HP OpenView
Service Quality Manager

mFormation Service Adapter

Installation, Configuration and User's Guide

Edition: 1.4

for the HP-UX Operating System

March 2007

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

Trademark notices

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.

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

UNIX® is a registered trademark of The Open Group.

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

Origin

Printed in France.

Contents

Preface

The HP OpenView Service Quality Manager (SQM) can use the mFormation Service Adapter to collect data from an mFormation platform in order to provide Mobile Subscriber Experience KPI in SQM.

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

A comprehensive model using the mFormation data is provided in SQM MDS value pack (MSE package). More information on how to uses mFormation probes data in SQM can be found in *HP OpenView SQM MDS Value Pack User's Guide*.

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
- mFormation platform environment
- HP OpenView SQM
- Service Level Management
- Network & Fault 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, "Software requirements".

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:

- *HP OpenView SQM – SQL Service Adapter Toolkit Installation, Configuration and User's Guide*
- *HP OpenView SQM MDS Value Pack User's Guide*

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

Support

You can visit the HP OpenView support web site at:

http://support.openview.hp.com/support.jsp

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

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

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

Chapter 1

Introduction

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

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

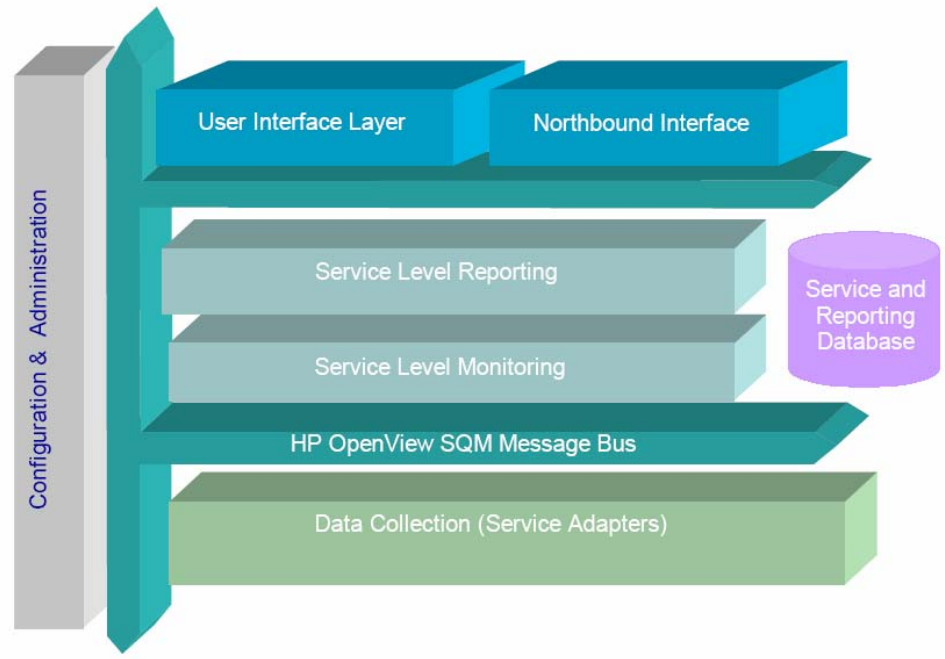
For a detailed description of Service Adapters, see the *HP OpenView Service Quality Manager Service Adapter User's Guide*.

1.1 Service Quality Manager

SQM provides a complete service quality management solution running under the HP implementation of UNIX®, HP-UX. 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 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 & Administration layer contains the configuration repository. This repository stores information on how to connect to mFormation platform and how to map the raw data from mFormation database to the SQM parameter format.

The Service and Reporting database contains the SQM Service Repository, which notably stores the parameters that the Service Adapter must collect.

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

1.2 mFormation Service Adapter

The mFormation Service Adapter connects to mFormation Service Manager to collect KPI related to Mobile Subscriber Experience.

It serves as a bridge between SQM and mFormation, collecting data from mFormation, calculating quality indicators, and then feeding them to SQM.

1.2.1 What the Service Adapter does

The mFormation Service Adapter provides to SQM some indicators deduced from the data stored in the mFormation database.

1.2.2 mFormation Probes overview

The mFormation Service Adapter provides indicators (KPI) that can be used in SQM to provide an end-to-end view of device management. This includes:

- The Carrier Network infrastructure that can be organized into:

- Regions
- Location Areas
- Cells
- Data Services (Email, HTTP, SMS, MMS, WAP)

- The customer Device Fleets that can be organized into:
  - Device Groups
  - End-User Devices

Based on mFormation collected Data, SQM can be used to analyze how the information is related. The *HP OpenView SQM MDS Value Pack* uses these indicators to provide such features:

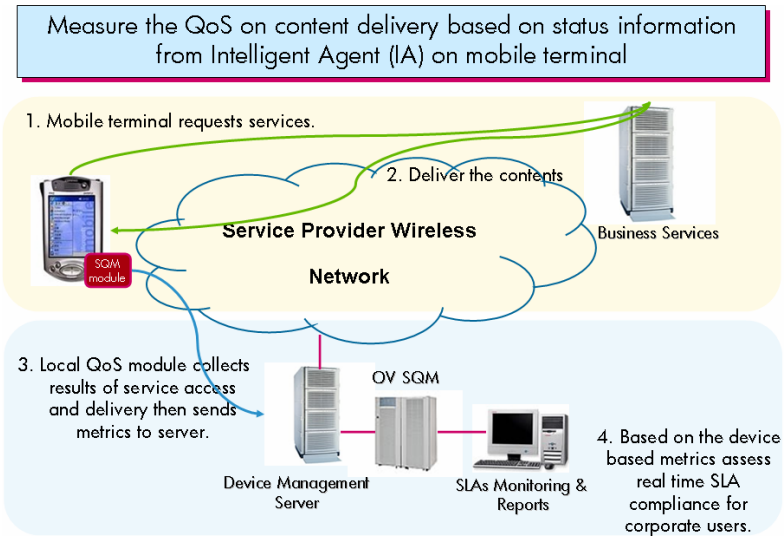
- Data is collected by the device MSISDN
- We can group devices into collections to represent corporate customers, device manufacturers.
- We can measure the performance of services that devices use (e.g. WAP, voice, MMS)
- We can see the device IMEI thus identify the device type
- We can see the device location and thus report on service by geography (cell, region etc)
- We can quantify the % call drop rate by customer, customer group, by cell, region or by network

1.2.3 mFormation Probes as a data source for SQM

Integration is achieved through an SQM SQL Service Adapter, called the mFormation Service Adapter. This Service Adapter collects mFormation data from a dedicated SQL view that computes required KPI for SQM.

This integration is illustrated in the following figure, “mFormation /SQM integration”.

Figure 2 mFormation /SQM integration



1.2.4 Product deployment

You must firstly install the mFormation platform with its associated Oracle database. Connection parameters to this database (host, user, password…) are required for configuring mFormation Service Adapters.

On the SQM platform, you must install and configure the Service Adapter with all of its prerequisites, which include the SQM Kernel, Service Adapter Common layer, and the SQL Service Adapter.

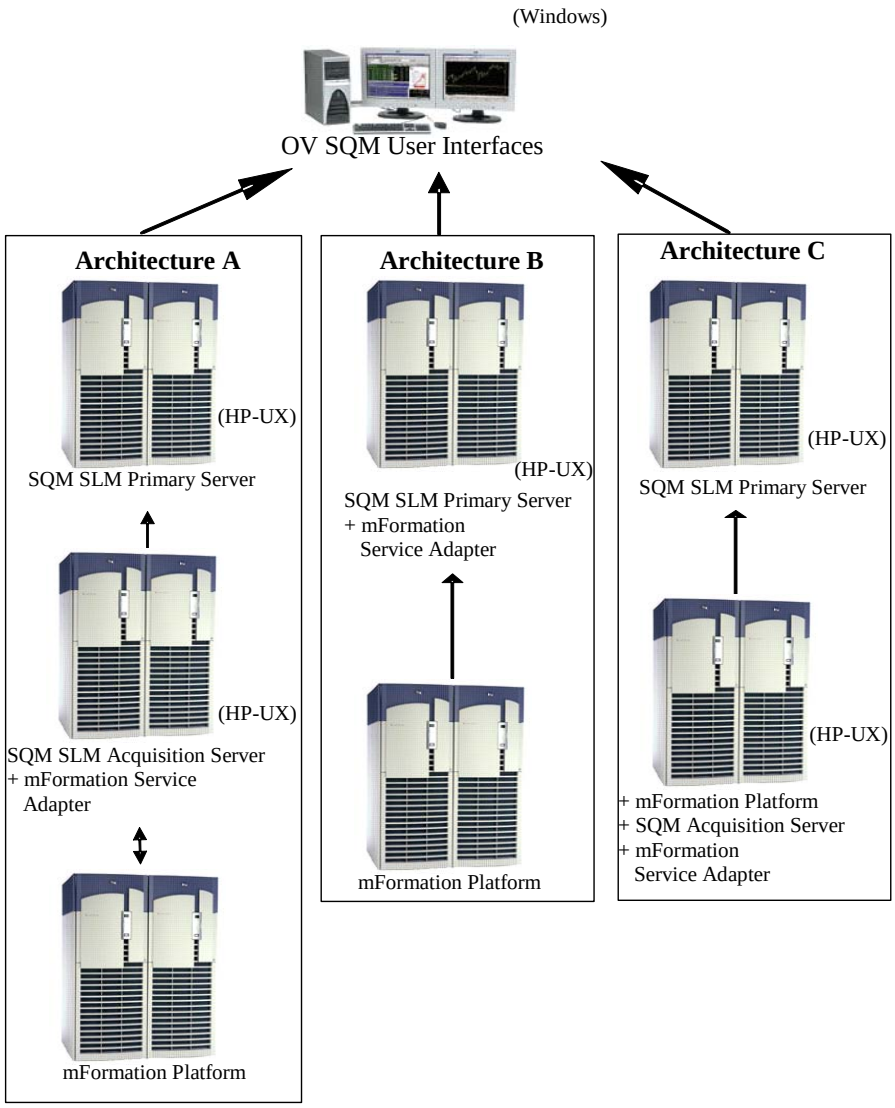
Once the Service Adapter has been configured and started, it begins to collect the specified quality indicators for SQM. The Service Adapter functions as an independent unit.

Each configured installation is called a Service Adapter Application or Service Adapter Instance (SAI). The Service Adapter Application connects to the mFormation 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 mFormation environment if you wish. 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 secure environments.

Figure 3 below shows how the Service Adapter works with SQM and mFormation Platform.

Figure 3 Service Adapter deployments



1.2.5 mFormation data collection

The Service Adapter uses data feeders to collect data. A data feeder is therefore 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**

A data structure which defines the source data collected; it is stored both in the SQM Service Repository Manager and in the SQM Central Repository.

- **DFD parameters**

Parameters that define what measurement data the Service Adapter sends to SQM; it is stored in the SQM Service Repository Manager repository.

- **DFD properties**

These properties further identify the managed probe whose data is collected. The properties 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 table shows the DFD configuration of the mFormation Service Adapter. Datafeeder's properties are in ***bold and italic*** font.

mFormation Probes DataFeeders

DataFeeder Device	Parameter	Type	Description
	Device Identifier	String	Identifier of the device.
	Group Identifier	String	Indicates whether this device is a 'user' device or a 'monitoring' device - a monitoring device is one that is used for monitoring networks. A 'user' device is one, which is normally used by the end users - employees of the customer.
	Network Equipment Identifier	String	Network equipment identifier, which also represents the Phone number assigned to the device in case of GSM/GPRS/CDMA networks.
	IMEI identifier	String	International Mobile Equipment Identification of the device.
	Available Memory	Int	Total available memory of the handheld.
	Device Make	String	Device make.
	Device Model	String	Device model.
	Device Group	String	Group where device is brought.
	Device Fleet	String	Fleet where device is brought.
	Device OS	String	Operating system of the device.
	Agent Version	String	Version of the agent located in the device.
	Home Network	String	Name the home network used by the device.
	Signal Strength	Int	Strength of the signal measured on the device.
	Device Status	Int	The status of the device: inactive(0), active(1).
	Coverage State	String	The current mobile state the device is in - Voice, Voice/Date or Emergency.
	Battery Level	Float	Battery level of the device, represented by a percentage (1 also meaning the battery is fully recharged).
	Roaming Network	String	Name the roaming network used by the device; equal to the home network means the device is actually working on the home network.
	Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
Cell coverage			
	Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
	Cell Identifier	String	Cell identifier of the test location.
	Region Identifier	String	This identifier represents the Region name. A region could contain several LAIs.
	#Devices "Voice"	Int	Number of devices in "Voice" state in the cell.
	#Devices "Voice/Data"	Int	Number of devices in "Voice/Data" state in the cell.
	#Devices "Emergency"	Int	Number of devices in "Emergency" state in the cell.

Cell call monitoring			
	Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
	Cell Identifier	String	Cell identifier of the test location.
	Region Identifier	String	This identifier represents the Region name. A region could contain several LAIs.
	#Calls	Int	Number of voice calls targeted to a cell for the past collection period.
	#Dropped calls	Int	Number of dropped calls in the cell for the past collection period.
	Cell Accessibility	Float	Cell %Call Success for the past collection period. (Number of calls - Number of dropped calls) / Number of calls
Cell signal strength			
	Cell Identifier	String	Cell identifier of the test location.
	Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
	Region Identifier	String	This identifier represents the Region name. A region could contain several LAIs.
	Worst signal strength	Float	Worst signal strength among all the devices connected to a cell.
	Average signal strength	Float	Average signal strength among all the devices connected to a cell.
	Percentile 10% signal strength	Float	"Percentile 10%" signal strength among all the devices connected to a cell => worst signal strength with ignorance of the 10% worst signal strength values.
	Percentile 70% signal strength	Float	"Percentile 70%" signal strength among all the devices connected to a cell => worst signal strength with ignorance of the 70% worst signal strength values.
SMS monitoring			
	Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
	Cell Identifier	String	Cell identifier of the test location.
	Region Identifier	String	This identifier represents the Region name. A region could contain several LAIs.
	Service signal strength	Float	Strength of the signal mesured while using a Data Service.
	#Active devices	Int	Number of active devices.
	Service response time	Int	Total time used by the service for responding.
	Service throughput	Float	Throughput of the service.
	#Data Service Requests	Int	Total Number of requests to SMS services
	#Success Data Requests	Int	Number of Data Requests that have succeeded.
	Error - Server connection	Int	11003 = Error - Server connection
	Error - Send	Int	11001 = Error - Send Failed

Failed		
Warning - Message delayed	Int	11012 = Warning - Message delayed
WAP monitoring		
Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
Cell Identifier	String	Cell identifier of the test location.
Region Identifier	String	This identifier represents the Region name. A region could contain several LAIs.
Service signal strength	Float	Strength of the signal measured while using a Data Service.
#Active devices	Int	Number of active devices.
Service response time	Int	Total time used by the service for responding.
Service throughput	Float	Throughput of the service.
#Data Service Requests	Int	Total Number of requests to WAP services
#Success Data Requests	Int	Number of Data Requests that have succeeded.
Error - Server connection	Int	11003 = Error - Server connection
Error - WAP gateway connection	Int	11006 = Error - WAP gateway connection
Error - WAP server connection	Int	11007 = Error - WAP server connection
Error - WAP request	Int	11008 = Error - WAP request
Error - WAP request timeout	Int	11009 = Error - WAP request timeout
Email monitoring		
Region Identifier	String	This identifier represents the Region name. A region could contain several LAIs.
Cell Identifier	String	Cell identifier of the test location.
Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
Service signal strength	Float	Strength of the signal measured while using a Data Service.
#Active devices	Int	Number of active devices.
Service response time	Int	Total time used by the service for responding.
Service throughput	Float	Throughput of the service.
#Data Service Requests	Int	Total Number of requests to EMail services
#Success Data Requests	Int	Number of Data Requests that have succeeded.
Error - Server connection	Int	11003 = Error - Server connection

HTTP monitoring		
Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
Cell Identifier	String	Cell identifier of the test location.
Region Identifier	String	This identifier represents the Region name. A region could contain several LAIs.
Service signal strength	Float	Strength of the signal measured while using a Data Service.
#Active devices	Int	Number of active devices.
Service response time	Int	Total time used by the service for responding.
Service throughput	Float	Throughput of the service.
#Data Service Requests	Int	Total Number of requests to HTTP services
#Success Data Requests	Int	Number of Data Requests that have succeeded.
Error - Server connection	Int	11003 = Error - Server connection
Error - HTTP request	Int	11004 = Error - HTTP request
Error - HTTP request timeout	Int	11005 = Error - HTTP request timeout
Error - PDP Activation	Int	11002 = Error - PDP Activation
MMS monitoring		
Location Area Identity	String	Location Area Identity (LAI) is composed of: + MCC (Mobil Country Code) + MNC (Mobil Network Code) + LAC (Location Area Code) Sub net identifier = MCC + MNC.
Cell Identifier	String	Cell identifier of the test location.
Region Identifier	String	This identifier represents the Region name. A region could contain several LAIs.
Service signal strength	Float	Strength of the signal measured while using a Data Service.
#Active devices	Int	Number of active devices.
Service response time	Int	Total time used by the service for responding.
Service throughput	Float	Throughput of the service.
#Data Service Requests	Int	Total Number of requests to MMS services
#Success Data Requests	Int	Number of Data Requests that have succeeded.
Error - Server connection	Int	11003 = Error - Server connection
Error - Send Failed	Int	11001 = Error - Send Failed
Error - PDP Activation	Int	11002 = Error - PDP Activation
Warning - Message delayed	Int	11012 = Warning - Message delayed

Chapter 2

Installing the Service Adapter

This chapter explains how to install the Service Adapter on HP-UX. Once 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
- mFormation Platform
- Oracle 10g
- HP OpenView Service Quality Manager V1.4 (Kernel subset)
- HP OpenView SA Common V1.4 (SQMSAGTWCOMMON)
- HP OpenView SQL Service Adapter Runtime V1.4 (SQMSASQL)
- Oracle JDBC driver related to Oracle platform in use.

2.1.2 Hardware requirements

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

For the Service Adapter:

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

2.2 Installing the mFormation SA

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

If the OV SQM Kernel has already been installed, go directly to “Installing the SA Common subset” on page 18.

2.2.1 Installing the OV SQM Kernel

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

To install the Kernel, do the following:

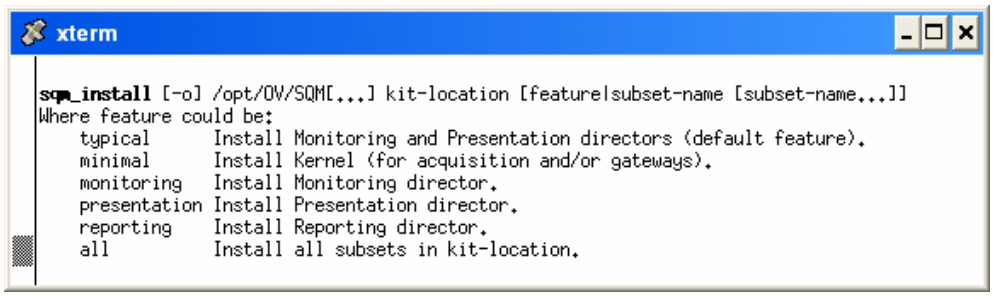
1. Create the SQM home directory, if it does not already exist.

2. 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.

3. Mount the SQM Core CD-ROM on your system.
4. Go to <*mounted directory*>/SQM-1.40.00
where <*mounted directory*> is the name of your mounted directory.
5. Run the *sqm_install* tool under the root account.



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

```
# sqm_install /opt/OV/SQM<xxx> <mount directory>/SQM-1.40.00/HP
UX/KIT
```

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

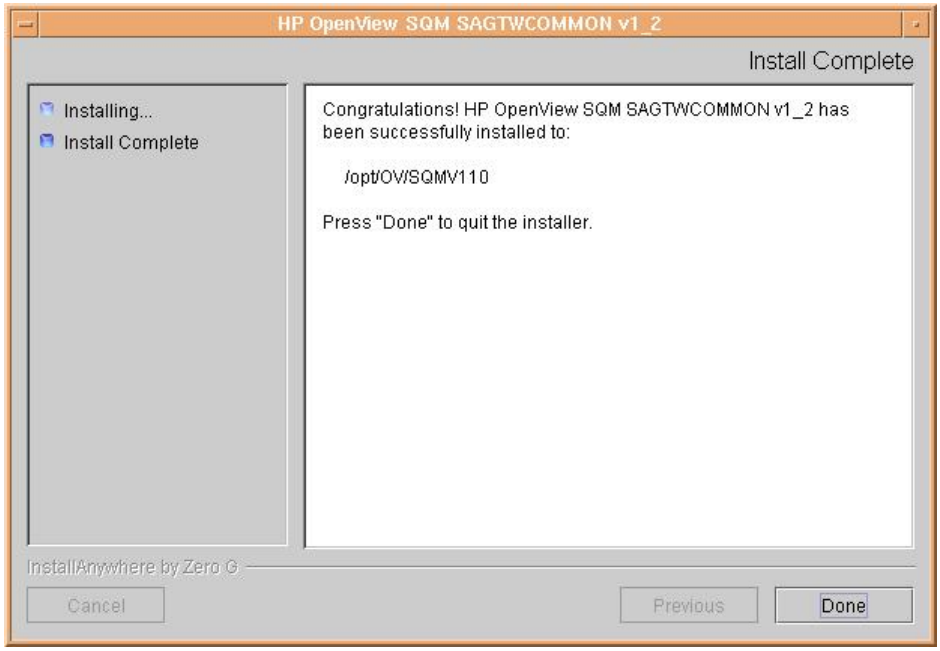
2.2.2 Installing the SA Common subset

If necessary, install the SA Common component by doing the following. If this has already been done, go directly to “Installing the SQL Service Adapter Runtime” on page 19.

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-1.40.00
and execute the following command:

```
# ./SQMSAGTWCOMMON-1.40.00.bin
```

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



2.2.3 Installing the 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 mFormation ” below.

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-1.40.00`
   where `<mount directory>` is your mount directory.
4. Run the installer by entering the following command:

```
# ./SQMSASQL-1.40.00.bin
```

The software is installed and the **Install Complete** window is displayed as shown below.

5. To finish, click **Done**.

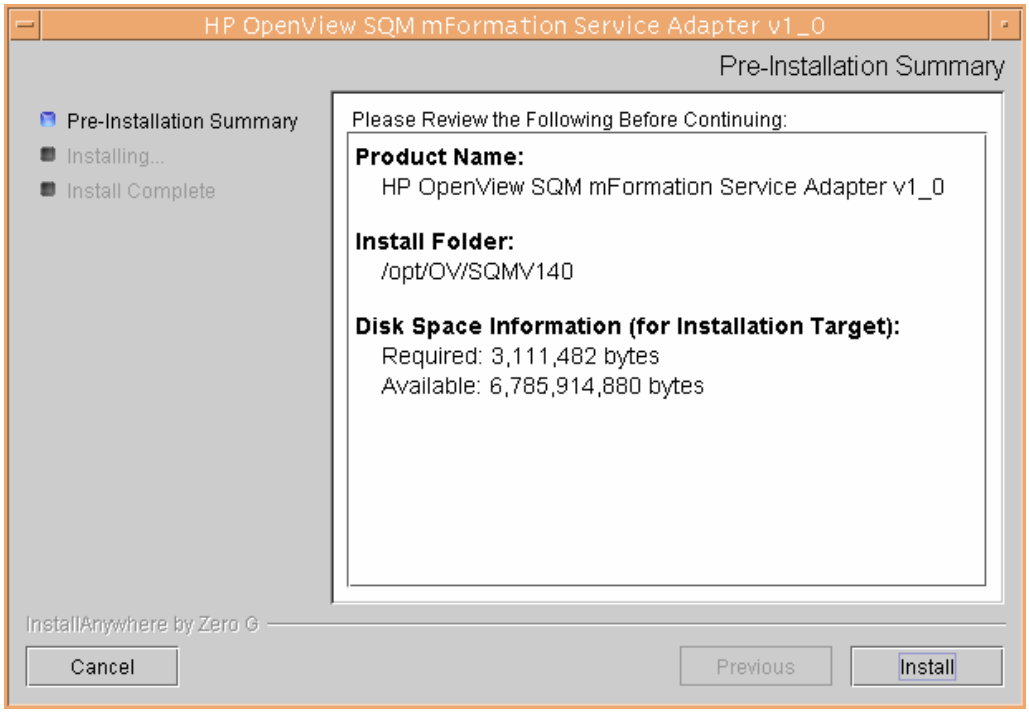
2.2.4 Installing the mFormation Service Adapter

To install the mFormation 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-1.40.00`
   where `<mount directory>` is your mount directory.
4. Run the installer by entering the following command:

```
# ./ SQMSAMFORMATION-1.40.00.bin
```

The **Pre-Installation Summary** window is displayed as shown below. Click on **Install** to start the installation.



5. At the end of the installation process, check the installation status in the **Install Complete** window. Click **Done** to close the installer.

2.2.5 Uninstalling the mFormation Service Adapter

To uninstall the mFormation Service Adapter 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 mFormation Service Adapter has been installed:

```
# cd $TEMIP_SC_HOME
#
./ServiceAdapters/Sql/v1_4/mFormation_v1_0/UninstallerDatamFormation/Uninstall_mFormation
```

4. See 4.3.2 to delete the application and eventually the DFD

Chapter 3

Setting up and Configuring the Service Adapter

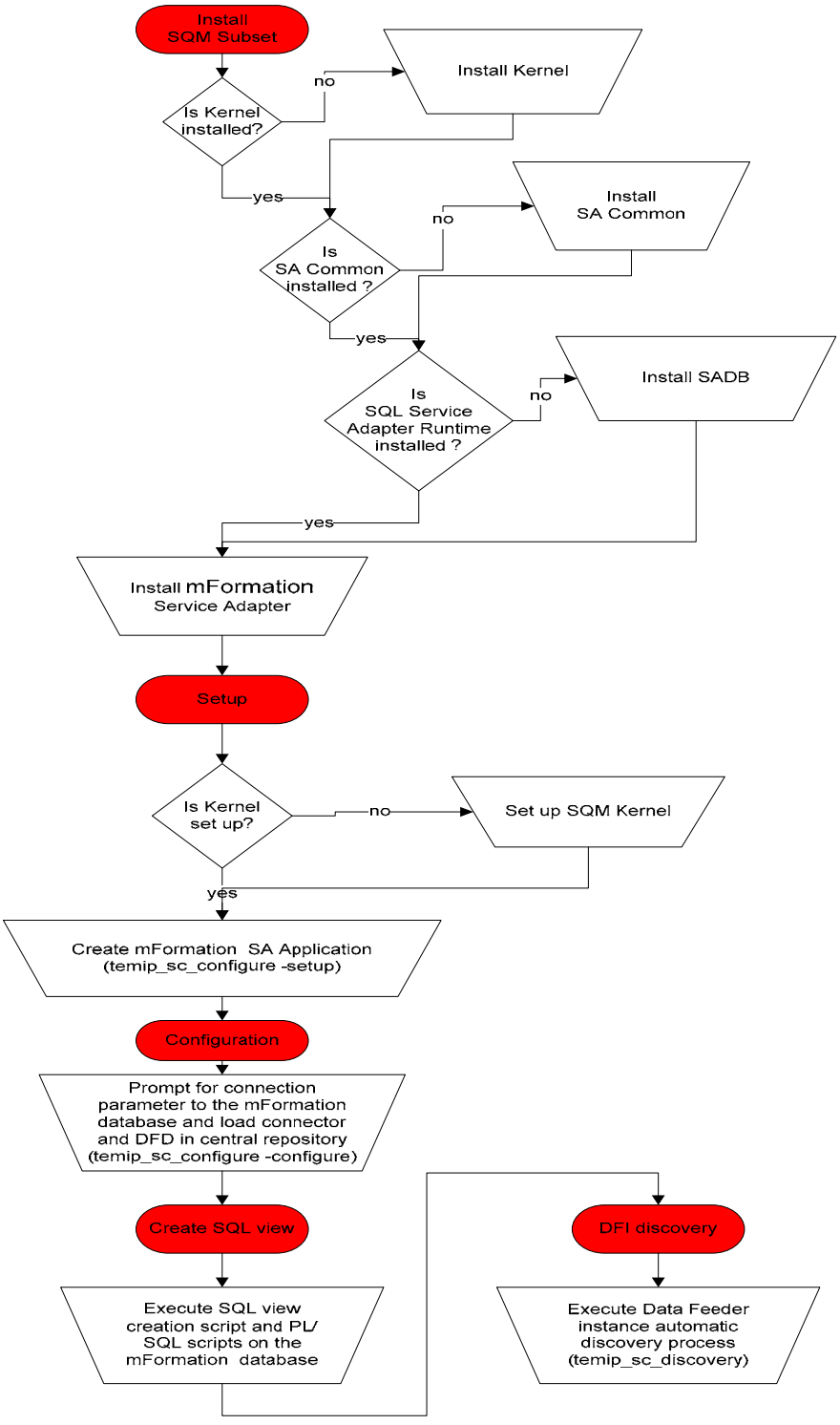
A set of tools are provided with the mFormation Service Adapter to make it easier to install, set up and configure. These are used to create the application and upload the Data Feeder Definition (DFD) and connection settings into the central repository. They are also used to execute SQL view creation scripts on the mFormation database server, as well as to discover Data Feeder instances automatically.

Figure 4 on page 24 shows you each of the Service Adapter configuration steps.

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

1. Setting up the application by declaring the mFormation Service Adapter application in the SQM Central Repository and creating the mFormation Service Adapter application data tree in `TEMIP_SC_VAR_HOME`
2. Configuring the application by replying to user prompts requesting the mFormation database's connection settings, 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 mFormation Service Adapter directory `$TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/database`
4. Discovering and loading DFIs. DFI discovery is an important feature provided by the `mFormation` Service Adapter. This feature is used to discover and load the Data Feeder Instances that will be managed by the `mFormation` SA application, completely automatically.
   This is done by querying the mFormation database and loading into SQM a DFI for each entity mFormation platform. This SQL query is based on a view of the YYY.`XXX` table. Before you execute this query to discover the DFI.

Figure 4 mFormation 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 25.

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 `mFormation` SA can be configured and run. You can set up the SQM Kernel in any of the following configurations:

- If the mFormation SA is installed on the HP-UX SQM SLM Primary Server:
  In this case, see the *HP OpenView SQM Installation Guide* for instructions on how to set up the SQM Server.
- If the mFormation 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.

  1. Create the **sqmadm** administration user on the target Unix system (see the *HP OpenView SQM Installation Guide* for instructions on how to create the user account)
  2. 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`
  3. 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 mFormation Service AdapterApplication

This phase consists in creating a mFormation Service Adapter application on the SQM platform (on a specified director). To create the mFormation Service Adapter application, do the following.

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

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/bi
n
#temip_sc_configure.sh -setup -dirName <director name> <applica
tion 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.

This command declares the mFormation Service Adapter application to the SQM Central Repository and creates the application's data tree in:
`$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0`

The following output is displayed:

```
[sqmadm_hotel]/opt/OV/SQMV120/ServiceAdapters/Sql/v1_4/mFormation
_v1_0/bin>temip_sc_configure.sh -setup MSE_host1

Setup the "MSE_host1" application ...
Create the application (this may take several minutes).
Create the data tree.
/var/opt/OV/SQM/slmv14/ServiceAdapters   (already exist)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql   (already exist)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4   (already exist)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0
 (already exist)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/c
onfig   (already exist)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/d
iscovery   (created)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/d
iscovery/filter   (created)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/d
iscovery/filter/slmv14_acquisition_mFormationSA_filter.sh   (crea
ted)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/d
iscovery/inventory   (created)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/d
iscovery/inventory/raw   (created)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/d
iscovery/inventory/filtered   (created)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/d
iscovery/repository   (created)
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0/r
epository   (created)
Initializing Discovery Tracing and Logging ...
Setup succeed.

Update the access permissions on the data tree... succeed
```

3.2 Configuring the application

The application must be configured by defining the mFormation Service Adapter 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 **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 commands:

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/bin
#temip_sc_configure.sh –configure <application name>
```

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

4. Enter the appropriate SQL database connection information (login, password, database URL, and JDBC driver), and then load the SQL Service Adapter's Data Feeder definitions and connector definition into the SQM Central Repository.

You must provide the following default connection parameters for the mFormation database:

- URL: jdbc:oracle:thin:@*<mFormation database hostname>*:1521:*<schema name>*
- Username: *<Username>*
- Default Password: *<User Password>*
- JDBC Driver: oracle.jdbc.driver.OracleDriver

The following output is displayed:

```
[sqmadm_hotel]/opt/OV/SQMV120/ServiceAdapters/Sql/v1_4/mFormati
on_v1_0/bin>temip_sc_configure.sh -configure MSE_host1

Configure the "MSE_host1" application ...
Please enter the database URL [jdbc:oracle:thin:@helios.vbe.cpq
corp.net:1521:xx]:
Please enter the database username [yy]:
Please enter the database password [********]:
Please enter the database JDBC Driver [oracle.jdbc.driver.Oracl
eDriver]:
Load the Connector in the Tibco Repository
Backup written at the following location: /var/opt/OV/SQM/slmv1
4/ServiceAdapters/Sql/v1_4/mFormation_v1_0/repository/connector
s_data.exp.2004_8_5_17_27_53
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0
/repository/connectors_data.exp has been imported into the Repo
sitory
Load the Data Feeder Definitions in the SRM
load DFD: DTO_APN_Gn - v1_0 (../repository/NewDFDReq_
DTO_APN_Gn.v1_0.xml) ... succeed.
Load the Data Feeder Definitions in the SRM
…
Load the Data Feeder Definitions in the Tibco Repository
Backup written at the following location: /var/opt/OV/SQM/slmv1
4/ServiceAdapters/Sql/v1_4/mFormation_v1_0/repository/
mFormation_dfds_data.exp.2004_8_5_17_28_18
../repository/
mFormation_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 mFormation database, even if multiple mFormation Service Adapter instances are

created. This is because all mFormation Service Adapter instances connected to a mFormation database use the same SQL view.

mFormation statistics are computed through an SQL view and PL/SQL scripts which must be deployed in the mFormation database. These scripts are located in the following directory:
`/opt/OV/SQMV120/ServiceAdapters/Sql/v1_4/mFormation_v1_0/database`

To deploy these scripts, enter the following two commands:

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/bin
# temip_sc_configure.sh –view <application name>
```

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

Enter the appropriate SQL database connection information (login, password, database URL, and JDBC driver). You must log in as a user with creation privileges on the mFormation database to execute these scripts. By default, the **system** user has these privileges.

You must provide the following default connection parameters for the mFormation database:

- `Username: system`
- `Default Password: manager`

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

The following output is displayed:

```
#  ./temip_sc_configure.sh -view -dir acqMSE MSE_host1

Create the view ...
/var/opt/OV/SQM/slmv14/ServiceAdapters/Sql/v1_4/mFormation_v1_0
/config/MSEHabit_DBConnector.cfg
=====================================================
Executing SQL Script on remote SQL database.
=====================================================
=> Please enter the database username: xxx
=> Please enter the database password: yyy
```

3.4 Discovering and Loading DFIs

The SQM DFI discovery is performed by the mFormation Service Adapter. This function is used to discover and load all the Data Feeder Instances that will be managed by the mFormation Service Adapter application, completely automatically. This is done by retrieving the MRP properties from an SQL table in the target database.

Discovery script

1. On the SQM director, the discovery is activated with the following script:

PATH:
`$TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/bin`

FILE: `temip_sc_discovery.sh`

SQM discovery phase

- **Raw discovery pha**se, in which the SQL discovery query is executed and details of all discovered DFIs are recorded in a raw (unprocessed) inventory file.
- Filtering phase, in which a script is executed to filter the DFIs declared in the raw inventory file. A new filtered inventory file is generated containing only the DFIs selected for management. By default, the script does not apply any filtering. You must therefore modify it to filter the details contained in the raw inventory file as required.
- Loading phase, in which the details of the filtered DFIs are loaded into the SQM 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 you select this option.
    - *-diff offline*
      The list of all discovered/filtered Data Feeder Instances is compared with a reference discovery file (provided by you) if you select this option.
      If a Data Feeder Instance exists in the inventory file but it 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 the inventory file and the reference file, it is not reloaded.
    - *-diff online*
      The same Data Feeder Instance comparisons are performed as those carried out in offline mode if you select this option. Instead of checking the DFIs against a reference file, however, whether DFI is declared depends only on whether the Data Feeder Instance exists in the SQM Central Repository in this case.

Note

The next chapters provide details of each of the above phases.

The same processing can be performed by executing a single command (by default, all filtered Data Feeder Instances are then loaded, through the *-diff no* switch). For details of this command, see “One-shot discovery and loading” on page 32.

3.4.1 Raw discovery phase

In this phase, the mFormation Service Adapter loads the discovery query definition file and executes the discovery queries to retrieve all DFI for all DFD managed by the mFormation SA.

A set of dedicated SQL view was deployed on the mFormation Server when you created the SQL view. These views are used to retrieve mFormation probes and declare associated DFI.

The DFI are discovered by running query on the view used for the collection. There is a specific query for each DFD.

For example:

```
select distinct APN, SGSNID from DTO_APN_SGSN_GB_VIEW
```

All these queries are defined in the following file:

`$TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/properties/TeSCSqlDiscovery.xml`

Tips

A simple way to perform filtering is to update the SQL script defining the discovery view by adding further conditions to the default SQL query's 'WHERE' clause, rather than using a filtering script. You must ensure that you filter out only the appropriate mFormation entities when you modify this script, however.

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:

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

3. Enter the following commands:

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/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:
  `$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/discovery/inventory/raw/<platform name>_<director name>_<application name>.xml`
- The DFI declaration XML files:
  `$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/discovery/repository/DeclareDFIReq_DTO_xxx.v1_0.<DFIID>.xml`
- The DFI deletion XML files:
  `$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/discovery/repository/DelDFIReq_DTO_xxx.v1_0.<DFIID>.xml`

3.4.2 Filtering phase

Not Applicable.

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 the 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 it 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 carried out in offline mode if you select this option. Instead of checking the DFIs against a reference file, however, whether DFI is declared depends only on whether the Data Feeder Instance exists in the SQM Central Repository in this case.

Input

- The DFI filtered inventory file output by the previous command is required as input for this phase.
  Its pathname is:
  `$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/discovery/inventory/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 and named as follows:
  `$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/discovery/repository/<`*platform name*`>_<`*director name*`>_<`*application name*`>_discovery_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/slmv14/temip_sc_env.sh
```

3. Enter the following two commands:

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/bi
n
# temip_sc_discovery.sh -platform <platform name> -director <di
rector 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 for each DFI (Successful, Failure, or Partial) is logged. The discovery loading procedure logs the result of each DFI declaration in the following file:
  `$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/discovery/repository/<platform name>_<director name>_<application name>_discovery_cmds.log`
- If the script fails, you can restart the DFI loading process by executing the following script manually:
  `$TEMIP_SC_VAR_HOME/ServiceAdapters/Sql/mFormation_v1_0/discovery/repository/<platform name>_<director name>_<application name>_discovery_cmds.sh`

3.4.4 One-shot discovery and loading

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

Command

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 commands:

```
# cd $TEMIP_SC_HOME/ServiceAdapters/Sql/v1_4/mFormation_v1_0/bi
n
#temip_sc_discovery.sh -platform <platform name> -director <dir
ector 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's repository

3.4.5 Scheduling DFI discovery

The mFormation Service Adapter can be deployed in environments in which the configuration changes daily (for example when new entities are added).
HP recommends scheduling automatic discovery by running special scripts containing all of the DFI discovery commands as a *crontab* in such situations.

Scheduled DFI discovery is run in batch mode, and loads newly discovered DFIs automatically from the mFormation platform.

3.5 Advanced configuration

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

Chapter 4

Service Adapter Operation

4.1 Starting and Stopping mFormation Service Adapter

Starting and stopping the mFormation 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> -dire
ctor <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 in 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. Enter the following command:

```
# temip_sc_stop_application -platform <platform name> -dire
ctor <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 the Collection

The data collection 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 mFormation Service Adapter Application, do the following:

1. Create the SA application. For details, see “Creating the mFormation Service AdapterApplication” on page 25.
2. Configure the SA Application. For details, see “Configuring the application” on page 26.
3. Create the SQL view. This step is necessary if the new application must be connected to a mFormation platform that is not already linked to an mFormation SA. For details, see “Creating the SQL view” on page 27.
4. Discover and load the DFIs. For details, see “Discovering and Loading DFIs” on page 28.
5. Start the SA application. For details, see “Starting and Stopping mFormation Service Adapter” on page 35.

4.3.2 Deleting an existing application

To delete the Service Adapter application, do the following:

1. Execute the following command:

```
# temip_sc_delete_application -platform <platform name> -direct
or <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 entered when the application was set up.

Chapter 5

Service Adapter Advanced Customization

Because an intermediate SQL view is used between the mFormation Server and the mFormation Service Adapter, you can customize the PL/SQL scripts to:

- Perform special filtering when discovering which entities mFormation manages
- Filter out mFormation that should not be included in statistic calculations

These aspects are not described in further details since it is not in the scope of this document.

Appendix A

Installation directory structure

The following directories and files are created under $TEMIP_SC_HOME/ServiceAdpaters/Sql/v1_4/mFormation_v1_0:

```
./UninstallerDatamFormation
./UninstallerDatamFormation/uninstaller.jar
./UninstallerDatamFormation/Uninstall_mFormation
./UninstallerDatamFormation/Uninstall_mFormation.lax
./UninstallerDatamFormation/.com.zerog.registry.xml
./config
./config/SCPlatform_SCDirector_SCApplication.properties
./config/SaSqlDiscoveryTraceLogging.properties
./config/SaSqlDiscoveryMtLogging.properties
./bin
./bin/mFormation_v1_0_launch.sh
./bin/temip_sc_create_db_view.sh
./bin/temip_sc_complete_install.sh
./bin/temip_sc_update_datatree_access_rights.sh
./bin/temip_sc_setup_connector.sh
./bin/temip_sc_discovery_filter_template.sh
./bin/temip_sc_create_datatree.sh
./bin/temip_sc_configure.sh
./bin/temip_sc_discovery.sh
./properties
./properties/TeSCmFormation_Version.properties
./properties/TeSCmFormation_Messages.properties
./properties/TeSCmFormation.properties
./properties/TeSCSql_Version.properties
./properties/saname.properties
./properties/TeSCSqlDiscovery.xml
./jar
./jar/TeSCSAmFormation.jar
./repository
./repository/mFormation_dfds_data.exp
./repository/mFormation_v1_0_template.exp
./repository/mFormation_v1_0_setup.cfg
./repository/mFormation_Connectors_data.exp
./repository/NewDFDReq_CellCall.v1_0.xml
./repository/DelDFDReq_CellCall.v1_0.xml
./repository/NewDFDReq_CellCoverage.v1_0.xml
./repository/DelDFDReq_CellCoverage.v1_0.xml
./repository/NewDFDReq_CellSignal.v1_0.xml
./repository/DelDFDReq_CellSignal.v1_0.xml
./repository/NewDFDReq_EmailDFD.v1_0.xml
./repository/DelDFDReq_EmailDFD.v1_0.xml
./repository/NewDFDReq_HTTPDFD.v1_0.xml
./repository/DelDFDReq_HTTPDFD.v1_0.xml
./repository/NewDFDReq_SMSDFD.v1_0.xml
./repository/DelDFDReq_SMSDFD.v1_0.xml
```

```
./repository/NewDFDReq_MMSDFD.v1_0.xml
./repository/DelDFDReq_MMSDFD.v1_0.xml
./repository/NewDFDReq_WAPDFD.v1_0.xml
./repository/DelDFDReq_WAPDFD.v1_0.xml
./repository/NewDFDReq_Device.v1_0.xml
./repository/DelDFDReq_Device.v1_0.xml
./util
./util/TeSCSql_Version.properties
./lib
./lib/ojdbc14.jar
./database
./database/MSE_01_CELL_CALL.sql
./database/MSE_02_CELL_COVERAGE.sql
./database/MSE_03_CELL_EMAIL.sql
./database/MSE_04_CELL_SIGNAL.sql
./database/MSE_05_CELL_HTTP.sql
./database/MSE_06_CELL_SMS.sql
./database/MSE_07_CELL_MMS.sql
./database/MSE_08_CELL_DEVICE.sql
./database/MSE_08_CELL_WAP.sql
```

Glossary

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

Term	Description
API	Application programming interface
DFD	Data feeder definition
DFI	Data feeder instance
MRP	Measurement reference point
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


invent


Java™
COMPATIBLE