

---

# HP OpenView Service Quality Manager



## Service Adapter for HP OpenView Operations Installation, Configuration and User's Guide

**Edition: 1.4**

**December 2006**

© Copyright 2006 Hewlett-Packard Development Company, L.P.

---

## Legal notices

### Warranty

The information contained herein is subject to change without notice. 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.

### License requirement, and U.S. Government legend

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

### Copyright notices

© Copyright 2000-2006 Hewlett-Packard Development Company, L.P.

### Trademark notices

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

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Microsoft®, Windows®, and Windows NT® 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.

### Origin

Printed in France.

# Contents

Preface .....	5
<b>Chapter 1 .....</b>	<b>7</b>
<b>Introduction .....</b>	<b>7</b>
1.1 Service Quality Manager.....	7
1.2 The Service Adapter .....	8
1.2.1 What the Service Adapter does .....	8
1.2.2 OpenView Operations overview .....	8
1.2.3 OVO integration as a data source for SQM.....	9
1.2.4 Product deployment.....	10
1.2.5 OVO data collection.....	11
1.2.6 Generic DFD configuration .....	12
1.2.7 DFD parameters .....	13
1.2.8 DFD properties.....	14
1.2.9 Measurement Reference Point.....	15
<b>Chapter 2 .....</b>	<b>17</b>
<b>Installing the Service Adapter.....</b>	<b>17</b>
2.1 Software and Hardware requirements .....	17
2.1.1 Software requirements.....	17
2.1.2 Hardware requirements .....	17
2.2 Installing the OVO Fault Statistics SA .....	17
2.2.1 Installing the OV SQM Kernel.....	18
2.2.2 Required environment .....	18
2.2.3 Installing the SA Common subset.....	19
2.2.4 Installing SQL Service Adapter Runtime .....	20
2.2.5 Installing the OVO Fault Statistics Service Adapter .....	20
2.2.6 Patch installation.....	21
2.2.7 Uninstalling the software.....	22
<b>Chapter 3 .....</b>	<b>25</b>
<b>Setting up and configuring the Service Adapter .....</b>	<b>25</b>
3.1 Creating the application .....	27
3.1.1 Configuring the SQM Kernel.....	27
3.1.2 Setting up the required environment.....	27
3.1.3 Creating the OVO SA application .....	27
3.2 Configuring the application .....	29
3.3 Creating the SQL view .....	30
3.4 Discovering and loading DFIs.....	31
3.4.1 Raw discovery phase.....	32
3.4.2 Filtering phase .....	33

3.4.3	Loading phase .....	35
3.4.4	One-shot discovery and loading .....	36
3.4.5	Scheduling DFI discovery .....	37
3.5	Advanced configuration .....	37
<b>Chapter 4</b>	<b>.....</b>	<b>39</b>
<b>Service Adapter operation</b>	<b>.....</b>	<b>39</b>
4.1	Starting and Stopping OVO SA.....	39
4.2	Starting and Stopping collection by the Service Adapter.....	40
4.3	Maintenance.....	40
4.3.1	Creating a new application .....	40
4.3.2	Deleting an existing application .....	40
<b>Chapter 5</b>	<b>.....</b>	<b>43</b>
<b>Service Adapter advanced customization</b>	<b>.....</b>	<b>43</b>
5.1	Customizing the Discovery view .....	43
5.2	Customizing the Statistics view.....	44
<b>Appendix A</b>	<b>.....</b>	<b>45</b>
<b>Installation directory structure</b>	<b>.....</b>	<b>45</b>
<b>Glossary</b>	<b>.....</b>	<b>47</b>

# Preface

This document describes how to install and configure the HP OpenView Service Quality Manager (SQM) Service Adapter for HP OpenView Operations. The Service Adapter is the application that provides SQM with the performance data used to measure the quality of service.

This document describes how to:

- Install the Service Adapter (and associated subsets)
- Set up the Service Adapter
- Start and Stop the Service Adapter
- Customize the Service Adapter

## Intended audience

This document is intended for Service Quality Manager Administrators.

## Required knowledge

It is assumed that the reader is familiar with the functionality of Service Quality Manager and has previous experience of the following:

- System administration and operations
- HP OpenView Operations
- Service Level Management

It is assumed that the reader is familiar with the concepts described in the following books:

- *HP OpenView Service Quality Manager Overview*
- *HP OpenView Service Quality Manager Service Adapter User's Guide*
- *HP OpenView Service Quality Manager Administration Guide*

## Software versions

The software versions referred to in this document are specified in chapter 2.1.1 below.

## Typographical conventions

The following typographical conventions have been used throughout this document.

Courier font:

- Source code and examples of file contents
- Commands that you enter on the screen
- Pathnames

*Italic text:*

- Filenames, programs and parameters in the text
- The names of other documents referred to in this guide

**Bold text:**

- New terms
- Emphasized words
- Keyboard key names

## Associated documents

The following documents contain useful reference information:

- *SQL Service Adapter Toolkit Installation, Configuration and User's Guide*
- *HP OpenView VantagePoint Operations for UNIX Reporting and Database Schema*

For a full list of Service Quality Manager user documentation, see the *HP OpenView Service Quality Manager Product Family Introduction*.

## Support

Please visit our HP OpenView web site at [openview.hp.com/](http://openview.hp.com/)

There you will find contact information as well as details about the products, services, and support HP OpenView has to offer.

The OpenView support area of the HP OpenView web site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

# Chapter 1

## Introduction

The HP OpenView Service Quality Manager (SQM) uses a Service Adapter to collect data from HP OpenView Operations (OVO) and feed that data to SQM. The data provides fault statistic information on nodes managed by OpenView Operations.

The Service Adapter provides a common interface through which SQM works with external components.

This section provides a brief overview of SQM and the Service Adapter, and describes the elements the Service Adapter uses to collect data.

For a detailed description of SQM, see the *HP OpenView Service Quality Manager Overview*.

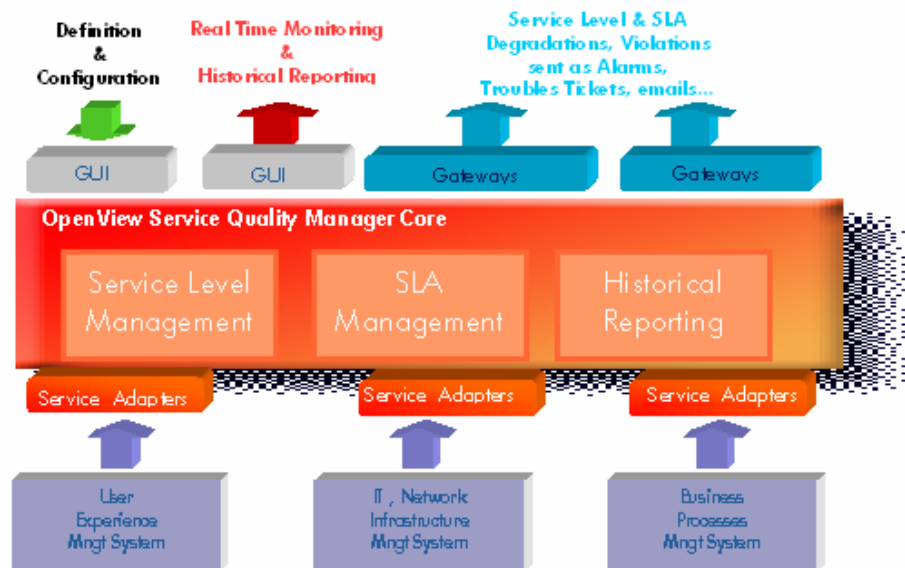
For a detailed description of Service Adapters, see the *HP OpenView Service Quality Manager Service Adapter User's Guide*.

### 1.1 Service Quality Manager

SQM provides a complete service quality management solution running under the HP implementation of UNIX®, HP-UX. HP-UX, which is compatible with the major industry standards, is based on the UNIX System V Release 4 operating system and includes important features from the Fourth Berkeley Software Distribution. SQM consolidates quality indicators across all domains—telecom, IT networks, servers, and applications—providing end-to-end visibility on service quality. It links service quality degradations to potential impacts on business, allowing network support personnel to address problems and prioritize actions proactively.

SQM monitors the service quality by aggregating information coming from all data sources, such as the network, the IT infrastructure, and the service provider's business processes. Using this information, service operators can pinpoint infrastructure problems and identify their potential effect on customers, services, and service level agreements (SLAs).

Figure 1 below shows the main SQM components.



**Figure 1 Service Quality Manager main components**

Service Adapters have two main interfaces to SQM:

- An interface to the Configuration and Administration layer
- An interface to the Service Level Monitoring (SLM) layer

The Configuration and Administration layer contains the Configuration Repository. This repository stores information on how to connect to OpenView Operations as well as how to map the raw data from OVO to the SQM parameter format.

The Service and Reporting Database contains the SQM Service Repository, which notably stores the parameters that are to be collected by the Service Adapter.

The layer receives all measurement data from the Service Adapter. It then calculates the status of each service to determine whether a service has failed to meet a service level (SL).

## 1.2 The Service Adapter

SQM uses a Service Adapter to connect to OVO and collect data. A Service Adapter serves as a bridge between SQM and OVO, collecting data from OVO, calculating quality indicators and feeding them to SQM.

### 1.2.1 What the Service Adapter does

The OVO Service Adapter provides fault statistic indicators deduced from the data stored in the OVO database. With OVO, the customer can detect faults occurring on a network or a system. This Service Adapter computes statistical indicators for the faults occurring on elements managed by OVO.

### 1.2.2 OpenView Operations overview

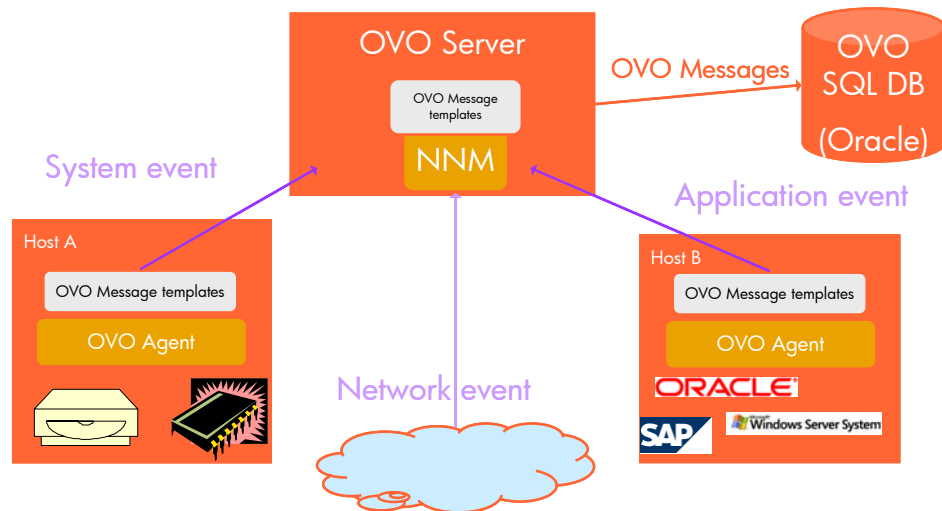
HP OpenView Operations (OVO) is a distributed client-server software solution designed to help system administrators detect, solve, and prevent problems occurring in networks, systems, and applications in any enterprise.

The OVO management concept is based on communication between a management server and managed nodes. Management server processes running on the central management server communicate with OVO agent processes running on managed



nodes throughout the environment. The OVO agent processes collect and process events on the managed nodes, and then forward relevant information in the form of OVO messages to the management server. A database serves as the central data repository for all OVO messages and configuration data. The database processes run on the management server.

**Figure 2 OpenView Operations overview**



### 1.2.3 OVO integration as a data source for SQM

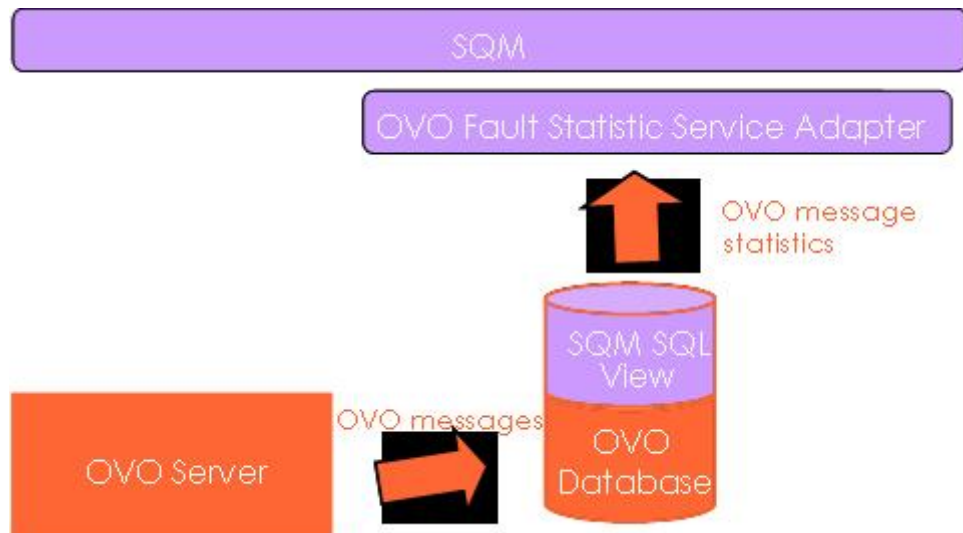
Integration is achieved through the OVO SQL database by implementing an SQM SQL Service Adapter and the OVO Fault Statistic Service Adapter, and by defining on the OVO SQL database a dedicated SQL view that computes statistics on the OVO messages. Note that the integration focuses only on active OVO messages and not on historical data, to reflect the status of current OVO nodes only.

This SQL view, named `SQM_OVO_FAULT_STAT`, computes for each OVO managed node defined on the OVO Server the following statistics on the OVO messages associated to the node:

- Worst severity
- Severity of the last message
- Number of messages for each level of severity
- Number of outstanding messages
- Problem duration

The SQM OVO Fault Statistics Service Adapter retrieves this statistical data and feeds it to the SQM. This integration is illustrated in Figure 3.

These statistics provide a snapshot of the faults occurring on the nodes managed by OVO.



**Figure 3 OVO/SQM integration**

## 1.2.4 Product deployment

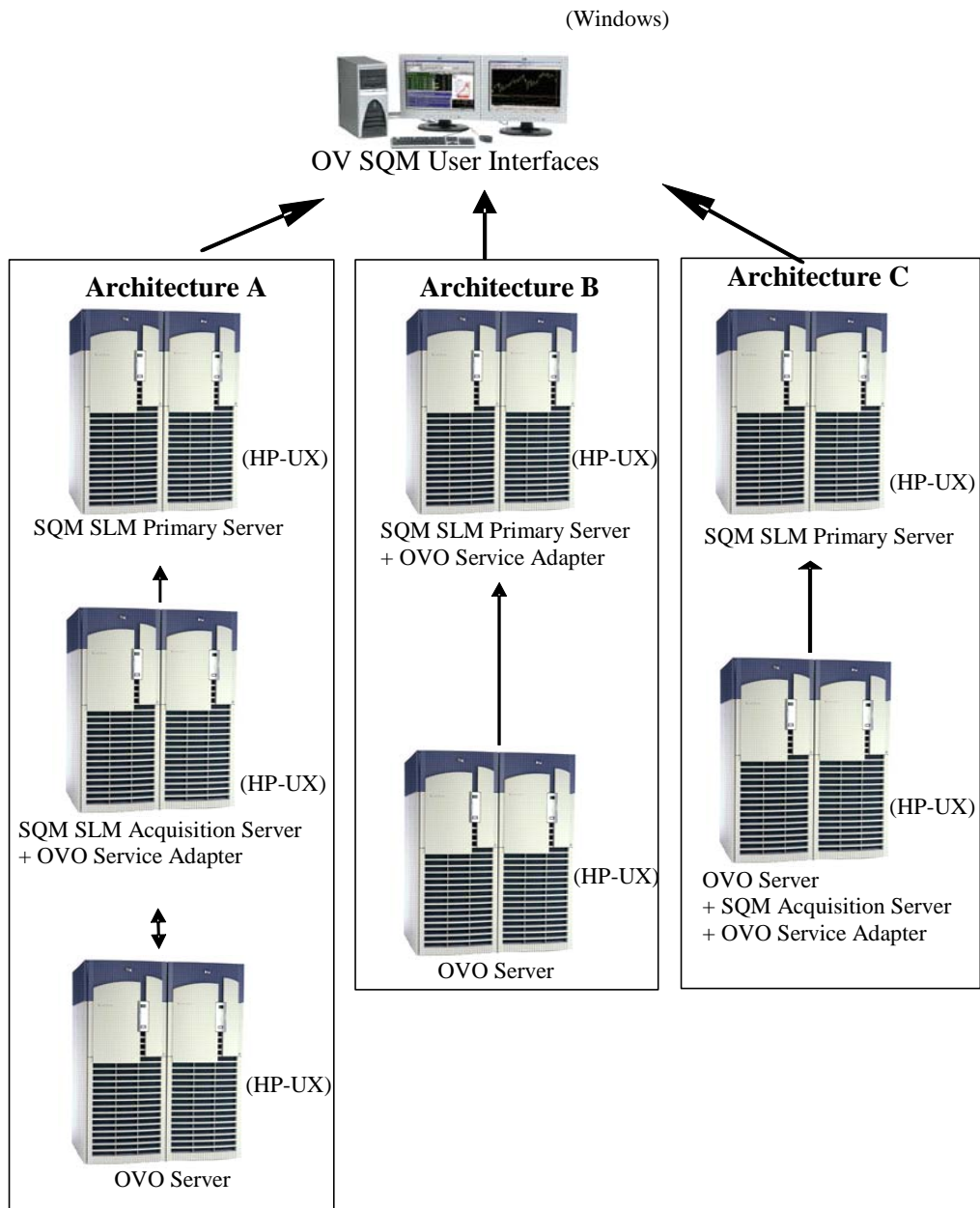
Once OVO, SQM and the Service Adapter are installed and configured, the Service Adapter starts to collect the specified quality indicators for SQM. The Service Adapter functions as an independent unit.

Each configured installation is called a Service Adapter Application. The Service Adapter Application connects to the OVO database using Java Database Connectivity (JDBC), and then collects data from the database.

You can install the Service Adapter on a computer that is not part of the OVO environment. This might be useful if you want to balance network load or have dedicated computers for specific tasks, for example. The Service Adapter Application can also connect across a firewall, providing greater flexibility in a secure environment.

Figure 4 below shows how the Service Adapter works with SQM and OVO.

**Figure 4 Service Adapter deployment**



### 1.2.5 OVO data collection

The Service Adapter uses data feeders to collect data. A data feeder is a source of data for SQM. Each data feeder is defined by a Data Feeder Definition (DFD). Some DFD related terms are explained below.

- **DFD**

The DFD is a data structure that defines what source data is collected from OVO. The DFD configuration information is stored both in the SQM Service Repository Manager and in the SQM Central Repository.

- **DFD parameters**

These parameters define what measurement data is sent by the Service Adapter to SQM. The DFD parameters are stored in the SQM Service Repository Manager's repository.

- **DFD properties**

DFD properties further define what source data is collected from OVO. They define the OVO managed node and are stored in the SQM Service Repository Manager’s repository.

- **Measurement Reference Point**

The Measurement Reference Point (MRP) identifies the source of the data that is to be collected.

The following tables show the DFD configuration of the OVO Fault Statistics Service Adapter.

## 1.2.6 Generic DFD configuration

**Table 1 DFD configuration**

Name	Type	Description	Value
<i>DfdName</i>	String	DFD’s name, used to uniquely identify the DFD within SQM. This name does not exceed 16 characters.	OVOSTATDFD
<i>DfdLabel</i>	String	DFD’s name, used for display purposes	OVO Statistics DFD
<i>Descr</i>	String	Description of the Data Feeder	This DFD collects OVO message statistics for an OVO managed node
<i>DfdVersion</i>	String	Version of the Data Feeder	“v1_0” by default for the first version of this SA
<i>SaName</i>	String	Service Adapter name	Ovo
<i>SaLabel</i>	String	Service Adapter’s name, used for display purposes	Equal to SaName by default
<i>SaVersion</i>	String	Version of the Service Adapter	“v1_0” by default for the first version of this SA
<i>Parameter Defs</i>	List	List of parameters for this DFD. Each parameter is composed of characteristics that are described below.	See “DFD parameters” on page 12.
<i>Property Defs</i>	List	List of property names filled at Data Feeder instantiation time. Each property name is a string.	See “DFD Properties” on page 12.
<i>MRP Naming Schema</i>	List	Sub-list of Data Feeder “Instance properties” that are used to construct the MRP (Measurement Reference Point). MRP properties must be	See “Measurement Reference Point” on page

Name	Type	Description	Value
		chosen from among the DFI properties so that uniqueness of a given DFI relative to all DFIs within the DF taken as a whole can be guaranteed. Each property name is a string. The order of the property names in this list is used to construct the MRP.	15.

## 1.2.7 DFD parameters

**Note:** The default values shown in the following table, “DFD parameters”, are displayed when the parameter concerned cannot be retrieved from the database. For instance, when there is no OVO message for a given OVO managed node, the *LAST\_SEV* parameter is set to Normal.

**Table 2 DFD parameters**

Parameter Name	Parameter Label	Data type	Description	Default value
<i>NB_OUTSTANDING</i>	Number of outstanding messages	Counter	Number of outstanding OVO messages on the managed node	0
<i>PB_DURATION</i>	Problem duration	Relative Time	Elapsed time between the oldest OVO message and the collection time	0
<i>WORST_SEV</i>	Worst Severity	Enum: Normal=0 Unknown=1 Warning=2 Minor=3 Major=4 Critical=5	Worst severity in all of the node’s active messages	Normal (0)
<i>WORST_SEV_TEXT</i>	Worst severity message text	String	OVO Message text displayed for the message with the worst severity	“No Message text available.”
<i>WORST_TIMESTAMP</i>	Worst Message’s Creation Time	AbsTime	Reaction time for the message with the worst severity	No default value

Parameter Name	Parameter Label	Data type	Description	Default value
<i>LAST_SEV</i>	Last message's severity	Enum: Normal=0 Unknown=1 Warning=2 Minor=3 Major=4 Critical=5	Severity of the last message	Normal (0)
<i>LAST_TIMESTAMP</i>	Last message creation time	AbsTime	Time at which last message received was created (in GMT)	No default value
<i>LAST_MSG_TEXT</i>	Last message's text	String	Text associated with last message received	"No Message text available."
<i>NB_SEV_NORM</i>	Number of normal messages	Int	Number of Normal active messages	0
<i>NB_SEV_UNK</i>	Number of unknown messages	Int	Number of Unknown active messages	0
<i>NB_SEV_WAR</i>	Number of warning messages	Int	Number of Warning active messages	0
<i>NB_SEV_MIN</i>	Number of minor messages	Int	Number of Minor active messages	0
<i>NB_SEV_MAJ</i>	Number of major messages	Int	Number of Major active messages	0
<i>NB_SEV_CRIT</i>	Number of critical messages	Int	Number of Critical active messages	0

## 1.2.8 DFD properties

Table 3 DFD Properties

Property Name	Property Label	Description	Data type
<i>NODE_ID</i>	Node Identifier	OVO node's Unique Identifier	String
<i>NODE_NAME</i>	OVO node name	OVO managed node's name	String
<i>LABEL</i>	OVO node label	Label of the OVO managed node	String

Property Name	Property Label	Description	Data type
<i>MACHINE_TYPE</i>	OVO machine type	Machine name (for example “Sun SPARCstation”)	String
<i>OS_NAME</i>	Operating System Name	Name of the OVO node’s operating system	String
<i>GROUPS</i>	OVO node groups	List of OVO node groups in which the node is defined	String

## 1.2.9 Measurement Reference Point

The Service Adapter collects performance measurements for a set of DFIs. Each set of measurements sent is then uniquely identified through its Measurement Reference Point (MRP).

An MRP serves as an identifier for a Data Feeder Instance (DFI). It is constructed by assembling a subset of the properties, following a pattern defined within the Data Feeder Definition.

It is used to distinguish easily between two Data Feeder Instances of the same Data Feeder Definition (DFD) and version.

During the raw information discovery phase, the DFI identifiers are constructed from the MRP and the DFD name and version.

When measurements are collected, a DFI’s MRP is included in every performance measurement message.

The OVO Service Adapter uses the MRP as a Unique Identifier for the OVO node.

An example of such a node UUID is **4e155362-c443-71d8-10de-10bc9e1b0000**





# Chapter 2

## Installing the Service Adapter

This chapter describes how to install the Service Adapter on HP-UX. After you have completed the installation, you must follow the instructions in the next chapter to configure the Service Adapter.

### 2.1 Software and Hardware requirements

#### 2.1.1 Software requirements

- HP-UX V11.11
- HP OpenView Operations A.07.10 or later
- Oracle 8.1.2 or 9.2.0.6 (Oracle 9.2 support is available with the OVO A.07.12 Management Server patch)
- HP OpenView Service Quality Manager V1.2 (Kernel subset)
- HP OpenView SA Common V1.2 (*SQMSAGTWCOMMON*)
- HP OpenView SQL Service Adapter Runtime V1.2 (*SQMSASQL*)
- Oracle JDBC driver 9.0.2 (the driver is provided in the OVO SA subset)

#### 2.1.2 Hardware requirements

For complete hardware requirements, see the *HP OpenView Service Quality Manager Installation Guide*.

- 5 MB of available hard disk space (10 MB recommended).

### 2.2 Installing the OVO Fault Statistics SA

If the OV SQM Kernel has not been installed already, you must install it before you install the Service Adapter, by following the instructions in “Installing the OV SQM Kernel” below.

#### 2.2.1 If the OV SQM Kernel has been installed already, go directly to “Required environment

1. **Sqmadm user/group**  
'sqmadm' group and user is necessary for the kernel/application setup and management. Please refer to *HP OpenView Service Quality Manager Installation Guide* for instructions on how to create the 'sqmadm' group and user.
2. Environment variables

If the kernel setup has already been done, you have to source the **temip\_sc\_env.sh** file located under \$TEMIP\_SC\_VAR\_HOME directory.

```
# . $TEMIP_SC_VAR_HOME/temip_sc_env.sh
```

If the kernel setup is not yet done (and before to install the SA Common and SA Ovis), you have to set your environment. Perform the following actions:  
- source environment variables:

```
# export TEMIP_SC_HOME=/opt/OV/SQM<xxx>  
# . $TEMIP_SC_HOME/jre/jre-setup.sh
```

Installing the SA Common subset” on page 18.

## 2.2.2 Installing the OV SQM Kernel

The SQM Core Kernel subset is a prerequisite for installation of the OVO SA.

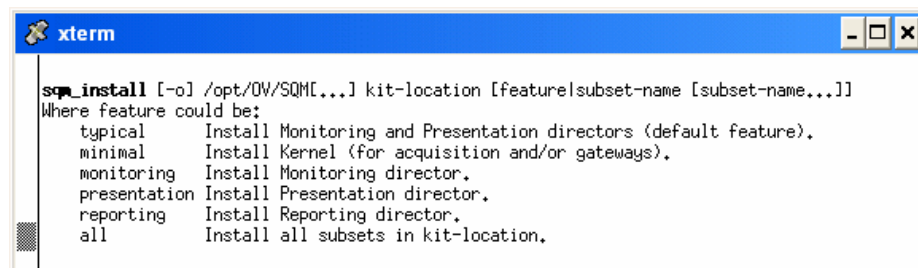
To install the Kernel, do the following.

1. Create the SQM home directory, if it does not already exist. Under the **root** account, enter the following command:

```
# mkdir /opt/OV/SQM<xxx>
```

where <xxx> is the label you want to assign to the current release.

2. Mount the SQM Core CD-ROM on your system.
3. Go to <mount directory>/SQM-1.20.00  
and run the *sqm\_install* tool under the **root** account.



```
xterm  
sqm_install [-o] /opt/OV/SQM[...] kit-location [feature|subset-name [subset-name...]]  
Where feature could be:  
  typical      Install Monitoring and Presentation directors (default feature).  
  minimal      Install Kernel (for acquisition and/or gateways).  
  monitoring    Install Monitoring director.  
  presentation  Install Presentation director.  
  reporting     Install Reporting director.  
  all           Install all subsets in kit-location.
```

4. Select the *minimal* feature by entering the following command:

```
# sqm_install /opt/OV/SQM<xxx> <mount directory>/SQM-1.20.00/HP  
UX/KIT minimal
```

5. Press **Enter** to install the Kernel.

## 2.2.3 Required environment

3. **Sqmadm user/group**  
'sqmadm' group and user is necessary for the kernel/application setup and management. Please refer to *HP OpenView Service Quality Manager Installation Guide* for instructions on how to create the 'sqmadm' group and user.
4. Environment variables

If the kernel setup has already been done, you have to source the **temip\_sc\_env.sh** file located under \$STEMIP\_SC\_VAR\_HOME directory.

```
# . $STEMIP_SC_VAR_HOME/temip_sc_env.sh
```

If the kernel setup is not yet done (and before to install the SA Common and SA Ovis), you have to set your environment. Perform the following actions:  
- source environment variables:

```
# export TEMIP_SC_HOME=/opt/OV/SQM<xxx>  
# . $STEMIP_SC_HOME/jre/jre-setup.sh
```

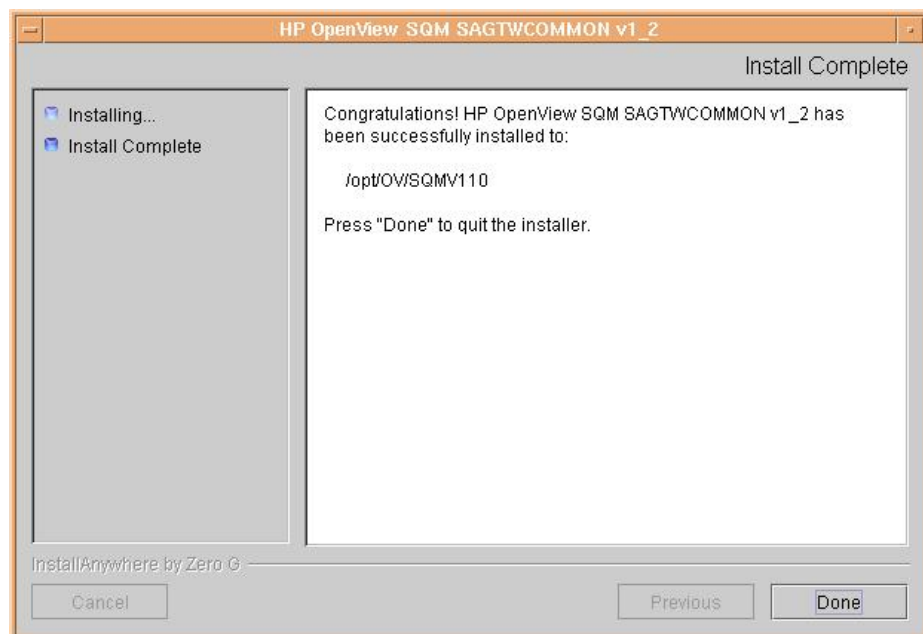
## 2.2.4 Installing the SA Common subset

If necessary, install the SA Common component by doing the following. If this has already been done, go directly to “Installing SQL Service Adapter Runtime” on page 20.

1. First, log on to the system as **root** user.
2. Mount the HP OpenView Service Adapters and Gateways CD-ROM on your system.
3. Go to `<mount directory>/SQM-1.20.00` and execute the following command:

```
# ./SQMSAGTWCCOMMON-1.20.00.bin
```

4. The software is installed and the **Install Complete** window is displayed.



## 2.2.5 Installing SQL Service Adapter Runtime

If necessary, install the SQL Service Adapter Runtime component by doing the following. If this has already been done, go directly to “Installing the OVO Fault Statistics Service Adapter” on page 17.

1. First, log on to the system as **root** user.
2. Mount the HP OpenView SQM Service Adapters and Gateways CD-ROM on your system.
3. Go to `<mount directory>/SQM-1.20.00`
4. Run the installer by entering the following command:

```
# ./SQMSASQL-1.20.00.bin
```

5. The software is installed and the **Install Complete** window is displayed.



6. To finish, click **Done**.

## 2.2.6 Installing the OVO Fault Statistics Service Adapter

To install the OVO SA on HP-UX, do the following.

1. Log on as **root** user.
2. Mount the HP OpenView SQM Service Adapters and Gateways CD-ROM.
3. Go to `<mount directory>/SQM-1.20.00`
4. Run the installer by entering the following command:

```
# ./SQMSAOVO-1.20.00.bin
```

5. The software is installed and the **Install Complete** window is displayed:



6. To end the installation process, click **Done**.
7. Execute the installation script to set the correct access rights to the installed files by entering the following commands:

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin
# sh temp_sc_complete_install.sh
```

## 2.2.7 Patch installation

To install a Patch, do the following.

1. Log on as **root** user.
2. Go to `<mount directory>/SQM-1.20.00`
3. Run the `sqm_install` tool with the `-p` option and the `minimal` feature by entering the following command:

```
# sqm_install -p /opt/OV/SQM<xxx> /SQM-1.20.00/HPUX/Patches minimal
```

This will install all Patches for the Kernel component.

4. Run the `sqm_install` tool with the `-p` option and the `SAGTWCCOMMON` feature by entering the following command:

```
# sqm_install -p /opt/OV/SQM<xxx> /SQM-1.20.00/HPUX/Patches SAGTWCCOMMON
```

This will install all Patches for the `SAGTWCCOMMON` component.

5. Run the `sqm_install` tool with the `-p` option and the `SQMSASQL` feature by entering the following command:

```
# sqm_install -p /opt/OV/SQM<xxx> /SQM-1.20.00/HPUX/Patches SQMSASQL
```

This will install all Patches for the `SQMSASQL` component.

6. Run the `sqm_install` tool with the `-p` option and the `SAOVO` feature by entering the following command:

```
# sqm_install -p /opt/OV/SQM<xxx> /SQM-1.20.00/HPUX/Patches SAOVO
```

This will install all Patches for the *OVO* component.

The installation on Unix is now complete. You must next configure the Service Adapter by following the instructions in Chapter 3.

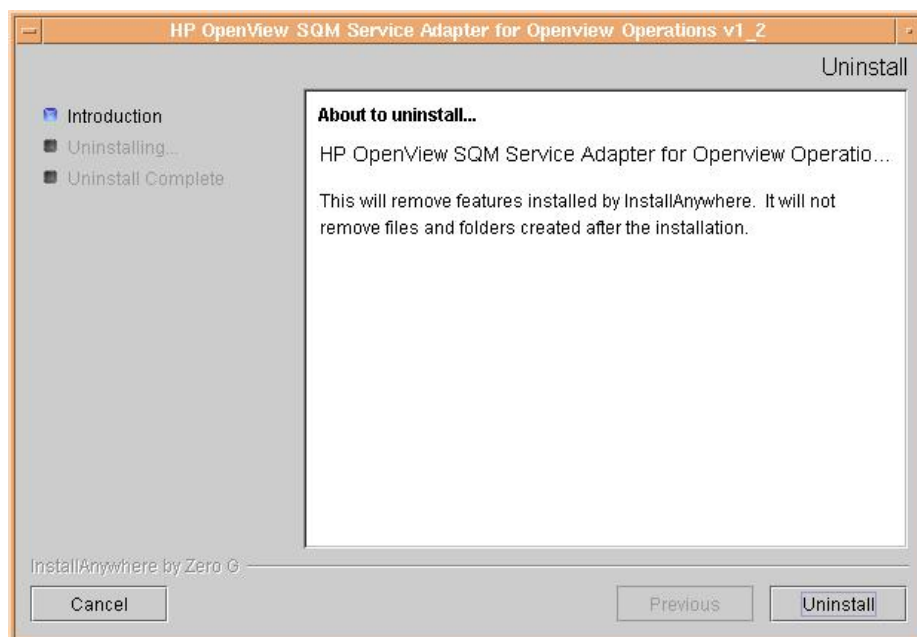
## 2.2.8 Uninstalling the software

To uninstall the OVO SA software, do the following.

1. Log on as **root** user.
2. Load the SQM environment variables  
(`$TEMIP_SC_VAR_HOME/temip_sc_env.sh`)
3. Enter the following two commands on the SQM platform where the OVO SA has been installed:

```
# cd $TEMIP_SC_HOME
#
./ServiceAdapters/Sql/v1_2/Ovo_v1_2/UninstallerDataOvo/Uninstall_Ovo
```

The **Uninstall** window is displayed.



4. If no additional SQL Service Adapters are running on your system, uninstall the SQL SA v1\_2 Runtime kit by entering the following two commands:

```
# cd $TEMIP_SC_HOME
# ./ServiceAdapters/Sql/v1_2/UninstallerDataSql/Uninstall_Sql
```

5. If no Service Adapters or Gateways are running on your system, uninstall the SA Common v1\_2 kit by entering the following two commands:

```
# cd $TEMIP_SC_HOME
```

```
#!/ServiceAdapters/Common/v1_2/Uninstaller_SAGTWCCOMMON/Uninstall_SAGTWCCOMMON
```





## Chapter 3

# Setting up and configuring the Service Adapter

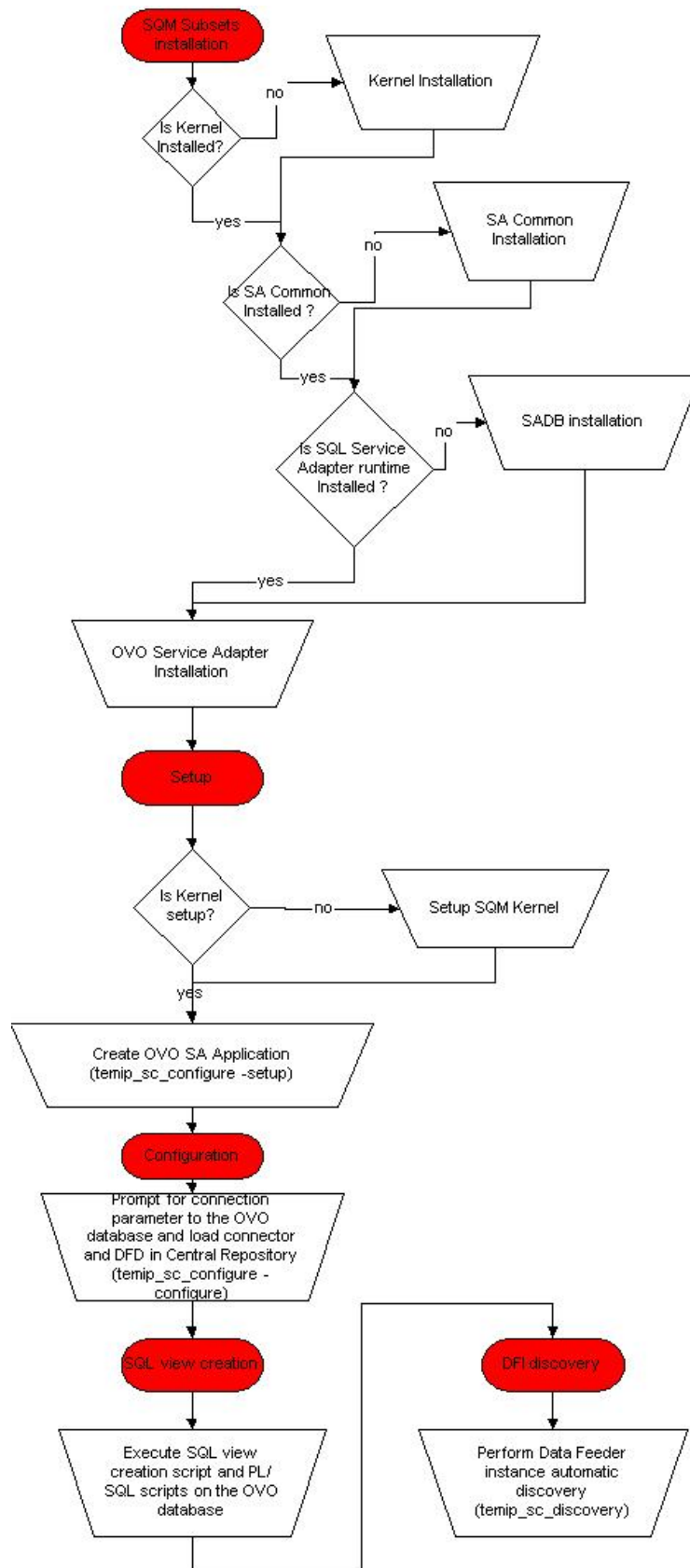
The OVO Fault Statistics Service Adapter provides a set of tools to ease product installation, setup and configuration, for creating the application, loading DFD definition and connection information into the central repository, executing SQL view creation scripts on the OVO SQL Server and also for automatically discovering Data Feeder Instances from the OVO SQL Server.

Figure 5 shows the steps you take to configure the Service Adapter.

The OVO Service Adapter is set up and configured in four steps:

1. Setting up the application by declaring the OVO SA application into the SQM Central Repository and creating the OVO SA application Data Tree in `TEMIP_SC_VAR_HOME`
2. Configuring the application by replying to user prompts requesting the connection settings to the OVO database, and loading the connection and Data Feeder definitions into the SQM Central Repository.
3. Creating the database views by executing the SQM SQL view creation files supplied in the OVO SA directory:  
`$TEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/database`
4. Discovering and loading DFIs. DFI discovery is an important feature provided by the OVO Service Adapter. This feature is used to discover and load the Data Feeder Instances that will be managed by the OVO SA application, completely automatically. This is done by querying the OVO database configuration and loading into SQM a DFI for each OVO managed node.

**Figure 5 OVO SA configuration overview**



## 3.1 Creating the application

If your Kernel is not yet set up, you must configure it by doing the following. If your Kernel is set up already, go directly to “Setting up the required environment” on page 27.

---

### Caution

---

A Service Adapter application name has to be unique cross platform and director!

---

### 3.1.1 Configuring the SQM Kernel

You must set up the SQM Kernel before the OVO SA can be configured and run. You can set up the SQL Kernel in any of the following configurations:

- The OVO SA is installed on the HP-UX SQM SLM Primary Server. (For instructions on how to set up the SQM Server, see the *HP OpenView SQM Installation Guide*.)
- The OVO SA is installed on a HP-UX system separate from the SQM SLM Primary Server on which the SQM Kernel has not yet been configured. In this case, you must retrieve the SLM Server platform description file by doing the following.
  - Create the **sqmadm** administration user on the target Unix system. (For instructions on how to create the user account, see the *HP OpenView SQM Installation Guide*.)
  - From the SQM SLM Primary Server, retrieve the file `$TEMIP_SC_VAR_HOME/setupconfig/platform_desc.cfg` and copy it to the SQL SA customization HP-UX system in `$TEMIP_SC_HOME/tmp`
  - Log on as **root** user, and then enter the following commands:

```
# export TEMIP_SC_HOME=<SQM installation directory>
# cd $TEMIP_SC_HOME/setup/bin
# temip_sc_setup -all -NI
```

### 3.1.2 Setting up the required environment

To set up the required environment, you must do the following.

1. Create the **Sqmadm** group and user.

The **sqmadm** group and user are used to set up and manage the kernel and application. See the *HP OpenView Service Quality Manager Installation Guide* for instructions on how to create the **sqmadm** group and user.

2. Set up the environment variables.

If the kernel has been set up already, you must source the `temip_sc_env.sh` file contained in the `$TEMIP_SC_VAR_HOME` directory by entering the following command:

```
# . $TEMIP_SC_VAR_HOME/temip_sc_env.sh
```

### 3.1.3 Creating the OVO SA application

This phase consists in creating an OVO SA application on the SQM platform (on a specified director).

To create the OVO SA application, do the following.

1. Log on as **root** user.
2. Enter the following two commands:

```
# cd $STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin
# temip_sc_configure.sh -setup -dirName <director name>
<application name>
```

where:

- *<director name>* is the director on which the application will be created. (By default, the director name is **acquisition**)
- *<application name>* is the application name provided by the user.

The second command declares the OVO Service Adapter application to the SQM Central Repository and creates the application's Data Tree in the following directory:

```
$STEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2
```

The following output is displayed.

```
[sqmadm_hotel1]/opt/OV/SQMv110/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin
>temip_sc_configure.sh -setup OVOSA

Setup the "OVOSA" application ...
Create the application (this may take several minutes).
Create the datatree.
/var/opt/OV/SQM/slmv12/ServiceAdapters (already exist)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql (already exist)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2 (already exist)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2 (already
exist)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/config
(already exist)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery
(created)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/
filter (created)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/
filter/slmv12_acquisition_OVOSA_filter.sh (created)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/
inventory (created)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/
inventory/raw (created)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/
inventory/filtered (created)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/
repository (created)
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/repository
(created)
Initializing Discovery Tracing and Logging ...
Setup succeed.

Update the access permissions on the datatree... succeed
```

## 3.2 Configuring the application

Configuration consists in defining the OVO SA connection parameters and loading the connection and Data Feeder configuration into the SQM Central Repository.

To configure the application, do the following.

1. Log on as **root** user.
2. Load the SQM environment variables by entering the following command:

```
# . $STEMIP_SC_VAR_HOME/temip_sc_env.sh
```

3. Enter the following two commands:

```
# cd $STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin
# temip_sc_configure.sh -configure <application name>
```

where *<application name>* is the application name entered in the setup command.

The second command prompts the user for SQL database connection information (the login, password, database URL, and JDBC driver), and then loads the SQL Service Adapter's Data Feeder definitions and connector definition into the SQM Central Repository.

The following default connection parameters must be provided for the OVO database:

- URL: jdbc:oracle:thin:@<OVO database hostname>:1521:openview
- Username: opc\_op
- Default Password: OpC\_op
- JDBC Driver: oracle.jdbc.driver.OracleDriver

The following output is displayed:

```
[sqmadm_hotel]/opt/OV/SQMv110/ServiceAdapters/Sql/v1_2/Ovo_v1_2
/bin>temip_sc_configure.sh -configure OVOSA

Configure the "OVOSA" application ...
Please enter the database URL
[jdbc:oracle:thin:@holly.vbe.cpqcorp.net:1521:openview]:
Please enter the database username [opc_op]:
Please enter the database password [*****]:
Please enter the database JDBC Driver
[oracle.jdbc.driver.OracleDriver]:
Load the Connector in the Tibco Repository
Backup written at the following location:
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/reposi
tory/connectors_data.exp.2004_8_5_17_27_53
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/reposi
tory/connectors_data.exp has been imported into the Repository
Load the Data Feeder Definitions in the SRM
load DFD: OVOSTATDFD - v1_0
(../repository/NewDFDReq_OVOSTATDFD.v1_0.xml) ... succeed.
Load the Data Feeder Definitions in the Tibco Repository
Backup written at the following location:
/var/opt/OV/SQM/slmv12/ServiceAdapters/Sql/v1_2/Ovo_v1_2/reposi
tory/Ovo_dfds_data.exp.2004_8_5_17_28_18
../repository/Ovo_dfds_data.exp has been imported into the
Repository
Configuration succeed.

Update the access permissions on the datatree... succeed.
```

## 3.3 Creating the SQL view

This step must be performed only once, when you create the first instance in the OVO database, even if multiple OVO Service Adapter instances are created. This is because all OVO SA instances connected to an OVO database use the same SQL view.

OVO message statistics are computed through a SQL view and PL/SQL scripts which must be deployed in the OVO database. These scripts are located in the following directory:

```
/opt/OV/SQMV110/ServiceAdapters/Sql/v1_2/Ovo_v1_2/database
```

To deploy these scripts, enter the following two commands:

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin
# temp_sc_configure.sh -view <application name>
```

where *<application name>* is the application name entered at the setup command.

The second command prompts the user for SQL database connection information (the login, password, database URL, and JDBC driver). The specified user must have creation privileges on the OVO database to execute these scripts. By default, the user **opc\_op** has these privileges.

The following default connection parameters must be provided for the OVO database:

```
Username:      opc_op
```

```
Default Password: OpC_op
```

The database URL and JDBC Driver information are not requested here, because the values specified to enable OVO SA to connect to the database are reused.

The following output is displayed.

```
[sqmadm hotel]/opt/OV/SQMV110/ServiceAdapters/Sql/v1_2/Ovo_v1_2
/bin>temp_sc_configure.sh -view OVOSA2

Create the view ...
=====
Executing SQL Script on remote SQL database.
=====
=> Please enter the database username: opc_op
=> Please enter the database password: OpC_op
INFO: Executing SQL scripts on
jdbc:oracle:thin:@holly.vbe.cpqcorp.net:1521:openview ...
INFO: Executing script
'../database/sqm_ovo_sa_01_drop_table_def.sql' ...
INFO: Script execution succeed.

INFO: Executing script
'../database/sqm_ovo_sa_02_record_def.sql' ...
INFO: Script execution succeed.

INFO: Executing script
'../database/sqm_ovo_sa_03_table_def.sql' ...
INFO: Script execution succeed.

INFO: Executing script '../database/sqm_ovo_sa_04_declare.sql'
...
INFO: Script execution succeed.

INFO: Executing script '../database/sqm_ovo_sa_05_body.sql' ...
INFO: Script execution succeed.
```

```
INFO: Executing script
'../database/sqm_ovo_sa_06_discovery_view.sql' ...
INFO: Script execution succeed.

INFO: Executing script
'../database/sqm_ovo_sa_07_statistic_view.sql' ...
INFO: Script execution succeed.

View creation succeed.
```

---

### Possible error

---

When the SQL View creation tool is run for the first time, the following error is displayed.

```
Failed to process request:--
=====
-----
-- Drop the definition of the statistic table type
--
=====
=====
DROP TYPE SQM_OVO_STAT_TABLE. Error:java.sql.SQLException: ORA-
04043: object SQM_OVO_STAT_TABLE does not exist
.
ERROR: Script execution failed.      (4)
```

This error does not affect the SQL View creation process and can be ignored.

---

## 3.4 Discovering and loading DFIs

Automatic discovery of DFIs is an important feature provided by the OVO Service Adapter. It enables the Data Feeder Instances that will be managed by the OVO SA application to be discovered and loaded fully automatically.

### Discovery script

The discovery script is contained in the following directory:

```
$STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temp_sc_d
iscovery.sh
```

### General processing

There are three steps in the discovery process:

1. **Raw discovery phase**, in which the SQL discovery request is executed and details of all DFIs discovered are retrieved in a raw (unprocessed) inventory file.
2. **Filtering phase**, in which a script that filters the DFIs declared in the raw inventory file is executed. This generates a new filtered inventory file containing details of only DFIs that are selected for management using the application. By default, no filtering is performed. This script must therefore be modified to filter the details contained in the raw inventory file.

3. **Loading phase**, in which the filtered DFIs are loaded into the SQM Central Repository, according to one of the following algorithms.

- *-diff no*

All of the filtered Data Feeder Instances are loaded into the SQM Central repository if this option is selected.

- *-diff offline*

The list of all discovered/filtered Data Feeder Instances are compared with a discovery reference file (provided by the user) if this option is selected.

If a Data Feeder Instance exists in the inventory file but does not exist in the reference file, the Data Feeder Instance is created.

If the Data Feeder Instance does not exist in the inventory file but exists in the reference file, the Data Feeder Instance is deleted from the SQM repository.

If the Data Feeder Instance exists in both the inventory file and the reference file), it is not reloaded.

- *-diff online*

The same Data Feeder Instance comparisons are performed as those in offline mode if this option is selected. Instead of checking the DFIs against a reference file, however, the declaration depends in this case on whether the Data Feeder Instance exists in the SQM Central Repository.

---

#### Note

---

The next chapters provide details of each phase described above.

The same processing can be performed by executing a single command (by default, all filtered Data Feeder Instances are loaded, through the *-diff no* switch). See “One-shot discovery and loading” on page 36 for details of this command.

---

### 3.4.1 Raw discovery phase

In this initial phase, the discovery query definition file is loaded and the discovery query is executed to retrieve all OVO SA Data Feeder Instance definitions.

A dedicated SQL view was deployed on the OVO Server when you created the SQL view. This view is named `SQM_OVO_NODES`. It is used with the OVO configuration SQL table to retrieve OVO node information including the node name, label, and machine type.

The raw data is discovered in this SQL view through the following SQL query:

```
SELECT DISTINCT NODE_NAME, LABEL, MACHINE_TYPE_STR, OS_NAME,
GROUPS, NODE_ID FROM SQM_OVO_NODES
```

This query is defined in the following file:

```
$STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/properties/TeS
CSqlDiscovery.xml
```



---

## Tips

---

A simple way to perform filtering is to update the SQL script defining the discovery view, by adding further conditions to the SQL 'WHERE' clause in the default SQL query, instead of using a filtering script. You must take care to ensure that you only filter out certain OVO nodes when you modify this script. See "Customizing the Discovery view" for more details.

---

### Command

To execute the discovery query, do the following.

1. Log on as **sqmadm** user.
2. Load the SQM environment variables by entering the following command:

```
# . $STEMIP_SC_VAR_HOME/temip_sc_env.sh
```

3. Enter the following commands:

```
# cd $STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin
# temip_sc_discovery.sh -platform <platform name> -director <director name> -application <application name> -discover
```

where:

- *<platform name>* is the name that was defined when the SQM Server was set up and can be read from the variable *\$KERNEL\_ID*.
- *<director name>* is the director on which the application was created during the setup phase. (By default, the director name is **acquisition**).
- *<application name>* is the name that was defined when the application was set up.

The following files are generated in the raw discovery phase.

- The discovered DFI inventory file:  
\$STEMIP\_SC\_VAR\_HOME/ServiceAdapters/Sql/v1\_2/Ovo\_v1\_2/discovery/inventory/raw/<platform name> <director name> <application name>.xml
- The DFI declaration and deletion XML files:  
\$STEMIP\_SC\_VAR\_HOME/ServiceAdapters/Sql/v1\_2/Ovo\_v1\_2/discovery/repository/DeclareDFIReq\_OVOSTATDFD.v1\_1.<DFIID>.xml  
\$STEMIP\_SC\_VAR\_HOME/ServiceAdapters/Sql/v1\_2/Ovo\_v1\_2/discovery/repository/DelDFIReq\_OVOSTATDFD.v1\_1.<DFIID>.xml

## 3.4.2 Filtering phase

In the discovery filtering phase, the discovery tool executes a filtering script.

This script parses the raw discovery file output from the previous command. It removes all DFI definitions that are not managed by the OVO SA application. The script can be used to select only OVO managed nodes that belong to a specific OVO node group, for example.

This filtering is mainly used for load balancing (to share the DFI load over several OVO SA applications).

The script generates a new DFI inventory file containing only DFIs that are managed by the OVO SA application.

A filtering script is provided with the OVO SA. This default script simply copies the input raw inventory file to the filtered inventory file without applying any filtering. You must adapt this script according to your needs.

The filtering script is contained in the following file:

```
$STEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/  
filter/<platform name>_<director name>_<application name>_filter.sh
```

---

### Note

The filtering script can be customized by the integrator. It accepts two input arguments:

- Full pathname of the raw inventory file
- Full pathname of the filtered inventory file generated

An example of a filtering script is provided in the following directory:

```
$STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/examples
```

---

The raw DFI inventory file's pathname is:

```
$STEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/inv  
entory/raw/<platform name>_<director name>_<application name>.xml
```

### Command

To filter the discovery file, do the following.

1. Log on as **sqmadm** user.
2. Load the SQM environment variables by entering the following command:

```
# . /var/opt/OV/SQM/s1mv12/temip_sc_env.sh
```

3. Enter the following commands:

```
# cd $STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin  
# temip_sc_discovery.sh -platform <platform name> -director <  
director name> -application <application name> -filter
```

where:

- *<platform name>* was defined when the SQM Server was set up and can be read from the variable *\$KERNEL\_ID*.
- *<director name>* is the director on which has been created the application at the setup phase. (By default, the director name is **acquisition**).
- *<application name>* is the name that was defined when the application was set up.

### Output

When the raw DFI discovery file has been filtered, the script generates the filtered inventory file and stores it at the following location:

```
$STEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_2/discovery/inventory/f  
iltered/<platform name>_<director name>_<application name>.xml
```

### 3.4.3 Loading phase

Depending on which “-diff” option is selected when the discovery script is launched, the following actions are performed (by default, the *-diff no* option is used to load all filtered Data Feeder Instances):

- *-diff no*

This option loads all the filtered Data Feeder Instances into SQM central repository.

- *-diff offline*

This option compares the list of discovered/filtered Data Feeder Instances against a DFI reference file.

If a Data Feeder Instance exists in the inventory file but does not exist in the reference file, the Data Feeder Instance is created.

If the Data Feeder Instance does not exist in the inventory file but it exists in the reference file, the Data Feeder Instance is deleted from the SQM central repository.

If the Data Feeder Instance exists in both the inventory file and the reference file, it is not reloaded.

- *-diff online*

The same Data Feeder Instance comparisons are performed as those in offline mode if this option is selected. Instead of checking the DFIs against a reference file, however, whether the DFIs are declared depends in this case on whether the Data Feeder Instance exists in the SQM Central Repository.

#### Input

The DFI filtered inventory file (the output from the previous command) is required as input for this phase.

Its path and name is:

```
$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/Ovo_v1_2/discovery/inventor  
y/filtered/<platform name>_<director name>_<application name>.xml
```

The inventory reference file is needed for the *-diff offline* loading option.

The file must be stored as follows:

```
$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/rep  
ository/<platform name>_<director name>_<application name>_discove  
ry_reference.xml
```

#### Command

To run the discovery loading query, do the following.

1. Log on as **sqmadm** user.
2. Load the SQM environment variables by entering the following command:

```
# . /var/opt/OV/SQM/slmv12/temip_sc_env.sh
```

3. Execute the following commands:

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin  
# temip_sc_discovery.sh platform <platform name> -director <d  
irector name> -application <application name> -load -  
diff [no|offline|online]
```

where:

- *<platform name>* is the name that was defined when the SQM Server was set up, and can be read from the variable *\$KERNEL\_ID*.
- *<director name>* is the director on which the application was created during the setup phase. (By default, the director name is **acquisition**.)
- *<application name>* is the name that was defined when the application was set up.

### Output

- The loading status of each DFI (Successful, Failure, or Partial) is logged. The discovery loading procedure logs the result of each DFI declaration in the following file:  
`$STEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/discovery/repository/<platform name>_<director name>_<application name>_discovery_cmds.log`
- If the script fails, the following script can be run manually by the user to restart the DFI loading process:  
`$STEMIP_SC_VAR_HOME/ServiceAdapters/Sql/Ovo_v1_2/discovery/repository/<platform name>_<director name>_<application name>_discovery_cmds.sh`

## 3.4.4 One-shot discovery and loading

If the user does not want to call each step in the DFI discovery process described above (discovery, filtering, and loading), DFI discovery can be performed by executing a single command as described below.

### Command

To perform DFI discovery, filtering and loading as a single process, do the following.

1. Log on as **sqmadm** user.
2. Load the SQM environment variables by entering the following command:

```
# . $STEMIP_SC_VAR_HOME/temip_sc_env.sh
```

3. Execute the following commands:

```
# cd $STEMIP_SC_HOME/ServiceAdapters/Sql/v1_1/Ovo_v1_2/bin  
# temip_sc discovery.sh -platform <platform name> -director <director name> -application <application name> -all
```

where:

- *<platform name>* is the name that was defined when the SQM Server was set up and can be read from the variable *\$KERNEL\_ID*.
- *<director name>* is the director on which the application was created during the setup phase. (By default, the director name is **acquisition**).
- *<application name>* is the name that was defined when the application was set up.

### Output

The one-shot discovery process performs the following tasks:

- Requesting discovery of all DFI and creating a raw discovery file
- Filtering all discovered DFIs by applying the appropriate filters

- If the default load option `-diff no` is selected, loading all discovered DFIs into the SQM Service Repository Manager

### 3.4.5 Scheduling DFI discovery

The OVO Service Adapter can be deployed in an OVO environment where the OVO configuration changes daily (with the addition of new OVO nodes, for example). In this situation, scheduling automatic discovery by running special scripts encapsulating all of the DFI discovery commands for running in a crontab is recommended.

Scheduled DFI discovery is run in batch mode, and loads newly discovered DFIs automatically from the OVO Server database.

## 3.5 Advanced configuration

See chapter 4.2.5 of the *SQL Service Adapter Toolkit Installation, Configuration and User's Guide*.



## Service Adapter operation

### 4.1 Starting and Stopping OVO SA

Starting and stopping an OVO Service Adapter application is done through the standard SQM management commands described in the *HP OpenView SQM Administration Guide*.

#### To start the application

To start the application, do the following.

1. Log on as **sqmadm** user.
2. Load the SQM environment variables by entering the following command:

```
# . $TEMIP_SC_VAR_HOME/temip_sc_env.sh
```

3. Enter the following command:

```
# temip_sc_start_application -platform <platform name> -director <director name> -application <application name>
```

where:

- *<platform name>* is the name that was defined when the SQM Server was set up and can be read from the variable `%KERNEL_ID%`
- *<director name>* is the director on which the application was created during the setup phase. (By default, the director's name is **acquisition**.)
- *<application name>* is the name that was entered when the application was set up

#### To stop the application

To stop the application, do the following.

1. Log on as **sqmadm** user.
2. Load the SQM environment variables by entering the following command:

```
# . $TEMIP_SC_VAR_HOME/temip_sc_env.sh
```

3. Enter the following command:

```
# temip_sc stop_application -platform <platform name> -director <director name> -application <application name>
```

where:

- *<platform name>* is the name that was defined when the SQM Server was set up, and can be read from the variable `%KERNEL_ID%`
- *<director name>* is the director on which the application was created during the setup phase. (By default, the director name is **acquisition**)
- *<application name>* is the name that was defined when the application was set up

## 4.2 Starting and Stopping collection by the Service Adapter

Collection of OVO data starts only when the Service Adapter Application is instructed to do so by the SQM Service Repository Manager. This is controlled by the SQM SLA Admin User Interface. For more information, see the *HP OpenView SQM SLA Admin User Interface User's Guide*.

## 4.3 Maintenance

### 4.3.1 Creating a new application

To create a new OVO SA application, do the following.

1. Create the SA application. For details, see “Creating the OVO SA application” on page 27.
2. Configure the SA Application. For details, see “Configuring the application” on page 29.
3. If the new application will be connected to an OVO Server that is not already linked to an OVO SA, create the SQL views. For details, see “Creating the SQL view” on page 30.
4. Discover and load the DFIs. For details, see “Discovering and loading DFIs” on page 31.
5. Start the SA application. For details, see “Starting and Stopping OVO SA” on page 39.

### 4.3.2 Deleting an existing application

To delete the Service Adapter application, do the following.

1. Log on as **sqmadm** user.
2. Load the SQM environment variables by entering the following command:

```
# . $TEMIP_SC_VAR_HOME/temip_sc_env.sh
```

3. Execute the following command:

```
# temip_sc delete_application -platform <platform name> -director <director name> -application <application name>
```



where:

- *<platform name>* is the name that was defined when the SQM Server was set up, and can be read from the *%KERNEL\_ID%* variable.
- *<director name>* is the director on which the application was created during the setup phase. (By default, the director name is **acquisition**)
- *<application name>* is the name that was defined when the application was set up.



## Service Adapter advanced customization

The use of an intermediate SQL view between the OVO Server and the OVO Service Adapter means it is possible to customize the PL/SQL scripts to:

- Perform special filtering when discovering which nodes are managed by OVO
- Filter out OVO messages that should not be included in statistic calculations

### 5.1 Customizing the Discovery view

The OVO Service Adapter Discovery tool discovers which nodes are managed by OVO by using an SQL query runs in a special SQL view called *SQM\_OVO\_NODES*. This view is created by the SQL script located in

```
$STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/sqm_ovo_sa_06_discovery_view.sql
```

The view joins the following OVO tables:

- *OPC\_NODES*: This table contains details of the nodes in the Node Bank for both external nodes and normal nodes.
- *OPC\_NODES\_NAMES*: This table contains the IP addresses (if the node is in an internet network) and the identifying name of nodes. It also contains the addresses for normal nodes contained in *opc\_nodes*. It additionally contains the node addresses of messages that matched an external node.
- *OPC\_NET\_MACHINE*: This table contains the main details for agent platforms available on this Management Server. It is used by the OVO GUI, to display the platform name and to define possible values, for example.

For more information about the OVO database structure, see *HP OpenView VantagePoint Operations for UNIX Reporting and Database Schema*.

By default, all nodes managed by OVO are discovered through this view. To filter out specific OVO nodes, you must do the following.

1. Edit the SQL query to modify the 'WHERE' clause so that specific OVO nodes are excluded by the filter.
2. Reload the script on the OVO database as described in "Creating the SQL view" on page 30.
3. Perform the DFI discovery process as described in "Discovering and loading DFIs" on page 31.

For instance, if you require fault statistics on a limited number of nodes, the recommended method is to create a OVO node group specifically for this purpose, put the targeted OVO node into this group and modify the SQL script to select only OVO nodes belonging to this group.

Where only statistics on nodes belonging to the group **hp\_ux** are required, for example, the SQL query defining the **SQM\_OVO\_NODES** view is as follows:

```
SELECT DISTINCT opc_nodes.node_id, node_name, OPC_NODES.label,  
machine_type_str, os_name,  
SQM_OVO_SA_PKG.get_groups(opc_nodes.node_id) groups  
FROM OPC_NODES, OPC_NODE_NAMES, OPC_NET_MACHINE,  
OPC_NODES_IN_GROUP, OPC_NODE_GROUPS  
WHERE opc_node_names.node_id=opc_nodes.node_id AND  
opc_nodes.machine_type=opc_net_machine.machine_type AND  
opc_node_names.network_type=opc_net_machine.network_type AND  
OPC_NODES_IN_GROUP.NODE_ID=opc_nodes.node_id AND  
OPC_NODES_IN_GROUP.node_group_id=OPC_NODE_GROUPS.node_group_id AND  
OPC_NODE_GROUPS.node_group_name='hp_ux'
```

## 5.2 Customizing the Statistics view

In the default implementation, the OVO Service Adapter calculates fault statistics based on all OVO messages stored in the OVO database. These messages include ones that are not relevant, meaning that invalid statistics might result. This is prevented by filtering, the set of OVO messages on which the statistics are computed. The fault statistics are computed by a PL/SQL function named *compute\_statistics* defined in SQL script in the following file:

```
$STEMIP_SC_HOME/ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/sqm_o  
vo_sa_05_body.sql
```

The default SQL function that defines the set of OVO messages from which statistics are computed is as follows:

```
CURSOR c1 IS SELECT * FROM opc_act_messages WHERE node_id=node  
and ACKN_FLAG = 0;
```

To filter out messages from the statistic computations, do the following.

1. Modify the SQL query defined at the beginning of the *compute\_statistics* PL/SQL function.
2. Reload the script on the OVO database as described in “Creating the SQL view” on page 30.

For example, to filter out messages belonging to the **OpC** OVO node group, you must modify the SQL query as follows:

```
CURSOR c1 IS SELECT * FROM opc_act_messages WHERE node_id=node  
and ACKN_FLAG = 0 AND MESSAGE_GROUP != 'OpC';
```

# Appendix A

## Installation directory structure

```
ServiceAdapters/Sql/v1_2/Ovo_v1_2/UninstallerDataOVO
ServiceAdapters/Sql/v1_2/Ovo_v1_2/UninstallerDataOVO/Uninstall_Ovo
ServiceAdapters/Sql/v1_2/Ovo_v1_2/UninstallerDataOVO/Uninstall_Ovo.lax
ServiceAdapters/Sql/v1_2/Ovo_v1_2/UninstallerDataOVO/uninstaller.jar
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/Ovo_v1_2_launch.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_create_db_view.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_complete_install.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_discovery.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_configure.pl
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_discovery_filter_template.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_configure.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_setup_connector.pl
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_setup_connector.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_create_datatree.pl
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_create_datatree.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/bin/temip_sc_update_datatree_access_rights.sh
ServiceAdapters/Sql/v1_2/Ovo_v1_2/config
ServiceAdapters/Sql/v1_2/Ovo_v1_2/config/SCPlatform_SCDirector_SCAApplication.properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/config/SaSqlDiscoveryTraceLogging.properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/config/SaSqlDiscoveryMtLogging.properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/sqm_ovo_sa_01_drop_table_def.sql
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/sqm_ovo_sa_02_record_def.sql
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/sqm_ovo_sa_03_table_def.sql
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/sqm_ovo_sa_04_declare.sql
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/sqm_ovo_sa_05_body.sql
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/qm_ovo_sa_06_discovery_view.sql
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/sqm_ovo_sa_07_statistic_view.sql
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/jar
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/jarTeSCSAOvo.jar
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/lib
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/lib/ojdbc14.jar
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/properties/TeSCOvo.properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/properties/TeSCOvo_Version.properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/properties/TeSCSql_Version.properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/properties/TeSCOvo_Messages.properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/properties/TeSCSqlDiscovery.xml
ServiceAdapters/Sql/v1_2/Ovo_v1_2/database/propertyessaname.properties
ServiceAdapters/Sql/v1_2/Ovo_v1_2/repository
ServiceAdapters/Sql/v1_2/Ovo_v1_2/repository/DelDFDReq_OVOSTATDFD.v1_0.xml
ServiceAdapters/Sql/v1_2/Ovo_v1_2/repository/Ovo_Connectors_data.exp
ServiceAdapters/Sql/v1_2/Ovo_v1_2/repository/Ovo_v1_2_setup.cfg
```

```
ServiceAdapters/Sql/v1_2/Ovo_v1_2/repository/NewDFDReq_OVOSTATDFD.v1_0.xml  
ServiceAdapters/Sql/v1_2/Ovo_v1_2/repository/Ovo_dfds_data.exp  
ServiceAdapters/Sql/v1_2/Ovo_v1_2/repository/Ovo_v1_2_template.exp  
ServiceAdapters/Sql/v1_2/Ovo_v1_2/util
```

# Glossary

The following table lists the acronyms that are commonly used in this document.

Term	Description
API	Application programming interface
DFD	Data feeder definition
DFI	Data feeder instance
HP-UX	HP implementation of Unix. HP-UX, which is compatible with various industry standards, is based on the UNIX® System V Release 4 operating system and includes important features from the Fourth Berkeley Software Distribution.
JDBC	Java Database Connectivity
MRP	Measurement reference point
OVO	OpenView Operations
SAI	Service Adapter Application Name (or Service Adapter instance)
SLA	Service Level Agreement
SLM	Service level management
SLO	Service level objective
SRM	Service Repository Manager
XML	eXtensible Mark-up Language







