Application Discovery Manager Repository Reference Guide

vCenter Application Discovery Manager 6.1

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

vmware®

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Contents

About This Book

The VMware vCenter™ Application Discovery Manager (ADM) Repository Reference Guide provides information about setting up and configuring the ADM external repository. It describes the table structure and dependencies within the ADM external database and also illustrates the tables. To help you with the custom reports, the book provides examples of reports that can be generated from the ADM external database.

Intended Audience

This document is part of the VMware vCenter Application Discovery Manager documentation set, and is intended for use by corporate Information Technology (IT) personnel who need to monitor enterprise applications and resources and make decisions about acquiring, allocating, and modifying these resources. Readers of this guide are expected to be familiar with these topics:

- Oracle database
- SQL

VMware Technical Publications Glossary

VMware® Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation go to http://www.vmware.com/support/pubs.

Document Feedback

VMware welcomes your suggestions for improving our documentation. If you have comments, send your feedback to docfeedback@vmware.com.

Technical Support and Education Resources

The following sections describe the technical support resources available to you. To access the current version of this book and other books, go to http://www.vmware.com/support/pubs.

Online and Telephone Support

To use online support to submit technical support requests, view your product and contract information, and register your products, go to http://www.vmware.com/support.

Customers with appropriate support contracts should use telephone support for the fastest response on priority 1 issues. Go to http://www.vmware.com/support/phone_support.

Support Offerings

To find out how VMware support offerings can help meet your business needs, go to http://www.vmware.com/support/services.

VMware Professional Services

VMware Education Services courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. Courses are available onsite, in the classroom, and live online. For onsite pilot programs and implementation best practices, VMware Consulting Services provides offerings to help you assess, plan, build, and manage your virtual environment. To access information about education classes, certification programs, and consulting services, go to http://www.vmware.com/services.

Introduction 1

This chapter introduces the VMware vCenter™ Application Discovery Manager (ADM) repository and provides an overview that includes descriptions of data collection, table types, and key fields.

This chapter includes the following topics:

- "Overview" on page 7
- "Synchronizing the External Database" on page 8

Overview

The ADM provides standard reports based on the data collected and stored in the ADM Repository. The schema of the new ADM External Database is simple to understand and allows you to write reporting queries over it without disrupting the performance of the other ADM services.

How Data Is Collected

The ADM collects, manages, and stores data through the components listed in Table 1-1

Table 1-1. ADM Components

Component	Description
Active Discovery-Unix	Collects data from the configuration objects in your data center. The following discovery types apply: ■ IP discovery — Detects hosts or other configuration items with a specific IP address when passive discovery fails to discover them. ■ Detail discovery — Extends the information obtained using passive and IP discovery. It uses common network protocols to remotely query servers in the managed network and obtains supplementary information about network hosts that is added to the database.
Active Discovery-Windows	A discovery engine that uses WMI based discovery policies for performing active discovery on Windows machines.
Passive Discovery	Passively observes the network traffic by performing a deep-packet analysis to discover applications and component relationships in physical and virtual environments. Passive discovery also allows you to do the following: ■ Map dependencies. ■ Count the activity of these dependencies. ■ Identify services.
Aggregator	Receives data from the discovery components and reconciles the data before transferring it to the database component. The aggregator also provides the user interface for using ADM and is the integration point for various integrations, for example, ERDB.
Database	An Oracle RDBMS used to store discovered data and ADM configuration.

Figure 1-1 illustrates how data is collected and stored in the ADM database.

Figure 1-1. Collection, Management, and Storage of ADM Data
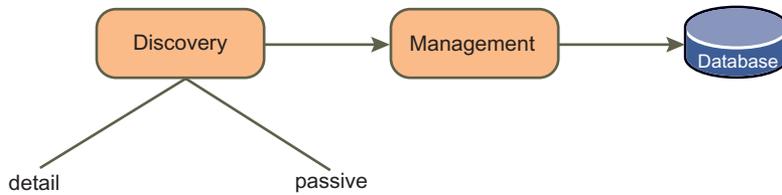
The data stored in the ADM database cannot be customized or manipulated for creating custom reports. Therefore, ADM provides functionality for exporting the data into an ADM external repository, from which custom reports are created.

The data transferred to the external repository is not real-time data. It is synchronized at regular intervals as defined in the External Reports Scheme Configuration policy.

Synchronizing the External Database

Figure 1-2 illustrates how the ADM external database is synchronized with the ADM repository through the ETL (Extract-Transform-Load) process. Extraction and transformation is done through ADM and loading is done by the adm_olap component on the external database host.

Figure 1-2. External Database Synchronization Process
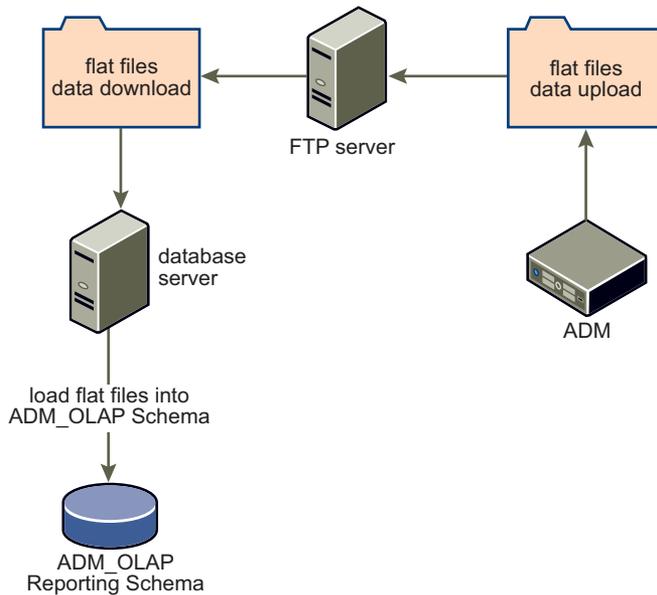
The synchronization process comprises of the following steps:

1   The synchronization is triggered according to the settings in the External Reports Scheme Configuration policy.

2   ADM checks the status of the replication to ensure that the previous synchronization and that any report requests against the external database has completed.

3   ADM extracts the data files for the load process as follows:

   a   The Report Schema is extracted and translated to `.csv` files.

   b   The data files are compressed and placed together in a single Zip archive.

   c   The Zip file is uploaded to the assigned FTP server.

4   The system running the external database periodically checks the FTP server for data files based on the synchronization schedule defined on the external database system.

5   When a data file is found, the following process takes place on the external database system:

   a   The latest Zip file is downloaded.

   b   The Zip file is extracted to the decompressed `.csv` data files.

   c   The database loading is performed using the database specific loader.

Performance and Scalability

The following steps indicate the amount of time required to complete each process:

- Extraction and flat file creation — This step utilizes the ADM resources to the maximum. Therefore, it is scheduled to extract the data at night or some other period of low resource consumption and to allow enough time to complete.

- Transferring the data files — This step involves uploading of the flat files from the ADM Repository appliance to the external database host. This step does not consume many ADM resources. However, the transfer time might vary greatly because of network conditions at the external database host site.

- Loading the database — This step might take few hours to complete.

Set Up the External Reports Scheme Configuration Policy

2

The ADM Console provides the necessary functionality to:

- "Set Up and Monitor the External Reports Scheme Configuration Policy" on page 11
- "Check the Status of the Synchronization Process" on page 13

Set Up and Monitor the External Reports Scheme Configuration Policy

The External Reports Scheme Configuration policy defines the FTP site where the database files are sent and how often extraction is performed. The External Reports Scheme Configuration provides functionality for running the policy automatically at a scheduled time, or starting the synchronization now.

This policy only controls the following steps of the synchronization process:

- Database data extraction
- Database files creation
- Posting the database files to the FTP site

You must configure the external database host to retrieve the data files from the FTP site and complete the synchronization process. Chapter 3, "External Database Setup," on page 15 provides complete details.

Creating the External Reports Scheme Configuration Policy

To set up the External Reports Scheme Configuration policy

1   Log in to the ADM Console as *admin* user.

2   From the ADM Console, navigate to **Connectors** > **Reports** and click **Configuration** in the left pane.

3   In the **Publisher** tab, enter the FTP site to which the data files are published as shown below.
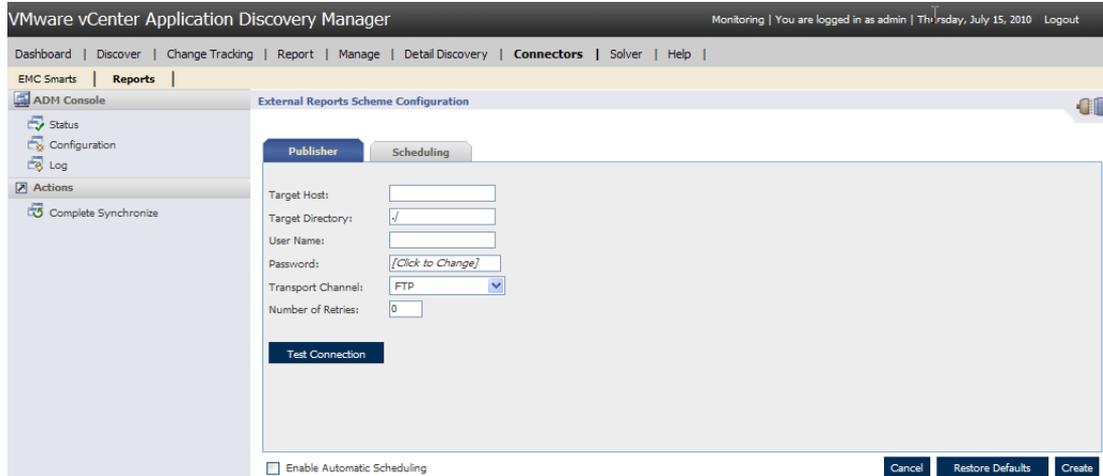
where:

- **Target Host** — the system hosting the FTP server.

- **Target Directory** — the directory on the FTP server host to which the files are sent.

- **User Name** — the name of a user with access to the FTP server.

- **Password** — the password associated with the user name.

  NOTE   Ensure that the user name and password are the same, as used to configure the synchronization on the external database system as described in "Installing and Configuring the External Database for Oracle" on page 16 for Oracle databases user and "Installing and Configuring the External Database for MS SQL" on page 20 for MS SQL Server users.

- **Transport Channel** — allows user to choose between FTP and Secured FTP (SFTP) for data upload from the drop-down menu.

- **Number of Retries** — the number of times the synchronization process attempts, in case of failure.

4   Click **Test Connection** to test the connection to the FTP server based on the publishing credential you entered. If you have entered invalid credentials or if the FTP server is down, the test times out or fails.

5   Click **Create policy**. If you have previously defined the policy, click **Update**.

Running the Synchronization Process at Scheduled Times

To schedule the synchronization process

1   After creating the External Reports Scheme Configuration policy, select the **Enable Automatic Scheduling** option at the bottom of the page as shown below.
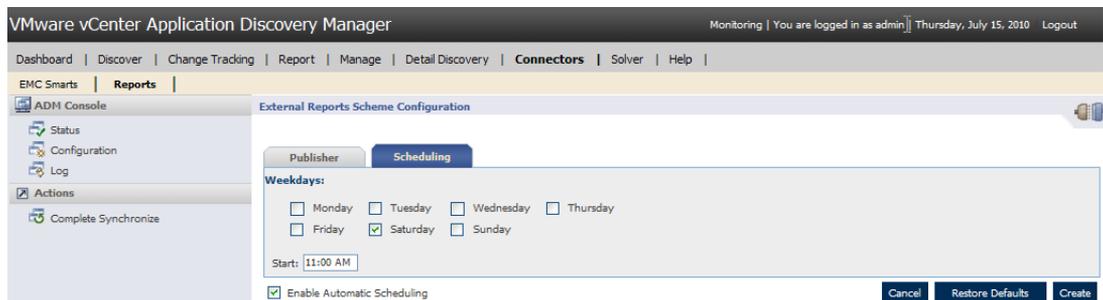
2   In the **Scheduling** tab, select one or more days of the week to schedule the start of the synchronization.

3   Enter the time of day the synchronization occurs.

In the above example the synchronization automatically begins every Saturday at 11:00 A.M.

4   Click **Update** to save the setting.

Running the Synchronization Process Now

To start the synchronization process on demand

After creating the External Reports Scheme Configuration policy, click **Complete Synchronization** under the **Actions** menu in the left pane.

IMPORTANT   If an existing synchronization process has not completed, then ADM does not start the synchronization process.

Check the Status of the Synchronization Process

The ADM Console displays the status of the External Reports Schema Configuration policy that includes the following synchronization steps:

■   Extraction of the database file.

■   Creation of the database file.

■   Posting the database files to the FTP site.

To check the status of the synchronization process

1   Log in to the ADM Console as *admin* user.

2   From the ADM Console, navigate to **Connectors** > **Reports** and click **Status** in the left pane.

3   The display pane lists the following parameters:

■   **Last Sync. Time** — the last time the external database was synchronized with the ADM repository. The time is based on the location of the appliance.

■   **Synchronization Status** — the status of the synchronization process, which is one of the following:

■   **Idle** — the process is idle. Nothing is currently running.

■   **Complete** — the ADM synchronization is complete and the FTP files are loaded to the FTP Server.

■   **Sync in progress** — the ADM synchronization process is in progress.

■   **Scheduled Synchronization Status** — the status of the synchronization schedule, which is one of the following:

Active — the synchronization schedule is set and runs at the scheduled time.

Not Active — there is no schedule associated with the policy.

■   **Populated Elements** — lists the elements that were populated to the external database during the last synchronization.

■   **Count** — gives the number of elements that were populated to the external database.

4   Click **Log** in the left pane to view the messages, describing the events of the synchronization process.

External Database Setup 3

To use the external database functionality you need to perform the following steps:

External Database System Requirements

To incorporate an external database into your ADM environment, you need the following components:

- Access to an FTP server.
- The external reporting database client has to be installed on the same server as the database.
- The hardware and software requirements as follows.

Minimum Hardware Requirements

Ensure that your system meets the following hardware requirements:

- 2 GB memory.
- Dual core 1.8 GHZ CPU.
- 500 MB for the client installation. If the data Zip files are retained on the server, more disk space is needed.
- 1 GB for the database instance.

Software Requirements

The software requirements for each type of database are as follows:

Oracle as the External Database

Ensure the system meets the following requirements when using Oracle as external database:

- Red Hat Enterprise Linux 4.0
- Oracle 10g R2 (10.2.0.1.0)

- Perl with the Archive::Zip module:

  Perl is bundled with RHEL 4.0, but the Archive::Zip module for handling Zip files needs to be installed manually.

  To verify that the Archive::Zip module is loaded, run the following:

  ```
  perl -MArchive::Zip -e 'print'
  ```

  If the result is: **Archive::Zip module is installed**, then the module is installed correctly.

  If the result starts with: **can't find Archive/Zip**, then it is not installed.

  To install this, execute the following command from the command prompt, with root permissions:

  ```
  cpan -i Archive::Zip
  ```

- Perl with Expect module:

  To install, run the following:

  ```
  yum install perl-Expect
  ```

MS SQL as the External Database

Ensure the system meets the following requirements when using MS SQL as the external database:

- Windows 2003 Service Pack 2

- MS SQL 2005

- psftp installed and added to the PATH env properties

- Perl with the Archive::Zip module

  Common Perl distribution for Windows is ActivePerl from ActiveState. The necessary Archive::Zip module is bundled with ActivePerl.

Installing and Configuring the External Database for Oracle

Use the following procedures to install and configure the external database for Oracle.

Install the Client-Side Script

To install the Client-side script

1 Copy the ADM external database installation file (`adm_olap_v.X.X-XXX.zip`, where *v.X.X–XXX* is the ADM version and build number) from http://downloads.vmware.com/Application Discovery Manager.

2 Log in to the Linux Server.

3 Change directories to the location of the installation file.

4 Unzip the `adm_olap_v.X.X-XXX.zip` to a directory of your choice.

   A directory named `adm_olap` is created with all the client-side scripts in it.

5 Change the `adm_olap` directory permission to allow read access to all users.

   To enable read permission, in Linux run:

   ```
   chmod 777 -R adm_olap
   ```

Create the External Reporting Database User

To create an external reporting database user

NOTE You must have database administrator permissions to create an adm_olap user and have $ORACLE_HOME/bin dir in the PATH to install and configure the host on which the external database is installed.

---

1   Switch to a user with oracle permissions for example:

```
$ su – oracle
$Password:
```

2   Change directory to *unzip_path*/adm_olap/schema/oracle/:

```
cd unzip_path/adm_olap/schema/oracle/
```

where *unzip_path*/adm_olap/schema/oracle/ is the directory in which the adm_olap_v.X.X–XXX.zip file was extracted.

3   Log in to the SQLplus server:

```
$ sqlplus '/as sysdba'
SQL*Plus: Release 10.1.0.2.0 – Production on Wed Dec 12 14:17:01 2007
Copyright (c) 1982, 2004, Oracle.  All rights reserved.
Connected to:
Oracle Database 10g Release 10.1.0.2.0 – Production
```

4   Verify that the instance is running, the database is mounted, and the listener is started.

5   Run the SQL script to create the adm_olap user:

```
sql> @create_olap_user.sql

User created.
Grant succeeded.
```

NOTE The user created here must be the same user that was defined when setting up the External Reports Scheme Configuration policy. The External Reports Scheme Configuration defines the FTP server where the extracted data files are transferred and how often the synchronization occurs.

---

6   Type **quit** to exit the sqlplus prompt.

Create the adm_olap Tables

To create the adm_olap tables

NOTE The adm_olap schema creation is meant to be managed by the client Database Administrator (DBA). Simple SQL scripts are provided to support the basic ADM tables. To support additional information add external tables. Changes to the tables provided with ADM are overwritten with each load. The scripts are located where the adm_olap_v.x.X–XXX.zip file was extracted followed by **/schema/oracle**.

---

1   Log in to SQLplus as adm_olap:

```
sqlplus adm_olap/adm_olap
```

2   Run the SQL script to create the tables:

```
SQL> @create_tables.sql
```

3   Type **quit** to exit the sqlplus prompt.

NOTE Do not add constraints. With the constraints turned on, the loader fails to load the data.

Configure the Synchronization Process

To configure the synchronization process

1   Configure the sync.properties file located in the `adm_olap` directory:

NOTE   An example file, `sync.properties.example`, is provided in adm_olap, to use as a template to configure the synchronization process.

---

a   Make a copy of the sync.properties.example file and name it sync.properties:

```
cp sync.properties.example   sync.properties
```

b   Modify the following information in the sync.properties file:

i   For the FTP information, enter the proper credentials, host remote directory, and whether or not to delete the files from the FTP server after the synchronization process completes:

```
ftp.user=ftp-user
ftp.password=ftp-password
ftp.host=ftp-hostname
ftp.dir=remote-dir
ftp.local.dir=tmp
# filter FTP files: consider only files starting with the prefix.
ftp.file.prefix=adm_olap
# delete files from FTP when done?
ftp.delete.files=true
```

ii   Ensure that the loader information points to the Oracle:

```
loader.exec=./load_oracle.pl
loader.dir=load
loader.data.dir=data
```

NOTE   Use the same user credentials when setting up the External Reports Scheme Configuration policy as described in "Creating the External Reports Scheme Configuration Policy" on page 11.

---

iii   Modify the archiving if necessary:

```
# archive option will save a local copy of the compressed data at the specified
              directory
archive=true
archive.dir=archive
```

archive and archive.dir defines whether or not to store the zipped files locally for history tracking.

If necessary add the hooks: hook.before.load= and hook.after.load= allow custom behavior just before the data loading and after the loading is complete. Such hooks are used to add or drop indices, to add client specific data for loading together with ADM data, or any other desired actions. The property contains the path to an executable script:

```
# This can be used, for example, to remove indexes before the data loading
# and re-create the indexes after the load is complete.
#hook.before.load=
#hook.after.load=
```

2   Configure the database loader for Oracle.

The Oracle loader script is configured using the `oracle.properties` file, that is located in the *unzip_path*/`adm_olap/load/` directory. An example file, `oracle.properties.example`, has been provided in *unzip_path*/`adm_olap/load/`:

a   Make a copy of the `oracle.properties.example` file and name it `oracle.properties`:

```
cp oracle.properties.example oracle.properties
```

    b   If you had defined a custom `adm_olap` user name and password when you had created the database user as described in "Create the External Reporting Database User" on page 17, then change the user name and password credentials accordingly:

```
user.id=adm_olap/adm_olap
```

    c   Change the delete.data.files parameter to =true if you want to delete the data files after they are loaded. If you keep the default, you need additional dataspace than defined in "External Database System Requirements" on page 15:

```
delete.data.files=false
```

3   Load the time data.

The current `adm_olap` schema has a time dimension table that is static. Fill the time table once and leave it unchanged thereafter. Perform the following steps before the loader first run:

```
cd adm_olap
unzip static_data/time.csv.zip –d data/
```

The `time.csv` data is loaded the next time the synchronization process runs.

4   Test the configuration to ensure that you have installed and configured the client correctly:

    a   Run a complete synchronization from the ADM Console as described in "Running the Synchronization Process Now" on page 13.

    b   After the upload completes, perform the following steps on the system on which the database is installed:

        i   Log in as a user with Oracle permissions.

        ii   Change directories to *unzip_path*/`adm_olap`/ and run the synchronization process:

```
cd <unzip_path>/adm_olap/
chmod 755 sync_adm_olap.pl
$ ./sync_adm_olap.pl
```

> **NOTE** For secure ftp, use the `sync_adm_olap_sftp_oracle.pl` script instead of `sync_adm_olap.pl`

5   Schedule the synchronization process on Linux.

    a   Log in as a user with oracle permissions and enter:

```
crontab –e
```

    b   Verify the following information. It has to be part of the `crontab` for it to run properly:

```
PATH=... <oracle home full path/bin>
ORACLE_SID=<>
ORACLE_HOME=<>
<cron expression><install dir>/adm_olap/sync_adm_olap.pl >>
                 <install_dir>/adm_olap/log/sync.log 2>&1
```

> **NOTE** For secure ftp, use the `sync_adm_olap_sftp_oracle.pl` script instead of `sync_adm_olap.pl` and for more information about cron expression please contact your unix admin.

The following is an example of a cron entry in Linux, where the process is scheduled to run after every 15 minutes:

```
PATH=...
ORACLE_SID=...
ORACLE_HOME=...
*/15 * * * * <install dir>/adm_olap/sync_adm_olap.pl >> <install dir>/adm_olap/log/sync.log
             2>&1
```

Installing and Configuring the External Database for MS SQL

Use the following procedures to install and configure the external database for MS SQL.

Install the Client-Side Scripts

To install the client-side script

> **NOTE** You must have administrator privileges to install the client-side scripts on Windows.

1 Copy the ADM external database installation file (adm_olap_*v.X.X–XXX*.zip) from http://downloads.vmware.com/Application Discovery Manager.

2 Unzip the adm_olap_v.X.X–XXX.zip to a directory of your choice.

A directory named adm_olap is created with all the client-side scripts in it.

Install the External Database for MS SQL Server

To install the external database for MS SQL

> **NOTE** You can use either SQL Server Management Studio or SQLCMD command to run the installation scripts. The following steps demonstrate the commands using SQLCMD.

1 From the Windows **Start** menu, select **Run** and type **cmd**. The command prompt appears.

2 Change directories to *unzip_path*/adm_olap/schema/sqlserver/:

```
cd <path>\adm_olap\schema\sqlserver
```

3 Run the create_database.sql script to create the ADM external database:

```
sqlcmd –S <serverName> –i create_database.sql
```

4 Run the create_olap_schema.sql script to create the adm_olap schema:

```
sqlcmd –S <serverName> –i create_olap_schema.sql
```

5 Run the create_adm_olap_login.sql script to create the login:

```
sqlcmd –S <serverName> –i create_adm_olap_login.sql
```

6 Run the create_user_for_login.sql script to create the login user:

```
sqlcmd –S <serverName> –i create_user_for_login.sql
```

> **NOTE** The user created here, must be the same user that was defined when setting up the External Reports Scheme Configuration policy. The External Reports Scheme Configuration policy defines the FTP server where the extracted data files are transferred and how often the synchronization occurs.

7 Run the create_tables.sql script to create the adm_olap tables:

```
sqlcmd –S <serverName> –i create_tables.sql
```

> **NOTE** Do not add constraints. With the constraints turned on, the loader fails to load the data.

Configure the Synchronization Process

To configure the synchronization process

1   Configure the sync.properties file located in the *unzip_path*/adm_olap directory.

NOTE   An example file, sync.properties.example, is provided in *unzip_path*adm_olap to use as a template for configuring.

---

a   Make a copy of the sync.properties.example file and name it sync.properties:

```
copy sync.properties.example  sync.properties
```

b   Modify the following information in the sync.properties file:

i   For the FTP information, enter the proper credentials, host remote directory, and whether or not to delete the files from the FTP server after the synchronization process completes:

```
ftp.user=ftp-user
ftp.password=ftp-password
ftp.host=ftp-hostname
ftp.dir=remote-dir
ftp.local.dir=tmp
# filter FTP files: consider only files starting with the prefix.
ftp.file.prefix=adm_olap
# delete files from FTP when done?
ftp.delete.files=true
```

NOTE   Use the same user credentials when setting up the External Reports Scheme Configuration policy as described in "Creating the External Reports Scheme Configuration Policy" on page 11.

---

ii   Ensure that the loader information points to the MS SQL Server:

```
loader.exec=load_sqlserver.pl
loader.dir=load
loader.data.dir=data
```

where:

loader.dir and loader.exec provide the location and script to run for loading the data.

loader.data.dir is the data directory where the decompressed files are stored (relative to the *unzip_path*/adm_olap directory).

unzip_path/adm_olap/data is the default directory and is left unchanged.

iii   If necessary modify the archives:

```
# archive option will save a local copy of the compressed data at the specified
              directory
archive=true
archive.dir=archive
```

archive and archive.dir defines whether or not to store the zipped files locally for history tracking.

If necessary add the hooks, hook.before.load= and hook.after.load= allow custom behavior just before the data loading and after the loading is complete. Such hooks are used to add or drop indices, to add client specific data for loading together with ADM data, or any other desired actions. The property contains the path to an executable script:

```
# This can be used, for example, to remove indexes before the data loading
# and re-create the indexes after the load is complete.
#hook.before.load=
#hook.after.load=
```

2   Configure the database loader for MS SQL Servers.

The SQL Server loader script is configured using the sqlserver.properties file, that is located in the *unzip_path*/adm_olap/load/ directory. An example file, `sqlserver.properties.example`, is provided in *unzip_path*/adm_olap/load/:

a   Make a copy of the `sqlserver.properties.example` file and name the file as `sqlserver.properties`.

b   Change the `delete.data.files` parameter to =true if you want to delete the data files after they are loaded. If you keep the default, you need additional dataspace than is defined in "External Database System Requirements" on page 15:

`delete.data.files=false`

3   Load the time data.

The current `adm_olap` schema has a time dimension table that is static. Fill the time table once and leave it unchanged thereafter. To have the loader load the time data upon initiation:

Unzip the `static_data/time.csv.zip` file into the *unzip_path*/adm_olap/data directory.

The `time.csv` data is loaded the next time the synchronization process runs.

4   Test your configuration:

a   Run a complete synchronization from the ADM Console as described in "Running the Synchronization Process Now" on page 13.

b   After the upload completes perform the following steps on the system on which the database is installed:

    i   Log in as a user with administrator permissions.

    ii   From the Windows **Start** menu, select **Run** and type **cmd**. The command prompt appears.

    iii   Change directories to *unzip_path*\adm_olap\and run the synchronization process:

```
cd <unzip_path>\adm_olap\
<perl_dir>\bin\perl.exe <install_dir>\adm_olap\sync_adm_olap.pl
```

NOTE For secure ftp, use the `sync_adm_olap_sftp_sqlserver.pl` script instead of `sync_adm_olap.pl`

Avoid id key verification for the first time by creating the id key as follows:

Run `psftp` manually against the ftp server using command line `psftp –pw` *password user@host* and then type **yes** when prompted.

5   Schedule the synchronization process to run automatically for the MS SQL database:

NOTE For the Windows scheduled task functionality, the flexibility to schedule tasks is limited. The schedules available are daily, weekly, monthly, and one time only, when the computer starts, or when the user logs in.

a   From the Windows **Start** menu, navigate to **Setting** > **Control Panel** > **Scheduled tasks**.

b   Double-click **Add Scheduled Task** to open the **Schedule Task Wizard**.

c   Click **Next** to select the task you are scheduling.

d   Browse to the *unzip_path/*adm_olap directory and select `sync_adm_olap.pl`.

NOTE For secure ftp use the `sync_adm_olap_sftp_sqlserver.pl` script instead of `sync_adm_olap.pl`

e   Enter a name for the `syn_adm_olap.pl` task schedule.

f   Select how often you want to run the task from the options and click **Next**.

g   Depending on the interval selected, define the time, days, and the date to start the task.

h   Enter the user credentials of the host administrator to ensure that environment settings are configured correctly.

i   If the user name and password is recognized, a message is displayed confirming the task has been scheduled.

j   Select **Open advanced properties for this task when I click Finish** option.

k   Click **Finish**.

l   From the Windows **Start** menu, navigate to **Setting** > **Control Panel** > **Scheduled tasks**.

m   Right-click the task and click **Properties.** The task Properties dialog appears.

n   The script is a Perl script and has to be run by Perl. Change the **Run** field to obtain the full path of the perl executable. For example:

`C:\Perl\bin\perl.exe <install_dir>\adm_olap\sync_adm_olap.pl`

where: `C:\Perl\bin\` is the Perl installation directory.

If Perl is installed in another location, be sure to use the correct path.

o   View the loader logs under `unzip_path\adm_olap\load\sqlserver\log`

Manually Loading Data to the External Database

The data files are manually loaded to the external database.

To load the zip files manually for Oracle

1   Unzip the contents of the zip file to the `unzip_path`/adm_olap/data directory:

`unzip adm_olap_v.X.X–XXX.zip –d adm_olap/data/`

2   Change directories to the loader base directory and run the loader script:

```
cd <unzip_path>/adm_olap/load/
./load_oracle.pl
```

To load the zip files manually for MS SQL Server

1   From Windows Explorer, copy the `adm_olap_v.X.X–XXX.zip` file to `unzip_path\adm_olap\data\` folder.

2   Right-click the `adm_olap_v.X.X–XXX.zip` file and unzip it to the same location.

3   From the Windows **Start** menu, select **Run** and type **cmd**. The command prompt appears.

4   Change directories to the loader base directory and run the loader script:

```
cd <unzip_path>\adm_olap\load\
load_sqlserver.pl
```

Checking the Status of the External Database Load Process

Use the following log files to check the status of the load process to the external database:

■ For Oracle, check: */unzip_path*/adm_olap/load/oracle/log

■ For MS-SQL, check: *unzip_path*\adm_olap\load\sqlserver\log

Upgrading the External Database

IMPORTANT The upgrade process requires a complete uninstall of the existing database version and an installation of the new database version.

Perform the following steps to upgrade the external database.

Upgrading for MS SQL

When upgrading the external database for MS SQL:

1 Optionally, record the **Scheduled Task** configuration:

   a From the Windows **Start** menu, navigate to **Setting** > **Control Panel** > **Scheduled tasks** and open the **sync_adm_olap** task.

   b Record the configuration for later use.

2 Delete the **sync_adm_olap** task.

   a From the Windows **Start** menu, navigate to **Setting** > **Control Panel** > **Scheduled tasks** and open the **sync_adm_olap** task.

   b Right-click the **sync_adm_olap** task and click **Delete**.

3 Ensure that synchronization is not currently running as follows:

   a Log in to the ADM Console as *admin* user.

   b From the ADM Console, navigate to **Connectors > Reports > External Reports Scheme Integration Status**.

   c If **Synchronization Status** indicates synchronization in progress, wait till it completes and then proceed with the upgrade.

IMPORTANT The following step clears the data in the external report database. The data is recovered once you perform a full synchronization after the upgrade is complete.

Ensure that SQL scripts are associated with SQL Server Management Studio before continuing.

---

4 Execute the following SQL scripts located in the adm_olap/scheme/sqlserver folder using the user/password *adm_olap/adm_olap* (To execute, double-click the script):

■ drop_constraints.sql

■ truncate_data.sql

■ drop_tables.sql

5 Execute the following SQL scripts located in the adm_olap/scheme/sqlserver folder using a user with administrator permissions (To execute, double-click the script):

■ drop_olap_schema.sql

■ drop_database.sql

6 Backup the old adm_olap folder that includes the sync.properties and sqlserver.properties files (It contains your configuration settings).

7   Proceed with the installation instructions in "Installing and Configuring the External Database for MS SQL" on page 20.

> ⚠️ **CAUTION** Do *not* copy the new configuration files over the backed up configuration files. Only use the backed up configuration files as a reference to manually update the new file.

Upgrading for Oracle

When upgrading the external database for Oracle:

1   Ensure the **sync_adm_olap** task is removed from the cron job as follows:

    a   At the command prompt, type:

       `crontab –e`

    b   Delete the task from the file.

    c   Save and exit.

2   Log in to the ADM Console as *admin* user.

3   From the ADM Console, navigate to **Connectors > Reports > External Reports Scheme Integration Status** and check if the **Synchronization Status** is **complete**. If the status is **sync in progress**, wait for it to complete.

4   Backup the folder in which you originally installed the `adm_olap` files. This folder includes the sync.properties and the `oracle.properties` files, that contains your configuration settings.

5   Backup the folder in which you originally installed the `adm_olap` files. This folder includes the `sync.properties` and the `oracle.properties` files, that contains your configuration settings.

> **IMPORTANT** The data in the external report database is lost as a result of the following step. The data is recovered once you perform a full synchronization after the upgrade is complete.

6   Execute the following scripts located in `adm_olap/scheme/oracle`:

- `sqlplus adm_olap/adm_olap @ drop_constraints.sql`
- `sqlplus adm_olap/adm_olap @  truncate_data.sql`
- `sqlplus adm_olap/adm_olap @ drop_tables.sql`

7   Backup the old `adm_olap` folder (includes the sync.properties and oracle.properties files, and contains your configuration settings).

8   Proceed with the installation instructions in "Installing and Configuring the External Database for Oracle" on page 16.

> ⚠️ **CAUTION** Do not copy the new configuration files over the backed up configuration files. Use the backed up configuration files only as a reference to manually update the new file.

ADM ERD Schema

4

This chapter illustrates the table structure and dependencies within the ADM external database. It also describes the tables within the external database in alphabetical order:

This chapter includes the following topics:

Overview of External Database Tables

This chapter provides information about the external database tables.

Database Structure and Dependencies

Figure 4-1 illustrates the structure and dependencies of the ADM external database. Arrow heads are used to identify the table keys.

It is important to note that some of the commonly accessed data is duplicated in many tables to avoid unnecessary joins.

For display purposes, some data is aggregated to a comma-separated list. For example:

- hostnames – a flat list of one or more host names.

- host_ips – a flat list of all IPs of a host.

- groups – a flat list of groups the entity belongs to.

- A single relation called entity_id (in group membership, usage metric, etc.) could be joined with all three "entity" tables (network_element, service, and dependency) although it is not shown in the ERD as a foreign key.

Figure 4-1. External Database Structure and Dependencies
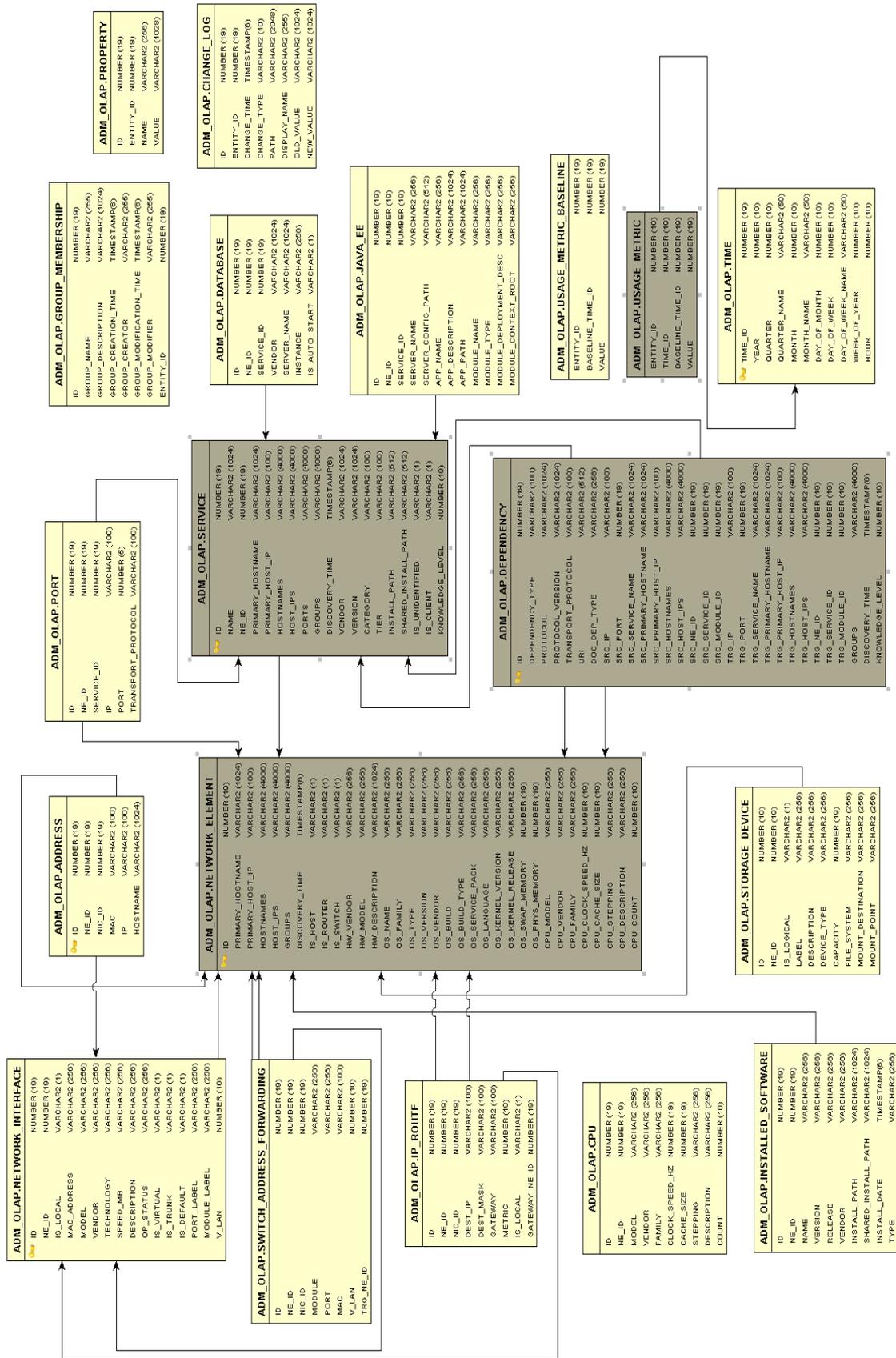
Address

This table identifies information of a network element consisting of the host name, IP address, and MAC address if known. Table 4-1 lists the contents of the Address table.

Table 4-1. Address Table

Name	Datatype	Description
ID	NUMBER(19)	Unique identifier
NE_ID	NUMBER(19)	Network element
NIC_ID	NUMBER(19)	Network interface
MAC	VARCHAR2(100)	MAC address
IP	VARCHAR2(100)	IP address
HOSTNAME	VARCHAR2(1024)	Host name

Where:

ID is a Primary key

NE_ID and NIC_ID are Foreign keys

Change_Log

This table captures the changes of data items on the entities. Table 4-2 lists the contents of the Change_log table

Table 4-2. Change_Log Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
ENTITY_ID	NUMBER(19)	Network Element
CHANGE_TIME	TIMESTAMP(6)(11)	Time of change
CHANGE_TYPE	VARCHAR2(10)	Type of change (added, modified, removed)
PATH	VARCHAR(22048)	"Path" of changed data from the containing entity
DISPLAY_NAME	VARCHAR2(255)	Name of the data property changed
OLD_VALUE	VARCHAR2(1024)	Old value
NEW_VALUE	VARCHAR2(1024)	New value

Where:

ID is a Primary key

ENTITY_ID is a Foreign key

CPU

This table contains the processor information of a network element. The information is identical to what is in the network element table unless the machine has multi-processor types (more than one different architecture on the same machine). Table 4-3 lists the contents of the CPU table.

Table 4-3. CPU Table

Name	Datatype	Description
ID	NUMBER	Unique Identifier
NE_ID	NUMBER	Network Element Identifier
MODEL	VARCHAR2	

Table 4-3. CPU Table (Continued)

Name	Datatype	Description
VENDOR	VARCHAR2	
FAMILY	VARCHAR2	
CLOCK_SPEED_HZ	NUMBER	Max clock speed in Hertz
CACHE_SIZE	NUMBER	Cache size in Bytes
STEPPING	VARCHAR2	Identifies the type and version of the CPU.
DESCRIPTION	VARCHAR2	
COUNT	NUMBER	Number of processors of this type

Where:

ID is a Primary key

NE_ID is a Foreign key

Database

This table contains database server and instance information. Several instances might exist for each server. Table 4-4 lists the contents of the Database table.

Table 4-4. Database Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
NE_ID	NUMBER(19)	Network element
SERVICE_ID	NUMBER(19)	Service
VENDOR	VARCHAR2(1024)	Vendor of the database
SERVER_NAME	VARCHAR2(1024)	Database Server name
INSTANCE	VARCHAR2(256)	Database instance
IS_AUTO_START	VARCHAR2(1)	Autostart flag (Y/N)

Where:

ID is a Primary key

NE_ID and SERVICE_ID are Foreign keys

Dependency

This table represents the relationship between a source host/service and target host/service (capturing topology). Table 4-5 lists the contents of the Dependency table.

Table 4-5. Dependency Table

Name	Datatype	Description
ID	NUMBER(19)	Unique identifier
DEPENDENCY_TYPE	VARCHAR2(100)	Network Connection, Documented Dependency
PROTOCOL	VARCHAR2(1024)	Network connection protocol (HTTP, FTP, etc.)
PROTOCOL_VERSION	VARCHAR2(1024)	Protocol version
TRANSPORT_PROTOCOL	VARCHAR2(100)	Transport protocol (TCP, UDP)
URL	VARCHAR2(512)	Documented Dependency URL
DOC_DEP_TYPE	VARCHAR2(256)	Type of the documented dependency

Table 4-5. Dependency Table (Continued)

Name	Datatype	Description
SRC_IP	VARCHAR2(100)	Source IP Address of the dependency
SRC_PORT	NUMBER(19)	Source port of the dependency
SRC_SERVICE_NAME	VARCHAR2(1024)	Source service name
SRC_PRIMARY_HOSTNAME	VARCHAR2(1024)	Source primary host name (see network element)
SRC_PRIMARY_HOST_IP	VARCHAR2(100)	Source primary IP address (see network element)
SRC_HOSTNAMES	VARCHAR2(4000)	Source hostnames (see network element)
SRC_HOST_IPS	VARCHAR2(4000)	Source IP addresses (see network element)
SRC_NE_ID	NUMBER(19)	Source network element
SRC_SERVICE_ID	NUMBER(19)	Source service
SRC_MODULE_ID	NUMBER(19)	Source module (java_ee, database) usually of a documented dependency
TRG_IP	VARCHAR2(100)	Target IP Address of the dependency
TRG_PORT	NUMBER(19)	Target port of the dependency
TRG_SERVICE_NAME	VARCHAR2(1024)	Target service name
TRG_PRIMARY_HOSTNAME	VARCHAR2(1024)	Target primary host name (see network element)
TRG_PRIMARY_HOST_IP	VARCHAR2(100)	Target primary IP address (see network element)
TRG_HOSTNAMES	VARCHAR2(4000)	Target host names (see network element)
TRG_HOST_IPS	VARCHAR2(4000)	Target IP addresses (see network element)
TRG_NE_ID	NUMBER(19)	Target network element
TRG_SERVICE_ID	NUMBER(19)	Target service
TRG_MODULE_ID	NUMBER(19)	Target module (java_ee, database) usually of a documented dependency
GROUPS	VARCHAR2(4000)	
DISCOVERY_TIME	TIMESTAMP	
KNOWLEDGE_LEVEL	NUMBER(10)	

Where:

ID is a Primary key

SRC_NE_ID is a Foreign key (network_element (id))

SRC_SERVICE_ID is a Foreign key (service (id))

SRC_MODULE_ID is a Foreign key (java_ee, database (id))

Group_Membership

ADM Groups contain entities as members. This table marks an entity (network element, service, dependency) as a member of a group. Table 4-6 lists the contents of the Group_Membership table.

Table 4-6. Group_Membership Table

Name	Datatype	Description
ID	NUMBER(19)	Group identifier (unique to groups, not to entity memberships)
GROUP_TYPE	VARCHAR2(255)	Group type
GROUP_NAME	VARCHAR2(255)	Group name

Table 4-6. Group_Membership Table (Continued)

Name	Datatype	Description
GROUP_DESCRIPTION	VARCHAR2(1024)	Group description
GROUP_CREATION_TIME	TIMESTAMP	Creation time
GROUP_CREATOR	VARCHAR2(255)	group creator user name
GROUP_MODIFICATION_TIME	TIMESTAMP	Group last modification time
GROUP_MODIFIER	VARCHAR2(255)	Group last modifier user name
ENTITY_ID	NUMBER(19)	Member entity

Where:

ENTITY_ID is a Foreign key (network_element, service, dependency (id))

Installed_Software

This table contains the information of softwares installed on a network element. The software information might also exist in the service table (installed software can appear as both installed software and as a service on the network). Table 4-7 lists the contents of the Installed_Software table.

Table 4-7. Installed_Software Table

Name	Datatype	Description
ID	NUMBER(19)	Unique identifier
NE_ID	NUMBER(19)	Network Element
NAME	VARCHAR2(256)	Software name
VERSION	VARCHAR2(256)	
RELEASE	VARCHAR2(256)	
VENDOR	VARCHAR2(256)	
INSTALL_PATH	VARCHAR2(1024)	
SHARED_INSTALL_PATH	VARCHAR2(1024)	
INSTALL_DATE	TIMESTAMP	
TYPE	VARCHAR2(256)	Type of software (application, driver, patch)

Where:

ID is a Primary key

NE_ID is a Foreign key

IP_Route

This table contains IP (layer 3) routing information of a network element. Table 4-8 lists the contents of the IP_Route table.

Table 4-8. IP_Route Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
NE_ID	NUMBER(19)	Network Element
NIC_ID	NUMBER(19)	Network Interface
DEST_IP	VARCHAR2(100)	Destination IP

Table 4-8. IP_Route Table

Name	Datatype	Description
DEST_MASK	VARCHAR2(100)	Destination Mask
GATEWAY	VARCHAR2(100)	Gateway IP
METRIC	NUMBER(10)	
IS_LOCAL	VARCHAR2(1)	Y/N
GATEWAY_NE_ID	NUMBER(19)	Gateway Network Element

Where:

ID is a Primary key

NE_ID, NIC_ID, and GATEWAY_NE_ID are Foreign key

Java_EE

This table contains Java Enterprise Edition components: Application Server, JEE Application, and modules (jar, war, ear, etc). Table 4-9 lists the contents of the Java_EE table.

Table 4-9. Java_EE Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
NE_ID	NUMBER(19)	Network Element ID
SERVICE_ID	NUMBER(19)	
SERVER_NAME	VARCHAR2(256)	Application Server
SERVER_CONFIG_PATH	VARCHAR2(512)	Application Server Configuration path
APP_NAME	VARCHAR2(256)	JEE Application
APP_DESCRIPTION	VARCHAR2(1024)	Application description
APP_PATH	VARCHAR2(1024)	Application path
MODULE_NAME	VARCHAR2(256)	JEE module name
MODULE_TYPE	VARCHAR2(256)	JEE module type (jar, war, ear, etc)
MODULE_DEPLOYMENT_DESC	VARCHAR2(256)	Deployment descriptor
MODULE_CONTEXT_ROOT	VARCHAR2(256)	Context root (war only)

Where:

ID is a Primary key

NE_ID and SERVICE_ID are Foreign key

Network_Element

This table contains computer system information, that could be any combination of host, router, and switch. Table 4-10 lists the contents of the Network_Element table.

Table 4-10. Network_Element Table

Name	Datatype	Description
ID	NUMBER(19)	Unique identifier
PRIMARY_HOSTNAME	VARCHAR2(1024)	Primary (single) hostname
PRIMARY_HOST_IP	VARCHAR2(100)	Primary (single) IP address in dot notation string (192.168.2.5).

Table 4-10. Network_Element Table (Continued)

Name	Datatype	Description
HOSTNAMES	VARCHAR2(4000)	Comma-separated list of all hostnames
HOST_IPS	VARCHAR2(4000)	Comma-separated list of all IP addresses
GROUPS	VARCHAR2(4000)	Comma-separated list of all groups the element belongs to.
DISCOVERY_TIME	TIMESTAMP	Time the element was first introduced to the system
IS_HOST	VARCHAR2(1)	Flag identifying the network element as "host" (values: Y/N)
IS_ROUTER	VARCHAR2(1)	Flag identifying the network element as "router" (values: Y/N)
IS_SWITCH	VARCHAR2(1)	Flag identifying the network element as "switch" (values: Y/N)
IS_VIRT_CONTAINER	VARCHAR2(1)	Flag identifying the network element as container (for virtual machines) (values: Y/N)
IS_VIRTUAL	VARCHAR2(1)	Flag identifying the network element as virtual machine (values: Y/N)
HW_VENDOR	VARCHAR2(256)	Name of the hardware vendor
HW_MODEL	VARCHAR2(256)	Model of the hardware
HW_DESCRIPTION	VARCHAR2(1024)	Description of the hardware
OS_NAME	VARCHAR2(256)	Operating system name
OS_FAMILY	VARCHAR2(256)	Operating system family
OS_TYPE	VARCHAR2(256)	Operating system type
OS_VERSION	VARCHAR2(256)	Operating system version
OS_VENDOR	VARCHAR2(256)	Operating system vendor
OS_BUILD	VARCHAR2(256)	Operating system build (mostly MS)
OS_BUILD_TYPE	VARCHAR2(256)	Operating system build type (mostly MS)
OS_SERVICE_PACK	VARCHAR2(256)	Operating system service pack (mostly MS)
OS_LANGUAGE	VARCHAR2(256)	Operating system language
OS_KERNEL_VERSION	VARCHAR2(256)	Operating system kernel version
OS_KERNEL_RELEASE	VARCHAR2(256)	Operating system kernel release
OS_SWAP_MEMORY	NUMBER(19)	Swap memory size used by the OS (in Byte)
OS_PHYS_MEMORY	NUMBER(19)	Physical memory size as seen by the OS (in Byte)
OS_REGISTRATION_KEY	VARCHAR2(256)	Operating system registration key
OS_SERIAL_NUMBER	VARCHAR2(256)	Operating system serial number
CPU_MODEL	VARCHAR2(256)	Processor model
CPU_VENDOR	VARCHAR2(256)	Processor vendor
CPU_FAMILY	VARCHAR2(256)	Processor family
CPU_CLOCK_SPEED_HZ	NUMBER(19)	Processor maximum speed (Hz)
CPU_CACHE_SIZE	NUMBER(19)	Processor cache size (Byte)
CPU_STEPPING	VARCHAR2(256)	Processor stepping
CPU_DESCRIPTION	VARCHAR2(256)	Processor description
CPU_COUNT	NUMBER(10)	Processor count as seen by the OS (multi-core CPU is usually seen by the OS as several separate CPUs)

Where:

ID is a Primary key

IS_VIRT_CONTAINER and IS_VIRTUAL are Foreign key (New in 6.0 and later versions)

Network_Interface

This table identifies a network interface card or a switch port. Table 4-11 lists the contents of the Network_Interface table.

Table 4-11. Network_Interface Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
NE_ID	NUMBER(19)	Network Interface ID
IS_LOCAL	VARCHAR2(1)	Y/N
MAC_ADDRESS	VARCHAR2(256)	
MODEL	VARCHAR2(256)	
VENDOR	VARCHAR2(256)	
TECHNOLOGY	VARCHAR2(256)	
SPEED_BIT_PS	VARCHAR2(256)	
DESCRIPTION	VARCHAR2(256)	
OP_STATUS	VARCHAR2(256)	
IS_VIRTUAL	VARCHAR2(1)	
IS_TRUNK	VARCHAR2(1)	
IS_DEFAULT	VARCHAR2(1)	
PORT_LABEL	VARCHAR2(256)	Switch port label
MODULE_LABEL	VARCHAR2(256)	Switch module label
V_LAN	NUMBER(10)	

Where:

ID is a Primary key

NE_ID is a Foreign key

SPEED_BIT_PS was speed_mb in version 5.3

Port

This table represents a logical network port to which a Service binds to (e.g., HTTP Server binds to port 80). Table 4-12 lists the contents of the Port table.

Table 4-12. Port Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
NE_ID	NUMBER(19)	Network element
SERVICE_ID	NUMBER(19)	Service
IP	VARCHAR2(100)	IP Address
PORT	NUMBER(5)	Port number
TRANSPORT_PROTOCOL	VARCHAR2(100)	TCP/UDP

Where:

ID is a Primary key

NE_ID and SERVICE_ID are Foreign keys

Property

This table contains general purpose properties (name-value pairs), such as URLs on http-servers and tables on database-related dependencies. Table 4-13 lists the contents of the Property table.

Table 4-13. Property Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
ENTITY_ID	NUMBER(19)	Network element, service, dependency
NAME	VARCHAR2(256)	
VALUE	VARCHAR2(1028)	

Where:

ID is a Primary key

ENTITY_ID is a Foreign key

Service

This table contains information of a software service running on the network element (usually exposed through a network port). Table 4-14 lists the contents of the Service table.

Table 4-14. Service Table

Name	Datatype	Description
ID	NUMBER(19)	Unique identifier.
NAME	VARCHAR2(1024)	Service name if known (null if unknown).
NE_ID	NUMBER(19)	Foreign key to the network-element containing the service.
PRIMARY_HOSTNAME	VARCHAR2(1024)	Primary host name of the containing host (see network element).
PRIMARY_HOST_IP	VARCHAR2(100)	Primary IP address of the containing host (see network element).
HOSTNAMES	VARCHAR2(4000)	Host names list of the containing host (see network element).
HOST_IPS	VARCHAR2(4000)	IP list of the containing host (see network element).
PORTS	VARCHAR2(4000)	Ports which the service is consumed by
GROUPS	VARCHAR2(4000)	Comma-separated list of all groups the element belongs to.
DISCOVERY_TIME	TIMESTAMP	Time the element was first introduced to the system.
VENDOR	VARCHAR2(1024)	Vendor of the service.
VERSION	VARCHAR2(1024)	Version of the service.
CATEGORY	VARCHAR2(100)	Category of the service.
TIER	VARCHAR2(100)	Tier of the service.
INSTALL_PATH	VARCHAR2(512)	Installation path.

Table 4-14. Service Table (Continued)

Name	Datatype	Description
SHARED_INSTALL_PATH	VARCHAR2(512)	Shared installation path.
IS_UNIDENTIFIED	VARCHAR2(1)	Indicating this is a placeholder for an unidentified service (values: Y/N).
IS_CLIENT	VARCHAR2(1)	Indicating a client service (values: Y/N).
KNOWLEDGE_LEVEL	NUMBER(10)	Indicates the level of knowledge of the service. Value of zero indicates speculative information only (used to filter out speculative information)

Where:

ID is a Primary key

NE_ID is a Foreign key (network_element (id))

Storage_Device

This table represents network element storage device. It contains both logical (such as partitions) and physical drives. Table 4-15 lists the contents of the Storage_Device table

Table 4-15. Storage_Device Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
NE_ID	NUMBER(19)	Network element
IS_LOGICAL	VARCHAR2(1)	(Y/N)
LABEL	VARCHAR2(256)	
DESCRIPTION	VARCHAR2(256)	
DEVICE_TYPE	VARCHAR2(256)	
CAPACITY	NUMBER(19)	Size In bytes
FILE_SYSTEM	VARCHAR2(256)	
MOUNT_DESTINATION	VARCHAR2(256)	
MOUNT_POINT	VARCHAR2(256)	

Where:

ID is a Primary key

NE_ID is a Foreign key

Switch_Address_Forwarding

This table contains address forwarding information of a switch (layer 2). Table 4-16 lists the contents of the Switch_Address_Forwarding table.

Table 4-16. Switch_Address_Forwarding Table

Name	Datatype	Description
ID	NUMBER(19)	Unique Identifier
NE_ID	NUMBER(19)	Origin Network element
NIC_ID	NUMBER(19)	Origin network interface
MODULE	VARCHAR2(256)	Switch module
PORT	VARCHAR2(256)	Switch port

Table 4-16. Switch_Address_Forwarding Table

Name	Datatype	Description
MAC	VARCHAR2(100)	Forward MAC address
V_LAN	NUMBER(10)	Virtual LAN
TRG_NE_ID	NUMBER(19)	Forward network element

Where:

ID is a Primary key

NE_ID, NIC_ID and TRG_NE_ID are Foreign keys

Time Dimension

This table provides the time dimension. Table 4-17 lists the contents of the Time dimension table.

Table 4-17. Time Dimension Table

Name	Datatype	Description
TIME_ID	NUMBER(19)	Human readable number: yyyymmddhh
YEAR	NUMBER(10)	yyyy
QUARTER	NUMBER(10)	1-4
QUARTER_NAME	VARCHAR2(50)	Q1-2008
MONTH	NUMBER(10)	1-12
MONTH_NAME	VARCHAR2(50)	January-December
DAY_OF_MONTH	NUMBER(10)	1-31
DAY_OF_WEEK	NUMBER(10)	1-7
DAY_OF_WEEK_NAME	VARCHAR2(50)	Sunday-Saturday
WEEK_OF_YEAR	NUMBER(10)	1-52
HOUR	NUMBER(10)	0-23

Where:

TIME_ID is a Primary key

Usage Metric

This table is a multi-dimensional OLAP fact table for capturing the demand of Dependency, Service, and Network Element. The granularity of the data is of a single hour. Together with the time dimension and the related entities, it is sliced and diced as needed. Table 4-18 lists the contents of the Usage metric table.

Table 4-18. Usage Metric Table

Name	Datatype	Description
ENTITY_ID	NUMBER(19)	The entity of the usage metric. The data contains foreign keys to all entity types: network_element, service, and dependency.
TIME_ID	NUMBER(19)	The time dimension id: numeric value but human readable: yyyymmddhh.
BASELINE_TIME_ID	NUMBER(19)	Baseline time id contains the hour part of the time_id (baseline is for 24 hours).
VALUE	NUMBER(19)	The number of network transactions in the given time (hour granularity).

Where:

ENTITY_ID is a Foreign key (network_element service, dependency (id))

TIME_ID is a Foreign key (time (time_id))

BASELINE_TIME_ID is a Foreign key (usage_metric_baseline (baseline_time_id))

Usage_Metric_Baseline

This table describes the baseline usage, composed of the entity's usage metric accumulated within the first 24 hours since detected by the system. Table 4-19 lists the contents of the Usage_Metric_Baseline table.

Table 4-19. Usage_Metric_Baseline Table

Name	Datatype	Description
BASELINE_TIME_ID	NUMBER(19)	Hour portion of the time_id
ENTITY_ID	NUMBER(19)	Network element, service, dependency
VALUE	NUMBER(19)	Number of network transactions

Where:

BASELINE_TIME_ID and ENTITY_ID are Primary keys

ENTITY_ID is a Foreign key

Virtual_System_Settings

This table represents the virtual system settings on a specific virtual container machine.Table 4-20 lists the contents of the Virtual_System_Settings table (Introduced in version 6.0 of ADM).

Table 4-20. Virtual_System_Settings Table

Name	Datatype	Description
ID	NUMBER(19)	Unique identifier.
CONTAINER_ID	NUMBER(19)	Network element id on which the current virtual system settings reside.
UUID	VARCHAR2(256)	Unique id of the virtual host.
STATE	VARCHAR2(256)	State of the virtual host.
VM_NAME	VARCHAR2(256)	Name of the virtual host.
OS_FAMILY	VARCHAR2(256)	OS family of the virtual host.
OS_NAME	VARCHAR2(256)	OS name of the virtual host.
OS_TYPE	VARCHAR2(256)	OS type of the virtual host.
TOOLS_STATUS	VARCHAR2(256)	VM tools status.
TOOLS_VERSION	VARCHAR2(256)	VM tools version.
TOTAL_MEM_SIZE	NUMBER(19)	Total memory size of the virtual host.
IPS	VARCHAR2(1024)	List of IPs that reside on the virtual host.
MACS	VARCHAR2(4000)	List of MACs that reside on the virtual host.
VIRTUAL_NE_ID	NUMBER(19)	Network element id of the virtual host connected on the other side.

Where:

ID is a Primary key

CONTAINER_ID and VIRTUAL_NE_ID are Foreign keys (network_element)

ADM Custom Reports

5

This chapter provides information on

Creating Custom Reports

Most reports are generated from the four core tables demonstrated in Figure 4-1. The reports cover inventory, dependency, and usage. The rest of the data is joined for special reports without much complication. The following are examples of reports that are generated from the ADM external database.

Host Inventory

The following script generates a report of all hosts that were discovered between 4:54 P.M., October 29, 2006 and 5:04 P.M., October 13, 2007:

```
select ne.host_ips, ne.hostnames,  ne.groups, ne.discovery_time
from network_element ne
where discovery_time between
to_date('2006-29-10 16:54:00', 'yyyy-dd-mm hh24:mi:ss') and  to_date('2007-13-10 17:04:00',
                'yyyy-dd-mm hh24:mi:ss')
```

Connection Inventory

For the specified type of connection, source service, and IP addresses, the report displays the connection type, version, source service, source host, port, IP address, and discovery date:

```
select dep.protocol, dep.groups, dep.src_service_name, dep.src_hostnames, dep.src_port,
                dep.trg_service_name, dep.trg_hostnames, dep.trg_port, dep.discovery_time
from dependency dep
where discovery_time between
to_date('2006-29-10 16:54:00', 'yyyy-dd-mm hh24:mi:ss') and  to_date('2007-13-10 17:04:00',
                'yyyy-dd-mm hh24:mi:ss')
```

Most Used Services

This report list the 10 most used services used for a selected date and time:

```
select * from
(
select service.primary_host_ip, service.primary_hostname, service.name, service.version,
                service.vendor, service.category, max(value)as peak, sum(value) as transactions
from usage_metric metric inner join service service on service.id = metric.entity_id
where metric.time_id between 2006102916  and 2007101317 and service.name is not null
group by service.name, service.version, service.vendor, service.category,
                service.primary_hostname, primary_host_ip
)
where rownum <= 10
```

Application Dependency

For the specified Business Application, this report shows each host and service the application depends on and each host and service that depends on it (if any):

```
select distinct 0 as degree, grp.group_name, service.name, service.version, service.vendor,
            service.category, service.primary_host_ip, service.primary_hostname,
            service.discovery_time
from dependency dep inner join group_membership grp on grp.entity_id = dep.id
inner join service service on service.id =dep.src_service_id
where service.name is not null and grp.group_name = '111'
union
select distinct 1 as degree, grp.group_name, service.name, service.version,  service.vendor,
            service.category,
service.primary_host_ip, service.primary_hostname, service.discovery_time
from dependency dep inner join group_membership grp on grp.entity_id = dep.id
inner join service service on service.id =dep.trg_service_id
where service.name is not null and grp.group_name = '111'
```