml`
	`SCHEMAS_DIR/entityNotFoundException.xml`

Network inventory exception response schema	`SCHEMAS_DIR/networkInventoryException.xsd`

NnmSnmpCollectPluglet

The `NnmSnmpCollectPluglet` allows you to obtain data directly from NNM SNMP collection files. Because of this, the pluglet must run in an instance of OVI residing on the same machine as the NNM collection station.

When a `PerformanceMetricRequest` (containing system names, interface names or numbers, a time range, and the NNM collected metrics that are of interest) is sent, `NnmSnmpCollectPluglet` retrieves the requested data and returns a `PerformanceMetricResponse`. `NnmSnmpCollectPluglet` searches all NNM SNMP collect files for those that contain the metrics specified in the request during the specified time range and for the specified systems and interfaces. If any SNMP collect files are found containing

metrics that match these criteria, the contents of the file are sent back in the response. If no matching files are found or if an error occurs, a `performanceMetricException` is returned either by itself or as part of the performance metric response.

The NnmSnmpCollectPluglet Configuration topic has additional information about this pluglet.

NnmSnmpCollectPluglet Configuration

Example `NnmSnmpCollectPluglet .deploy`, `.config`, and `.in` files have been provided in the `OVI_BASE_DIR/examples/OVI/configs/NNM/SnmpCollect` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory tree before editing. The `NnmSnmpCollectPluglet.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<nnmSnmpCollectPluglet xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="/opt/OV/schema/OVI/
NnmSnmpCollectPluglet.xsd">
      <!--
          The SNMP collect data directory.
       -->
      <snmpCollectDir>/var/opt/OV/share/databases/snmpCollect/</
snmpCollectDir>
      <!--
          Switch on message trace logging
       -->
      <traceMessages>true</traceMessages>
      <!--
          Switch on message route logging
       -->
      <traceRoute>true</traceRoute>
</nnmSnmpCollectPluglet>
```

The following table describes elements in the `NnmSnmpCollectPluglet.config` file:

Element Name	Value

`<snmpCollectDir>`
NNM Collect Dir
`</snmpCollectDir>`
	The local location of the NNM SNMP collect data files.

`<traceMessages>`
boolean
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
boolean
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet will be collected and, eventually, displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

If you use `TestRequesterPluglet` to generate a test request for `NnmSnmpCollectPluglet`, you can modify `TestRequesterPlugetExample.in` to contain your desired message parameters. The `TestRequesterPlugetExample.in` file contains the performance metric request message that is sent to `NnmSnmpCollectPluglet` by `TestRequesterPluglet`. This file's contents appear below, and elements that you may want to change are indicated.

```
<?xml version="1.0" encoding="UTF-8"?>
<performanceMetricRequest xmlns="http://openview.hp.com/xmlns/ico/
performance"
xmlns:ovit="http://openview.hp.com/xmlns/ico/types" version="1.0">
    <!--
        System(s) from which to acquire SNMP metrics
    -->
        You may want to modify this:
    <snmpEntity>
        <entityId>system1.xyz.com</entityId>
        <snmpId>
            <snmpInstance>1</snmpInstance>
        </snmpId>
        <snmpId>
            <interfaceName>lan2</interfaceName>
        </snmpId>
    </snmpEntity>
    <snmpEntity>
        <entityId>system2.xyz.com</entityId>
        <snmpId>
            <snmpInstance>1</snmpInstance>
        </snmpId>
    </snmpEntity>

    <!--
        Metrics desired
    -->
        You may want to modify this:
    <metric>
        <name>If%util</name>
    </metric>
    <metric>
        <name>IfHCInPktPctErrors</name>
    </metric>
    <metric>
        <name>IfOutPktPctErrors</name>
    </metric>

    <!--
        Data source (must be SNMP)
    -->
        DO NOT change this:
    <dataSource>SNMP</dataSource>

    <!--
        Time frame (in ISO 8601 format) within which metrics
        are gathered
    -->
        You may want to modify this:
    <timeRange>
        <ovit:startTime>2003-02-25T01:00:00</ovit:startTime>
        <ovit:endTime>2003-02-25T01:10:59</ovit:endTime>
    </timeRange>

    <!--
        Metric granularity is set to 600 seconds (10 minutes)
    -->
        You may want to modify and uncomment this:
    <!--
        Optional summarization
    <summarization>600</summarization>
    -->
</performanceMetricRequest>
```

Most elements of the performance metric request sent to the NnmSnmpCollectPluglet in the TestRequesterPlugletExample.in file are the same as in the Performance Metric Request sent to the OvpmQueryPluglet. Differences are described in the table below. Metric filtering is not supported by the NnmSnmpCollectPluglet.

Element Name	Value
<snmpEntity> <entityId>system1.xyz.com</entityId> <snmpId> <snmpInstance>1</snmpInstance> </snmpId> <snmpId> <interfaceName>lan2</interfaceName> </snmpId> </snmpEntity>	Unlike the performance metric request sent to OvpmQueryPluglet, the request sent to NnmSnmpCollectPluglet wraps the entity (system) name with the interfaces (NICs) on that system that are of interest for collecting SNMP metrics. Interfaces can either be specified by SNMP index number or the corresponding interface name if it is assigned in NNM.
<metric> <name>*If%util*</name> </metric>	Specification of performance metric data to retrieve. Multiple <metric> elements may exist in this file. Unlike the performance metric request sent to the OvpmQueryPluglet, specifying the metric group and filter is not necessary.
<dataSource>SNMP</dataSource>	The data source for NNM SNMP collect requests must always be set to SNMP.

Namespaces and Performance Metric Request/Response Message Format

The namespaces and the performance metric request/response message format for the NnmSnmpCollectPluglet is identical to that of the OvpmQueryPluglet. See Performance Metric Messages for more details.

Performance metric exception handling for the NnmSnmpCollectPluglet is identical to that in OvpmQueryPluglet. See Performance Metric Exception Handling for more details.

NnmInventorySnmpEventPluglet

The NnmInventorySnmpEventPluglet allows you to receive raw or correlated SNMP inventory events from NNM. Event types available via OVI include NODE_ADDED, NODE_DELETED, NODE_UP, NODE_DOWN, INTERFACE_ADDED, INTERFACE_DELETED, INTERFACE_UP and INTERFACE_DOWN. When an event is received from NNM, the NnmInventorySnmpEventPluglet creates ovInventoryChangeEvent XML (conforming to SCHEMAS_DIR/entity/ovInventoryChangeEvent.xsd) that represents the event and passes it to its targetPluglet:

- If a new network node is discovered by NNM or if a node goes up or down, OVI transmits an ovInventoryChangeEvent containing information about the affected node and all its associated interfaces
- When NNM detects that a node has been deleted, the transmitted ovInventoryChangeEvent contains information about the deleted node but not its interfaces

- If OVI receives an NNM event indicating that an interface has been added/deleted or has gone up/down, it sends an `ovInventoryChangeEvent` that has information about the affected interface as well as the node to which the interface belongs.

By default, the `NnmInventorySnmpEventPluglet` is also configured to send a periodic heartbeat message to:

- Provide the status of OVI's connection to the NNM postmaster daemon (pmd), which supplies SNMP event information
- Indicate when the connection to the NNM postmaster daemon is lost
- Indicate when OVI is in the process of trying to reestablish its connection to NNM

The heartbeat message contains XML that conforms to `SCHEMAS_DIR/types/ovCanYouHearMeNowEvent.xsd`.

See NnmInventorySnmpEventPluglet Configuration for further information.

NnmInventorySnmpEventPluglet Configuration

Example `NnmInventorySnmpEventPluglet .deploy` and `.config` files have been provided in the `OVI_BASE_DIR/examples/OVI/configs/NNM/InventoryEvent` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory tree before editing. The `NnmInventorySnmpEventPluglet.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<nnmInventorySnmpEventPluglet
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/NnmInventorySnmpEventPluglet.xsd">
    <!--
        Specify the Pluglet that the NnmInventorySnmpEventPluglet should
        acquire
    -->
    <targetPluglet>PublisherPluglet</targetPluglet>
    <!--
        Indicate whether only correlated NNM SNMP events will be handled.
        If false, raw NNM SNMP events will be processed.
    -->
    <correlatedEventsOnly>false</correlatedEventsOnly>
    <!--
        Time (in milliseconds) between heartbeat messages
    -->
    <pulseRate>300000</pulseRate>
    <!--
        Number of attempts that will be made to reestablish connection with
        the NNM SNMP event source
    -->
    <numReconnectTries>10</numReconnectTries>
    <!--
        Time (in milliseconds) to wait between attempts to reestablish a
        connection with the NNM SNMP event source
    -->
    <delayBetweenRetries>15000</delayBetweenRetries>
    <!--
        Switch on message trace logging
    -->
    <traceMessages>true</traceMessages>
    <!--
        Switch on message route logging
    -->
```

```
    <traceRoute>true</traceRoute>
</nnmInventorySnmpEventPluglet>
```

The following table describes elements in the `NnmInventorySnmpEventPluglet.config` file:

Element Name	Value

`<targetPluglet>`
targetPlugletName
`</targetPluglet>`
	Name of this pluglet's target pluglet. This is the pluglet to which messages are passed.

`<correlatedEventsOnly>`
false
`</correlatedEventsOnly>`
	Indicates whether only correlated NNM SNMP events are handled. If `false`, raw NNM SNMP events are processed by this pluglet. When running NNM, correlated events are those which appear in the NNM browser.

`<pulseRate>`
milliseconds
`</pulseRate>`
	Time, in milliseconds, between heartbeat pulse messages. The initial setting is `300,000`, meaning that hearbeats are sent every 5 minutes.

`<numReconnectTries>`
tries
`</numReconnectTries`
	Number of attempts that are made to reestablish a connection with the NNM SNMP event source if it is lost. The initial setting is `10`. If the specified number of retry attempts fail, the pluglet stops attempting to reconnect.

`<delayBetweenRetries>`
milliseconds
`</delayBetweenRetries`
	Time, in milliseconds, to wait between attempts to reestablish a connection with the NNM SNMP event source. The initial setting is `15,000`, meaning that there is a 15 second delay between retry attempts.

`<traceMessages>`
boolean
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
boolean
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

Namespaces and NnmInventorySnmpEventPluglet Message Formats

The network inventory change event message schema and example XML can be found in `OVI_BASE_DIR/examples/OVI/schemas`. (This directory is referred to as `SCHEMAS_DIR` below.)

`NnmInventorySnmpEventPluglet` XML messages containing NNM event information conform to the schema in `SCHEMAS_DIR/entity/ovInventoryChangeEvent.xsd`, which uses these namespace designations: `xmlns=http://openview.hp.com/xmlns/ico/entity` and `xmlns=http://openview.hp.com/xmlns/ico/types`.

`NnmInventorySnmpEventPluglet` heartbeat XML messages conform to the schema in `SCHEMAS_DIR/types/ovCanYouHearMeNowEvent.xsd` and use the `http://openview.hp.com/xmlns/ico/types` namespace.

A sample network inventory change message is available at `SCHEMAS_DIR/entity/ovInventoryChangeEvent.xml`.

NnmEventReadPluglet

The `NnmEventReadPluglet` enables retrieval of top-level SNMP events from NNM. This allows an application that is tracking events, but goes down for a period of time, to resynchronize with the NNM event database. It also permits applications to poll NNM for SNMP events.

Event retrieval is accomplished by sending an event read request to the `NnmEventReadPluglet`. Due to the potentially large volume of event data that may be returned in response to a given request, the request contains event filtering criteria. Criteria include a time range and an optional application filter (i.e., a string passed to HP Network Node Manager that further determines which events are returned). See the *OVsnmpEventOpen* section of the *Network Node Manager DevKit documentation* **(http://support.openview.hp.com/selfsolve/document/KM6890/binary/nnm70_man3_pdf)** for a description of the syntax of this filter.

When the `NnmEventReadPluglet` receives an event read request (`ovMessageReadRequest`), it gathers event data meeting the designated start time, end time and optional application filter criteria and returns that data to the requester in an `ovMessageReadResponse`. The response also contains an `allRequestedMessagesReturned` flag, which indicates whether all available event data meeting the filtering criteria has been returned. If the flag is set to `false`, the number of available events exceeded the maximum number of messages that `NnmEventReadPluglet` is permitted to return. (The maximum messages setting can be adjusted in the `NnmEventReadPluglet.config` file.) When this happens, you can receive the remaining available events by noting the time of the last message returned and then making another request with this time as the request's start time. If an error occurs while processing an `ovMessageReadRequest`, an `ovMessageReadException` is returned as the response.

See NnmEventReadPluglet Configuration_ for further information.

NnmEventReadPluglet Configuration

Example `NnmEventReadPluglet .deploy` and `.config` files have been provided in the `OVI_BASE_DIR/examples/OVI/configs/NNM/EventRead` directory. If you need to make changes, copy them to the `OVI_DATA_DIR/conf/OVI` directory tree before editing. The `NnmEventReadPlugletExample.config` file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<nnmEventReadPluglet
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
    "/opt/OV/schema/OVI/NnmEventReadPluglet.xsd">
    <!--
        Optional element containing the maximum number of
        messages to return in a response. The JVM heap
        size may need to be increased to accommodate the
        specified maxMessagesToReturn. The heap size can
        be set through the JVM_ARGS variable in the
        OVI.env file
        <maxMessagesToReturn>250</maxMessagesToReturn>
    -->
    <!--
        Switch on message trace logging
    -->
    <traceMessages>true</traceMessages>
    <!--
        Switch on message route logging
    -->
    <traceRoute>true</traceRoute>
</nnmEventReadPluglet>
```

The following table describes elements in the `NnmEventReadPluglet.config` file:

Element Name	Value

`<maxMessagesToReturn>`
250
`</maxMessagesToReturn>`
	Optional element containing the maximum number of events to return in an `ovMessageReadResponse`. The heap size of the JVM may need to be increased to accommodate larger `maxMessagesToReturn` values. (Default: `250`)

`<traceMessages>`
boolean
`</traceMessages>`
	Controls whether an `INFO` level log message is created when a message arrives at this pluglet. Valid values are `true` or `false`. See traceMessages Settings section for more details. (Default: `true`)

`<traceRoute>`
boolean
`</traceRoute>`
	Controls whether route-tracing information containing the time of message arrival at this pluglet is collected and displayed. Valid values are `true` or `false`. See the traceRoute Settings section for more details. (Default: `true`)

If you use `TestRequesterPluglet` to generate a test (event read) request for `NnmEventReadPluglet`, you can modify its associated `NnmEventReadPlugletExampleRequest.in` file to contain your desired time range and application filter values. This file's contents appear below, and elements that you may want to change are indicated.

```
<?xml version="1.0" encoding="UTF-8"?>
<ovMessageReadRequest
    xmlns="http://openview.hp.com/xmlns/ico/message"
    xmlns:ovit="http://openview.hp.com/xmlns/ico/types"
    version="1.0">
    <messageReadCriteria>
        <!--
            Time range used to determine which messages to
            return. Only messages that arrived on the OV
            management server after startTime and before
            endTime will be returned. Both startTime and
            endTime are specified in ISO 8601 format.
            You will want to modify this:
        -->
        <timeRange>
            <ovit:startTime>2004-05-10T09:10:00</ovit:startTime>
            <ovit:endTime>2004-05-10T10:30:00</ovit:endTime>
        </timeRange>

        <!--
            Application-specific string used to further filter
            which messages are read.
            You will want to modify this:
            <applicationFilter>.1.3.6.1.4.*;.1.3.6.1.5.7.*
            </applicationFilter>
        -->
    </messageReadCriteria></ovMessageReadRequest>
```

The following table describes elements in the `NnmEventReadPlugletExampleRequest.in` file:

Element Name	Value

`<timeRange>`
`  <ovit:startTime>`
`  2004-05-10T09:10:00`
`  </ovit:startTime>`
`  <ovit:endTime>`
`  2004-05-10T10:30:00`
`  </ovit:endTime>`
`</timeRange>`
	Time range, in ISO 8601 format, used to determine which NNM messages to retrieve.

`<applicationFilter>`
`  .1.3.6.1.4.*`
`</applicationFilter>`
	Optional element that holds an application-specific string used to further filter which messages are retrieved from NNM.

Event Read Namespaces

The namespaces used in the event read request/response XML messages for HP Network Node Manager are `xmlns=http://openview.hp.com/xmlns/ico/message` and `xmlns=http://openview.hp.com/xmlns/ico/types`.

These namespaces are required for `<ovMessageReadRequest>`, `<ovMessageReadResponse>`, and `<ovMessageReadException>` messages.

The contents of the event read request XML message is only partially validated. Full validation is not performed for performance reasons. Instead, specific parts of each event read request are checked.

Event Read Request/Response Message Format

The event read request and response message schemas and example XML can be found in `OVI_BASE_DIR/examples/OVI/schemas/message`. (This directory is referred to as `SCHEMAS_DIR` below.)

XML Message Description	**Location**
Event read request message	`SCHEMAS_DIR/ovMessageReadRequest.xml`
Event read request schema	`SCHEMAS_DIR/ovMessageReadRequest.xsd`
Event read response message	`SCHEMAS_DIR/ovMessageReadResponse-NNM.xml`
Event read response schema	`SCHEMAS_DIR/ovMessageReadResponse.xsd`

Event read exception response message	`SCHEMAS_DIR/ovMessageReadOperationException.xml`
	`SCHEMAS_DIR/ovMessageReadInvalidXmlException.xml`
	`SCHEMAS_DIR/ovMessageReadTimeOutException.xml`

Event read exception response schema	`SCHEMAS_DIR/ovMessageReadException.xsd`

A Sample Applications

OVI installs several sample applications that illustrate how to use the information provided by OVI. These are not production quality samples, but they show how to query or receive OV information using OVI. Some examples are in Java and others in Perl to show the flexibility of OVI. There are also some SOAP examples written in C#.

Type	Description
Socket requester	This sample shows how you communicate with the socket-based response pluglet of an OVI instance with a Perl script acting as the requester. See: `OVI_BASE_DIR/examples/OVI/sampleCode/SocketRequester/SocketSnRequesterDesign.html`
HTTP requester	This sample shows how you communicate with an HTTP-based OVI response pluglet with a Perl script acting as the requester. See: `OVI_BASE_DIR/examples/OVI/sampleCode/HttpRequester/HttpSnRequesterDesign.html`.
Service Navigator query	This Java application and its source code are provided to demonstrate how to integrate with the Service Navigator Query Pluglet (`SNQueryPluglet`) using JMS. An accompanying white paper describes the application's purpose and design, how to run it, and example scenarios using `SNQueryPluglet`. The whitepaper is located at: `OVI_BASE_DIR/examples/OVI/sampleCode/ServiceStatusRequest/ServiceStatusRequesterDesign.html`

JMS responder	This sample Java application and its source code show how to enrich OVI messages with additional data using a JMS client in the enrichment request/response scenario. See: `OVI_BASE_DIR/examples/OVI/sampleCode/JMSResponder/ResponderClientDesign.html`
	
	When configured for a request/response scenario, OVI utilizes only a single topic for replies, regardless of number of pending requests. To properly correlate these replies with their associated request, a reply must contain the JMS message ID of the request as the correlation ID of the response. This correlation ID is then used by OVI `RequesterProxyPluglet` to locate the original message and properly generate a response. This mechanism is built into the `ResponderProxyPluglet`, the primary consumer of request messages. The code that performs this correlation is:
	`String correlationID = jmsRequest.getJMSMessageID();`
	`jmsResponse.setJMSCorrelationID(correlationID);`

Socket event create requester	This sample shows how you communicate with the socket-based response pluglet of an OVI instance with a Perl script acting as the requester for the creation of OVO events. See: `OVI_BASE_DIR/examples/OVI/sampleCode/SocketEventCreateRequester/SocketEventCreateDesign.html`.

Type	Description
Socket performance metric requester	This sample application demonstrates how to communicate with the socket-based response pluglet of an OVI instance with a Perl script acting as the requester for retrieving performance metrics from OVPM. See: `OVI_BASE_DIR/examples/OVI/sampleCode/SocketPerformanceMetricRequester/SocketPerformanceMetricRequester.html`.
Socket transaction metric requester	This sample code shows how to communicate with the socket-based response pluglet of an OVI instance through a Perl script, which is acting as the requester for retrieving transaction metrics from OVTA. See: `OVI_BASE_DIR/examples/OVI/sampleCode/SocketTransactionMetricRequester/SocketTransactionMetricRequester.html`.
HTTP network inventory requester	This sample shows how you communicate with an HTTP-based response pluglet of an OVI instance with a Perl script acting as the requester for retrieving network inventory information from NNM. See: `OVI_BASE_DIR/examples/OVI/sampleCode/HttpNetworkInventoryRequester/HttpTopoInventoryRequesterDesign.html`.
SOAP client samples (Java and .NET C#)	These samples, written in Java for Apache Axis and .NET C#, illustrate OVI's support for receiving SOAP-formatted requests and returning SOAP-formatted responses over HTTP. Web Services Description Language (WSDL) files define the format of the SOAP OVI request and response messages, and are available in the `OVI_BASE_DIR/examples/OVI/wsdl` subdirectories. The sample Java and .NET C# clients are located in the `OVI_BASE_DIR/examples/OVI/sampleCode/soapClients` subdirectories. Within the `csharp` subdirectory, `.bat` files are included for generating the .NET client executables. The .NET Framework and the .NET SDK are required for the proper execution of these `.bat` files, which rely on `wsdl.exe` and the C# compiler, csc.exe. Within the `java` subdirectory, script files are included for generating the Java for Apache Axis clients. The environment variable `AXIS_HOME` must be set prior to executing these scripts.

B Troubleshooting

At times, you may want to alter pluglet settings that regulate the amount of debug information generated as OVI runs and messages are passed from pluglet to pluglet. To do this, you must change values of the `traceMessages` flag, `traceRoute` flag, and logging and tracing settings. See tables in the traceMessages Settings, traceRoute Settings and Logging and Tracing Settings sections below for details.

traceMessages Settings

The `traceMessages` flag controls whether an `INFO` level log message is created when a message arrives at a pluglet. To enable `traceMessages`, OVI logging must be set to `INFO` or `DEBUG`. For further information on how to change the OVI logging level, see the Logging and Tracing Settings section. Each pluglet has its own `traceMessages` flag, which can be changed by editing the pluglet's `.config` file or via the JMX browser. In addition, a message-specific `traceMessages` flag exists in each OVI message that is created. This flag is set by the pluglet that originates the message (based on the `traceMessages` flag setting in its `.config` file), and is passed from pluglet to pluglet. The combination of the message-specific and pluglet-specific flag settings determines if a log message is generated when a message arrives at a pluglet, as shown in this table:

	Pluglet's traceMessages = true	**Pluglet's traceMessages = false**
message's traceMessages = true	Tracing occurs	Tracing occurs
message's traceMessages = false	Tracing occurs	No tracing

traceRoute Settings

The `traceRoute` flag controls whether route-tracing information containing the time of message arrival at a pluglet is collected and displayed. Each pluglet has its own `traceRoute` flag, which can be changed by editing the pluglet's `.config` file or via the JMX browser. In addition, a message-specific `traceRoute` flag exists in each OVI message that is created. This flag is set by the first pluglet in the message path (based on the `traceRoute flag setting` in its `.config` file), and is passed from pluglet to pluglet. It is the combination of the message-specific and pluglet-specific flag settings that determines if routing information is generated when a message arrives at a pluglet, as shown in this table:

	Pluglet's traceRoute = true	**Pluglet's traceRoute = false**
message's traceRoute = true	Pluglet *is* added to route trace	Pluglet *is not* added to route trace
message's traceRoute = false	Pluglet *is not* added to route trace	Pluglet *is not* added to route trace

Logging and Tracing Settings

The changes will become effective after restarting of OVI.

OVI logging and tracing settings are read from `OVI_DATA_DIR/conf/OVI/OviLogger.config`. You can edit this file to modify current values for `logLevel`, `logTarget`, `logFile`, and `logFileMaxSize`. Valid values for these elements are provided in the following table. After OVI is installed and if you make no changes to logging configuration, default settings are used. This means that log messages are sent to `STDOUT` and that the log level is set to `INFO`.

Element Name	**Use**	**Valid Values**	**Default Value**
`<logLevel>`	Sets the threshold at which message logging occurs. For example, if `logLevel` is set to `INFO`, all `INFO`, `WARNING`, and `ERROR` messages are logged. `DEBUG` messages are not generated.	`ERROR, WARNING, INFO, DEBUG`	`INFO`

Element Name	Use	Valid Values	Default Value
<logTarget>	Sets the destination of logging messages.	STDERR, STDOUT, FILE	STDOUT
<logFile>	If logTarget is set to FILE, this setting specifies the file to which logging messages are written. This file is rolled whenever OVI is started, and whenever it grows to its maximum size (explained below). When a log file is rolled, its contents are copied from ovi.log to ovi.log.1, which is a backup log file. Up to three backup log files are maintained. When the ovi.log file is rolled, the contents of ovi.log.1 (if it exists) are copied to ovi.log.2. Similarly, if there is an ovi.log.2, its contents are copied to ovi.log.3. If there is an existing ovi.log.3, the previous contents are discarded.	Any valid filename with a path to a directory that is writeable.	OVI_DATA_DIR/log/OVI/ovi.log
<logFileMaxSize>	If logTarget is set to FILE, this setting specifies the maximum size (in megabytes) to which the log file can grow before its contents are rolled.	Any positive integer.	10 MB

Making Temporary Configuration Changes with the JMX Browser

An alternative way to make temporary changes to OVI settings is via a JMX browser. It is a handy tool to employ when you are debugging a problem because it allows you to quickly alter pluglet attributes to provide additional debug information. Changes made to pluglet attributes via the browser are *not* written to the pluglet's .config file.

As of version 3.0, OVI no longer includes an integrated JMX browser. You may obtain any one of your choice, however, only EJTools has been verified to work with OVI. To integrate EJTools:

1. After JMX browser installation has completed, navigate to *lib/ext* under the EJTools installation directory.
   - **UNIX**: Copy /opt/OV/nonOV/OVI/java/jmx/mx4j-jmx.jar and /opt/OV/nonOV/OVI/java/jmx/mx4j-tools.jar into this directory.
   - **Windows**: Copy OVI_BASE_DIR\nonOV\OVI\java\jmx\mx4j-jmx.jar and OVI_BASE_DIR\nonOV\OVI\java\jmx\mx4j-tools.jar into this directory.

2 Edit the `/var/opt/OV/conf/OVI/OviAgent.config` (UNIX) or `OVI_DATA_DIR\conf\OVI\OviAgent.config` (Windows) file. Uncomment the section beginning with `<adaptor>`. This can be done by changing:

```
<!-- JMX adaptor disabled by default
<adaptor>
    <jrmpAdaptor>
        <jndi>
            <factory>
                com.sun.jndi.rmi.registry.
                     RegistryContextFactory
            </factory>
            <port>1099</port>
            <binding>jrmp</binding>
        </jndi>
   </jrmpAdaptor>
</adaptor>
-->
```

to:

```
<!-- JMX adaptor disabled by default -->
<adaptor>
    <jrmpAdaptor>
        <jndi>
            <factory>
                com.sun.jndi.rmi.registry.
                     RegistryContextFactory
            </factory>
            <port>1099</port>
            <binding>jrmp</binding>
        </jndi>
   </jrmpAdaptor>
</adaptor>
```

3 Restart OVI.

4 Start EJTools, and select **File** → **New**. In the Connect to a Server dialog, choose the file connection profile for `MX4J JRMP`. This causes the EJTools JMX Browser Untitled Connection window to appear. Leave the default values in place (Factory: `com.sun.jndi.rmi.registry.RegistryContextFactory`, Packages: `com.sun.jndi.rmi.registry`, URL: **rmi://localhost:1099**, JNDI Binding: `jrmp`). Optionally, you can change the connection name to something meaningful.

5 Click **Connect to the JMX Server**. An object tree appears in the left panel. Expanding the top level `OVI` item and its `OVI` child item reveals a list of running pluglets that are JMX-enabled.

6 Click on a pluglet of interest. Its JMX-managed attributes and operations are displayed in the right panel. Some attributes are read-only, while others are writeable. You can temporarily alter the value of one that is writeable by double-clicking on it, entering a new value, and clicking **Update**. Similarly, you can cause pluglet operations to be invoked by scrolling down to the **Operations** list in the right pane and selecting the desired operation's **Invoke** button. If the operation requires parameters, you are prompted to enter them.

OVI Version Utility

To see version information about the OVI components currently installed, enter:

UNIX: `/opt/OV/bin/OVI.sh -v`

Windows: `OVI_BASE_DIR\bin\OVI.wsf -v`

or, select **Start** → **Program** → **HP** → **HP Interconnect** → **Start HP Interconnect** and click the OVI Version tab.

C HP Interconnect Directory Structure

This appendix describes the OVI directory structure for both UNIX and Windows.

Installation Directory Structure for UNIX

The HP Interconnect (OVI) directories listed in this section are placed on your UNIX system during installation.

OVI Startup, Shutdown, and Uninstall Scripts Directories

```
/opt/OV
/opt/OV/bin
/opt/OV/lbin
/opt/OV/lbin/OVI
/opt/OV/license-agreements/OVI
/opt/OV/Uninstall
/opt/OV/Uninstall/OVI
```

OVI .jar Files Directories

```
/opt/OV/java
/opt/OV/java/OVI
```

Non-OpenView .jar Files Directories

```
/opt/OV/nonOV
/opt/OV/nonOV/OVI
/opt/OV/nonOV/OVI/java
/opt/OV/nonOV/OVI/java/http
/opt/OV/nonOV/OVI/java/jmx
/opt/OV/nonOV/OVI/java/log4j
/opt/OV/nonOV/OVI/java/xml
/opt/OV/nonOV/OVI/java/jms
```

OVI Shared Libraries Directory

/opt/OV/lib

OVI .config and .deploy Files Directories

The files in these directories are used by HP Support as a backup of the originally installed `.config` and `.deploy` files.

/opt/OV/newconfig

/opt/OV/newconfig/OVI

OVI Example Files Directories

/opt/OV/examples

/opt/OV/examples/OVI

/opt/OV/examples/OVI/configs

/opt/OV/examples/OVI/configs/CommandResponder

/opt/OV/examples/OVI/configs/FileReader

/opt/OV/examples/OVI/configs/NNM

/opt/OV/examples/OVI/configs/NNM/EventRead

/opt/OV/examples/OVI/configs/NNM/Inventory

/opt/OV/examples/OVI/configs/NNM/InventoryEvent

/opt/OV/examples/OVI/configs/NNM/SnmpCollect

/opt/OV/examples/OVI/configs/PublishSubscribeTest

/opt/OV/examples/OVI/configs/RequestResponseTest

/opt/OV/examples/OVI/configs/OVO

/opt/OV/examples/OVI/configs/OVO/Inventory

/opt/OV/examples/OVI/configs/OVO/PublishEvent

/opt/OV/examples/OVI/configs/OVO/EventAction

/opt/OV/examples/OVI/configs/OVO/EventEnrichment

/opt/OV/examples/OVI/configs/OVO/FilterExamples

/opt/OV/examples/OVI/configs/OVO/EventCreation

/opt/OV/examples/OVI/configs/OVO/EventChange

/opt/OV/examples/OVI/configs/OVO/EventModify

/opt/OV/examples/OVI/configs/OVO/EventRead

/opt/OV/examples/OVI/configs/OVPM/QueryMetric

/opt/OV/examples/OVI/configs/OVPM/QueryMetricInfo

/opt/OV/examples/OVI/configs/OVPM/QueryReport

/opt/OV/examples/OVI/configs/OVPM/QuerySourceInfo

/opt/OV/examples/OVI/configs/OVTA/Query

/opt/OV/examples/OVI/configs/SD/Inventory

/opt/OV/examples/OVI/configs/SD/ServiceInfoEnrichment

/opt/OV/examples/OVI/configs/SN

/opt/OV/examples/OVI/configs/SN/QueryStatus

/opt/OV/examples/OVI/configs/SN/QueryHierarchy

/opt/OV/examples/OVI/configs/Transformer

/opt/OV/examples/OVI/configs/Transformer/PublishSubscribe

/opt/OV/examples/OVI/configs/Transformer/RequestResponse

/opt/OV/examples/OVI/configs/Transformer/xsl

/opt/OV/examples/OVI/sampleCode/HttpNetworkInventoryRequester

/opt/OV/examples/OVI/sampleCode/HttpRequester

/opt/OV/examples/OVI/sampleCode/HttpRequester/config

/opt/OV/examples/OVI/sampleCode/HttpTransactionMetricRequester

/opt/OV/examples/OVI/sampleCode/JMSResponder

/opt/OV/examples/OVI/sampleCode/ServiceStatusRequest

/opt/OV/examples/OVI/sampleCode/SocketEventCreateRequester

/opt/OV/examples/OVI/sampleCode/SocketPerformanceMetricRequester

/opt/OV/examples/OVI/sampleCode/SocketRequester

/opt/OV/examples/OVI/sampleCode/SocketRequester/config/

/opt/OV/examples/OVI/sampleCode/soapClients

/opt/OV/examples/OVI/sampleCode/soapClients/config/

/opt/OV/examples/OVI/sampleCode/soapClients/config/entitystatus/

/opt/OV/examples/OVI/sampleCode/soapClients/config/inventory/

/opt/OV/examples/OVI/sampleCode/soapClients/config/ovinventory/

/opt/OV/examples/OVI/sampleCode/soapClients/config/ovnodeinventory/

/opt/OV/examples/OVI/sampleCode/soapClients/config/messageaction/

/opt/OV/examples/OVI/sampleCode/soapClients/config/messagecreate/

/opt/OV/examples/OVI/sampleCode/soapClients/config/messagemodify/

/opt/OV/examples/OVI/sampleCode/soapClients/config/messageread/

/opt/OV/examples/OVI/sampleCode/soapClients/config/messageread/NNM/

/opt/OV/examples/OVI/sampleCode/soapClients/config/messageread/OVO/

/opt/OV/examples/OVI/sampleCode/soapClients/config/performance/

/opt/OV/examples/OVI/sampleCode/soapClients/config/transactionmetric/

/opt/OV/examples/OVI/sampleCode/soapClients/java/

/opt/OV/examples/OVI/sampleCode/soapClients/java/axis/

OVI .config File XML Schema Directories

```
/opt/OV/schema
/opt/OV/schema/OVI
/opt/OV/schema/OVI/csf
/opt/OV/schema/OVI/http
/opt/OV/schema/OVI/jms
/opt/OV/schema/OVI/socket
/opt/OV/schema/OVI/transformer
```

OVI misc Files Directories

```
/opt/OV/misc
/opt/OV/misc/OVI
/opt/OV/misc/OVI/ConfigTest
```

OVI Environment, .config, and .deploy Files Directories

```
/var/opt/OV/conf
/var/opt/OV/conf/OVI
```

OVI Keystore Files Directories

```
/var/opt/OV/certificates
/var/opt/OV/certificates/OVI
```

OVI Temporary Files Directories

```
/var/opt/OV/tmp
/var/opt/OV/tmp/OVI
/var/opt/OV/tmp/OVI/ExampleReadDir
/var/opt/OV/tmp/OVI/ExampleWriteDir
```

OVI Logs Directories

```
/var/opt/OV/log
/var/opt/OV/log/OVI
```

Documentation Directories

```
/opt/OV/paperdocs/C/OVI/autopass
```

```
/opt/OV/paperdocs/C/OVI
/opt/OV/paperdocs/C/OVI/toolkit
/opt/OV/paperdocs/C/OVI/toolkit/javadoc
/opt/OV/paperdocs/C/OVI/toolkit/javadoc/sdk
```

OVI Message Schemas and Web Services Description Language (WSDL)

```
/opt/OV/examples/OVI/schemas
/opt/OV/examples/OVI/schemas/command
/opt/OV/examples/OVI/schemas/entity
/opt/OV/examples/OVI/schemas/message
/opt/OV/examples/OVI/schemas/performance
/opt/OV/examples/OVI/schemas/types
/opt/OV/examples/OVI/wsdl
/opt/OV/examples/OVI/wsdl/entity
/opt/OV/examples/OVI/wsdl/message
/opt/OV/examples/OVI/wsdl/performance
```

OVI Toolkit

```
/opt/OV/examples/OVI/toolkit
/opt/OV/examples/OVI/toolkit/configs
/opt/OV/examples/OVI/toolkit/configs/RequesterResponder
/opt/OV/examples/OVI/toolkit/configs/SampleSNExtender
/opt/OV/examples/OVI/toolkit/configs/SourceSink
/opt/OV/examples/OVI/toolkit/sdk
```

Installation Directory Structure for Windows

The directories listed in this section are placed on the OVI installation drive of your Windows system during installation. The base installation directory, `OVI_BASE_DIR` defaults to `\Program Files\HP OpenView`. The installation data directory, `OVI_DATA_DIR`, defaults to `\Program Files\HP OpenView\data`. You can change the location of both the installation directory and data directory if required.

OVI Startup, Shutdown, and Uninstall Scripts Directories

```
OVI_BASE_DIR\bin
OVI_BASE_DIR\lbin
OVI_BASE_DIR\license-agreements\OVI
```

```
OVI_BASE_DIR\Uninstall
OVI_BASE_DIR\Uninstall\OVI
OVI_BASE_DIR\Uninstall\OVI\resource
```

OVI .jar Files Directories

```
OVI_BASE_DIR\java
OVI_BASE_DIR\java\OVI
```

Non-OpenView .jar Files Directories

```
OVI_BASE_DIR\nonOV\OVI\java\http
OVI_BASE_DIR\nonOV\OVI\java\jmx
OVI_BASE_DIR\nonOV\OVI\java\log4j
OVI_BASE_DIR\nonOV\OVI\java\xml
OVI_BASE_DIR\nonOV\OVI\java\jms
```

OVI Libraries Directory

```
OVI_BASE_DIR\lib
```

OVI .config and .deploy Files Directories

The files in these directories are used by HP Support as a backup of the originally installed `.config` and `.deploy` files.

```
OVI_BASE_DIR\newconfig
OVI_BASE_DIR\newconfig\OVI
```

OVI Example Files Directories

```
OVI_BASE_DIR\examples
OVI_BASE_DIR\examples\OVI
OVI_BASE_DIR\examples\OVI\configs
OVI_BASE_DIR\examples\OVI\configs\FileReader
OVI_BASE_DIR\examples\OVI\configs\NNM
OVI_BASE_DIR\examples\OVI\configs\NNM\EventRead
OVI_BASE_DIR\examples\OVI\configs\NNM\Inventory
OVI_BASE_DIR\examples\OVI\configs\NNM\InventoryEvent
OVI_BASE_DIR\examples\OVI\configs\NNM\SnmpCollect
OVI_BASE_DIR\examples\OVI\configs\OVO
```

OVI_BASE_DIR\examples\OVI\configs\OVO\EventAction

OVI_BASE_DIR\examples\OVI\configs\OVO

OVI_BASE_DIR\examples\OVI\configs\OVO\EventCreation

OVI_BASE_DIR\examples\OVI\configs\OVO\EventModify

OVI_BASE_DIR\examples\OVI\configs\OVO\FilterExamples

OVI_BASE_DIR\examples\OVI\configs\OVO\PublishEvent

OVI_BASE_DIR\examples\OVI\configs\OVO\EventChange

OVI_BASE_DIR\examples\OVI\configs\

OVI_BASE_DIR\examples\OVI\configs\OVPM\QueryMetric

OVI_BASE_DIR\examples\OVI\configs\OVPM\QueryMetricInfo

OVI_BASE_DIR\examples\OVI\configs\OVPM\QueryReport

OVI_BASE_DIR\examples\OVI\configs\OVPM\QuerySourceInfo

OVI_BASE_DIR\examples\OVI\configs\OVTA

OVI_BASE_DIR\examples\OVI\configs\OVTA\Query

OVI_BASE_DIR\examples\OVI\configs\PublishSubscribeTest

OVI_BASE_DIR\examples\OVI\configs\RequestResponseTest

OVI_BASE_DIR\examples\OVI\configs\SD

OVI_BASE_DIR\examples\OVI\configs\SD\Inventory

OVI_BASE_DIR\examples\OVI\configs\SD\ServiceInfoEnrichment

OVI_BASE_DIR\examples\OVI\configs\SN

OVI_BASE_DIR\examples\OVI\configs\SN\QueryHierarchy

OVI_BASE_DIR\examples\OVI\configs\SN\QueryStatus

OVI_BASE_DIR\examples\OVI\configs\Transformer

OVI_BASE_DIR\examples\OVI\configs\Transformer\PublishSubscribe

OVI_BASE_DIR\examples\OVI\configs\Transformer\RequestResponse

OVI_BASE_DIR\examples\OVI\configs\Transformer\xsl

OVI_BASE_DIR\examples\OVI\sampleCode

OVI_BASE_DIR\examples\OVI\sampleCode\HttpTransactionMetricRequester

OVI_BASE_DIR\examples\OVI\sampleCode\JMSResponder

OVI_BASE_DIR\examples\OVI\sampleCode\ServiceStatusRequest

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\entityhierarchy\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\entitystatus\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\inventory\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\messageaction\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\messagecreate\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\messagemodify\

```
OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\messageread\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\messageread\NNM

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\ovinventory\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\performance\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\config\transactionmetri
c\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\csharp\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\java\

OVI_BASE_DIR\examples\OVI\sampleCode\soapClients\java\axis\

OVI_BASE_DIR\examples\OVI\sampleCode\SocketEventCreateRequester

OVI_BASE_DIR\examples\OVI\sampleCode\SocketPerformanceMetricRequester

OVI_BASE_DIR\examples\OVI\sampleCode\SocketPerformanceMetricRequester\co
nfig

OVI_BASE_DIR\examples\OVI\sampleCode\SocketRequester
```

OVI .config File XML Schema Directories

```
OVI_BASE_DIR\schema

OVI_BASE_DIR\schema\OVI

OVI_BASE_DIR\schema\OVI\csf

OVI_BASE_DIR\schema\OVI\http

OVI_BASE_DIR\schema\OVI\jms

OVI_BASE_DIR\schema\OVI\socket

OVI_BASE_DIR\schema\OV\transform
```

OVI misc Files Directories

```
OVI_BASE_DIR\misc

OVI_BASE_DIR\misc\OVI

OVI_BASE_DIR\misc\OVI\ConfigTest

OVI_BASE_DIR\misc\OVI\ConfigTest\RequestResponseTest
```

OVI Environment, .config, and .deploy Files Directories

```
OVI_DATA_DIR\conf

OVI_DATA_DIR\conf\OVI
```

OVI Keystore Files Directories

```
OVI_DATA_DIR\certificates
```

```
OVI_DATA_DIR\certificates\OVI
```

OVI Temporary Files Directories

```
OVI_DATA_DIR\tmp
OVI_DATA_DIR\tmp\OVI
OVI_DATA_DIR\tmp\OVI\ExampleReadDir
OVI_DATA_DIR\tmp\OVI\ExampleWriteDir
```

OVI Log Files Directories

```
OVI_DATA_DIR\log
OVI_DATA_DIR\log\OVI
```

Documentation Directories

```
OVI_BASE_DIR\paperdocs\C\OVI\autopass
OVI_BASE_DIR\paperdocs\C\OVI
OVI_BASE_DIR\paperdocs\C\OVI\toolkit
OVI_BASE_DIR\paperdocs\C\OVI\toolkit\javadoc
OVI_BASE_DIR\paperdocs\C\OVI\toolkit\javadoc\sdk
```

OVI Message Schemas and Web Services Description Language (WSDL)

```
OVI_BASE_DIR\examples\OVI\schemas
OVI_BASE_DIR\examples\OVI\schemas\command
OVI_BASE_DIR\examples\OVI\schemas\entity
OVI_BASE_DIR\examples\OVI\schemas\message
OVI_BASE_DIR\examples\OVI\schemas\performance
OVI_BASE_DIR\examples\OVI\schemas\types
OVI_BASE_DIR\examples\OVI\wsdl
OVI_BASE_DIR\examples\OVI\wsdl\entity
OVI_BASE_DIR\examples\OVI\wsdl\message
OVI_BASE_DIR\examples\OVI\wsdl\performance
```

OVI Toolkit

```
OVI_BASE_DIR\examples\OVI\toolkit
OVI_BASE_DIR\examples\OVI\toolkit\configs
OVI_BASE_DIR\examples\OVI\toolkit\configs\RequesterResponder
```

```
OVI_BASE_DIR\examples\OVI\toolkit\configs\SampleSNExtender
OVI_BASE_DIR\examples\OVI\toolkit\configs\SourceSink
OVI_BASE_DIR\examples\OVI\toolkit\sdk
```