hp OpenView
Service Quality Manager

Service Adapters

User's Guide

Edition: 1.2

January 2005

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company

United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

Trademark Notices

Adobe® and Acrobat® 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™ is a U.S. trademark of Sun Microsystems, Inc.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

Netscape is a U.S. trademark of Netscape Communications Corporation.

NMOS™ is a trademark of RiverSoft Technologies Limited.

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

Oracle7™ and Oracle7 Server™ are trademarks of Oracle Corporation, Redwood City, California.

PostScript® is a trademark of Adobe Systems Incorporated.

Riversoft™ is a trademark of RiverSoft Technologies Limited.

UNIX® is a registered trademark of The Open Group.

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

X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Contents

Preface 5

Chapter 1 Introduction to SQM Service Adapters Concepts 7

1.1 Overview 7
1.1.1 SQM 7
1.1.2 Service Adapters 8
1.2 Definitions 10
1.2.1 Service Adapters 10
1.2.2 Service Adapter applications 11
1.2.3 Data Feeders 11
1.2.4 Data Feeder Definitions 11
1.2.5 Data Feeder Parameters 12
1.2.6 Data Feeder Property 12
1.2.7 Data Feeder Instance 12
1.2.8 Connector 13
1.2.9 Measurement Reference Point 13
1.2.10 Subscriber and customer 16
1.2.11 Example 16
1.3 Service Adapter lifecycle 18
1.3.1 Deploying Service Adapters in OpenView SQM 19
1.3.2 Service Adapter installation and configuration 22
1.3.3 Service Adapter Setup and Startup 22

Chapter 2 Data Feeder Definitions 25

2.1 SQM Repository Part of a DFD 25
2.1.1 Characteristic of DFDs 27
2.2 Configuration Part of a DFD 28

Chapter 3 Data Feeder Instances 30

3.1 SQM Repository Part of a Data Feeder Instance 30
3.2 Characteristics of Data Feeder Instances 32

Chapter 4 Service Adapters Installation 33

4.1 Software and Hardware Requirements 33
4.1.1 Required Software 33
4.1.2 Hardware Requirements 34
4.2 Installation Tool 34
4.3 Directory Structure 34
4.3.1 Versioning 34
4.3.2 Global Directory Structure 35

Table of figures

Figure 1: SA positioning in the global architecture 8
Figure 2: SA and OpenView SQM Layers 9
Figure 3: Service Adapters distribution capabilities 10
Figure 4: MRP Example 13
Figure 5: When MRP is different from metric access 15
Figure 6: When MRP equals metric access 16
Figure 7: Hierarchy of Platform, Director, and Application 19
Figure 8: SA application Deployment Architecture 21
Figure 9: DFD Configuration Data Example 29
Figure 10: Service Adapters Directory Structure 35

Table of tables

Table 1 List of Terms and Acronyms 6
Table 2 : DFD Characteristics 27
Table 3 : DFD Parameter Characteristics 28
Table 4 : DFD Property Characteristics 28
Table 5 : DFI Characteristics 32

Preface

This document provides reference information to help you configure and use HP OpenView Service Quality Manager (SQM) Service Adapters.

Intended Audience

This document is intended for the following personnel:

- Administrators and technical staff in charge of administrating an OpenView SQM platform.

Prerequisite Reading

This document assumes that you have read the OpenView SQM Overview and are familiar with OpenView SQM terminology and components.

Supported Software

The supported software referred to in this document is as follows:

Product Version	Operating System
OpenView Service Quality Manager 1.2	HP-UX 11.11 Windows XP

The term UNIX is used as a generic reference to the operating system, unless otherwise specified

Typographical Conventions

`Courier` Font:

- Source code and examples of file contents.
- Commands that you enter on the screen.
- Path names
- Keyboard key names

Italic Text:

- Filenames, programs and parameters.
- The names of other documents referenced in this manual.

Bold Text:

- To introduce new terms and to emphasize important words.

Associated Documents

The OpenView SQM documentation set includes the following:

- *OpenView SQM Overview*
- *OpenView SQM Installation Guide*
- *OpenView SQM Administration Guide*

Support

Please visit our HP OpenView web site at: HP OpenView

There you will find contact information as well as details about OpenView products, services and support.

The OpenView support area of the OpenView web site includes:

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

Terms and acronyms

The following abbreviations are used:

Term	Description
CDR	Call Data Record
DF	Data Feeder
DFD	Data Feeder Definition
DFI	Data Feeder Instance
EMS	Element Management System
MOM	Message Oriented Middleware
MRP	Measurement Reference Point
NMS	Network Management System
RDB	Relational DataBase
SLA	Service Level Agreement
SIG	Service Instance Group
SA	Service Adapter
SACE	Service Adapters Central Engineering
SAI	Service Adapter Application (or Service Adapter Instance)
SC	Service Component
SD	Service Definition
SPD	Service Parameter Definition
SQM	Service Quality Manager
3PP	Third-Party Product

Table 1 List of Terms and Acronyms

Chapter 1

Introduction to SQM Service Adapters Concepts

This chapter introduces the concepts used in SQM Service Adapters.

Service Adapters are the southbound components of SQM that feed the core of SQM with performance measures coming from any source of data (such as EMS, NMS, and other similar products). Each Service Adapter is dedicated to a third party product interface.

The aim of Service Adapters is to convert performance indicators provided by a third party product into meaningful information for SQM.

Service Adapters expose performance data values to SQM, and if connected to Fault Management or Billing applications, the data flow (Faults or usage records) will be filtered, qualified, and only Quality of Service information will be exposed to SQM. For Service Adapters, the goal is not to know the cause of a network outage, but instead to give the most accurate information reflecting the quality of the infrastructure.

Thanks to Service Adapters, SQM is not dedicated to a specific category of performance indicators. Service Adapters can be connected to equipments that address the following domains: network (agents technology), traffic (sniffers), systems, applications, and end-user QoS probes (used to monitor the perceived quality).

New Service Adapters are developed on-demand and rely on the existing portfolio of product-specific Service Adapters.

1.1 Overview

1.1.1 SQM

SQM is a Service Level management software that ensures Service quality meets Service Level Objectives and Service Level Agreement.

SQM provides a complete service quality management solution, allowing enterprises to continually improve service levels without making additional investments in infrastructure. It consolidates quality indicators across all domains — telecom, IT networks, servers, and applications — providing end-to-end visibility on service quality. SQM links service quality degradations to potential affects 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, network support personnel can pinpoint

infrastructure problems and identify their potential affect on customers, services, and service level agreements (SLAs).

1.1.2 Service Adapters

A Service Adapter is a mediation layer between SQM and a third party product. It is used to convert the data model coming from the agent or database of a third party product into a predefined model fitting the needs of SQM. Figure 1 describes where Service Adapters fit into the OpenView SQM global architecture.

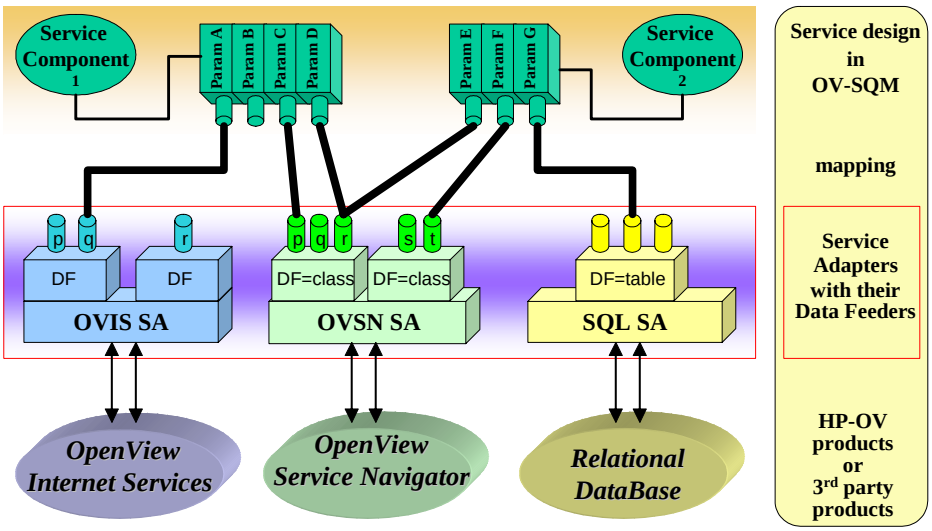


Figure 1: SA positioning in the global architecture

In an operational environment, a Service Adapter is an independent process connected to SQM. It exposes one or more **data feeders**.

Data feeders provide data to OpenView SQM. Each data feeder models service resources by defining one or more service parameters. These parameters contain values that are periodically updated and that can be used to monitor service components.

A **parameter binding** mechanism is available for associating data feeder parameters with service component parameters (performance indicators to quality indicators). See the *hp OpenView SQM Information Modeling* documentation for more details.

As illustrated in Figure 2, the Service Adapters communicate with the following components which are part of the SQM Service Level Monitoring layer (for more information about SQM layers, please refer to the ***OpenView SQM Administration Guide***):

- The SQM Performance Data Collector, which receives the performance messages.
- The Central Repository which stores SQM application-specific configuration (accessible through Tibco Designer).
- The SQM Service Repository Manager, which stores all Service definitions and instances.

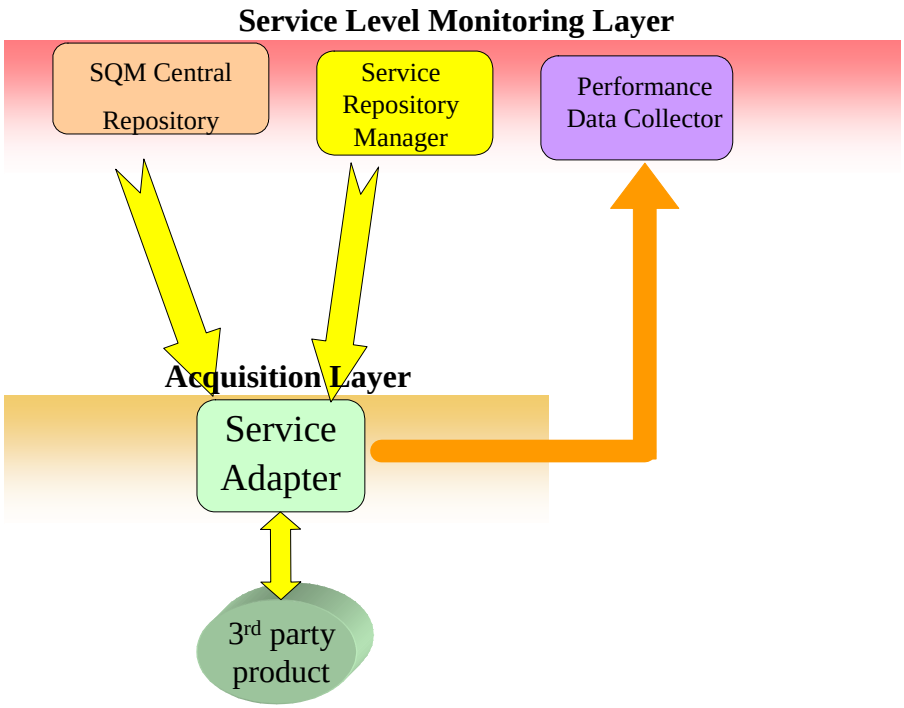


Figure 2: SA and OpenView SQM Layers

Service Adapters can be installed and run on a different system from the SQM SLM Primary Server. Depending on the protocol used to get the performance indicators, some Service Adapters may need to be installed on the same host as the third party product they monitor. Others can connect remotely to the third party product. To handle both cases, Service Adapters are designed to connect remotely to SQM SLM Primary Server.

Different Service Adapters can be installed on the same host. Moreover, on the same host several applications of the same Service Adapter can be started. Depending on the Service Adapter implementation, one Service Adapter application can be connected to several third party product agents. Figure 3 presents the different architecture possibilities offered by the Service Adapters. The Service Adapters used in the figure are fictitious. Refer to the documentation for your Service Adapter to know if it supports remote connectivity to a third party product. The distribution of Service Adapters depends on the end-user configuration and on performance tuning.

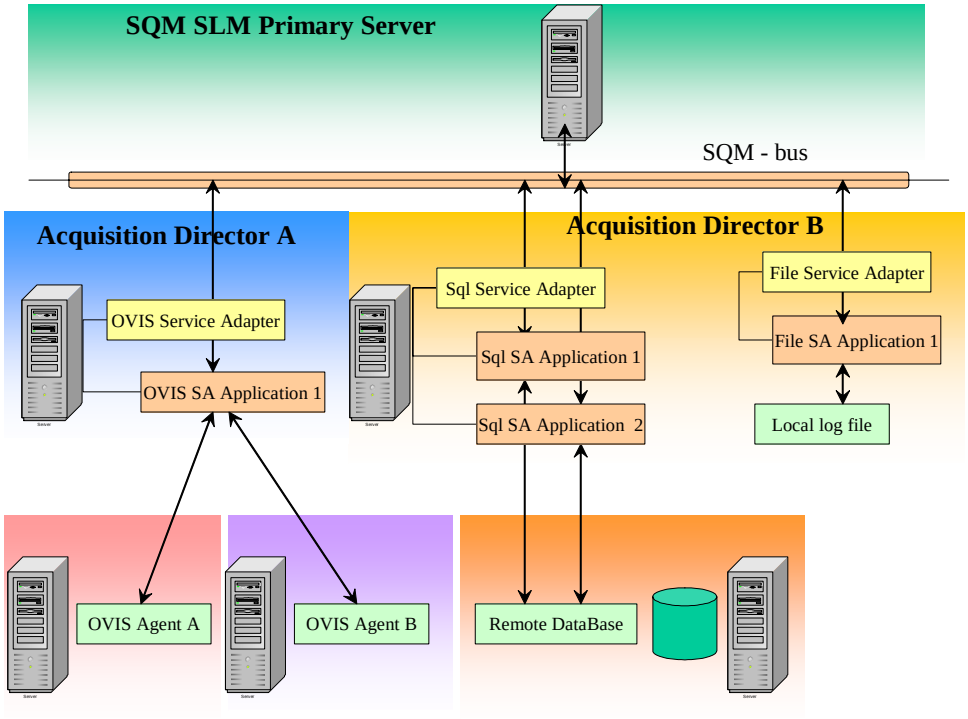


Figure 3: Service Adapters distribution capabilities

The Service Adapters do not provide clock synchronization. The machines where Service Adapters are installed (as well as the ones where the third party agents are installed) need to be time synchronized to avoid meaningless timestamps. Software like Network Time Protocol (NTP) must be installed and configured independently from the OpenView SQM Service Adapters. The Service Adapters rely on the timestamp of the third party product. If a timestamp is not available, it issues the performance measure with the current timestamp of the local host where it is installed.

1.2 Definitions

Some Service Adapter definitions provided in this chapter are also available in the ***SQM Information Modeling*** document. A previous reading of this documentation can provide you with an overview of all SQM definitions.

1.2.1 Service Adapters

A Service Adapter (SA) is an execution environment for data feeders. A Service Adapter is implemented for a specific technology, such as OVIS, TeMIP, or SQL.

There are different kinds of Service Adapters:

- **Generic Service Adapters**. A generic SA can accept new data feeders by configuration process (its implementation is generic enough to be able to collect on new Data Feeders).
- **Customized Service Adapters.** These SAs enrich a generic Service Adapter for a dedicated customer. A generic SA is customized either with a pre-configured set of data feeder definitions, or with some code enrichments and the associated configuration information.
- **Specific Service Adapters.** These SAs come with a hard-coded set of data feeders that require a patch or an implementation change to modify it. For

this kind of Service Adapter, the configuration information is not enough to modify or enhance its data feeder definition.

1.2.2 Service Adapter applications

A Service Adapter application is a running process of a Service Adapter. It is this process that does the data collection. One Service Adapter can be instantiated in several Service Adapter applications that are the running processes of this SA because one SA can be started several times on several hosts.

Service Adapter applications (also called Service Adapter Instance: SAI) are distributed to collect measures as close to the targeted equipment as possible.

The Service Adapter application is the process that hosts the data feeder instances.

For example, there is one SA for OVIS and one SA for SQL technology, but there can be several OVIS Service Adapter applications and several SQL Service Adapter applications.

The Service Adapter application name is "Platform1_director1_saApplication1". In the OpenView SQM environment, the Service Adapter application name is composed of the OpenView SQM platform name, the director name, and the application name: `<platform>_<director>_<application>`

1.2.3 Data Feeders

A Data Feeder is the source of data for OpenView SQM. It consists of a set of indicators, which are made up of data feeder parameters that give information about the different performance aspects of a specific resource type.

A Data Feeder is linked to a third party product and, as a consequence, to a Service Adapter. One Service Adapter can implement several data feeders.

For examples in the SQL SA (referring to Figure 1), a data feeder is mapped to a database table. Each Data Feeder parameter is mapped to a column in this table. In the TeMIP Service Adapter a data feeder is mapped to a TeMIP class and a data feeder parameter is mapped to a TeMIP attribute.

1.2.4 Data Feeder Definitions

A Data feeder definition (DFD) is the description of a data feeder. It contains the list of parameters to be collected and all the associated information needed to access that data.

The following information is mandatory for each DFD (refer to 1.2.3 for further details):

- A name: the data feeder naming convention is described in the next chapter.
- A version (by default: "v1_0").
- A set of data feeder definition parameters (as described in the next chapter).
- A measurement reference point (MRP) naming scheme. The MRP naming scheme describes how this DFD defines the measurement points of all its data feeder instances (for example, by concatenating data feeder property values with fixed strings, or by using a constant fixed string).

Other data are optional:

- Data feeder properties (explained in detail in the following chapters).
- Information about how to access performance indicators (data stored in the SQM Central Repository).

Because the data feeder is linked to a Service Adapter, the prefix of the data feeder definition name contains the name of the Service Adapter it relies on. There are families of DFD depending on the SA they rely on: “sql_DFD1”, “sql_DFD2”, etc. The DFD name must identify uniquely a data feeder within OpenView SQM.

Not all the DFD information is stored in the same place. The part of DFD that is used by OpenView SQM is stored in the SQM Service Repository Manager, and the part of the DFD that is only used by the Service Adapters is stored in the configuration part of the Central Repository (see Figure 2: SA and OpenView SQM Layers).

The OpenView SQM Service Repository Manager stores two types of information about the DFD: static data and collected data. Static data never changes and includes information such as parameter names. Collected data have values that depend on the instantiations of the data feeder, such as DFD properties.

1.2.5 Data Feeder Parameters

A data feeder parameter measures the performance of the monitored Service. It is a metric, a key performance indicator.

The data feeder parameters are used to build service component parameters. They are either global or customer specific. Global parameters are not linked to any subscriber information. Customer specific parameters are collected for a particular customer. When a parameter is customer specific, one parameter can have different values for different customers.

1.2.6 Data Feeder Property

A data feeder property belongs to the DFD and describes data feeder data that is known only during data feeder application configuration. A data feeder property is defined with a name and a type. The value of the property is set during configuration of the data feeder instance.

A property is not a performance indicator.

Properties cannot be used in the collection chain like data feeder parameters. That is, a DFD property cannot be bound to a service parameter. However, they can be displayed in the OpenView SQM user interfaces to help you distinguish data feeder instances.

1.2.7 Data Feeder Instance

A data feeder instance (DFI) publishes the runtime values of a data feeder. There are several data feeder instances for a unique DFD.

Data feeder instances collect DFD parameters (performance indicators) for a specific measurement reference point (MRP). When a DFD parameter is customer dependent, it measures the perceived QoS from an MRP for all the subscribers.

Data feeder instances are stored in the OpenView SQM Service Repository Manager, and they do not need any extra configuration information.

A data feeder instance is uniquely identified by the concatenation of:

- The DFD name the data feeder instance depends on.
- The DFD version.
- The measurement reference point.

1.2.8 Connector

All the data used to connect to a third party product are gathered into a connector. There is one connector for each opened connection to the third party product. A connection can be remote or local on the same host where the Service Adapter application is running. Some Service Adapters applications are connected remotely from the host they monitor, while others need to be installed on the same machine.

The parameters of the connector are purely configuration information. These parameters are optional and can be hard coded, hidden in a property file, or stored in the Central Repository.

The connector parameters do not belong to the OpenView SQM Service Repository Manager. Instead, they are stored in the Central Repository Server (see Figure 2: SA and OpenView SQM Layers) of the repository layer.

Connectors are part of Service Adapter application configuration data.

1.2.9 Measurement Reference Point

The role of the measurement reference point (MRP) is to give the physical location of a performance indicator. It describes all the information about where the Service Adapter measured the performance data value pushed to OpenView SQM. The MRP is the logical access point in the service infrastructure.

An example illustrates the role of the MRP in OpenView SQM data collection.

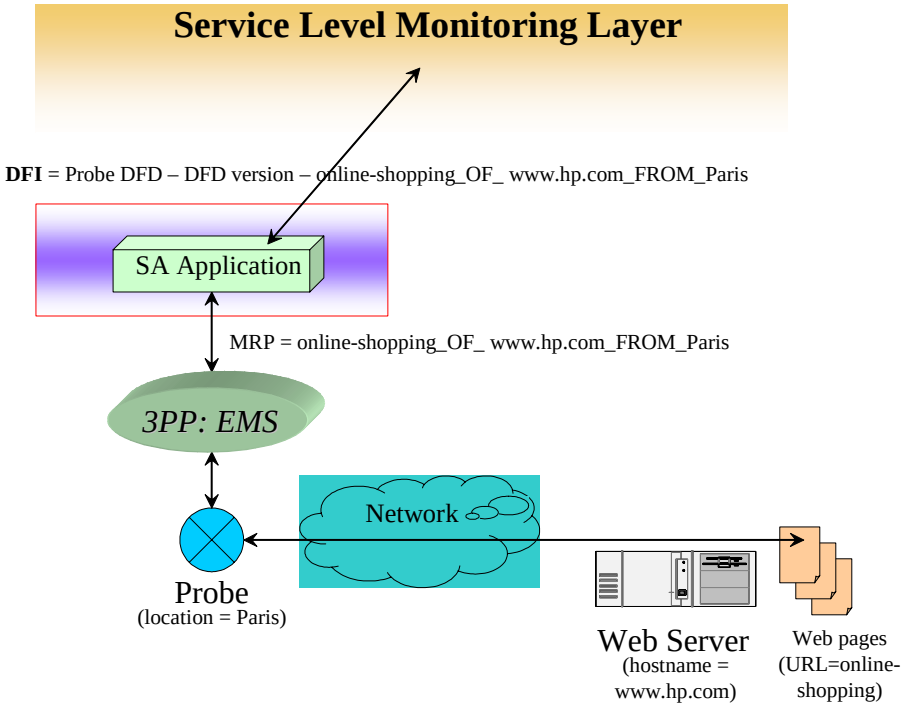


Figure 4: MRP Example

The probe accesses specific Web pages using HTTP and measures the performance of IP and HTTP using QoS parameters. These indicators are collected by the EMS and the Service Adapter application collects them. To define precisely where the measure was made, we need to know the following:

- What is measured: the URL of the HTML pages.
- Where the measure is made: the Web server.
- From where the measure is made: the probe location.

So, the MRP is identified by the probe location, the Web server and the URL of the pages that are accessed. The MRP naming schema for this DFD is:

<page URL> + "_OF_" + <web server> + "_FROM_" + <probe location>

The MRP of the DFI chosen in this example is:

"Online-shopping_OF_www.hp.com_FROM_Paris"

The Measurement Reference Point (MRP) should gather all information that distinguishes two performance data values coming from the same data feeder and for the same parameter name, except:

- The timestamps.
- The subscriber ID and subscriber domain.

The timestamp and the subscriber information are kinds of data that go into specific fields of the OpenView SQM performance data value messages (associated to the value of each data feeder parameter).

The MRP is used to distinguish two data feeder instances built from the same data feeder definition (with the same version).

The data feeder instance and its MRP are related as follows:

- The data feeder instance ID is computed from:

DFD name + DFD version + MRP

  - As a result, you can have several data feeder instances that use the same MRP if these DFIs come from different DFDs

The MRP is a string that can be displayed in the OpenView SQM User Interfaces. It can be constant (like "Paris") or coming from a concatenation of the values of some Data Feeder Properties.

The MRP is one of the most critical concepts of Service Adapters because it drives the number of data feeder instances monitored in OpenView SQM and their accuracy. An MRP should describe precisely where the measure is made, and not how it is made.

Fo example, the same FTP server is being monitored by two different third party product agents. Each agent is connected to two different Service Adapter applications as illustrated in Figure 5.

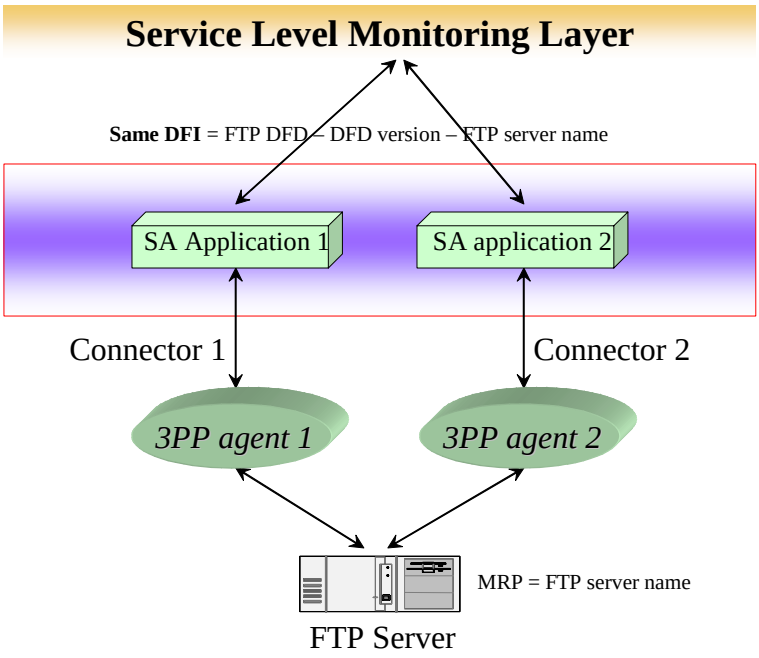


Figure 5: When MRP is different from metric access

The Service Adapter application name and the connector information (including the name of the third party product agent) are not of interest to OpenView SQM. Instead, OpenView SQM needs to know whether the FTP service is available or not. The MRP describes where the measure is made using the FTP server name (IP address, FTP port number, and any other information required to identify the FTP service being monitored). The MRP does not determine how the information is retrieved (connector attributes like the 3PP agent name). As a result, the two Service Adapter applications (processes) will collect information for the same data feeder instance, because the DFI identifier is constructed from the data feeder definition name, the version of the data feeder definition, and the MRP. OpenView SQM will receive duplicated messages for the same performance indicators, which is a normal behavior.

In real life, you would not want to receive the same information from several sources. This example illustrates that even if information is collected twice, it should not influence the MRP you choose.

As another example, one SA application is connected to two different hosts to measure their respective CPU loads. This configuration is illustrated in Figure 6.

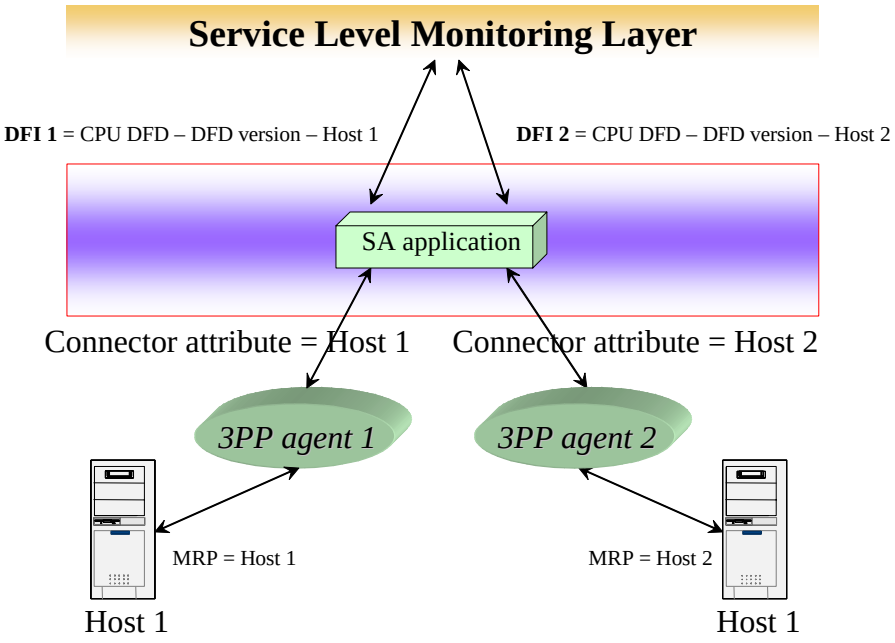


Figure 6: When MRP equals metric access

The connector attribute refers to the agent and the host where the CPU measure is made. These attributes constitute the MRP. No distinction is made between the way performance indicator are retrieved and where these indicators are measured. Because the MRPs are different, the Service Adapter application collects information from the two data feeder instances.

1.2.10 Subscriber and customer

A subscriber is the person who uses the service. Subscribers are given access to the telecommunication services by a customer. The customer pays for the service and owns the contractual relationship. For example an HTTP service can be used by Mr.Dupont identified by his IP address Dupont@hp.com and paid by HP Corporation. The subscriber is Dupont and the customer is HP.

Service Adapters deal with subscribers (even if in some cases subscribers are equivalent to customers), while OpenView SQM is interested only by customers and handles the resolution between subscribers and customers.

1.2.11 Example

The following example is illustrating the previous definitions.

1.2.11.1 Input

A customer database contains the following tables as input for the indicators to be monitored in SQM. This is the input for the Service Adapter.

Table of Voice Call Detail Reports:

Calling number	Local switch node	Remote switch node	Creation date	Time stamp	Average elapsed time	Number of failures
600007	b4dns20	b4dns175	12/06/2002	18:11:53	8	5
600007	b4dns20	b4dns175	12/06/2002	18:40:10	3	2

Table of Data Call Detail Reports:

Calling number	Local switch node	Remote switch node	Creation date	Time stamp	Average elapsed time	Number of failures
600004	b4dns20	b4dns19	12/06/2002	18:33:24	12	1
600002	b4dns21	b4dns104	12/06/2002	18:33:45	1	0

1.2.11.2 Output

The Service Adapter produces the following message. This message is sent to SQM Service Level Monitoring Layer.

Msg ID	52
DFIPerfMeasure	

DFI Id	Cdr_voice – v1_0 - b4dns20 - b4dns175
DFD Name	Cdr_voice
DFD Version	V1_0
SA Application Id	Platform1_director1_saApplication1
DFIMeasure	

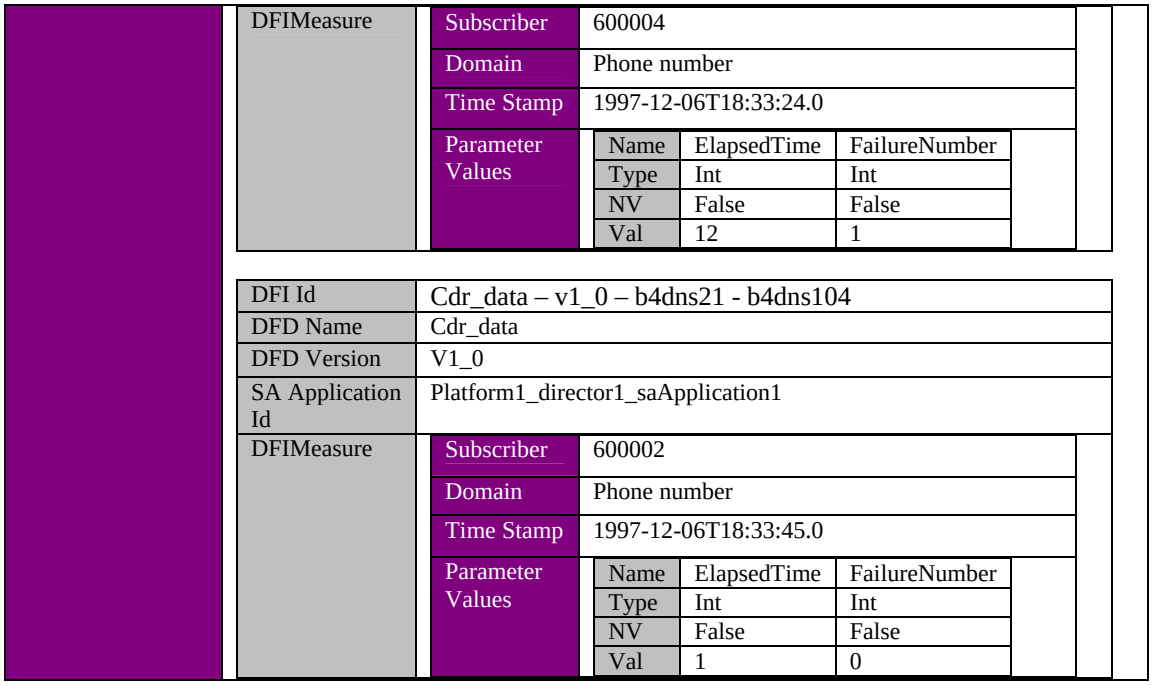
Subscriber	600007
Domain	Phone number
Time Stamp	2002-12-06T18:11:53.0
Parameter Values	

Name	ElapsedTime	FailureNumber
Type	Int	Int
NV	False	False
Val	8	5

Subscriber	600007
Domain	Phone number
Time Stamp	2002-12-06T18:40:10.0
Parameter Values	

Name	ElapsedTime	FailureNumber
Type	Int	Int
NV	False	False
Val	3	2

DFI Id	Cdr_data – v1_0 - b4dns20 - b4dns19
DFD Name	Cdr_data
DFD Version	V1_0
SA Application Id	Platform1_director1_saApplication1



DFIMeasure		
	Subscriber	600004
	Domain	Phone number
	Time Stamp	1997-12-06T18:33:24.0
	Parameter Values	Name: ElapsedTime, FailureNumber Type: Int, Int NV: False, False Val: 12, 1

DFI Id	Cdr_data – v1_0 – b4dns21 - b4dns104	
DFD Name	Cdr_data	
DFD Version	V1_0	
SA Application Id	Platform1_director1_saApplication1	
DFIMeasure	Subscriber	600002
	Domain	Phone number
	Time Stamp	1997-12-06T18:33:45.0
	Parameter Values	Name: ElapsedTime, FailureNumber Type: Int, Int NV: False, False Val: 1, 0

OpenView SQM uses the “elapsed time” and the “number of failures” parameters. These parameter values will be bound to service component parameters and an SLA can be constructed using them.
In this example, there is one message coming from one Service Adapter application (called Platform1_director1_saApplication1) and two different data feeders (Cdr_voice and Cdr_data). For the first data feeder, there is only one data feeder instance containing two parameter values with different timestamps (18:11:53 and 18:40:10). The second Data Feeder has two data feeder instances (b4dns20 - b4dns19 and b4dns21 - b4dns104).

1.3 Service Adapter lifecycle

In this chapter we will enumerate the steps to be followed when configuring a SQM Platform to host a Service Adapter:

- **deployment preparation**: as the Service Adapter will need to be configured to connect to Third Party products, run on a different system from the SQM SLM Primary Server, support multiple DFIs on multiple SA Applications, it is really important to plan in advance, where it will be installed and how it will be configured to provide the best performance for data acquisition.
- **installation and configuration**: depending of the type of Service Adapter that will be installed, some configuration steps will need to be performed to
  - Create one or more applications associated to this Service Adapter
  - Discover DFD and DFIs from the Third Party Product and prepare the input files for the setup of the SA applications.
- **setup and startup**: this step consists in
  - loading each defined Service Adapter Application configuration into the SQM Repository Manager and Repository Server,
  - Start the Service Adapter application(s).

1.3.1 Deploying Service Adapters in OpenView SQM

1.3.1.1 OpenView SQM Concepts

Each Service Adapter application name is the concatenation of the SQM platform name, the director name, and the application name:

platform name + “_” + director name + ”_” + application name

Platform, director, and application are logical ways to group and identify the various components of OpenView SQM. The relation ship between these logical groups is:

$$\text{Applications} \subset \text{Directors} \subset \text{Platforms}$$

Applications are included in one director, and all the directors are included in one platform. A director is assigned to one host, and a host can handle several directors.

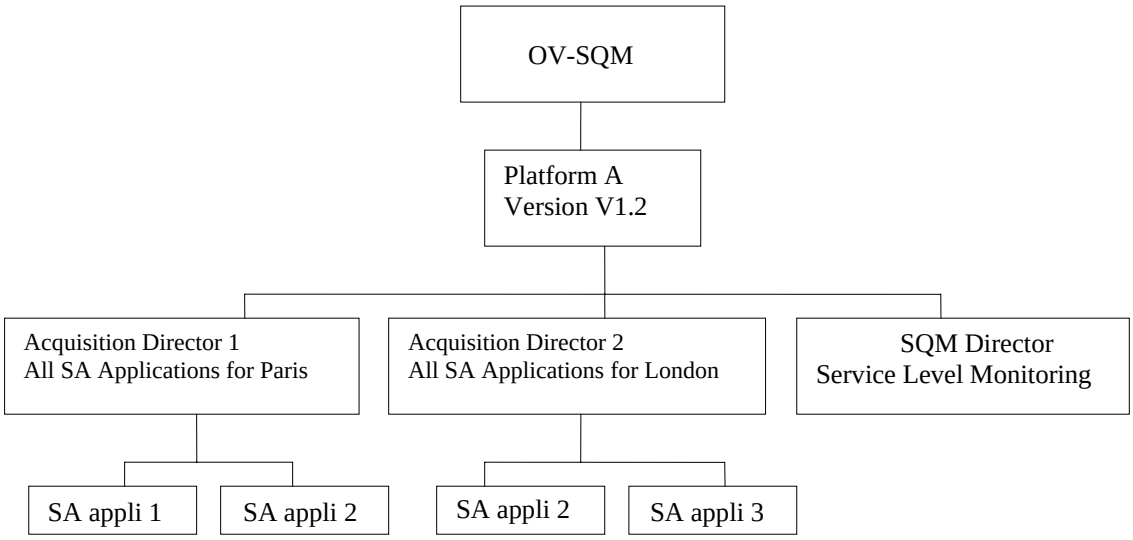


Figure 7: Hierarchy of Platform, Director, and Application

Each level acts as a different name space for each sub element. Two different directors can have applications with the same name (director1 contains **SA application 2** and director2 also contains **SA application 2**). However, these applications are different because they belong to a different director. That is the reason why the full SA application name equals: platform name + director name + application name. The SA application name identifies a given SA application uniquely for the whole OpenView SQM platform.

1.3.1.2 About the Platform

The platform is unique for one installation of the OpenView SQM. So, all the OpenView SQM components installed and belonging to the same OpenView SQM version have the same platform name. For a given customer, the platform concept is useful when migrating from one OpenView SQM version to a more recent version.

Two OpenView SQM components belonging to different platforms can be installed on the same host. This is because the SQM Data Directory ($TEMIP_SC_VAR_HOME directory) is composed of:
<user defined directory path> / <SQM platform name>.

1.3.1.3 About the Director

A director is a logical group of applications. All the applications of a given director can be managed by commands in the OpenView SQM kernel (refer to the ***SQM Administration Guide*** for more information about these administration commands). When you start or stop a director, you also start or stop all the applications belonging to that director.

The same director can group different SQM applications that are installed on different hosts. The commands you run on OpenView SQM to act on directors can be executed on remote systems.

The administrator decides how best to gather application in directors. The following section provides some recommendations.

1.3.1.4 About the Application

The application name corresponds to the name of the TiBCO adapter instance of the OpenView SQM component.

1.3.1.5 Deployment Recommendations

Service Adapter applications do not belong to the OpenView SQM core directors because they are meant to be installed close to the monitored NMS.

The following figure illustrates an example of deployment architecture:

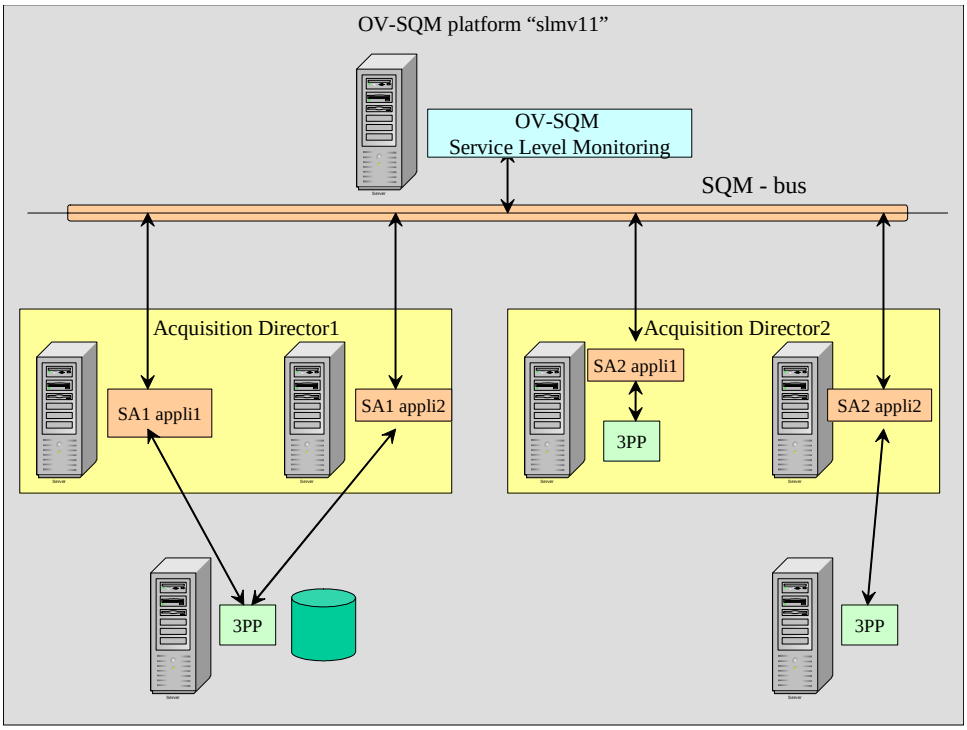


Figure 8: SA application Deployment Architecture

You can group Service Adapter applications into SQM directors using the following criteria:

- Technology driven. You use one director for each SA, meaning that all the Service Adapter applications belonging to the same Service Adapter are shared in one director.
- OS driven. You use one director for each OS, so that, for example, all the Service Adapter applications installed on Windows belong to Windows director and all the Service Adapter applications installed on HP-UX systems belong to the HPUX director.
- Geography driven. You use one director for each location, so that, for example, all the monitoring networks for the Service Adapter applications situated in Paris belong to the “Paris” director.

To gather Service Adapter applications into directors, keep in mind that all the Service Adapter applications of one director can be started or stopped in one command on that director. There is no link between the director concept and any performance recommendations. One director can have hundreds of Service Adapter applications, while another can have only ten Service Adapter applications. The number of applications has no affect on performance. So, we recommend that you group Service Adapter applications in directors based on the criteria that these applications need to be restarted at the same time. For example, if many Service Adapter applications handle data coming from the same database, you should group them in the same director so that each time the database is restarted they can all be restarted at once.

1.3.2 Service Adapter installation and configuration

1.3.2.1 Configuration and discovery Tools

Configuration tools depend on the implementation of each Service Adapter. The aim of these tools is to ease the configuration of data feeders, data feeder instances, and the Service Adapter applications for a given Service Adapter.

The user interacting with the configuration tools is the SQM Administrator. This user can configure a Service Adapter application using the tools, change the files produced by the tool, and directly make changes in the Central Repository that contains the configuration data.

Some tools offer the possibility to be connected to the third party product during the configuration phase (a feature called “auto-discovery”). Some Service Adapters offer configuration tools that automatically discover the third party product entities, map them to data feeder instances, and propose them to the user.

Each configuration tool is Service Adapter specific. They all offer a command line mode for remote use in a simple telnet environment.

The data feeder definition configuration tool is optional and only concerns the “generic” Service Adapters.

You can use the data feeder instance configuration tool to produce the registration messages sent to the OpenView SQM Service Repository Manager. The OpenView SQM SLA administration UI proposes a pre-registration of data feeder instances, but this does not prevent you from configuring data feeder instances at the Service Adapter level. It is mandatory that you register and configure data feeder instances in the Service Adapter. You provide additional data that are missing during DFI pre-registration.

Service Adapter application configuration sets the connection parameters between one application and the third party product agents or processes it is connected to. Moreover, during SA application configuration, the administrator needs to set the data feeder definitions that will be monitored by this application.

The documentation of each specific Service Adapter describes the configuration tools in more detail.

1.3.3 Service Adapter Setup and Startup

1.3.3.1 SA Setup

No matter what kind of Service Adapter you are using (the level of configuration tools provided, or if it comes with a predefined configuration), the data needs to be uploaded into the TiBCO Central Repository server and registered in the OpenView SQM Service Repository Manager.

The SQM platform and Acquisition director can be automatically setup with the SA application setup (for more information, refer to the Service Adapter documentation that explains the setup phase).

Uploading the Configuration Files

Depending of the SA type (Generic, Customized, or Specific), the configuration files to be uploaded into SQM Service Repository Manager or Tibco Central Repository Server will be:

- Generated by a discovery tool, for a generic SA
- Brought with the Service Adapter kit, if a Customized or Specific SA

Uploading DFD and DFI to the SQM Service Repository Manager

To upload the data feeder definition and data feeder instance information, use the Service Adapter convenient tools delivered with the SA kit. The DFD and DFI loading commands are generally encapsulated in user friendly menus.

Uploading configuration in Tibco Central Repository Server

This phase will load DFD and connector and SA application configuration into the Central Repository.

The loading of the Service Adapter Repository Server information is generally performed using the appropriate tools provided by the Service Adapter. Always refer to the specific Service Adapter User's Guide to perform this step.

1.3.3.2 SA Startup

Starting a Service Adapter Application

To start the Service Adapter application, you must be connected as the sqmadm user (for Unix System) and must have already set all of the OpenView SQM environment variables (refer to the ***OpenView SQM Administration Guide*** for more information).

The SQM User/Administrator may choose to perform a ***temip_sc_start_platform*** which will lead to the startup of all directors (and their associated applications) of the specified platform, or perform a ***temip_sc_start_director*** to start the acquisition director which hosts the Service Adapter application(s).

If, for troubleshooting reason, the SQM Administrator want to monitor the specific Service Adapter application, it is also possible to only start the application using ***temip_sc_start_application***:

```
$TEMIP_SC_HOME/bin/temip_sc_start_application
            -platform <platform_name>
            -director <director_name>
            -application <SA_application_name>
```

This command has the following parameters:

- `platform <platform name>`
  The SQM platform on which the Service Adapter has been configured.

- `director <director name>`
  The SQM director that hosts the application

- `application <SA_application_name>`
  The name of the application.
  Note that the Service Adapter application full name (also called Service Adapter Instance name) is <*platform_name*>_<*director_name*>_<*SA_application_name*>.

Stopping a Service Adapter Application

To stop the Service Adapter application, you must be connected as the sqmadm user (if it is a Unix host) and you must have set all of the OpenView SQM environment variable (refer to ***SQM Administration Guide*** for more information). Use the following command to start the application:

```
$TEMIP_SC_HOME/bin/temip_sc_stop_application
            -platform <platform_name>
            -director <director_name>
            -application <SA_application_name>
```

You can also stop the Service Adapter application by invoking the stop directive inside the self-management application (Refer the the ***OpenView SQM Administration Guide***).

Are also available the ***temip_sc_stop_director*** to stop the acquisition director and ***temip_sc_stop_platform*** to stop the entire platform and all applications.

Chapter 2

Data Feeder Definitions

Customized Service Adapters automatically contain pre-defined Data Feeder Definitions (DFDs). Otherwise, if the Service Adapter is generic you can create DFDs using a configuration tool. This chapter describes the structure of DFDs, their role, and their locations in the OpenView SQM system directory structure.

2.1 SQM Repository Part of a DFD

The data described below are stored in the internal database of SQM Repository Manager. The visible part is the XML message used to register data feeder definitions in OpenView SQM. This XML definition can be obtained using 2 ways:

- The SQM Service Designer User Interface, where the SQM architect defines the Service and the data feeders that will compose the Service to be managed. The generated output is an XML definition of a data feeder.
- The Service Adapter configuration tools. These tools come with the Generic Service Adapters and are able, in a discovery phase, to generate the Data Feeder XML definition and the Data Feeder Instance definitions depending on the user input information.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sc:NewDFDReq SYSTEM "DTD/tsc_DataFeederDefSrv.dtd">
<sc:NewDFDReq msg.id="14"
              xmlns:sc="http://www.openview.com/SQM">
    <sc:DataFeederDef dfd.label="Call Data Record Voice DFD"
                      dfd.name="Cdr_voice"
                      dfd.version="v1_0"
                      sa.name="Cdr"
                      sa.version="v1_0">

        <sc:Descr>SA for CDR Voice data feeder definition</sc:Descr>

        <sc:ParameterDefs>
            <sc:ParameterDef category="Rate"
                customerDepend.flag="True" datatype="Float"
                parameter.label="Elapsed Time"
                parameter.name="ElapsedTime"/>
            <sc:ParameterDef category="Counter"
                customerDepend.flag="True" datatype="Int"
                parameter.label="Number of failures"
                parameter.name="FailureNumber"/>
        </sc:ParameterDefs>
```

```
        <sc:PropertyDefs>
            <sc:PropertyDef datatype="String"
                            property.label="Local Node"
                            property.name="LocalNode"/>
            <sc:PropertyDef datatype="String"
                            property.label="Remote Node"
                            property.name="RemoteNode"/>
        </sc:PropertyDefs>

        <sc:MRPNamingSchema>
            <sc:FixedString>Cdr_voice – v1_0 - </sc:FixedString>
            <sc:PropertyName property.name="LocalNode"/>
            <sc:FixedString> - </sc:FixedString>
            <sc:PropertyName property.name="RemoteNode"/>
        </sc:MRPNamingSchema>
    </sc:DataFeederDef>
</sc:NewDFDReq>
```

The DFD is identified by its name and its version. The DFD label is displayed in the OpenView SQM user interfaces just like any other element label. The Service Adapter name and Service Adapter version tells you which Service Adapter can manage this DFD. All the Service Adapter applications coming from this Service Adapter can collect information from any data feeder instance built from this DFD.

Each parameter is defined by its name and several other properties. The parameter name is used by OpenView SQM. It can be different from the name coming from the third party product. Each Service Adapter allows you to change the parameter name provided by the third party product when defining it in OpenView SQM. The mapping between the two parameter names occurs during the configuration of the DFD.

The characteristics you can define for each parameter include:

- Its datatype, which can be one of the following:
  - Int
  - AbsTime
  - Float
  - RelativeTime
  - String
  - Enum
- The customer dependency flag. If True, this parameter is linked to the information for a subscriber.
- A category to describe the context in which this parameter is measured. The categories are:
  - Rate
  - Counter
  - Gauge
  - Percent
  - Other

The MRP naming schema is described in the DFD. The DFD shows how to construct an MRP for each data feeder instance built from this DFD. For more information on MRPs, refer to Section 1.2.9.

The property names used in the MRP must refer to existing properties in the properties list contained in this Data Feeder Definition.

2.1.1 Characteristic of DFDs

The following table describes all the fields required for a DFD.

Name	Type	Description	Value
dfd.name	String	Defines this DFD uniquely within OpenView SQM. This name does not exceed 16 characters. It is set at registration time and must contain the Service Adapter name as a prefix to ensure its uniqueness across the different Service Adapters.	Set at registration time.
dfd.label	String	Provides the display name of the DFD. This name can be internationalized.	Equal to DFD name by default
Descr	String	Describes the data feeder	
dfd.version	String	Provides the version of the data feeder. Each DFD is identified by the DFD name and the DFD version. The DFD version is mandatory.	"v1_0" by default for the first version of SQM
sa.name	String	Provides the name of the Service Adapter that handles this DFD.	
sa.label	String	Provides the display name of the Service Adapter. This name can be internationalized.	Equal to SA name by default
sa.version	String	Provides the version of the Service Adapter.	"v1_0" by default for the first version of SQM
ParameterDefs	**list**	Lists the parameters for this DFD. Each parameter is composed of characteristics that are described in 2.1.1.1.	
PropertyDefs	**list**	Lists the names of the properties that are filled at data feeder instanciation time. Each property name is a string.	
MRPNaming Schema	**list**	Provides a sub-list of data feeder property names which values are used to construct the MRP. You must choose the MRP properties so that the uniqueness of the DFI will be guaranteed among all the DFIs belonging to this data feeder. Each property name is a string. The order of the property names in this list is used to construct the MRP. Fixed strings can be added into this MRP naming scheme to make the MRP more readable.	

Table 2 : DFD Characteristics

2.1.1.1 DFD Parameter Characteristics

The following table describes all the characteristics of the parameters of a DFD.

Name	Type	Description	Value
parameter.name	String	Parameter name. This name can be different from the parameter name used by the third party product. By default (without any user action), this name is identical to the parameter name of the Third Party Product.	Equal to 3PP parameter name by default.
parameter.label	String	Name of the parameter used for display purposes. This parameter can be internationalized.	Equal to Parameter-Name by default
Descr	String	Parameter description.	
datatype	String	Data type of the parameter. The connectivity layer of the ServiceAdapters is responsible for type casting third party product parameter types to one of these OpenView SQM datatype.	Int, Float, String, AbsTime, RelativeTime, Enum
units	String	For display purposes only.	
category	String	Parameter category.	Rate, Counter, Gauge, Percent, Other

customer Depend.flag	boolean	Flag that signals if the parameter depends on a customer. This parameter is set to True if it is customer dependant. When set to true, the Service Adapter must give subscriber information and a subscriber domain for each parameter value published to OpenViewSQM.	False

Table 3 : DFD Parameter Characteristics

2.1.1.2 DFD property characteristics

A property is a couple {key, value}, where the key is the property name that is defined at DFD definition phase and the value is known when registering a DFI from this DFD. Each property must have a unique name for the same DFD.

Name	Type	Description	Value
property.name	String	Property name. It is defined during DFD creation and is used to valuate DFI properties during DFI registration.	
property.label	String	The property label is the internationalizable version of the property name, it is used for display purpose. Properties are used in SQM GUIs to sort different DFIs by categories.	Equal to Property-Name by default
Descr	String	Property description	
datatype	String	Can be one of the following: Int, Float, String, AbsTime, RelativeTime and Enum. The property value given during DFI registration must respect this datatype.	Int, Float, String, AbsTime, RelativeTime, Enum

Table 4 : DFD Property Characteristics

2.1.1.3 DFD MRP Naming Schema

The MRP naming schema is an indexed list of MRP elements. An MRP element can be either a **constant fixed string** or a **property name**. The list is indexed to impose an order and so that several MRP elements, and more especially the constant strings, can have the same values.

2.2 Configuration Part of a DFD

The following chapter describes the Service Adapter configuration that is stored in the SQM Central Repository Server. It is assumed that the reader has consulted the *SQM Administration Guide* where it is already explained how to start the Tibco Designer and access to SQM Repository Server information.

In the TiBCO Central Repository, you can access the Service Adapter DFD configuration information stored under each SA Application configuration URL:

/ tibco / private / adapter / ServiceCenter / ServiceAdapters /<*SAName*>/<*SA Version*>/<*Application Name*>_config/extendedProperties/DataFeederDefConfigList/InternalReference

This configuration is loaded (updated) at the Service Adapter installation and setup. In the case of a Specific Service Adapter, the DFD configuration is loaded once at the setup. In case of a Generic Service Adapter, the DFD configuration can be augmented when running the configuration tool and discovering new DFDs.

Figure 9 illustrates the location of this path inside the TiBCO Central Repository. All the DFD configuration data are gathered in this location.

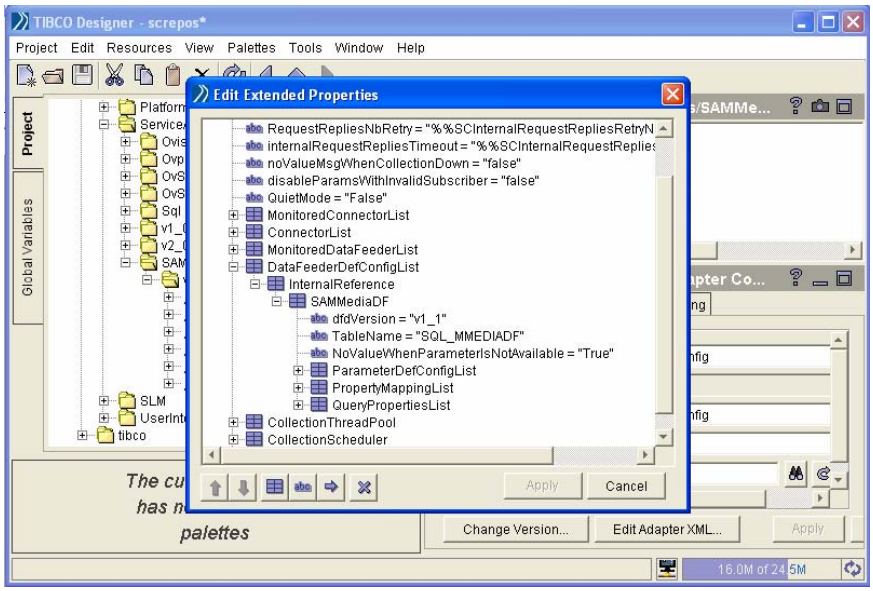


Figure 9: DFD Configuration Data Example

The figure shows the DFD configuration of a SQL Service Adapter application (this definition is specific to the SQL SA and each SA has is own way of defining DFD in the Central Repository Server)

The ParameterDefConfigList structure contains all the configuration data required to map OpenView SQM parameter names to the real parameter name in the Third Party Product (SQL database in this case).

The PropertyMappingList structure contains the entire configuration data needed to retrieve the properties used for each data feeder instance built from this DFD.

The QueryPropertiesList provides the Timestamp, Subscriber and SubscriberDomain fields used to retrieve this information from the Third Party Product (these data are optional if the third party product does not provide them).

The configuration data is different for each Service Adapter. This example tries to describe the main principles for the information stored in the Central Repository for each DFD.

Chapter 3

Data Feeder Instances

Data Feeder Instances (DFIs) are entities that rely on a given DFD. The DFD is the definition, while the DFI is the real element used to collect performance indicators.

3.1 SQM Repository Part of a Data Feeder Instance

DFIs have no associated configuration information. All of the data for DFIs are stored in the SQM Service Repository Manager. Nothing is stored in the TiBCO Central Repository.

DFIs are generally discovered at the configuration steps of the Service Adapter, using the Service Adapter configuration tool. The tool will generate an XML declaration of the DFI that can then be loaded into the Service Repository Manager. A DFI auto discovery process can be put in place (in batch mode) to automatically retrieve new DFI and augment the Service Repository Manager (this feature may or may not be offered by the Service Adapter, so you need to refer to the Service Adapter User's Guide for more information).

A DFI pre-registration is possible when the Service Adapter has not yet been installed and configured. This pre-registration can be done through the SQM SL Administration User Interface, at the service instanciation. Then the Service Adapter phase describe above will be performed to really create the DFI and be ready for the collection.

The DFI XML declaration described below is the output of the Service Adapter configuration tool that is stored into the SQM Service Repository Manager.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sc:DeclareDFIReq SYSTEM
         "DTD/tsc_DataFeederInstanceSrv.dtd">
<!-- Generated the Fri Feb  7 16:25:55  2003 -->

<sc:DeclareDFIReq xmlns:sc="http://www.openview.com/SQM" msg.id="24">
   <sc:DataFeederInstanceDeclarations>
      <sc:DataFeederInstanceDeclaration
            dfi.id="Cdr_voiceb4d_A25"
            dfd.name=" Cdr_voice " dfd.version="v1_0"
            mrp.name="Cdr_voice – v1_0 - b4dns20 - b4dns175"
            sai.id="defaultplatform_defaultdirector_defaultinstance"
            sa.name="SACdr" sa.version="v1_0"
            collState.admin="Locked"
            collState.operational="Disabled"
            collStatus.availability="Unknown">
```

```
         <sc:PropertyValues>
            <sc:PropertyValue
                  property.name="LocalNode"
                  datatype="String">b4dns20</sc:PropertyValue>
            <sc:PropertyValue
                  property.name="RemoteNode"
                 datatype="String">b4dns175</sc:PropertyValue>
         </sc:PropertyValues>

      </sc:DataFeederInstanceDeclaration>
   </sc:DataFeederInstanceDeclarations>
</sc:DeclareDFIReq>
<!-- End of File -->
```

A DFI is identified by an identifier that is computed from the concatenation of the following:

- DFD name + DFD version + MRP

As the identifier is not supposed to exceed 16 characters, some Service Adapter configuration tools may encode (hash-coding) the result of this concatenation. The beginning of the DFI identifier must be as readable as possible. This might give a result like: "`Cdr_voiceb4d_25A`", but the way DFI identifiers are computed is Service Adapter specific. The 16 character rule is applicable for all Service Adapters.

The DFD from which this DFI is built from, is identified by its name and its version.

The DFI MRP is the result of the computation of the MRP naming schema defined in the DFD with the property values that are described in the DFI.

Each DFI is linked to a Service Adapter application. When setting the Service Adapter application name, the administrator, responsible for the DFI registration, chooses the Service Adapter application that will collect on this DFI.

There are three collection states for each DFI:

- The administrative status
- The availability status
- The operational status

The administrative status indicates what SQM wants regarding the DFI collection (LOCKED means that no data needs to be collected for this DFI, and UNLOCKED means that collection must be started as soon as possible for this DFI).

The availability status comes from the Service Adapter application. The Service Adapter application is responsible for setting the status depending on the DFI collection health. The availability status can be one of the following:

- AVAILABLE
- DEGRADED
- FAILED

The operational status summaries the current DFI collection status: ENABLED or DISABLED.

In the example, we declare a new DFI that is LOCKED and DISABLED. Through the SQM SL Administration User Interface or through the command line, a user can choose, when needed, to start the collection for this DFI. The administrative status should always be set to LOCKED when registering a new DFI; SQM will then manage it depending on the binding of the DFI and on the status of the service instances that rely on it.

The properties of the DFI are set to their real values during DFI configuration. They are used to construct the MRP, and to consequently construct the DFI identifier.

3.2 Characteristics of Data Feeder Instances

The following table describes the characteristics of a DFI:

Name	Type	Description	Value
dfi.id	String	DFI identifier. This field defines this DFI uniquely within OpenView SQM. This ID is internal, does not exceed 16 characters, and is computed at registration time from the DFD name, the DFD version, and the MRP name.	Computed at registration time.
dfi.label	String	Name of the DFI, which can be displayed, easily understood, and internationalized.	By default it is the concatenation of DfdLabel, DfdVersion, and MrpLabel.
Descr	String	DFI description.	
version	String	Version of this DFI.	By default, “v1_0” for the first version of OpenView SQM.
dfd.name	String	Name of the DFD this DFI is associated to.	
dfd.label	String	Label of the DFD.	
dfd.version	String	Version of the DFD this DFI is associated to.	
sa.name	String	Service Adapter name. This Service Adapter name must match the name defined in the DFD.	
sa.label	String	Label of the Service Adapter.	
sa.version	String	Version of the Service Adapter.	By default, “v1_0” for the first version of OpenView SQM.
sai.id	String	Identifier of the Service Adapter application which will manage this DFI.	
mrp.name	String	MRP name, which uniquely identifies this DFI from the other DFIs that depend on the same DFD. This name is a concatenation of a subset of the DFI property values (following the MRP naming schema in the DFD) so that the uniqueness of this DFI can be guaranteed for the same data feeder.	
mrp.label	String	A label that is always the MRP name.	MRP name
collError .text	String	Error text that is posted when this data feeder instances is in error. This text may come from the third party product. If one or several parameters are in error, then this text contains the error explanations for each parameter with a line separating each.	Valued at runtime during data collection.
collState .admin	String	Current collection state as required by OpenView SQM.	Set by SQM when it asks to start or stop data collection on this DFI.
collState .operational	String	The collection status computed by the Service Adapter. It depends on the administrative state and the current collection availability.	Valued at runtime during collection.
collStatus .availability	String	The collection status set by the Service Adapter at run-time. This state can be: available, degraded, or failed.	Valued at runtime during collection
Property Values	**list**	List of values mapped to the properties described in the DFD.	

Table 5 : DFI Characteristics

Chapter 4

Service Adapters Installation

4.1 Software and Hardware Requirements

4.1.1 Required Software

4.1.1.1 SQM Acquisition director

Before installing a Service Adapter, remember that it must be installed on a machine where an OpenView SQM kernel subset is installed. This subset contains the minimum SQM Kernel package to run a Service Adapter on a host.

4.1.1.2 SA_Common Library

All the Service Adapters described in this document rely on the SA_Common library. SA Common library comes in the ***SQMSAGTWCOMMON*** subset. This library gathers all the common behaviors of all the Java SQM Service Adapters.

The two directories that contain SA_Common files are:

- $TEMIP_SC_HOME / ServiceAdapters / Common / <***SACommonVersion***> / jar /
- $TEMIP_SC_HOME / ServiceAdapters / Common / <***SACommonVersion***> / properties /

The SA Common library is installed in a versioned directory. Several versions of the Common library can be installed on the same system. Each installed Service Adapter may require a specific version of the SA Common library, and it is the appropriate ***SQMSAGTWCOMMON*** subset version that is to be installed to run the Service Adapter.

4.1.1.3 Generic SA

If the Service Adapter that is going to be installed is a customization of a generic one (see Section 1.2.1 Service Adapter), then the generic Service Adapter it relies on, should have been installed before.

For example, if the SQL SA v1_1 is already installed, then its customization, the "CDR SA" v1_1**Error! Reference source not found.** will be installed under the SQL directory.

- $TEMIP_SC_HOME / ServiceAdapters / Sql / <**Sql version**> / jar / TeSCSASql.jar
- $TEMIP_SC_HOME / ServiceAdapters / Sql / <***Sql version***> / Cdr _v1_1/ …

Like the SA Common library, Generic SA libraries are installed in a version directory. Several versions of a Generic SA can be installed on the same host.
The SA customization has also a version number that is contained in its installation directory. Several versions of the customization can be installed and run on the same host.

4.1.1.4 Third Party Products

In many cases, the Service Adapter requires a third party product software to connect on the local host or to connect remotely. Service Adapters may require any of the following:

- The SA requires the presence of the third party product on the same host where it will be installed.
- The SA requires the presence of a part of the third party product (the library of its open external interface for example).
- The SA does not require anything because the third party product library can be embedded in the Service Adapter installation (only if a legacy agreement allows it).

4.1.2 Hardware Requirements

As Service Adapters rely on the SQM kernel, they need to be installed on hardware and an operating system compatible with SQM kernel requirements. Two platforms are supported: Windows 2000/XP and HP-UX 11.i.

When the Service Adapter must be installed on the same host than the third party product it monitors, it must also be installed on hardware and an operating system compatible with SQM Kernel.

4.2 Installation Tool

The installer used to install the Service Adapter is specific for each Service Adapter. There is no generic “Install Anywhere” application for all Service Adapters. Refer to the documentation for each SA and follow the installation steps if you need further information.

4.3 Directory Structure

4.3.1 Versioning

As already introduced in the previous chapter, every Common libraries, Generic Service Adapters or Customizations have an associated version. This version is important to distinguish 2 levels of the same Service Adapter. The version information is contained in the Service Adapter installation directory structure. Two versions of the same Service Adapter can run on the same host, and this versioning implementation can help the user to smoothly migrate an existing Service Adapter to a higher version.

4.3.2 Global Directory Structure

The directory tree appears as follows on the file-system where you install the Service Adapter:

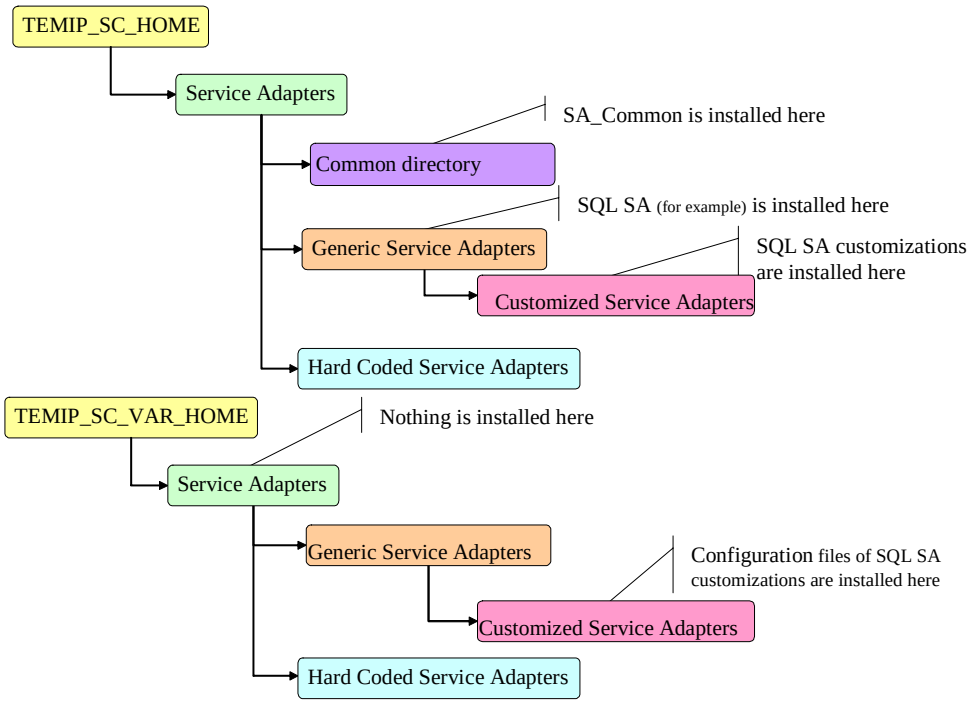


Figure 10: Service Adapters Directory Structure

The root directory, $TEMIP_SC_HOME, contains all of the installed files that are not to be modified by the user. Under the **`Common`** directory, there is the installation of the SA_Common library needed by all the Service Adapters (Each SA Common library is installed under its directory version). At the **`ServiceAdapters`** directory level, there is one directory per Service Adapter if they belong to the "generic" kind or if they do not rely on another generic Service Adapter (called "hard coded"). The "customized" Service Adapters are under the "generic" Service Adapter they rely on.

The $TEMIP_SC_VAR_HOME directory contains files that can be modified by the user. There are all the configuration files (generated by Service Adapter configuration tools) uploaded into SQM, and needed by the Service Adapter applications. This directory structure is the same than the other one, except that SA_Common library does not need any configuration file.
Refer to the appropriate Service Adapter documentation to have a detailed description of the installation directory and the data directory structure (TEMIP_SC_VAR_HOME).