Oracle® Data Mining

Administrator's Guide

11*g* Release 1 (11.1)

B28130-04

August 2008

ORACLE®

Oracle Data Mining Administrator's Guide, 11*g* Release 1 (11.1)

B28130-04

Contents

4 Users and Privileges for Data Mining

5 Installing Oracle Data Miner

6 Installing the Spreadsheet Add-In for Predictive Analytics

7 Mining Model Schema Objects

8 The Data Mining Sample Programs

Index

Preface

This manual explains how to install the various components of Oracle Data Mining and perform basic administration tasks. It also explains how to install and run the Data Mining sample programs.

The preface contains these topics:

- Audience
- Documentation Accessibility
- Related Documentation
- Where to Find Database Tuning Documentation
- Conventions

Audience

This guide can be used by a spectrum of users, however, it is primarily directed at an individual user who wants to install, configure, and use Oracle Data Mining on a laptop or personal computer.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documentation

The documentation set for Oracle Data Mining is part of the Oracle Database 11*g* Release 1 (11.1) Online Documentation Library. The Oracle Data Mining documentation set consists of the following documents:

- *Oracle Data Mining Concepts*
- *Oracle Data Mining Application Developer's Guide*
- *Oracle Data Mining Java API Reference* (javadoc)

For detailed information about the Oracle Data Mining PL/SQL interface, see *Oracle Database PL/SQL Packages and Types Reference*. Search for DBMS_DATA_MINING.

For detailed information about the SQL data mining functions, see *Oracle Database SQL Language Reference*.

For an introduction to application development in SQL and PL/SQL, see *Oracle Database Advanced Application Developer's Guide.*

For an introduction to application development in Java, see *Oracle Database Java Developer's Guide*

Where to Find Database Tuning Documentation

The Oracle Data Mining documentation set does not address the topic of database tuning for data mining. The *Oracle Database Performance Tuning Guide* is the best source of information for tuning Oracle Database (including when the goal is to invoke mining build/score operations).

The memory tuning parameters PGA_AGGREGATE_TARGET and SGA_TARGET can easily have the most impact on data mining. The correct setting of PGA is very important for model building routines, as well as complex queries and batch scoring. From a Data Mining perspective, the SGA is generally less of a concern, except that real-time scoring is enabled by having the model loaded into the shared cursor in the SGA, so this still should be sized accordingly.

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for getting started
>
> *Oracle Database Performance Tuning Guide* for more detail on memory configuration

In addition, if parallel execution is desired, then there are INIT.ORA parameters that control this functionality.

> **See Also:** Chapter 25 in *Oracle Database Data Warehousing Guide* for information about parallel execution tuning

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

x

What's New in Oracle Data Mining Administration

This section summarizes the new features of Oracle Data Mining that pertain to installation, administration, and upgrade.

> **See Also:**
>
> - "What's New" in *Oracle Data Mining Concepts* for a complete list of new and deprecated features in Oracle Data Mining 11*g* Release 1 (11.1).
>
> - Chapter 3 for information on upgrading and downgrading a database

This section contains the following topics:

- No DMSYS Schema

- Mining Models in the Oracle Data Dictionary

- Enhanced Security

- Scoping of Nested Data

- Enhanced Handling of Sparse Data

- Deprecated Features

No DMSYS Schema

Oracle Data Mining 11*g* Release 1 (11.1) has a tight integration with Oracle Database. Data Mining metadata and PL/SQL packages have been migrated from `DMSYS` to `SYS`. The `DMSYS` schema no longer exists in Oracle Database 11*g* Release 1 (11.1) fresh installations.

Mining Models in the Oracle Data Dictionary

New catalog views for Data Mining are introduced in 11*g* Release 1 (11.1):

- `ALL/DBA/USER_MINING_MODELS` provides information about mining models

- `ALL/DBA/USER_MINING_MODEL_ATTRIBUTES` provides information about data columns used to build a mining model

- `ALL/DBA/USER_MINING_MODEL_SETTINGS` provides information about configuration settings for mining models

The `ALL/DBA/USER_OBJECTS` catalog view now identifies mining models.

> **See Also:** "Obtaining Information from the Data Dictionary" in Chapter 7.

Enhanced Security

Security features of Oracle Data Mining are significantly enhanced in 11*g* Release 1 (11.1). Improved security for data mining has several aspects:

- Previously, Oracle Data Mining used `DMSYS` as its own repository schema. This necessitated the granting of advanced database privileges to `DMSYS`, a non-system account. In 11*g* Release 1 (11.1), the Oracle Data Mining metadata repository is in `SYS`, where it is accessible only to the system DBA.

- New system and object privileges for mining model objects are introduced in 11*g* Release 1 (11.1).

- The SQL auditing system can be used to track operations on mining model objects.

> **Note:** The privilege `CREATE MINING MODEL` is required for creating models in 11*g*. This privilege should be added to any accounts being upgraded to 11*g*.

> **See Also:** Chapter 4, "Users and Privileges for Data Mining".

Scoping of Nested Data

Oracle Data Mining supports nested data types for both categorical and numerical data. Multi-record case data must be transformed to nested columns for mining.

In Oracle Data Mining 10gR2, nested columns were processed as top-level attributes; the user had to ensure that two nested columns did not contain an attribute with the same name. In Oracle Data Mining 11g, nested attributes are scoped with the column name, which relieves the user of this burden.

> **See Also:** *Oracle Data Mining Application Developer's Guide*

Enhanced Handling of Sparse Data

Handling of sparse data and missing values has been standardized across algorithms in Oracle Data Mining 11g. Data is sparse when a high percentage of the cells are empty, but all the values are assumed to be known. Only nested data can be considered sparse. Missing values in simple numeric or character columns are considered missing at random.

> **See Also:** *Oracle Data Mining Application Developer's Guide*

Features Not Available in This Release

The following features are not supported in 11*g* Release 1 (11.1):

- `DMSYS` schema
- Oracle Data Mining Scoring Engine

- In Oracle 10.2, you could use Database Configuration Assistant (DBCA) to configure the Data Mining option. In Oracle 11*g*, you do not need to use DBCA to configure the Data Mining option.

- Basic Local Alignment Search Tool (BLAST)

Deprecated Features

The following features are deprecated in 11*g* Release 1 (11.1):

- Adaptive Bayes Network classification algorithm

- `DM_USER_MODELS` view is replaced by data dictionary views

- Several PL/SQL procedures have been deprecated.

  - `GET_DEFAULT_SETTINGS`

    Replaced with data dictionary views: `USER/ALL/DBA_MINING_MODEL_SETTINGS`

  - `GET_MODEL_SETTINGS`

    Replaced with data dictionary views: `USER/ALL/DBA_MINING_MODEL_SETTINGS`

  - `GET_MODEL_SIGNATURE`

    Replaced with data dictionary views: `USER/ALL/DBA_MINING_MODEL_ATTRIBUTES`

    See Also: *Oracle Database PL/SQL Packages and Types Reference*

    Note: Oracle recommends that you do not use deprecated features in new applications. Support for deprecated features is for backward compatibility only.

1

Installing the Software

This chapter is intended for anyone who wishes to install and use Oracle Data Mining on a personal computer for educational or demonstration purposes. It provides the basic information you will need to install the Data Mining software on Microsoft Windows and run the sample programs locally on your PC or laptop. To run the programs remotely, see the instructions in Chapter 2.

This chapter contains the following sections. Complete the instructions in each section before proceeding to the next section.

- Install Oracle Database
- Install Oracle Database Examples
- Create a Data Mining Demo User
- Run the Sample Programs

Install Oracle Database

Oracle Data Mining is part of Oracle Database Enterprise Edition. To perform data mining activities, you must be able to log on to an Oracle database, and your user ID must have the appropriate database privileges.

The instructions in this section explain how to perform a basic installation of Oracle Database with the sample schemas on your personal computer. The sample schemas are needed for the Data Mining sample programs.

> **Note:** These instructions are not intended as a replacement for Oacle Database installation documentation. If you have questions, if you encounter problems during the installation, or if you already have Oracle components installed on your PC, refer to:
>
> - *Oracle Database Installation Guide for Microsoft Windows*
> - *Oracle Database 2 Day DBA*
>
> Additional documentation is available on the *Installing and Upgrading* page of the Oracle Database 11*g* Online Documentation Library:
>
> http://www.oracle.com/pls/db111/db111.homepage

1. Before you begin the installation, ensure that your computer meets the system requirements described in *Oracle Database Installation Guide for Microsoft Windows*.

2. Stop any Oracle services that may be running on your computer.

In Windows Control Panel, choose **Administrative Tools**, then **Services**. Find the service names that start with "Oracle". Choose **Stop** for each one.

3. To start the installation, run SETUP.EXE from the Database installation directory.

Oracle Universal Installer opens and displays the Select a Product to Install dialog. Choose **Oracle Database 11***g*.

Choose **Next**.

4. The Installer displays the Select Installation Method page.
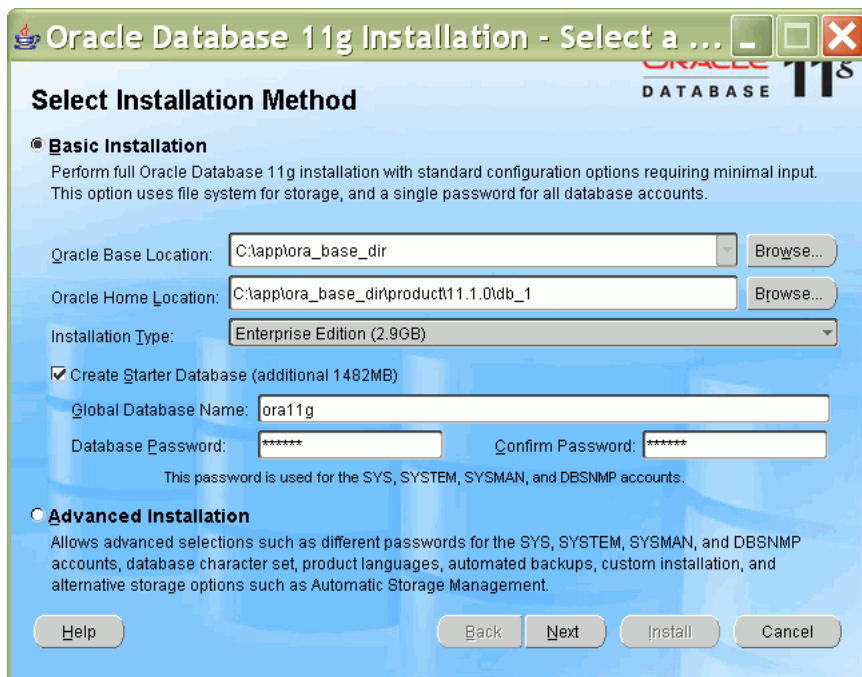
■ Choose **Basic Installation**.

- Specify the Oracle base and home directories. Oracle home is a subdirectory of the Oracle base directory. You may install Oracle Database in an existing Oracle base, but you must specify a new Oracle home.

  Oracle Installer creates the directories for Oracle base and Oracle home if they do not already exist on your computer.

- Choose **Enterprise Edition** (or **Personal Edition**) as the Installation Type.

- Check the **Create Starter Database** box.

- Specify a unique name for **Global Database Name**. You can use the default global database name provided by the Installer, as long as it does not already exist on your computer.

- Specify a password for the database system accounts. The password must have at least eight characters and include both alphabetic and numeric characters. For details about specifying passwords, refer to *Oracle Database Security Guide*.

  You will have the opportunity to change the passwords for the database system accounts at a later time.

- Click **Next**.

5. The Installer performs prerequisites checks.

   If the checks succeed, choose **Next** to advance to the next step.

   If any of the checks do not succeed, correct the problem and then click **Retry**. If more extensive changes are needed, cancel the installation, fix the problem, and then restart Oracle Installer

6. On the Oracle Configuration Manager Registration page, you can choose to register your installation with your Metalink account.

   This page is optional. You can simply choose **Next**.

7. The Summary page displays the settings and components for the installation.
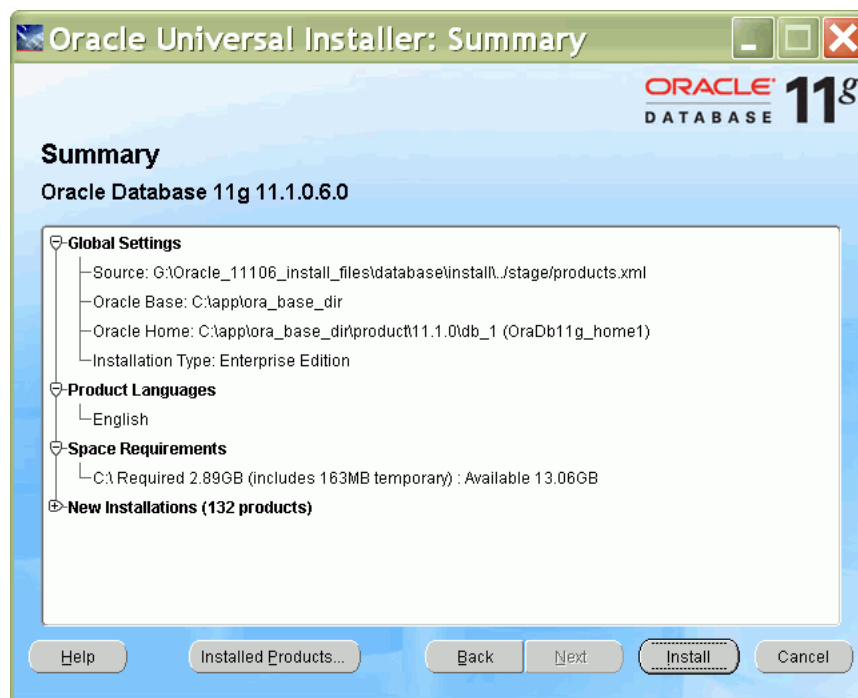
Click **Install**.

8. The Installer proceeds with the installation.
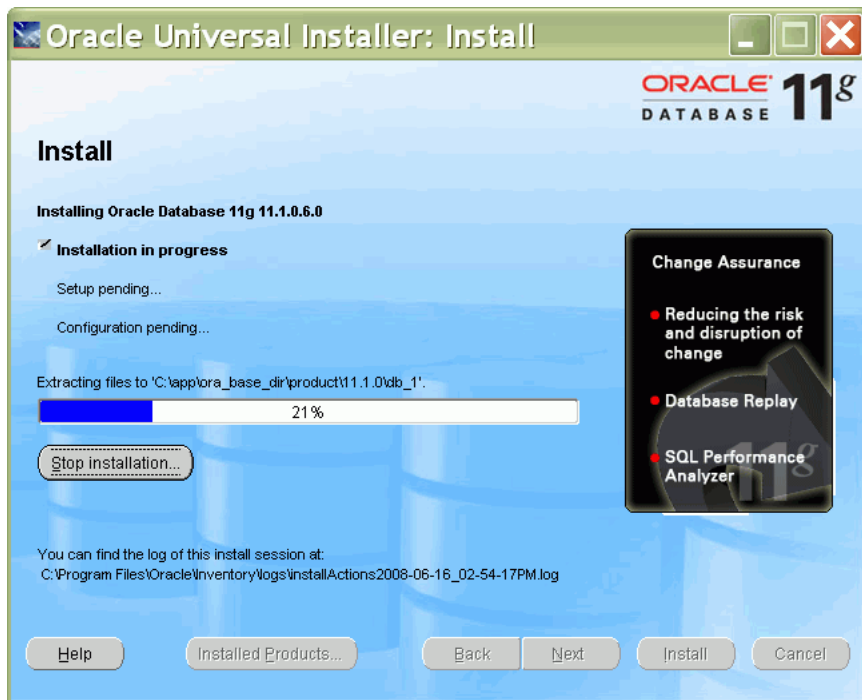
9. The Installer invokes the Configuration Assistants to configure and start the starter database.
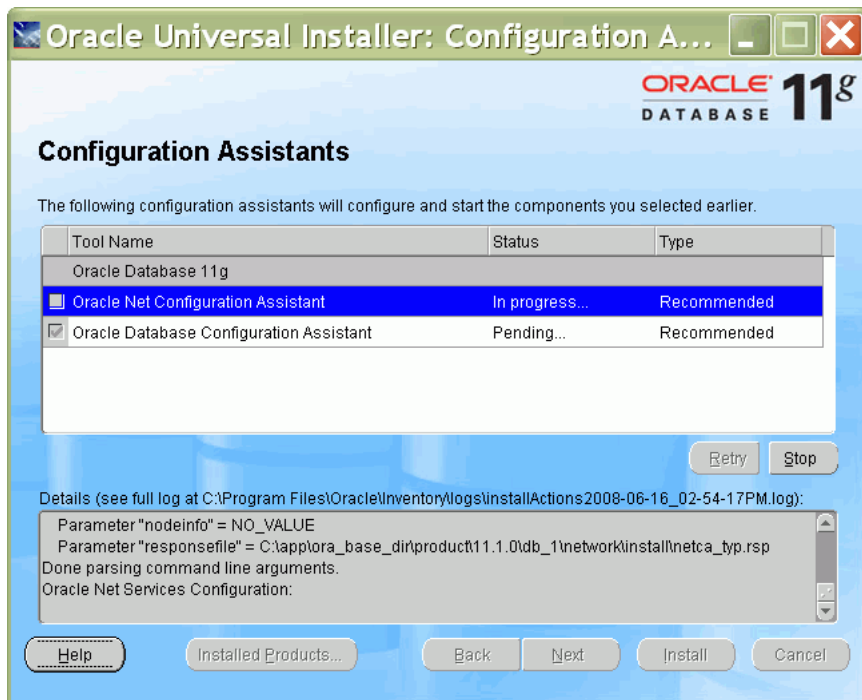
If the Configuration Assistants encounter an error, check the logs to determine the problem. You can choose to continue the installation and start the assistants manually later, or you can restart the installation. To continue the installation, click **Install**.

10. Database Configuration Assistant creates the starter database.
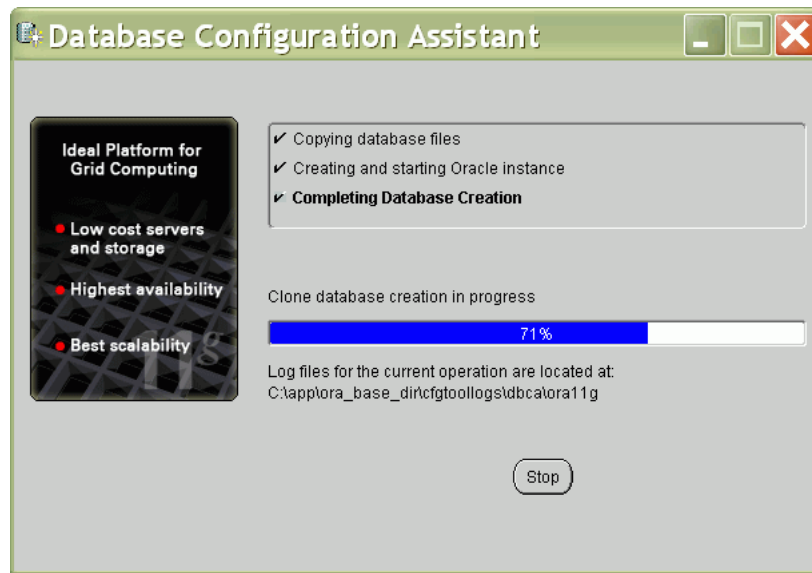
11. Database Configuration Assistant displays information about the starter database.
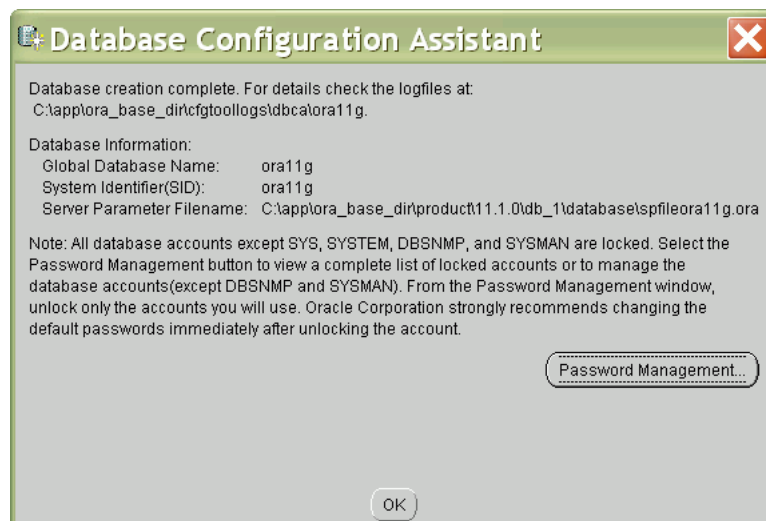
Click the **Password Management** button.

12. Unlock the SH account and specify a password. The SH schema is used by the sample programs.

You can change the passwords for SYS and SYSTEM if you wish. The password must have at least eight characters and include both alphabetic and numeric characters

For details about specifying passwords, refer to *Oracle Database Security Guide*.
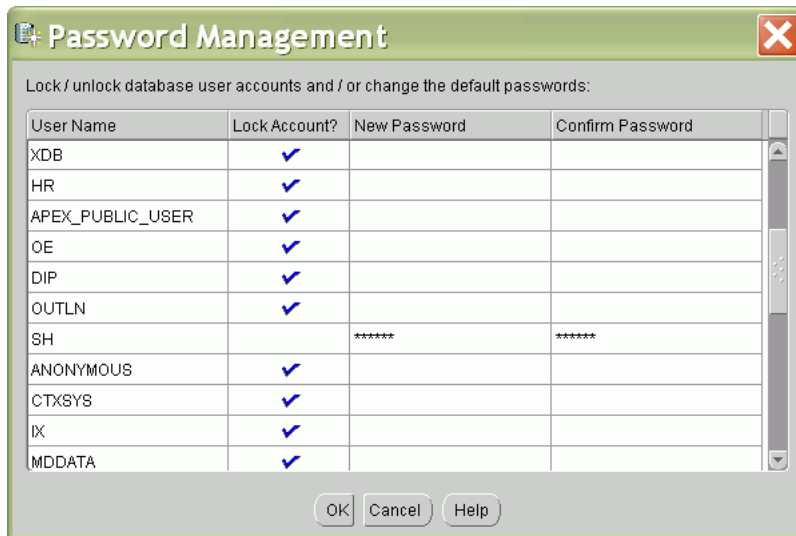
Click **OK** to return to the Database Configuration Assistant page.

On the Database Configuration Assistant page, click **OK**.

13. On the End of Installation page, confirm that the installation was successful.

Click **Exit** to exit the Installer.

Install Oracle Database Examples

The Oracle Data Mining sample programs are installed with Oracle Database Examples.

The Database Examples installation process copies the Oracle Data Mining sample programs, along with examples and demonstrations of other database features, to the `\RDBMS\demo` subdirectory of Oracle home.

To install Database Examples, perform these steps:

1. Ensure that your computer meets the system requirements described in *Oracle Database Examples Installation Guide*.

2. Stop any Oracle services that may be running on your computer.

    In Windows Control Panel, choose **Administrative Tools**, then **Services**. Find the service names that start with "Oracle". Choose **Stop** for each one.

3. To start the installation, go to the Examples installation directory and run SETUP.EXE.

    Oracle Universal Installer opens and displays the Welcome page. Click **Next** to advance to the next page.

4. On the Specify Home Details page, specify the Oracle home directory in which you installed Oracle Database. Do not assume that the directory displayed by the Installer is correct.
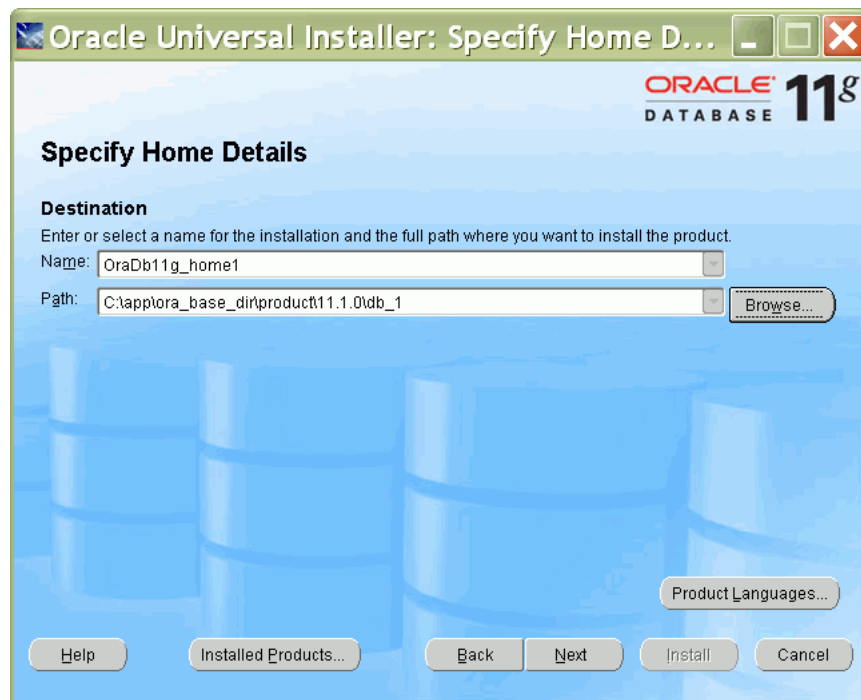
5. The Summary page displays the settings and components for the installation.
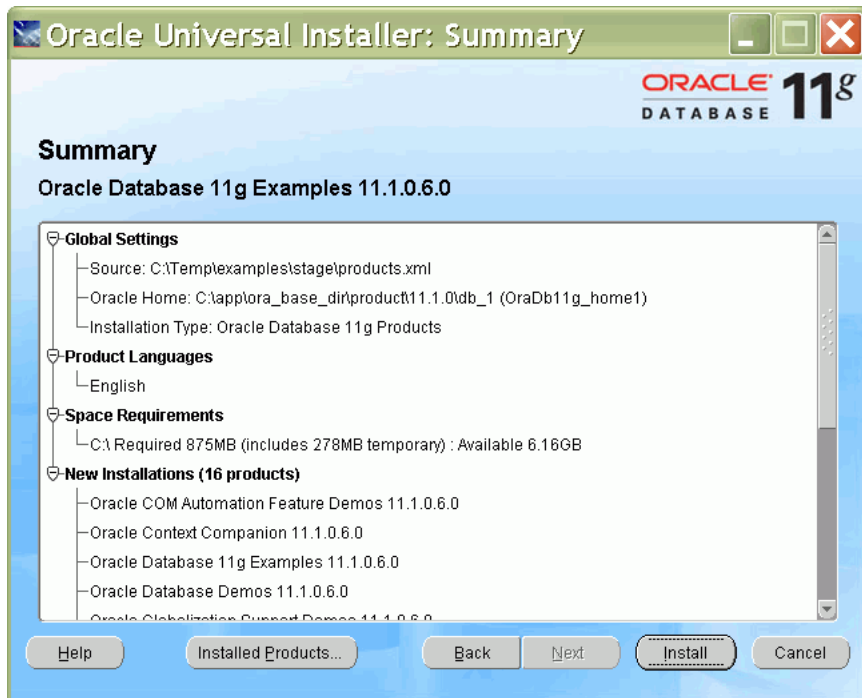
Click **Install**.

6. The Installer proceeds with the installation.
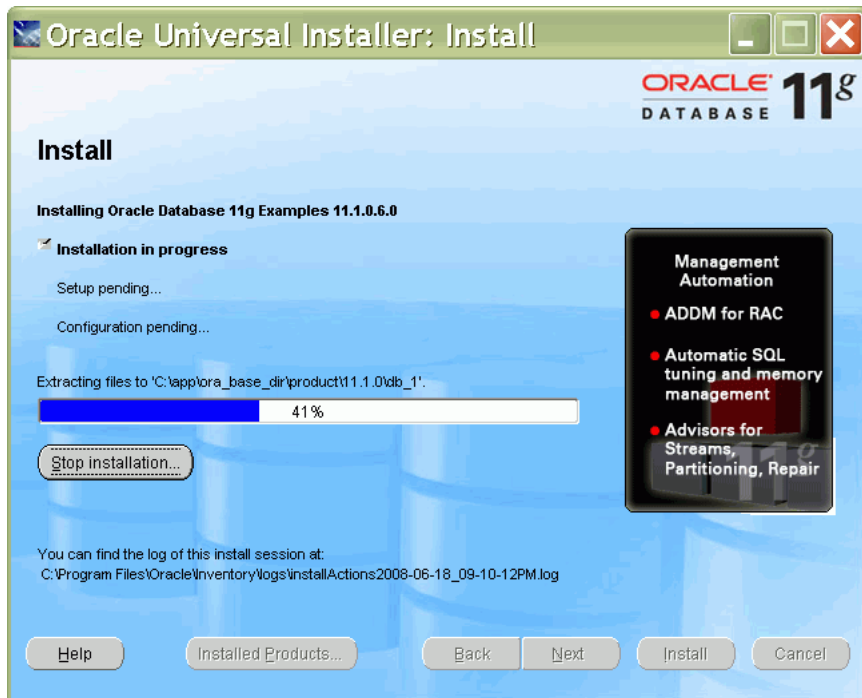
**7.** On the End of Installation page, confirm that the installation was successful.
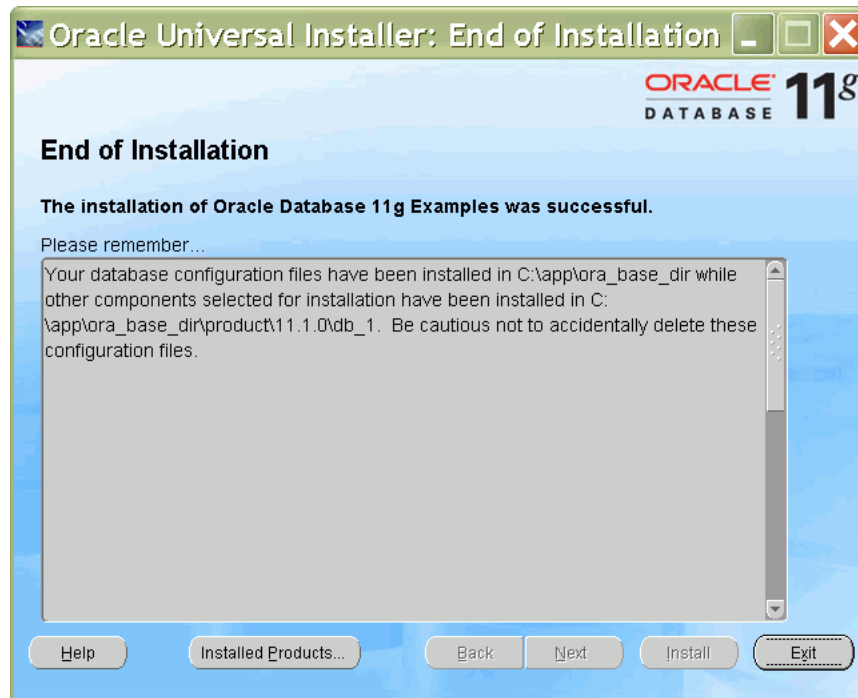
Click **Exit** to exit the Installer.

Create a Data Mining Demo User

To build and score Data Mining models, you must have an Oracle user ID with the appropriate privileges. Follow these instructions to create a demo user that has the privileges you need to run the sample programs and create and score models within your schema.

> **See Also:** Chapter 4, "Users and Privileges for Data Mining" to create data mining users that are capable of performing broader data mining tasks

> **Note on ORACLE_HOME:** In the following sections, you will find references to `ORACLE_HOME`, the environment variable that represents the Oracle home directory.
>
> The `ORACLE_HOME` environment variable is not required on Windows systems. If you wish to create it, you can do so by editing the system properties for your computer. In Control Panel, open **System** and choose the **Advanced** tab.
>
> If `ORACLE_HOME` is not defined, you must specify the full path when executing scripts that reside under Oracle home.

To create a demo user for Oracle Data Mining:

1. From the Windows Start menu, select the Oracle home directory of the local database.

2. Choose **Application Development**.

3. Choose **SQL*Plus**.

4. Log in with system privileges.

    ```
    Enter user-name: sys / as sysdba
    Enter password: password
    ```

5. To create the user, type a command like the following.

    ```
    CREATE USER dmuser IDENTIFIED BY password
           DEFAULT TABLESPACE USERS
           TEMPORARY TABLESPACE TEMP
           QUOTA UNLIMITED ON USERS;
    ```

6. Run `dmshgrants.sql` to grant access to the `SH` schema. Several tables in `SH` are used by the Data Mining sample programs. Specify the Data Mining user name as the parameter.

    ```
    @ %ORACLE_HOME%\RDBMS\demo\dmshgrants dmuser
    ```

    Note: *If you have not upgraded to 11.1.0.7*, run `dmshgrants` by specifying the `SH` password in addition to the Data Mining user name.

    ```
    @ %ORACLE_HOME%\RDBMS\demo\dmshgrants SH_password dmuser
    ```

7. Now connect to the database as the Data Mining user.

    ```
    CONNECT dmuser
    Enter password: password
    ```

8. Run `dmsh.sql` to populate the schema of the Data Mining user with tables, views, and other objects needed by the sample programs.

    ```
    @ %ORACLE_HOME%\RDBMS\demo\dmsh
    COMMIT;
    ```

Once you have completed these steps, you can run the Data Mining sample programs whenever you log in to the database as the Data Mining demo user.

Run the Sample Programs

To locate the sample programs on your computer, navigate to the `RDBMS\demo` subdirectory under Oracle home.

To display the Data Mining PL/SQL sample programs, search for the files that start with `dm` and end with `.sql`. (The list will include `dmsh.sql` and `dmshgrants.sql`, which are used to configure the Data Mining demo user ID.) The PL/SQL sample programs are listed in Table 1–1.

Table 1–1   Sample PL/SQL Data Mining Programs

Program File	Algorithm	Mining Function or Task
`dmaidemo.sql`	Minimum Descriptor Length	Attribute Importance
`dmardemo.sql`	Apriori	Association
`dmdtdemo.sql`	Decision Tree	Classification
`dmdtxvlddemo.sql`	Decision Tree (cross validation)	Classification
`dmglcdem.sql`	Binary Logistic Regression (GLM)	Classification
`dmglrdem.sql`	Multivariate Linear Regression (GLM)	Regression
`dmkmdemo.sql`	*k*-Means	Clustering

Table 1–1 (Cont.) Sample PL/SQL Data Mining Programs

Program File	Algorithm	Mining Function or Task
`dmnbdemo.sql`	Naive Bayes	Classification
`dmnmdemo.sql`	Non-Negative Matrix Factorization	Feature Extraction
`dmocdemo.sql`	O-Cluster	Clustering
`dmsvcdem.sql`	Support Vector Machine	Classification
`dmsvodem.sql`	Support Vector Machine	Anomaly Detection
`dmsvrdem.sql`	Support Vector Machine	Regression
`dmtxtfe.sql`	Term extraction using Oracle Text	Text transformation for mining
`dmtxtnmf.sql`	Non-Negative Matrix Factorization	Text mining using NMF
`dmtxtsvm.sql`	Support Vector Machine	Text mining using SVM

In the same directory, search for the files that start with `dm` and end with `.java` to display the Java samples. The Java sample programs are listed in Table 1–2.

Table 1–2 Sample Java Data Mining Programs

Program File	Algorithm	Mining Function or Task
`dmaidemo.java`	Minimum Description Length	Attribute importance
`dmapplydemo.java`	Naive Bayes	Illustrate scoring methods
`dmardemo.java`	Apriori	Association
`dmexpimpdemo.java`	export/import	Model Export/Import
`dmglcdemo.java`	Binary Logistic Regression (GLM)	Classification
`dmglrdemo.java`	Multivariate Linear Regression (GLM)	Regression
`dmkmdemo.java`	*k*-Means	Clustering
`dmnbdemo.java`	Naive Bayes	Classification
`dmnmdemo.java`	Non-Negative Matrix Factorization	Feature extraction
`dmocdemo.java`	O-Cluster	Clustering
`dmpademo.java`	Automated predict and explain	Predictive Analytics
`dmsvcdemo.java`	Support Vector Machine	Classification
`dmsvodemo.java`	Support Vector Machine (one class)	Classification
`dmsvrdemo.java`	Support Vector Machine	Regression
`dmtreedemo.java`	Decision Tree	Classification
`dmtxtnmfdemo.java`	Non-Negative Matrix Factorization	Text mining with NMF
`dmtxtsvmdemo.java`	Support Vector Machine	Text mining with SVM classification
`dmxfdemo.java`	Binning, clipping, and normalization	Data Transformations

View the Source Code

You will learn a great deal about the Data Mining APIs by investigating the source code of the sample programs. The programs illustrate typical approaches to data

preparation, algorithm selection, algorithm tuning, testing, and scoring. All the programs include extensive comments to help you understand what the code is doing.

You can view the source code simply by opening the files in a text editor.

Run the PL/SQL Sample Programs

Now that you have a user ID with the required privileges and a schema populated with the required objects, you can run the sample programs. Each program creates a Data Mining model.

While the program is running, it displays the program code and the program output.

You can run the sample programs as many times as you wish. The programs clean up the results of the previous run before executing the current run.

To run the PL/SQL programs:

1. Start SQL*Plus and log in as the Data Mining user.

   ```
   Enter user-name: dmuser
   Enter password: password
   ```

2. Run the program by specifying an at sign (@) followed by the fully-qualified path of the program. This example executes the program `dmnbdemo.sql`, which creates a Naive Bayes model.

   ```
   SQL>@ %ORACLE_HOME%\RDBMS\demo\dmnbdemo
   ```

Prepare to Run the Java Programs

Before you can run the Java programs, you must set up your Java environment and compile the programs. You can do this in an Integrated Development Environment such as Oracle JDeveloper, or you can execute the following commands at the operating system prompt.

1. Check that the version of Java you are using is 1.5 or higher. You can execute the following in a command window to check the version of Java.

   ```
   >java -version
   ```

2. Add `%ORACLE_HOME%\jdk\bin\` to your `PATH` variable before the paths of any other Java versions.

3. Add the following Data Mining JAR files to your Windows `CLASSPATH`:

   ```
   %ORACLE_HOME%\RDBMS\jlib\jdm.jar
   %ORACLE_HOME%\RDBMS\jlib\ojdm_api.jar
   %ORACLE_HOME%\RDBMS\jlib\xdb.jar
   %ORACLE_HOME%\jdbc\lib\ojdbc5.jar
   %ORACLE_HOME%\oc4j\j2ee\home\lib\connector.jar
   %ORACLE_HOME%\jlib\orai18n.jar
   %ORACLE_HOME%\jlib\orai18n-mapping.jar
   %ORACLE_HOME%\lib\xmlparserv2.jar
   ```

4. Compile the programs listed in Table 1–2. To use the `JAVAC` executable, open a command window and go to `\RDBMS\demo` in Oracle home.

   ```
   >javac program_name.java
   ```

   For example:

   ```
   >javac dmnbdemo.java
   ```

If `JAVAC` is not found, then check the value of the `PATH` variable.

Run the Java Programs

You can run a Java program from the operating system prompt with a command like this:

```
>java program_name host_name:port_number:database_identifier user password
```

View the Models Created by the Sample Programs

In SQL*Plus, you can query the `USER_MINING_MODELS` view to list the models in your schema. This example shows that there are two mining models in your schema. The model name, mining function, and algorithm are displayed

```
SQL> set linesize 100
SQL> SELECT model_name, mining_function, algorithm from user_mining_models;


MODEL_NAME                 MINING_FUNCTION            ALGORITHM
------------------------   -------------------------   ------------------------------
AI_SH_SAMPLE               ATTRIBUTE_IMPORTANCE        MINIMUM_DESCRIPTION_LENGTH
AR_SH_SAMPLE               ASSOCIATION_RULES           APRIORI_ASSOCIATION_RULES
```

To find all the columns defined in a view, use a `DESCRIBE` command.

```
SQL> DESCRIBE user_mining_models
```

You can query the `USER_MINING_MODEL_ATTRIBUTES` and `USER_MINING_MODEL_SETTINGS` views to obtain information about the attributes and settings for the models in your schema.

2

Connecting to a Data Mining Database

This chapter explains how to connect to an Oracle database to perform data mining and other database activities.

> **See Also:** *Oracle Database 2 Day DBA* for additional information

This chapter contains the following sections:

- Your User ID
- Connect to a Local Database
- Connect to a Remote Database
- Connect From Oracle Data Miner
- Connect From Oracle Spreadsheet Add-In For Predictive Analytics

Your User ID

To connect to Oracle Database, you must log on with a user ID and password. To perform data mining activities in the database, you must log on with a user ID that has been granted the necessary database privileges.

If you simply wish to run the Data Mining sample programs, see "Create a Data Mining Demo User" on page 1-9.

If you wish to perform broader data mining activities, refer to Chapter 4, "Users and Privileges for Data Mining".

Connect to a Local Database

In Chapter 1, you learned how to install Oracle Database on your personal computer. When you install Oracle Database locally, you do not need to install separate client software to connect to the database.

> **Note:** There is one exception. If you plan to use Oracle Spreadsheet Add-In for Predictive Analytics, you *do* need to install Oracle Client — even if the database is local. For more information, see:
>
> - Chapter 6, "Installing the Spreadsheet Add-In for Predictive Analytics".
> - "Install Oracle Client" on page 4
> - "Create a Net Service Name" on page 8
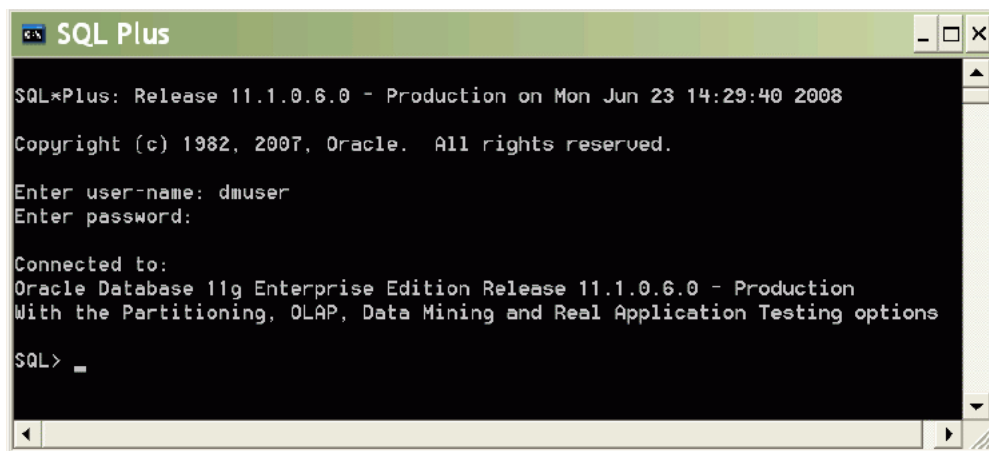
Connect to a Local Database Using SQL*Plus

SQL*Plus and other client tools are installed with Oracle Database. You can use SQL*Plus to connect to your local database by simply specifying a user name and password.

In SQL*Plus, you can run the Data Mining PL/SQL sample programs and perform data mining tasks using the PL/SQL API.

To start SQL*Plus:

1. From the Windows Start menu, select the Oracle home directory of the local database.

2. Choose **Application Development**.

3. Choose **SQL*Plus**.

4. Specify the user ID and password of the Data Mining user.

Connect to a Local Database Using SQL Developer

SQL Developer is a graphical development environment. You can use SQL Developer to run the Data Mining PL/SQL sample programs and perform data mining tasks using the PL/SQL API.

To start SQL Developer:

1. From the Windows Start menu, select the Oracle home directory of the local database.

2. Choose **Application Development**.

3. Choose **SQL Developer**.
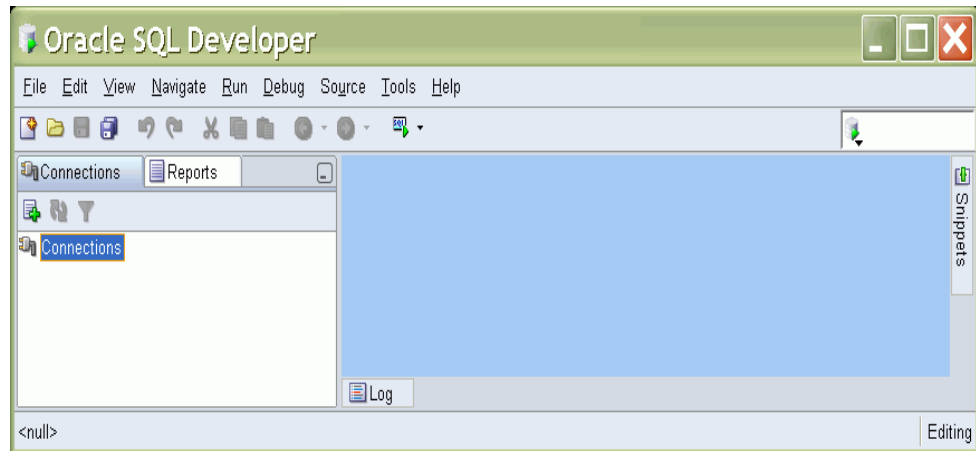
4. To define a connection to the local database, right click **Connections** and choose **New**.

5. The New/Select Database Connection dialog appears.
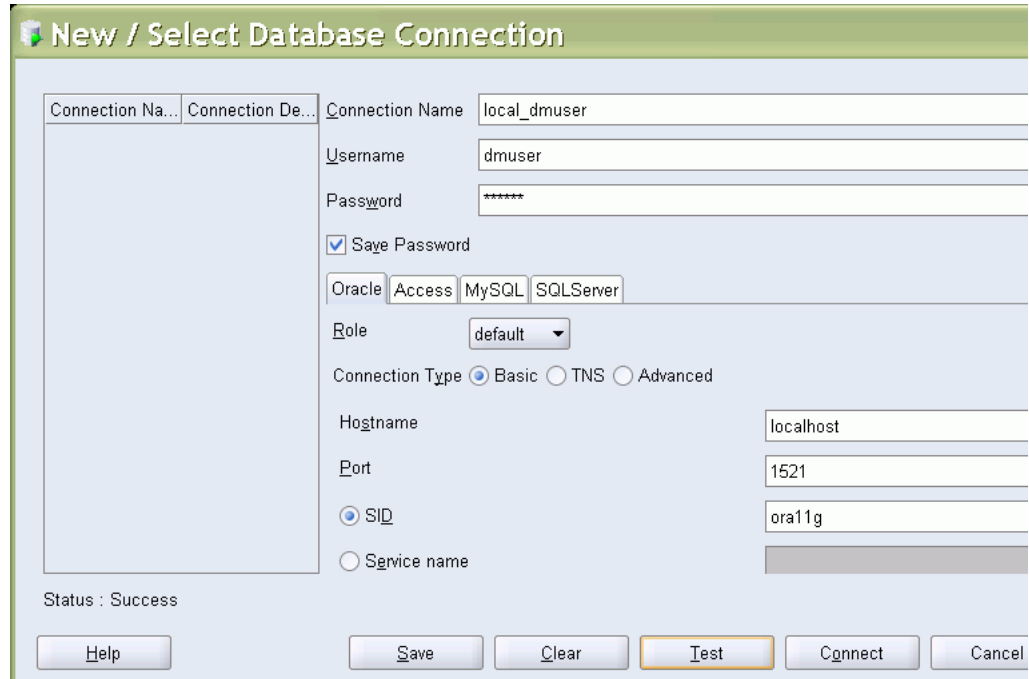
- For **Connection Name**, specify a meaningful name that you will remember.

- Provide a user name and password for the connection.

- For **Hostname**, specify `localhost` to indicate that the database is hosted on the local computer.

- For **Port**, specify the port that the database will use to listen to requests from the client. The default port is 1521.

- For **SID** (Service Identifier), specify the global database name that you created when you installed Oracle Database.

Click **Connect** to connect to the database using the specified credentials.

The connection that you created appears in the list of connections in the right-hand pane.In SQL Developer, you can have multiple connections open simultaneously.
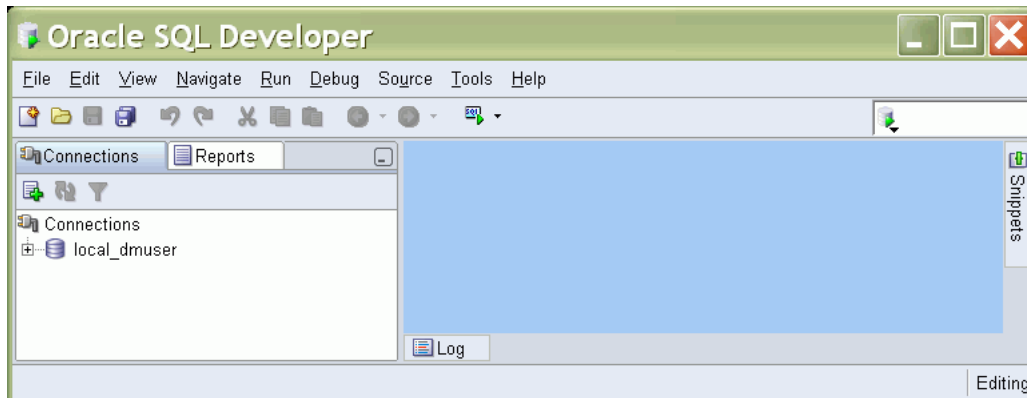
Connect to a Remote Database

If you do not have a local database, you must install Oracle Client to obtain SQL*Plus, SQL Developer, and other tools for connecting to a remote Oracle database.

To connect to a remote database, Oracle Client requires a host name, port number, service identifier or global database name, and a user name and password. If Oracle Client can use the Easy Connect method, these credentials can be presented when the connection request is made. Alternatively, Oracle Client can use a Net Service Name to connect to a remote database. A Net Service Name represents the login credentials for a given remote connection.

In the following sections, you will find instructions for installing Oracle Client and connecting to a remote database for data mining.

- Install Oracle Client

- Create a Net Service Name

- Perform Data Mining on a Remote Computer

- Run the Data Mining Sample Programs on a Remote Computer

Note: These instructions are not intended as a replacement for Oacle Client and Net Management documentation. For additional information, refer to:

- *Oracle Database 2 Day DBA*

- *Oracle Database Client Installation Guide for Microsoft Windows*

- *Oracle Database Net Services Administrator's Guide*

Additional documentation is available on the *Installing and Upgrading* page of the Oracle Database 11*g* Online Documentation Library:

http://www.oracle.com/pls/db111/db111.homepage

Install Oracle Client

Use the following steps to install Oracle Client on a Windows platform:

1. Ensure that your computer meets the system requirements described in *Oracle Database Client Installation Guide for Microsoft Windows*.

2. Stop any Oracle services that may be running on your computer.

   In Windows Control Panel, choose **Administrative Tools**, then **Services**. Find the service names that start with "Oracle". Choose **Stop** for each one.

3. To start the installation, go to the Client installation directory and run SETUP.EXE.

   Oracle Universal Installer opens and displays the Welcome page. Click **Next** to advance to the next page.

4. On the Select Installation Type page, choose **Administrator** .

5. On the Specify Home Details page, provide the path of a home directory for Oracle Client. Oracle home is a subdirectory of the Oracle base directory.

   You can install Oracle Client in an existing Oracle base, or you can specify a new one. However, you must install Oracle Client in a new Oracle home. Do not install Oracle Client in the Oracle Database home directory, or in any other pre-existing Oracle home.

   Oracle Installer creates the directories for Oracle base and Oracle home if they do not already exist on your computer.

6. The Installer performs prerequisite checks.

   If the checks succeed, choose **Next** to advance to the next step.

   If any of the checks do not succeed, correct the problem and then click **Retry**. If more extensive changes are needed, cancel the installation, fix the problem, and then restart Oracle Installer

7. The Summary page displays the settings and components for the installation.
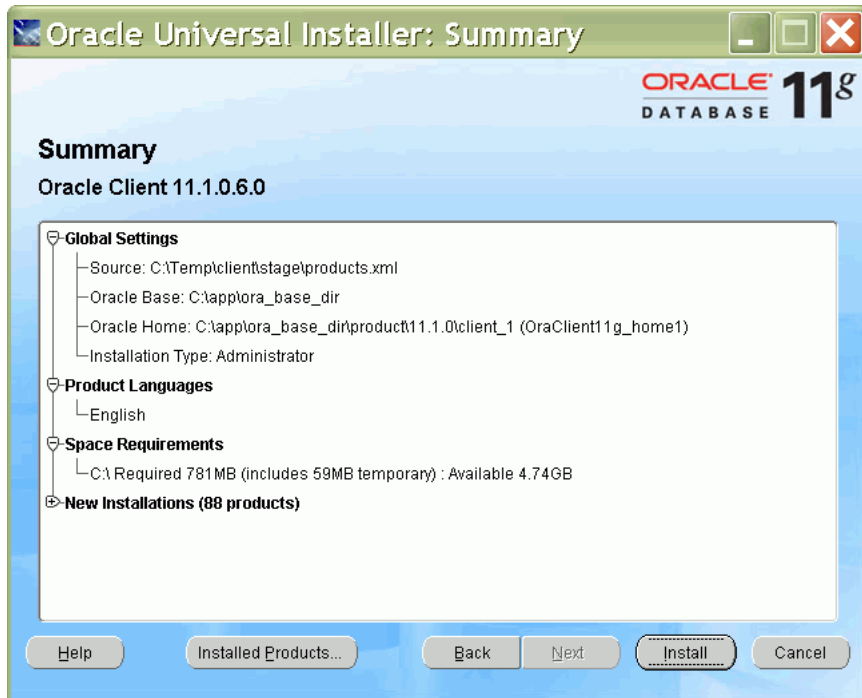
   Click **Install**.

8. On the Install page, the Installer displays a progress bar.

9. When the installation is complete, the Installer invokes the Configuration Assistants to create a database connection.
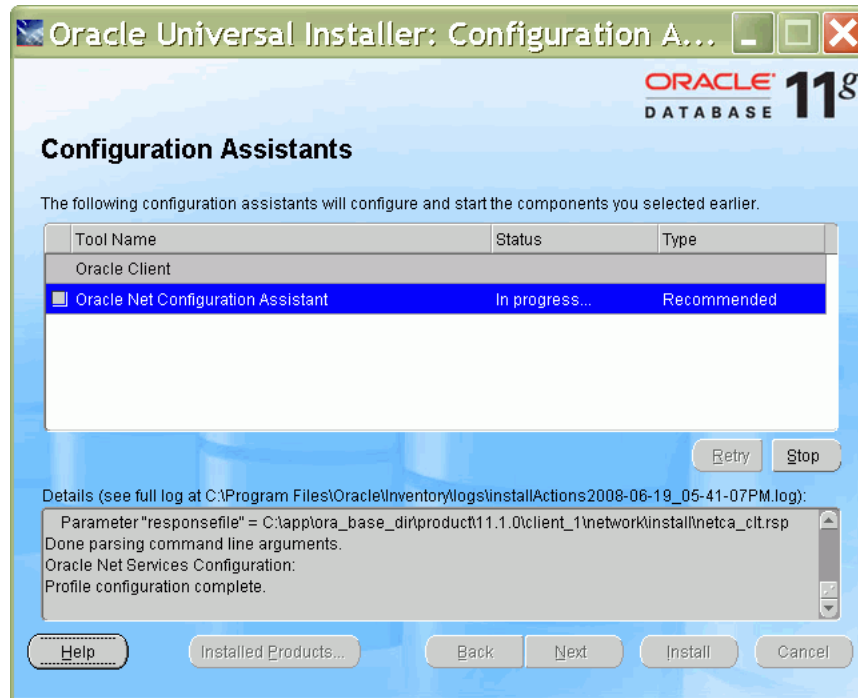
10. Oracle Net Configuration Assistant starts and displays the Welcome page.

Oracle Net Configuration Assistant determines that it is not necessary to create a specific database connection, because Oracle Client can use the Easy Connect naming method.

The Easy Connect naming method allows a client to present connection credentials at the time of the connection. The connection does not have to be specified as a Net Service Name or configured in advance.

For example, in SQL*Plus, with the Easy Connect method you can connect to a database with a statement like the following.

```
SQL>CONNECT username @ "hostname[:port][/servicename]"
```

The system will prompt for a password and establish a connection.

11. When the Oracle Net Configuration process is complete, click **Finish**.
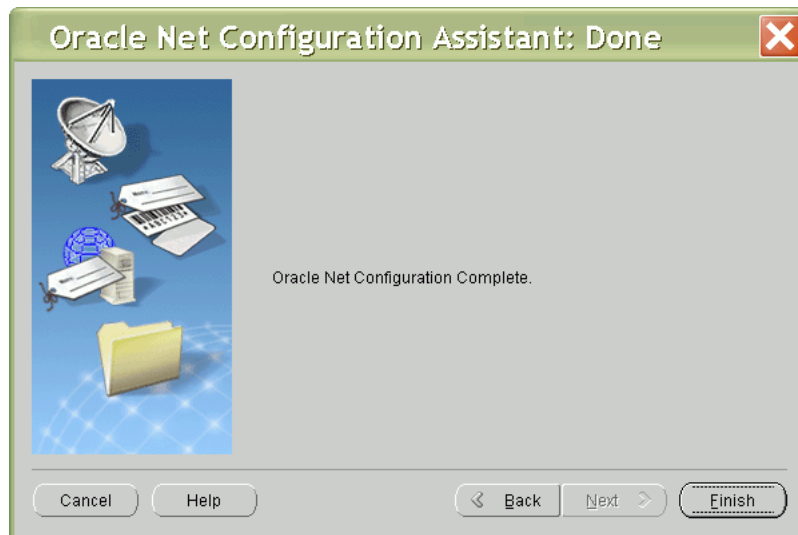
12. On the End of Installation page, confirm that the installation was successful. Then click **Exit** to exit the Installer.

Create a Net Service Name

Even if your Oracle Client installation supports Easy Connect Naming, you may need to create a Net Service Name for specific applications. For example, Oracle Spreadsheet Add-In for Predictive Analytics uses a Net Service Name to connect to an Oracle database. This connection is required whether you intend to work with data stored in Oracle or data stored in Excel.

To create a Net Service Name:

1. From the Windows Start menu, select the home directory of Oracle Client.

2. Choose **Configuration and Migration Tools**.

3. Choose **Net Configuration Assistant**.

4. Oracle Net Configuration Assistant displays the Welcome page.
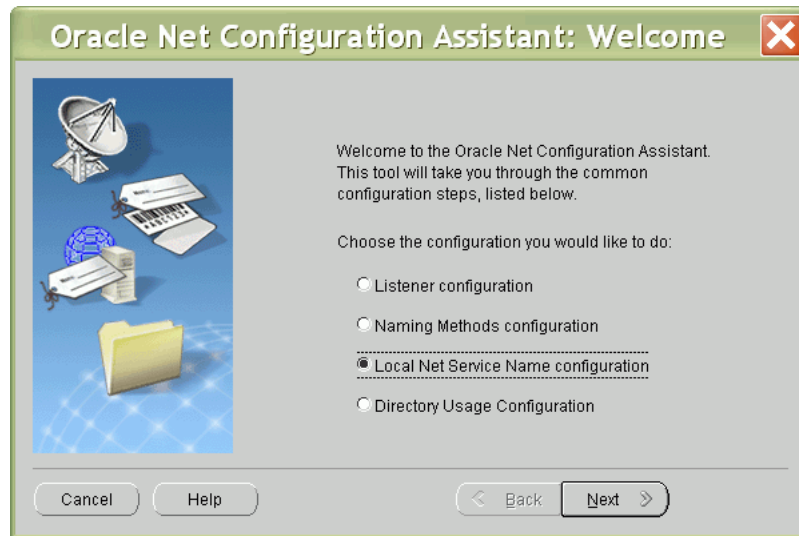
Choose **Local Net Service Name configuration**, then **Next**.

5. On the next page, choose **Add** to add a new Net Service Name. Then click **Next**.

6. Provide the name of the Oracle SID (Service Identifier). By default this is the global database name that you specified when you installed Oracle Database.
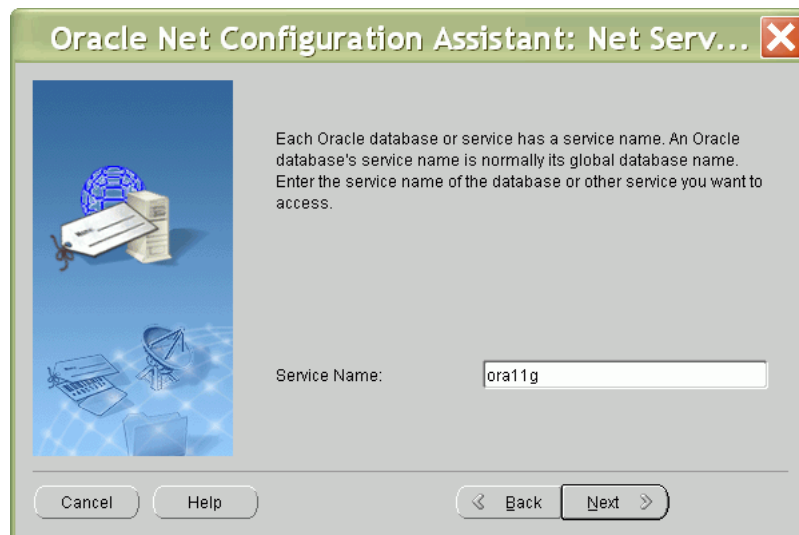
Choose **Next**.

7. On the next page specify the communication protocol. The default is TCP.

Choose **Next**.

8. On the next page, choose **Yes, perform a test** to test the new connection.
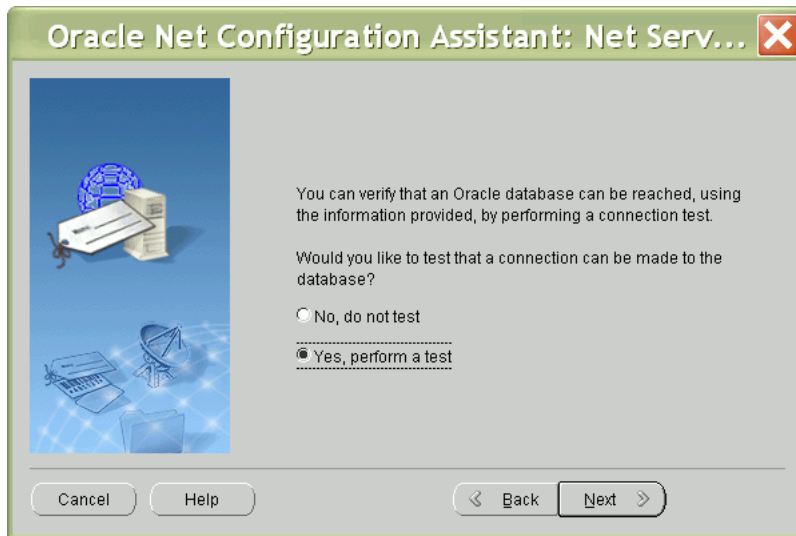
The connection will probably fail until you provide a new user ID and password.
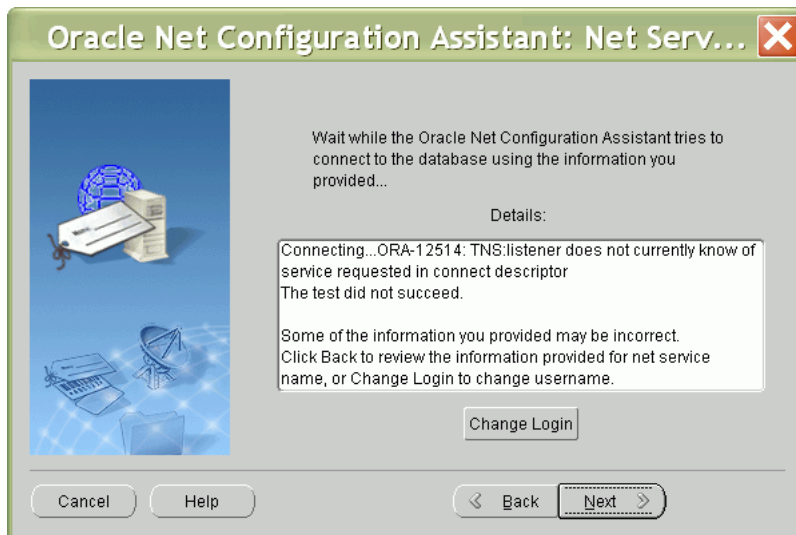
Choose **Change Login.**

Provide the user ID and password of an account that has data mining privileges on the host.

9. If the user name and password were the only problem with the connection, Oracle Net Configuration Assistant makes a successful test connection.
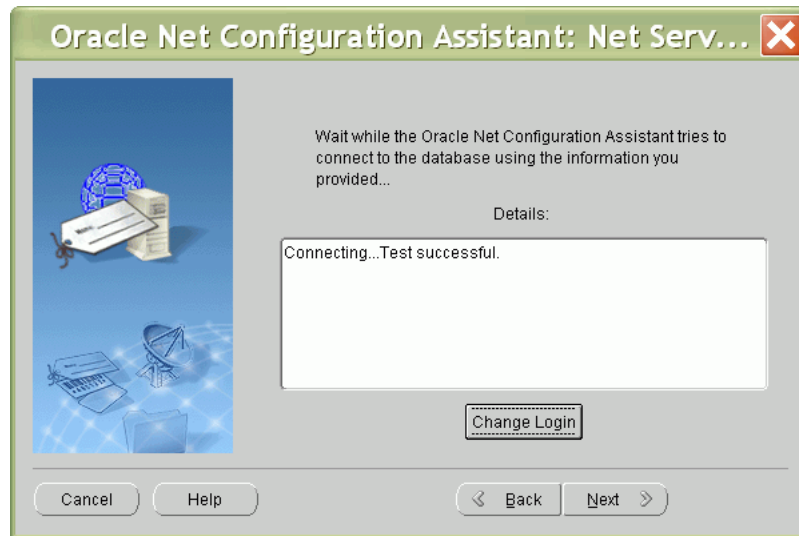
Choose **Next**.

10. Oracle Net Configuration Assistant reports that the connection configuration was successful. When you click **Next**, you are returned to the Welcome page, where you have the opportunity to perform additional configuration.

Click **Finish** to exit Oracle Net Configuration Assistant.

Perform Data Mining on a Remote Computer

You can use the Oracle Data Mining PL/SQL API in a remote instance of Oracle Database as long as:

■ SQL*Plus (or another client tool such as SQL Developer or JDeveloper) has been installed on your computer.

■ The system administrator has created a user ID for you in the remote database and given you the connection information.

■ The system administrator has granted the privileges to your user ID to allow the data mining activities that you will perform.

> **See Also:** *Oracle Data Mining Application Developer's Guide* to learn how to connect to a remote database and use the Java API

Run the Data Mining Sample Programs on a Remote Computer

If you wish to run the Data Mining sample programs on a remote computer:

1. Ask the system administrator to install the SH schema in the remote database, if it has not already been installed.

2. Ask the system administrator to run dmshgrants.sql to grant the appropriate privileges to your user ID.

3. Start SQL*Plus and log in to the remote database as the Data Mining user. Run dmsh.sql to populate your schema with objects used by the sample programs. (See "Create a Data Mining Demo User" on page 1-9 for information about dmshgrants.sql and dmsh.sql.)

4. Follow the instructions in "Run the Sample Programs" on page 1-10.

Download the Sample Programs from OTN

If you want to use local copies of the sample programs, you can download them from the Oracle Technology Network and run them in the remote database.

The programs are available in two different locations on OTN: the Oracle Data Mining page and the Sample Code page.

Oracle Data Mining page on OTN:

http://www.oracle.com/technology/products/bi/odm/index.html

Sample Code page on OTN:

http://www.oracle.com/technology/sample_code/index.html

To run the local program code in the remote database:

1. Unzip the sample program files to a directory on your computer.

2. Follow the instructions in "Run the Sample Programs" on page 1-10, but specify the local path of the sample program instead of the path under the remote Oracle home. For example, this statement would cause the `dmnbdemo.sql` program in your local directory to execute in the remote database:

   ```
   SQL>@ local_dir\dmnbdemo
   ```

Connect From Oracle Data Miner

Oracle Data Miner is the graphical user interface to Oracle Data Mining. It can connect to either a local or a remote database. It does not require Oracle Client.

When you start Oracle Data Miner, the Choose Connection dialog appears. To create a new connection, choose **New**.

The New Connection dialog appears.
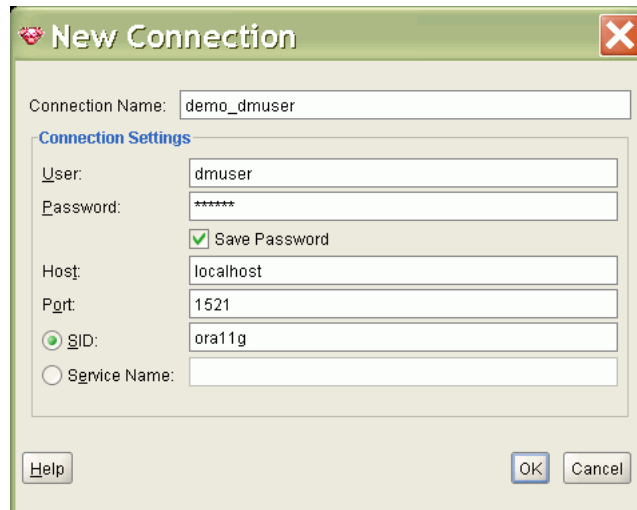
- For **Connection Name**, specify a meaningful name that you will remember.

- Provide the user name and password for the connection.

- For **Hostname**, specify `localhost` to indicate that the database is hosted on the local computer.

- For **Port**, specify the port that the database will use to listen to requests from the client. The default port is 1521.

- For **SID** (Service Identifier), specify the global database name that you created when you installed Oracle Database.

Click **OK** to create the connection and return to the Choose Connection dialog. When you select the connection and click **OK**, Oracle Data Miner is launched using the specified connection.

> **See Also:** Chapter 5, "Installing Oracle Data Miner"

Connect From Oracle Spreadsheet Add-In For Predictive Analytics

Oracle Spreadsheet Add-In for Predictive Analytics implements predictive analytics ("one-click" data mining) within Microsoft Excel. You can use the Spreadheet Add-In to analyze data stored in Excel or in Oracle. Either way, the Spreadsheet Add-In requires a connection to an Oracle database, which can be local or remote.

The Spreadsheet Add-In requires Oracle Client. It uses a Net Service Name to connect to the database. Even if the database is local, Oracle Client and a Net Service Name are required.

To connect to Oracle, follow these steps:

1. Install Oracle Client on your computer as described in "Install Oracle Client" on page 2-4.

2. Create a Net Service Name as described in "Create a Net Service Name" on page 2-8.

3. Add the Add-In to Excel as described in Chapter 6, "Installing the Spreadsheet Add-In for Predictive Analytics".

4. From the Add-In menu, choose **Connect**.
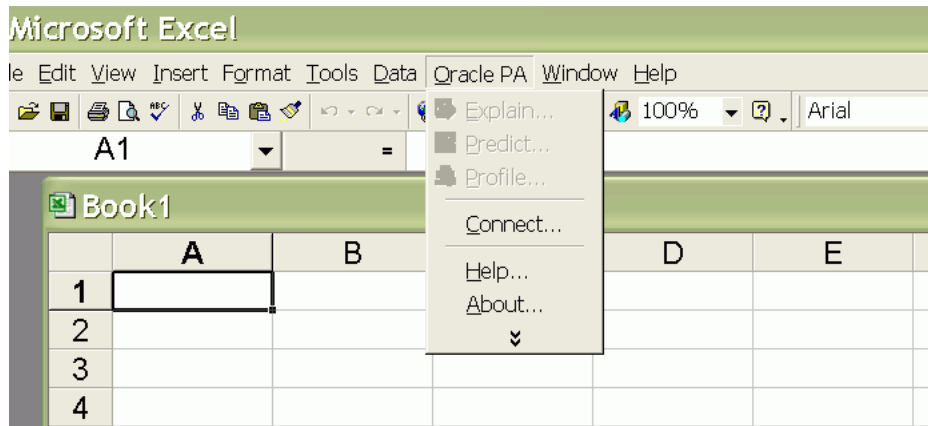
5. The Connect (Oracle Database) dialog appears.
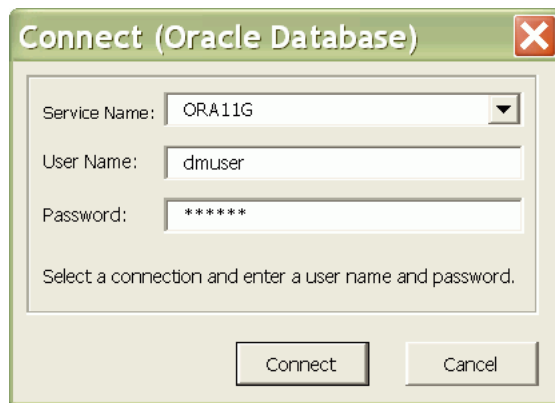
From the list of pre-defined Net Service Names, choose the connection you want to use. Provide a user name and password and click **Connect** to activate the Spreadsheet Add-In.

> **See Also:** Chapter 6, "Installing the Spreadsheet Add-In for Predictive Analytics"

3

Upgrading Oracle Data Mining

This chapter explains how to upgrade or downgrade your database.

> **See Also:** "What's New in Oracle Data Mining Administration" on page -xi for a summary of changes in Oracle Data Mining 11*g* that affect database administration

This chapter contains the following sections:

- Applying the 11.1.0.7 Patch Set
- Upgrading Oracle Data Mining
- Downgrading Oracle Data Mining

Applying the 11.1.0.7 Patch Set

If you already have an Oracle Database 11*g* Release 1 (11.1) installation, you can apply the 11.1.0.7 patch set. Instructions are provided in the Oracle Database Release Notes for your platform.

- *Oracle Database Patch Set Notes for Microsoft Windows (32-Bit)*
- *Oracle Database Patch Set Notes for Microsoft Windows x64*

Upgrading Oracle Data Mining

Oracle Data Mining upgrade is fully integrated with the upgrade of Oracle Database. In Oracle 11*g*, Data Mining metadata is migrated from DMSYS to SYS during the upgrade of Oracle Database.

Data Mining models are also upgraded during the upgrade of Oracle Database. The upgraded models are available for use in the new upgraded environment. Upgraded models will continue to work as they did in the prior release. All new mining functionality in 11*g* can be used in the upgraded environment.

In Oracle 11*g*, the CREATE MINING MODEL privilege must be granted to Data Mining user accounts. This privilege is required for creating Data Mining models in 11*g*. Refer to Chapter 4, "Users and Privileges for Data Mining" for more information.

> **See Also:** *Oracle Database Upgrade Guide* for information about upgrading Oracle Database

Supported Version Upgrades

Oracle Data Mining 10.2, including both metadata and models, can be upgraded to 11g.

Models created by the Oracle Data Mining PL/SQL API can be upgraded from 10.1 to 11*g*. Models created by the Oracle Data Mining Java API cannot be upgraded from 10.1 to 11*g*. The 10.1 version of the Java API was no longer supported in Oracle Data Mining 10.2.

Model upgrade from 9.2 to 11*g* is not supported.

Using Oracle Database Upgrade Assistant

You can use Oracle Database Upgrade Assistant to upgrade a database to 11*g*. During the upgrade, all Data Mining metadata that previously existed in the DMSYS schema is created in SYS. After the upgrade, all Data Mining metadata will reside in SYS and will no longer reside in DMSYS. Data Mining models, which reside in the owner's schema, will also be upgraded to 11*g*.

On Windows platforms, you can use the Database Upgrade Assistant graphical tool to upgrade to 11*g*. To start the Upgrade Assistant:

1. Go to the Windows **Start** menu and choose the Oracle home directory.

2. Choose the **Configuration and Migration Tools** menu.

3. Launch the **Upgrade Assistant**.

On Linux platforms, run the DBUA utility to upgrade Oracle Database.

After upgrading, check the upgrade log file and DBA_REGISTRY to ensure that the upgrade process completed successfully. Also check the DBA_MINING_MODELS view in the upgraded database. The newly upgraded mining models should be listed in this view.

After you have verified the upgrade and confirmed that there will be no need to downgrade, you should set the initialization parameter COMPATIBLE to 11.1. At that point, you can drop the DMSYS schema from the database. Once DMSYS is removed, the DBA_REGISTRY will no longer list Oracle Data Mining as a component.

> **Note:** After upgrading to 11g, you can no longer switch to the Data Mining Scoring Engine (DMSE). The Scoring Engine does not exist in 11*g*.

Exporting and Importing from a Dump File

If you wish, you can use a less automated approach to upgrading Oracle Data Mining. You can export the models created in a previous version of Oracle Database and import them into a new 11*g* database.

To export your Data Mining models to a dump file, follow the instructions in Chapter 7 on import/export.

Before importing from the dump file, run the DMEIDMSYS script to create the DMSYS schema in the 11g database.

```
SQL>CONNECT / as sysdba;
SQL>@ORACLE_HOME/RDBMS/admin/dmeidmsys.sql
SQL>EXIT;
```

> **Note:** The `TEMP` tablespace must already exist in the 11*g* database. The `DMEIDMSYS` script uses `TEMP` and `SYSAUX` to create the `DMSYS` schema.

To import the dump file into the database:

```
%ORACLE_HOME/bin/impdp system/password file_name = .....
SQL>CONNECT / as sysdba;
SQL>EXECUTE dmp_sys.upgrade_models('11.0.0');
SQL>ALTER SYSTEM FLUSH SHARED_POOL;
SQL>EXIT;
```

If you shutdown the database before operating on the upgraded mining models, this will also flush the shared pool.

After the import is complete, you can drop the `DMSYS` schema from the database. Once `DMSYS` is removed, the `DBA_REGISTRY` will no longer list Oracle Data Mining as a component.

Downgrading Oracle Data Mining

Before downgrading the database back to the previous version, ensure that there are no 11*g* mining models in the upgraded database. Issue the following SQL statement in `SYS` to verify this:

```
SQL>SELECT o.name FROM sys.model$ m, sys.obj$ o
               WHERE m.obj#=o.obj# AND m.version=2;
```

If there are any 11*g* mining models in the database, you must manually delete them using the `DBMS_DATA_MINING.DROP_MODEL` routine before downgrading the database. If you do not do this, the database downgrade process will be aborted.

4
Users and Privileges for Data Mining

This chapter explains how to create data mining users and control their mining activities through the use of database privileges.

> **See Also:** *Oracle Database 2 Day + Security Guide* for information about Oracle Database security

This chapter contains the following topics:

- Creating Data Mining Users
- System Privileges for Mining Models
- Object Privileges on Mining Models

Creating Data Mining Users

Data Mining should be performed by a user that is specifically configured for data mining activities.

The following SQL statement creates a database user and password. The user will use a tablespace called `default_tablespace` by default; its temporary tablespace is called `temp_tablespace`. Tablespaces for Data Mining users should be assigned according to standard DBA practices, depending on system load and system resources.

```
CREATE USER dmuser IDENTIFIED BY password
     DEFAULT TABLESPACE default_tablespace
     TEMPORARY TABLESPACE temp_tablespace
     QUOTA UNLIMITED on default_tablespace;
```

You must be logged in as a system administrator to create users and grant privileges.

> **Note:** To create a user that can use the Data Mining sample programs, you need to run two configuration scripts as described in "Create a Data Mining Demo User" on page 1-9.

Privileges Required for Data Mining

Oracle Corporation recommends a set of basic privileges for data mining. The following `GRANT` statements grant these privileges to a user named `dmuser`.

```
GRANT create mining model TO dmuser;
GRANT create procedure TO dmuser;
GRANT create session TO dmuser;
GRANT create table TO dmuser;
```

```
GRANT create sequence TO dmuser;
GRANT create view TO dmuser;
GRANT create job TO dmuser;
GRANT create type TO dmuser;
GRANT create synonym TO dmuser;
```

The `CREATE MINING MODEL` privilege is required for creating models. The other privileges may not be required for some mining activities, however it is prudent to grant them all as a group.

`SELECT` access to the data being mined is always required. The following example allows `dmuser` to mine data in the `CUSTOMERS` table in the `SH` schema.

```
GRANT SELECT ON sh.customers TO dmuser;
```

`EXECUTE` access to the Oracle Text package `ctxsys.ctx_ddl` is required for text mining.

```
GRANT EXECUTE ON ctxsys.ctx_ddl TO dmuser;
```

Additional access rights are required for exporting and importing mining models, as described in "Exporting and Importing Mining Models" on page 7-5.

Additional system and object privileges described in the following sections can be used to enable or restrict particular mining activities.

System Privileges for Mining Models

A **system privilege** confers the right to perform a particular action in the database or to perform an action on a type of schema objects. For example, the privileges to create tablespaces and to delete the rows of any table in a database are system privileges.

To grant a system privilege, you must either have been granted the system privilege with the `ADMIN OPTION` or have been granted the `GRANT ANY PRIVILEGE` system privilege.

The system privileges listed in Table 4–1 are required for performing specific operations on mining models.

Table 4–1   System Privileges for Mining Model

System Privilege	Allows you to ....
CREATE MINING MODEL	Create mining models in your own schema.
CREATE ANY MINING MODEL	Create mining models in any schema.
ALTER ANY MINING MODEL	Change the name or cost matrix of any mining model in any schema.
DROP ANY MINING MODEL	Drop any mining model in any schema.
SELECT ANY MINING MODEL	Apply a mining model in any schema, also view model details in any schema.
COMMENT ANY MINING MODEL	Add a comment to any mining model in any schema.
AUDIT ANY	Generate an audit trail for any mining model (or any object) in any schema.

Object Privileges on Mining Models

An object privilege confers the right to perform a particular action on a specific schema object. For example, the privilege to delete rows from the `SH.PRODUCTS` table is an example of an object privilege.

A user automatically has all object privileges for schema objects contained in his or her schema. A user can grant any object privilege on any schema object he or she owns to any other user or role.

Table 4–2   Object Privileges for Data Mining

Object Privilege	Allows you to ....
ALTER MINING MODEL	Change the name or cost matrix of the specified mining model object.
SELECT MINING MODEL	Apply or view the specified mining model object.

5

Installing Oracle Data Miner

This chapter provides instructions for installing Oracle Data Miner, the graphical user interface to Oracle Data Mining.

> **See Also:** Oracle Data Miner README

- About Oracle Data Miner
- Requirements
- Install Oracle Data Miner

About Oracle Data Miner

Oracle Data Miner is a full-featured graphical tool for developing data mining models for Oracle databases. You can deploy the models from within Oracle Data Miner or in SQL-based or Java applications. Using the Oracle Data Miner Code Generator, you can capture the entire process of model build, test, and apply in a PL/SQL package.

The main features of Oracle Data Miner are summarized in Table 5–1.

Table 5–1   Features of Oracle Data Miner

Feature	Description
Activities	In Oracle Data Miner, the creation, testing, and scoring of models is performed in units of work called activities. You can use activities to create, test, and apply models using any of the algorithms supported by Oracle Data Mining.
Defaults	Oracle Data Miner supplies reasonable default settings for model builds. However, a range of parameters are also available for advanced users.
Text Mining	Oracle Data Miner supports text mining.
Predictive Analytics	Oracle Data Miner supports predictive analytics operations: predict, explain, and profile.
Publish as Table	You can use the Publish as Database Table feature of Oracle Data Miner to publish mining results in a table or view for use by query and reporting tools.
Code Generator	You can use the Oracle Data Miner Code Generator to create PL/SQL packages that implement mining activites. The Code Generator is a useful tool for producing code to integrate into data mining applications.

Requirements

Oracle Data Miner requires access to an instance of Oracle Database installed with the Data Mining option. The database can be installed locally on your computer or it can be installed on a remote host.

The connecting user ID must have the database and object permissions described in Chapter 4. If you want to use the sample data provided with the sample programs, the SH schema must be installed in the database and the user ID must be enabled as described in "Create a Data Mining Demo User" on page 1-9.

Oracle Data Miner does not require Oracle Client, and a Net Service Name for the database connection is not needed. However, if you wish to import data from text files in Oracle Data Miner, SQL*Loader is required. SQL*Loader is installed with Oracle Client. Follow the instructions in "Install Oracle Client" on page 2-4 to obtain SQL*Loader.

Oracle Data Miner requires Java JDK version 1.5. To check the version of Java, you can execute the following at the command prompt:

```
> java -version
```

Install Oracle Data Miner

To install Oracle Data Miner:

1. Go to the Oracle Data Miner page on the Oracle Technology Network:

    http://www.oracle.com/technology/products/bi/odm/odminer.html

    You will find the following available for download:

    ■ Oracle Data Miner

    ■ Oracle Data Miner Tutorial

    ■ Oracle Data Miner PL/SQL Code Generator for JDeveloper

    ■ Oracle Data Miner PL/SQL Code Generator for SQL Developer

2. Download Oracle Data Miner and unzip to a new, empty directory. The zip file includes a readme.

3. Download and unzip the tutorial. The zip file includes sample data, a readme, and the tutorial PDF.

4. If you wish to use the code generators, follow the installation instructions in the Oracle Data Miner readme.

To start Oracle Data Miner:

1. Navigate to the bin directory in the Oracle Data Miner installation diretory.

2. Double click odminerw.exe to launch Oracle Data Miner.

    > **Note:** The odminer.exe executable, also in the bin directory, displays a console window that can be used for troublshooting.

3. To connect to an Oracle database, follow the instructions in "Connect From Oracle Data Miner" on page 2-12.

6

Installing the Spreadsheet Add-In for Predictive Analytics

This chapter provides instructions for installing Oracle Spreadsheet Add-In for Predictive Analytics.

> **See Also:** Spreadsheet Add-In for Predictive Analytics README

This chapter contains the following topics:

- About the Spreadsheet Add-In
- Requirements
- Install the Spreadsheet Add-In

About the Spreadsheet Add-In

The Oracle Spreadsheet Add-In for Predictive Analytics adds predictive analytics features to Microsoft Excel. Using simple "one click" data mining for `PREDICT`, `EXPLAIN`, and `PROFILE` operations, Excel users can mine data stored in Excel or in Oracle Database.

Predictive analytics provides automated methodologies that simplify data mining.

> **See Also:** *Oracle Data Mining Concepts* for information about Oracle predictive analytics

Requirements

The Spreadsheet Add-In requires access to an instance of Oracle Database installed with the Data Mining option. The database can be installed locally on your computer or it can be installed on a remote host.

The connecting user ID must have the database and object permissions described in Chapter 4. If you want to use the sample data provided with the sample programs, the `SH` schema must be installed in the database and the user ID must be enabled as described in "Create a Data Mining Demo User" on page 1-9.

Oracle Spreadsheet Add-In requires:

- Microsoft Excel 2000 or later
- Oracle Objects for OLE, installed with Oracle Client
- An Oracle Net Service Name for the database connection

Oracle Client

Oracle Spreadsheet Add-In for Predictive Analytics requires the presence of Oracle Objects for OLE and Oracle Net Configuration Assistant. To obtain these components, you must install Oracle Client on the computer where Excel and the Spreadsheet Add-In are installed.

Oracle Client is required even if your database is local.

> **See Also:** "Install Oracle Client" on page 4 for installation instructions

Oracle Net Service Name

Oracle Spreadsheet Add-In for Predictive Analytics uses a Net Service Name to connect to the database server. A Net Service Name is required even if your database is local.

> **See Also:** "Create a Net Service Name" on page 2-8 for instructions

Install the Spreadsheet Add-In

Once you have installed Oracle Client and created a Net Service Name, follow these steps to install the Spreadsheet Add-In:

1. Download the Spreadsheet Add-In from the Oracle Data Mining Web site at http://www.oracle.com/technology/products/bi/odm/index.html.

2. Open the ZIP file and extract the following files to the Microsoft Office Library directory:

   - `Predictive_Analytics.xla` if you have Microsoft Excel 2000, 2002, or 2003

   - `Predictive_Analytics_2007.xla` if you have Microsoft Excel 2007

   - `PAhelp.chm`, the help file for the Spreadsheet Add-In

   The Microsoft Office library has a path such as this one:

   `C:\Program Files\Microsoft Office\Office\Library`

   The ZIP file also includes a readme. You can extract the readme to a directory of your choosing.

3. Open Excel and click **Tools** > **Add-Ins**. Select **Oracle Predictive Analytics** from the Add-Ins dialog box.
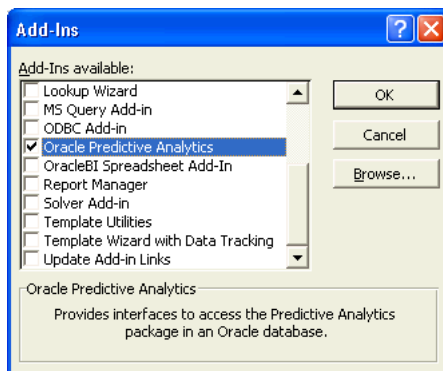
> **Note:** If you had a previous version of the Spreadsheet Add-In, uncheck **Oracle Predictive Analytics** and click **OK** to close the Add-Ins dialog. Then reopen the dialog and check the **Oracle Predictive Analytics** box.

4. The OraclePA menu is added to the Excel toolbar.

5. In Excel, click **OraclePA** > **Connect**.

6. On the Connect dialog box, select a Net Service Name from the drop-down list, and type in your database user name and password.

7

Mining Model Schema Objects

In this chapter, you will learn how to find information about mining models in the data dictionary and how to perform various operations on mining models.

> **See Also:** Chapter 4, "Users and Privileges for Data Mining" for information about system and object privileges associated with mining model objects.

This chapter contains the following topics:

- Obtaining Information from the Data Dictionary
- Adding a Comment to a Mining Model
- Auditing Mining Models
- Exporting and Importing Mining Models

Obtaining Information from the Data Dictionary

Mining models are database schema objects. They can be queried in the `ALL`, `DBA`, and `USER` data dictionary views.

The data dictionary views in Table 7–1 reveal information about mining models created by Oracle Data Mining.

Table 7–1   Oracle Data Mining Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
ALL_MINING_MODELS	DBA_MINING_MODELS	USER_MINING_MODELS
ALL_MINING_MODEL_ATTRIBUTES	DBA_MINING_MODEL_ATTRIBUTES	USER_MINING_MODEL_ATTRIBUTES
ALL_MINING_MODEL_SETTINGS	DBA_MINING_MODEL_SETTINGS	USER_MINING_MODEL_SETTINGS

Obtaining Information about Mining Models

You can query the `ALL_MINING_MODELS` data dictionary view to obtain information about all accessible mining model objects. `USER` and `DBA` versions of this view are also available.

```
SQL> DESCRIBE all_mining_models
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                     NOT NULL VARCHAR2(30)
 MODEL_NAME                                NOT NULL VARCHAR2(30)
 MINING_FUNCTION                                    VARCHAR2(30)
```

```
ALGORITHM                                          VARCHAR2(30)
CREATION_DATE                             NOT NULL DATE
BUILD_DURATION                                     NUMBER
MODEL_SIZE                                         NUMBER
COMMENTS                                           VARCHAR2(4000)
```

The COMMENTS column contains comments created by SQL COMMENT, if they exist. See "Adding a Comment to a Mining Model" on page 7-3.

> **See Also:**
>
> *Oracle Data Mining Concepts* for information about mining functions and algorithms
>
> *Oracle Data Mining Application Developer's Guide* for more information about ALL_MINING_MODELS

Obtaining Information about Mining Model Attributes

You can query the ALL_MINING_MODEL_ATTRIBUTES data dictionary view to obtain information about all accessible mining model attributes. USER and DBA versions of this view are also available.

```
SQL> DESCRIBE all_mining_model_attributes
 Name                                     Null?    Type
 ---------------------------------------- -------- ----------------------------
 OWNER                                    NOT NULL VARCHAR2(30)
 MODEL_NAME                               NOT NULL VARCHAR2(30)
 ATTRIBUTE_NAME                           NOT NULL VARCHAR2(30)
 ATTRIBUTE_TYPE                                    VARCHAR2(11)
 DATA_TYPE                                         VARCHAR2(12)
 DATA_LENGTH                                       NUMBER
 DATA_PRECISION                                    NUMBER
 DATA_SCALE                                        NUMBER
 USAGE_TYPE                                        VARCHAR2(8)
 TARGET                                            VARCHAR2(3)
```

> **Note:** The attributes listed in this data dictionary view are the physical columns in the build data that were used to construct the model. Some or all of these columns should be present for scoring. These **data attributes** are referred to as the **model signature**.
>
> The term *attribute* is more accurately used to designate the numericals and categoricals derived from the data attributes for manipulation by the algorithm. These **model attributes** may or may not correspond to data attributes, depending on transformations and on whether or not the column is nested. The model attributes can be viewed in the model details (GET_MODEL_DETAILS functions).

> **See Also:**
>
> *Oracle Data Mining Application Developer's Guide* for more information about attributes
>
> *Oracle Data Mining Application Developer's Guide* for more information about ALL_MINING_MODEL_ATTRIBUTES

Obtaining Information about Mining Model Settings

You can query the `ALL_MINING_MODEL_SETTINGS` data dictionary view to obtain information about all accessible mining model settings. `USER` and `DBA` versions of this view are also available.

```
SQL> DESCRIBE all_mining_model_settings
 Name                                      Null?    Type
 ----------------------------------------- -------- ---------------------------
 OWNER                                     NOT NULL VARCHAR2(30)
 MODEL_NAME                                NOT NULL VARCHAR2(30)
 SETTING_NAME                              NOT NULL VARCHAR2(30)
 SETTING_VALUE                                      VARCHAR2(4000)
 SETTING_TYPE                                       VARCHAR2(7)
```

Model settings can be specified in a settings table used in the model build. The settings all have default values. The defaults are used when they are not overridden by settings specified in the settings table, or when there is no settings table.

> **See Also:**
>
> *Oracle Database PL/SQL Packages and Types Reference* for descriptions of model settings
>
> *Oracle Data Mining Concepts* for details about algorithm settings
>
> *Oracle Data Mining Application Developer's Guide* for more information about `ALL_MINING_MODEL_SETTINGS`

Adding a Comment to a Mining Model

You can associate a comment with a mining model using a SQL `COMMENT` statement.

```
COMMENT ON MINING MODEL schema_name.model_name IS string;
```

> **Note:** To add a comment to a model in another schema, you must have the `COMMENT ANY MODEL` system privilege.

To drop a comment, set it to the empty `' '` string.

The following statement adds a comment to the model `DT_SH_CLAS_SAMPLE` in your own schema.

```
SQL> COMMENT ON mining model dt_sh_clas_sample IS
            'Decision Tree model predicts promotion response';
```

You can view the comment by querying the catalog view `USER_MINING_MODELS`.

```
SQL> COLUMN comments FORMAT a22
SQL> SELECT model_name, mining_function, algorithm, comments FROM user_mining_models;

MODEL_NAME        MINING_FUNCTION  ALGORITHM       COMMENTS
----------------- ---------------- --------------- ------------------------------------------------
DT_SH_CLAS_SAMPLE CLASSIFICATION   DECISION_TREE   Decision Tree model predicts promotion response
```

To drop this comment from the database, issue the following statement:

```
SQL> COMMENT ON mining model dt_sh_clas_sample '';
```

Auditing Mining Models

You can use the SQL auditing system to track operations on data mining models.

> **Note:** To audit a mining model in another schema, you must have the `AUDIT ANY` system privilege.

Enabling Auditing in the Database

The database initialization parameter `AUDIT_TRAIL` controls auditing capabilities in the database. To enable auditing, set `AUDIT_TRAIL` to `DB`, `DB_EXTENDED`, or `OS`.

Set `AUDIT_TRAIL` to `NONE` to prevent auditing information from being recorded. By default, `AUDIT_TRAIL` is set to `NONE`.

Opening an Audit Trail on Mining Models

Use the SQL `AUDIT` statement to open an auditing trail on a data mining model.

```
AUDIT {operation|ALL} ON mining model schema_name.model_name;
```

You can track the following operations on mining models.

Audit Operation	Description
AUDIT	Generate an audit trail for a mining model
COMMENT	Add a comment to a mining model
GRANT	Give permission to a user to access the model
RENAME	Change the name of the model
SELECT	Apply the model or view its signature.

For example, this statement generates an audit trail for all `GRANT` operations on the model `NB_SH_CLAS_SAMPLE` in the `DMUSER` schema.

```
SQL> AUDIT GRANT ON mining model dmuser.nb_sh_clas_sample;
```

This statement generates an audit trail for all operations on the same model.

```
SQL> AUDIT GRANT,AUDIT,COMMENT,RENAME,SELECT
         ON mining model dmuser.nb_sh_clas_sample;
```

You can refine the criteria for auditing with the following additional semantics.

```
AUDIT {operation|ALL} ON MINING MODEL schema_name.model_name
                              [BY [SESSION|ACCESS]]
                              [WHENEVER [NOT] SUCCESSFUL]];
```

Closing the Audit Trail

Use the `NOAUDIT` statement to stop one or more auditing operations previously enabled by the `AUDIT` statement.

```
NOAUDIT {operation| ALL} ON MINING MODEL model_name
                                  [WHENEVER [NOT] SUCCESSFUL]];
```

Viewing the Audit Trail

For each audited operation, Oracle Database produces an audit record containing:

- The name of the user performing the operation
- The type of operation
- The object involved in the operation
- The date and time of the operation

Several data dictionary views present auditing information. Some examples are:

- `DBA_AUDIT_OBJECT` displays audit trail records for all objects in the database.
- `USER_AUDIT_OBJECT` displays audit trail records for all objects accessible to the current user
- `DBA_OBJ_AUDIT_OPTS` describes auditing options for all objects in the database.
- `USER_OBJ_AUDIT_OPTS` describes auditing options for all objects owned by the current user.

> **Note:** The Oracle Database auditing system is a powerful, highly configurable tool for tracking operations on schema objects. Refer to the following manuals for more information:
>
> - *Oracle Database SQL Language Reference* for documentation of the `AUDIT` and `NOAUDIT` statements
> - *Oracle Database Reference* for documentation of the `AUDIT_TRAIL` initialization parameter and the data dictionary views for querying the database audit trail.
> - *Oracle Database Security Guide* for a comprehensive discussion of database auditing.

Exporting and Importing Mining Models

You can export data mining models to flat files to back up work in progress or to move models to a different instance of Oracle Database Enterprise Edition (such as from a development database to a production database). All methods for exporting and importing models are based in Oracle Data Pump technology.

Oracle Data Pump consists of two command-line clients and two PL/SQL APIs. The command-line clients, `EXPDP` and `IMPDP`, provide an easy-to-use interface to the Data Pump export and import utilities. The Data Mining APIs also use the Data Pump export and import utilities.

You can export and import models at different levels, depending on your access rights in the database:

- **Database**. When a DBA exports a full database using `EXPDP`, all data mining models in the database are exported. The `IMPDP` utility imports all the models with the other objects in the database.
- **Schema**. When a DBA or an individual user exports a schema using `EXPDP`, all the data mining models in the schema are exported. Likewise, `IMPDP` imports all the models with the other objects in the schema.

- **Models Only**. The Data Mining APIs contain utilities for exporting and importing mining models. You can select all the mining models in a schema or mining models that match specific criteria.

  You cannot select individual models using `EXPDP` or `IMPDP`.

The Data Pump export utility writes the tables and metadata that constitute a model to a dump file set, which consists of one or more files. The Data Pump import utility retrieves the tables and metadata from the dump file and restores them to the target database.

> **See Also:**
>
> - *Oracle Database Utilities* for a complete discussion of Oracle Data Pump and the `expdp` and `impdp` utilities
>
> - *Oracle Database PL/SQL Packages and Types Reference* for detailed information about the export and import procedures in the `DBMS_DATA_MINING` package
>
> - *Oracle Data Mining Java API Reference* for information about the export and import classes in the Oracle Data Mining Java API

Prerequisites

To export and import Data Mining models, you must have read and write access to a directory object, and you may need additional database permissions.

Directory Objects

A directory object is a logical name in the database for a physical directory on the host computer. Without read and write access to a directory object, you cannot access the host computer file system from within Oracle Database.

You must have the `CREATE ANY DIRECTORY` privilege to create directory objects.

The following SQL command creates, or re-creates if it already exists, a directory object named `dmtest`. The file system directory (in this example, `C:\oracle\product\11.1.0\dmining`) must already exist and have shared read/write access rights granted by the operating system.

```
CREATE OR REPLACE DIRECTORY dmtest AS 'c:\oracle\product\11.1.0\dmining';
```

This SQL command gives user `dmuser` both read and write access to `dmtest`.

```
GRANT ALL ON DIRECTORY dmtest TO dmuser;
```

For more information about creating database directories, refer to the `CREATE DIRECTORY` and `GRANT` commands in the *Oracle Database SQL Language Reference*.

Additional Database Privileges

You may need special privileges in the database to take full advantage of all Data Pump features, such as importing models and other objects into a different schema. These privileges are granted by the `EXP_FULL_DATABASE` and `IMP_FULL_DATABASE` roles, which are only available to privileged users (such as `SYS` or a user with the `DBA` role).

You do not need these roles to export models from your own schema. To import models, you must have the same database privileges as the user who created the dump file set. Otherwise, a DBA with full system privileges must import the models.

PL/SQL APIs for Exporting and Importing Models

The `DBMS_DATA_MINING` PL/SQL package contains these two procedures:

- `EXPORT_MODEL`

- `IMPORT_MODEL`

For more information about these procedures, refer to the *Oracle Database PL/SQL Packages and Types Reference*.

Java APIs for Exporting and Importing Models

Oracle Database implements the industry-standard Java Data Mining (JDM) API Specification, which includes these two interfaces:

- `javax.datamining.task.ExportTask`

- `javax.datamining.task.ImportTask`

For more information about the standard JDM API, refer to the Java Help for the JSR-73 Specification, which is available on the Oracle Technology Network at

`http://www.oracle.com/technology/products/bi/odm/JSR-73/index.html`

Tables Created By Exporting and Importing Models

The Data Mining export and import utilities create tables in the user's schema that are for internal use only:

- `DM$P_MODEL_EXPIMP_TEMP`. Used for internal purposes during export and import, and provides a job history.

- `DM$P_MODEL_IMPORT_TEMP`. Used only for internal purposes during import.

- `DM$P_MODEL_TABKEY_TEMP`. Used only for internal purposes during export and import.

Do not alter these tables. However, you may drop them when no export or import job is running. The utilities will re-create them for the next job.

Example: Exporting and Importing Models

This example creates a dump file containing two models and imports the models from the dump file.

This example was generated on a Linux system. The directory object `mydir` identifies the path `/scratch/dumpfiles`. (See .)

Exporting Models from the DMUSER Schema

In this example, the `dmuser` schema contains two mining models.

```
SQL> CONNECT dmuser
Enter password: password

SQL> SELECT model_name FROM user_mining_models;

MODEL_NAME
------------------------------
GLMR_SH_REGR_SAMPLE
SVMC_SH_CLAS_SAMPLE
```

The following command exports all models from `dmuser` to the directory identified by the directory object `mydir`.

```
SQL> EXECUTE dbms_data_mining.export_model('all_dmuser_models.dmp', 'mydir');
```

An export or import creates a log file in the same directory as the dump file. Error messages are returned to the current output device (such as the screen), and the log file may provide additional information.

This sample export created two files in the `mydir` directory:

■ A dump file named `ALL_DMUSER_MODELS01.DMP` (note the 2-digit suffix added to the name)

■ A log file with the name `DMUSER_EXP_920.LOG`

For detailed information about the default names of files, see *Oracle Database PL/SQL Packages and Types Reference*.

You can view the log file using a system command or editor. You must know the path of the physical directory in order to locate the file.

`DMUSER_EXP_920.LOG` lists the two exported mining models and supporting objects .

```
Starting "DMUSER"."DMUSER_exp_17":  DM_EXPIMP_JOB_ID=17
Estimate in progress using BLOCKS method...
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 1.062 MB
Processing object type TABLE_EXPORT/TABLE/TABLE
Processing object type TABLE_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type TABLE_EXPORT/TABLE/INDEX/INDEX
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type TABLE_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type TABLE_EXPORT/TABLE/POST_INSTANCE/PROCACT_INSTANCE
. . exported "DMUSER"."DM$PIGLMR_SH_REGR_SAMPLE"        7.085 KB      14 rows
. . exported "DMUSER"."DM$PISVMC_SH_CLAS_SAMPLE"        7.507 KB      17 rows
. . exported "DMUSER"."DM$PCGLMR_SH_REGR_SAMPLE"       53.27 KB    2278 rows
. . exported "DMUSER"."DM$PAGLMR_SH_REGR_SAMPLE"        5.796 KB      24 rows
. . exported "DMUSER"."DM$PBGLMR_SH_REGR_SAMPLE"       15.76 KB      67 rows
. . exported "DMUSER"."DM$PDGLMR_SH_REGR_SAMPLE"           8 KB      66 rows
. . exported "DMUSER"."DM$PDSVMC_SH_CLAS_SAMPLE"        9.023 KB      88 rows
. . exported "DMUSER"."DM$PFGLMR_SH_REGR_SAMPLE"        5.656 KB      10 rows
. . exported "DMUSER"."DM$POSVMC_SH_CLAS_SAMPLE"        5.320 KB       8 rows
. . exported "DMUSER"."DM$PXSVMC_SH_CLAS_SAMPLE"        7.265 KB      77 rows
. . exported "DMUSER"."DM$PZSVMC_SH_CLAS_SAMPLE"        6.164 KB       1 rows
. . exported "DMUSER"."DM$P_MODEL_EXPIMP_TEMP"          5.921 KB       2 rows
Master table "DMUSER"."DMUSER_exp_17" successfully loaded/unloaded
******************************************************************************
Dump file set for DMUSER.DMUSER_exp_17 is:
  /scratch/dumpfiles/all_dmuser_models01.dmp
Job "DMUSER"."DMUSER_exp_17" successfully completed at 10:30:19
```

Importing Models Into the Same Schema

The exported models still exist in `DMUSER`. In this example, we drop the models before importing from the dump file. An import will not overwrite an existing model with the same name.

```
SQL> EXECUTE dbms_data_mining.drop_model('GLMR_SH_REGR_SAMPLE');
SQL> EXECUTE dbms_data_mining.drop_model('SVMC_SH_CLAS_SAMPLE');
```

The following command restores all models from the dump file to the DMUSER schema.

```
SQL> EXECUTE dbms_data_mining.import_model('all_dmuser_models01.dmp', 'mydir');
```

Importing Models Into a Different Schema

A user with the necessary privileges can load the models from a dump file into a different schema. The target schema must have the same permissions and have access to the same tablespace as the schema from which the models were exported.

```
CREATE USER dmuser2 IDENTIFIED BY password
        default tablespace dmuser
        temporary tablespace temp
        quota unlimited on dmuser;

@ $ORACLE_HOME/RDBMS/demo/dmshgrants dmuser2
```

The import command, also executed as SYS, loads the two models into the DMUSER2 schema.

```
SQL> EXECUTE dbms_data_mining.import_model('all_dmuser_models01.dmp', 'mydir',
null, null, null, 'todmuser2', 'dmuser:dmuser2');
```

A parameter specifies TODMUSER2.LOG as the name of the log file; the .LOG extension is added automatically to the name. The log file shows the names of the imported models and supporting metadata.

```
Master table "SYS"."todmusr2" successfully loaded/unloaded
Starting "SYS"."todmusr4":  DM_EXPIMP_JOB_ID=21
Processing object type TABLE_EXPORT/TABLE/TABLE
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
. . imported "DMUSER2"."DM$PIGLMR_SH_REGR_SAMPLE"         7.085 KB      14 rows
. . imported "DMUSER2"."DM$PISVMC_SH_CLAS_SAMPLE"         7.507 KB      17 rows
. . imported "DMUSER2"."DM$PCGLMR_SH_REGR_SAMPLE"         53.27 KB    2278 rows
. . imported "DMUSER2"."DM$PAGLMR_SH_REGR_SAMPLE"         5.796 KB      24 rows
. . imported "DMUSER2"."DM$PBGLMR_SH_REGR_SAMPLE"         15.76 KB      67 rows
. . imported "DMUSER2"."DM$PDGLMR_SH_REGR_SAMPLE"            8 KB       66 rows
. . imported "DMUSER2"."DM$PDSVMC_SH_CLAS_SAMPLE"         9.023 KB      88 rows
. . imported "DMUSER2"."DM$PFGLMR_SH_REGR_SAMPLE"         5.656 KB      10 rows
. . imported "DMUSER2"."DM$POSVMC_SH_CLAS_SAMPLE"         5.320 KB       8 rows
. . imported "DMUSER2"."DM$PXSVMC_SH_CLAS_SAMPLE"         7.265 KB      77 rows
. . imported "DMUSER2"."DM$PZSVMC_SH_CLAS_SAMPLE"         6.164 KB       1 rows
. . imported "DMUSER2"."DM$P_MODEL_EXPIMP_TEMP"          5.921 KB       2 rows
Processing object type TABLE_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type TABLE_EXPORT/TABLE/INDEX/INDEX
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type TABLE_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type TABLE_EXPORT/TABLE/POST_INSTANCE/PROCACT_INSTANCE
Job "SYS"."todmusr2" successfully completed at 11:15:26
```

8

The Data Mining Sample Programs

A number of sample programs are available with Oracle Data Mining. These programs illustrate the many features of the PL/SQL and Java APIs.

The sample programs create a set of models in the database. You can examine the sample source code, which includes numerous comments, to familiarize yourself with the Oracle Data Mining APIs, and you can create your own models by modifying the samples.

> **Note:** This chapter provides general information about the sample programs and the sample data.
>
> Instructions for installing and running the sample programs are provided in Chapter 1.

This chapter includes the following sections:

- PL/SQL Programs
- Java Programs
- Text Mining Programs
- The Sample Data

PL/SQL Programs

The PL/SQL sample programs illustrate the use of the `DBMS_DATA_MINING` package for creating models and the `DBMS_DATA_MINING_TRANSFORM` package for performing transformations on the mining data.

> **See Also:**
>
> - *Oracle Database PL/SQL Packages and Types Reference* for syntax of the PL/SQL API
> - *Oracle Data Mining Application Developer's Guide* for information on the use of the APIs
> - *Oracle Database SQL Language Reference* for syntax of the SQL functions for model scoring

PL/SQL Program Summaries

Summary descriptions of the PL/SQL sample programs are provided in Table 8–1. For detailed descriptions of the sample programs, see the comments in the source code.

Table 8–1   Overview of the PL/SQL Sample Programs

Mining Function	Description
Classification	The classification programs demonstrate various preprocessing techniques and perform the following steps:
	■ Build a classification model using training data
	■ Display model details and settings
	■ Test the model by applying the model on the test data
	■ Compute test metrics, such as confusion matrix, lift, and ROC
	■ Apply the model on the scoring data
	■ Present apply results
	■ Present ranked apply results, influenced by a cost matrix
	`dmnbdemo.sql` illustrates Naive Bayes.
	`dmdtdemo.sql` illustrates Decision Tree.
	`dmsvcdem.sql` illustrates SVM classification.
	`dmglcdem.sql` illustrates GLM classification (binary logistic regression)
	The `dmdtxvlddemo.sql` program demonstrates cross-validation techniques for decision tree based-classification. With minor modifications, this program can be used to perform cross validation using other models/algorithms.
Regression	`dmsvrdem.sql` uses different test metrics, but otherwise performs most of the same steps used in the classification programs. Selected attributes of the input data are preprocessed (normalized).
	NOTE: `dmsvrdem.sql` illustrates the new Automatic Data Preparation feature.
	`dmglrdem.sql` illustrates GLM regression (multivariate linear regression)
Anomaly Detection	`dmsvodem.sql` illustrates one-class SVM
Association	`dmardemo.sql` builds an association model and presents frequent itemsets and association rules as output.
Clustering	`dmkmdemo.sql` (*k*-Means) and `dmocdemo.sql` (0-Cluster) build clustering models and present cluster details, such as rules, centroid, and histogram for each cluster as output. The models are scored, and the probabilities associated with each cluster are returned as output. Selected attributes of the input data are preprocessed.
	NOTE: `dmkmdemo.sql` illustrates the new Automatic Data Preparation feature.
Feature extraction	`dmnmdemo.sql` builds a feature extraction model and presents model details as the output. The model is scored, and each feature ID is associated with a probability. Selected attributes of the input data are preprocessed (normalized).
Attribute importance	`dmaidemo.sql` builds an attribute importance model and presents a list of important attributes as the output of model details. Selected attributes of the input data are preprocessed (binned).

Data Mining SQL Scoring Functions

Most of the PL/SQL sample programs use the Data Mining SQL functions for scoring. The Data Mining scoring functions can be used to apply models created with the PL/SQL API or with the Java API.

> **Note:** The SQL functions for Data Mining are documented in *Oracle Database SQL Language Reference*. Information about these functions is also provided in *Oracle Data Mining Application Developer's Guide*.

The programs that demonstrate the Data Mining functions are listed in Table 8–2.

Table 8–2   Data Mining SQL Functions in the Sample Programs

Program Name	Algorithm	SQL Functions Used
dmkmdemo.sql	*k*-Means	CLUSTER_ID, CLUSTER_PROBABILITY, CLUSTER_SET
dmocdemo.sql	O-Cluster	CLUSTER_ID
dmnmdemo.sql	NMF	FEATURE_ID, FEATURE_SET, FEATURE_VALUE
dmdtdemo.sql	Decision Tree	PREDICTION, PREDICTION_COST, PREDICTION_DETAILS, PREDICTION_SET
dmdtxvlddemo.sql	Decision Tree	PREDICTION
dmglcdem.sql	GLM classification	PREDICTION, PREDICTION_BOUNDS, PREDICTION_PROBABILITY,
dmglrdem.sql	GLM regression	PREDICTION, PREDICTION_BOUNDS
dmsvcdem.sql	SVM classification	PREDICTION, PREDICTION_PROBABILITY, PREDICTION_SET
dmsvodem.sql	One-Class SVM	PREDICTION, PREDICTION_PROBABILITY, PREDICTION_SET
dmsvrdem.sql	SVM regression	PREDICTION
dmtxtsvm.sql	Text mining	PREDICTION, PREDICTION_PROBABILITY

Java Programs

The Java demos illustrate the features of the Oracle Data Mining Java API, which implements Oracle-specific extensions to the Java Data Mining (JDM) 1.0.1.1 standard.The Java programs demonstrate data preprocessing and the basic mining functions. Additional Java samples demonstrate predictive analytics, import/export, and text mining.

> **See Also:** *Oracle Data Mining Java API Reference* (javadoc) and the *Oracle Data Mining Application Developer's Guide* for information on the Java API.

Java Program Summaries

Summary descriptions of the Java sample programs are provided in Table 8–3. For detailed descriptions, see the comments in the source code.

Table 8–3   Overview of the Java Sample Programs

Mining Function or Task	Description
Classification	The classification programs demonstrate various preprocessing techniques and perform the following steps: ■ Build a classification model using training data ■ Display model details and settings ■ Test the model by applying the model on the test data ■ Compute test metrics, such as confusion matrix, lift, and ROC ■ Apply the model on the scoring data ■ Present apply results ■ Present ranked apply results, influenced by a cost matrix The `dmapplydemo.java` program demonstrates several ways of applying a Naive Bayes model. `dmglcdemo.java` illustrates GLM classification (binary logistic regression)
Regression	`dmsvrdemo.java` uses different test metrics, but otherwise performs most of the same steps used in the classification programs. Selected attributes of the input data are preprocessed (normalized). `dmglrdemo.java` illustrates GLM regression (multivariate linear regression)
Association	`dmardemo.java` builds an association model and presents frequent itemsets and association rules as output. Selected attributes of the input data are preprocessed (binned).
Clustering	`dmkmdemo.java` (*k*-Means) and `dmocdemo.java` (0-Cluster) build clustering models and present cluster details, such as rules, centroid, and histogram for each cluster as output. The models are scored, and the probabilities associated with each cluster are returned as output. Selected attributes of the input data are preprocessed (normalized).
Feature extraction	`dmnmdemo.java` builds a feature extraction model and presents model details as the output. The model is scored, and each feature ID is associated with a probability. Selected attributes of the input data are preprocessed (normalized).
Attribute importance	`dmaidemo.java` builds an attribute importance model and presents a list of important attributes as the output of model details. Selected attributes of the input data are preprocessed (binned).
Data transformations	`dmxfdemo.java` demonstrates binning, clipping, and normalization transformations.
Predictive Analytics	`dmpademo.java` demonstrates `PREDICT`, `EXPLAIN,` and `PROFILE` functions.
Model import/export	`dmexpimpdemo.java` builds a Naive Bayes model, exports it to a dump file, then imports it from the dump file.

Text Mining Programs

Oracle Data Mining can mine text columns that have undergone pre-processing by Oracle Text routines.

Oracle Text is a technology for building text query and document classification applications. It provides indexing, word and theme searching, and viewing capabilities for text. Oracle Text is included in a general installation of Oracle Database Enterprise Edition, and therefore is already present in a database installed according to the instructions in Chapter 1.

The pre-processing steps for text mining create nested table columns of type DM_NESTED_NUMERICALS from columns of type VARCHAR2 or CLOB. Each row of the nested table specifies an attribute name and a value. The type definition is as follows.

```
CREATE OR REPLACE TYPE dm_nested_numerical AS OBJECT
  (attribute_name VARCHAR2(4000),
   value          NUMBER)
/
CREATE OR REPLACE TYPE dm_nested_numericals AS TABLE OF dm_nested_numerical
```

Terms extracted from text documents into nested tables can become generic attributes in training or scoring data.

Sample text mining programs in both PL/SQL and Java illustrate classification and feature extraction of a pre-processed text column.

Text Mining in PL/SQL

Three PL/SQL sample programs illustrate the process of text mining. One program illustrates the pre-processing that is required to prepare the data for mining. The other two programs build models that use the transformed text.

Text Transformation Demo

To prepare a column for text mining using the PL/SQL API, you must use Oracle Text routines to perform the following general steps:

1. Create a domain index on the column.

2. Use the index to extract terms from the column to a temporary table.

3. Populate a column of type DM_NESTED_NUMERICALS with the terms in the temporary table.

The process of term extraction using Oracle Text is illustrated in the sample program dmtxtfe.sql. The source code contains extensive comments that explain the steps involved in transforming text into a set of features that can be mined using Oracle Data Mining.

More details about text transformation are provided in the *Oracle Data Mining Application Developer's Guide*.

Text Transformation for the PL/SQL Text Mining Sample Programs

The dmsh.sql script performs the text transformation required by the PL/SQL text mining demos. There are two such sample programs: dmtxtnmf.sql, which builds a feature extraction model using Non-Negative Matrix Factorization, and dmtxtsvm.sql, which builds a classification model using Support Vector Machine. Both of these programs use the following tables, which have a nested table column of comment data:

```
MINING_BUILD_NESTED_TEXT
MINING_TEST_NESTED_TEXT
MINING_APPLY_NESTED_TEXT
```

The Sample Text Mining Models (PL/SQL)

You can run the PL/SQL text mining demo programs, `dmtxtnmf.sql` and `dmtxtsvm.sql`, like the other PL/SQL programs. The models created by these programs are listed in the following example.

```
SQL> @ %ORACLE_HOME%\RDBMS\demo\dmtxtnmf.sql
SQL> @ %ORACLE_HOME%\RDBMS\demo\dmtxtsvm.sql
SQL> SELECT model_name, mining_function, algorithm
         FROM user_mining_models;

MODEL_NAME         MINING_FUNCTION    ALGORITHM
----------------   -----------------  ------------------------
T_NMF_SAMPLE       FEATURE_EXTRACTION NONNEGATIVE_MATRIX_FACTOR
T_SVM_CLAS_SAMPLE  CLASSIFICATION     SUPPORT_VECTOR_MACHINES
```

Text Mining in Java

Two Java sample programs illustrate the process of text mining. One builds a feature extraction model, the other builds a classification model.

Text Transformation for the Sample Java Text Mining Programs

The Oracle Data Mining Java API provides an interface that handles the term extraction process. If you are developing data mining applications in Java, you do not need to use Oracle Text directly. However, you must ensure that Oracle Text is present in the database.

The `OraTextTransform` interface is used to perform text transformation within the Java text mining demos. There are two such sample programs: `dmtxtnmfdemo.java`, which builds a feature extraction model using Non-Negative Matrix Factorization, and `dmtxtsvmdemo.java`, which builds a classification model using Support Vector Machine. Both of these programs create build, test, and apply data sets from the following tables, which have a text column of comment data:

```
MINING_BUILD_TEXT
MINING_TEST_TEXT
MINING_APPLY_TEXT
```

The Sample Text Mining Models (Java)

You can run the Java text mining sample programs, `dmtxtnmfdemo.java` and `dmtxtsvmdemo.java`, like the other Java programs. The models created by these programs are shown in the following example.

```
java dmtxtnmfdemo host:port:SID dmuser password
java dmtxtsvmdemo host:port:SID dmuser password
SQLPLUS dmuser
Enter password: password
SQL> SELECT model_name, mining_function, algorithm
         FROM user_mining_models;

MODEL_NAME         MINING_FUNCTION    ALGORITHM
----------------   -----------------  ------------------------
txtnmfModel_jdm    FEATURE_EXTRACTION NONNEGATIVE_MATRIX_FACTOR
txtsvmModel_jdm    CLASSIFICATION     SUPPORT_VECTOR_MACHINES
```

The Sample Data

The `dmsh.sql` script creates views, tables, and indexes in the user's schema. The views define columns of customer data from tables in the `SH` schema. This data is used by the Data Mining sample programs. The tables reference the same columns in `SH`, but they include an extra `COMMENTS` column for text mining. The indexes are used to extract terms from the text in the `COMMENTS` column and build a nested table column.

Customer Data for Data Mining

Views in the data mining user's schema define columns of data from the `CUSTOMERS`, `SALES`, `PRODUCTS`, `COUNTRIES`, and `SUPPLEMENTARY_DEMOGRAPHICS` tables in the `SH` schema. You can list these views with the following SQL statements.

```
SQL>CONNECT dmuser
Enter password: password
SQL>SELECT view_name FROM user_views;
```

The views are listed in

Table 8–4   Views Used by the Data Mining Sample Programs

View Name	Description
`MINING_DATA_APPLY_STR_V`	Scoring data for o-cluster
`MINING_DATA_BUILD_STR_V`	Training data for o-cluster
`MINING_DATA_APPLY_V`	Scoring data for data mining (not text mining)
`MINING_DATA_BUILD_V`	Training data for data mining (not text mining)
`MINING_DATA_TEST_V`	Test data for data mining (not text mining)
`MARKET_BASKET_V`	Data for association rules
`MINING_DATA_ONE_CLASS_V`	Data for one-class SVM

You can see the references to tables in `SH` by listing the view definitions. The definition of the view `MINING_DATA_BUILD_V` is shown as follows.

```
SQL> set long 1000000
SQL> set longc 100000
SQL> set pagesize 100
SQL> SELECT text FROM all_views WHERE
    owner='dmuser3'AND view_name='mining_data_build_v';

    SELECT a.CUST_ID, a.CUST_GENDER, 2003-a.CUST_YEAR_OF_BIRTH AGE,
           a.CUST_MARITAL_STATUS, c.COUNTRY_NAME, a.CUST_INCOME_LEVEL,
           b.EDUCATION, b.OCCUPATION, b.HOUSEHOLD_SIZE, b.YRS_RESIDENCE,
           b.AFFINITY_CARD, b.BULK_PACK_DISKETTES, b.FLAT_PANEL_MONITOR,
           b.HOME_THEATER_PACKAGE, b.BOOKKEEPING_APPLICATION,
           b.PRINTER_SUPPLIES, b.Y_BOX_GAMES, b.OS_DOC_SET_KANJI
      FROM  sh.customers a,
            sh.supplementary_demographics b,
            sh.countries c
     WHERE a.CUST_ID = b.CUST_ID AND a.country_id = c.country_id
           AND a.cust_id between 101501 and 103000
```

The views are used to build, test, and score the sample models. Each view has a `CUSTOMER_ID` column, which is the case ID, and an `AFFINITY_CARD` column, which is the target used by the predictive models. Most of the views provide data for 1500

customers (1500 rows). The view used by the One-Class SVM model has data for 940 customers.

The columns of training data in the `MINING_DATA_BUILD_V` view are listed in the following example.

```
SQL> DESCRIBE mining_data_build_v

CUST_ID                      NOT NULL       NUMBER
CUST_GENDER                  NOT NULL       CHAR(1)
AGE                                         NUMBER
CUST_MARITAL_STATUS                         VARCHAR2(20)
COUNTRY_NAME                 NOT NULL       VARCHAR2(40)
CUST_INCOME_LEVEL                           VARCHAR2(30)
EDUCATION                                   VARCHAR2(21)
OCCUPATION                                  VARCHAR2(21)
HOUSEHOLD_SIZE                              VARCHAR2(21)
YRS_RESIDENCE                               NUMBER
AFFINITY_CARD                               NUMBER(10)
BULK_PACK_DISKETTES                         NUMBER(10)
FLAT_PANEL_MONITOR                          NUMBER(10)
HOME_THEATER_PACKAGE                        NUMBER(10)
BOOKKEEPING_APPLICATION                     NUMBER(10)
PRINTER_SUPPLIES                            NUMBER(10)
Y_BOX_GAMES                                 NUMBER(10)
OS_DOC_SET_KANJI                            NUMBER(10)
```

Market Basket Data for Association Rules

The association demos use the `MARKET_BASKET_V` data set, which includes columns of products from the `PRODUCTS` table and the `CUSTOMER_ID` column from the `CUSTOMERS` table in `SH`. The columns of the `MARKET_BASKET_V` view are listed in the following example.

```
SQL> DESCRIBE market_basket_v

CUST_ID                       NOT NULL      NUMBER
EXTENSION_CABLE                             NUMBER
FLAT_PANEL_MONITOR                          NUMBER
CD_RW_HIGH_SPEED_5_PACK                     NUMBER
ENVOY_256MB_40GB                            NUMBER
ENVOY_AMBASSADOR                            NUMBER
EXTERNAL_8X_CD_ROM                          NUMBER
KEYBOARD_WRIST_REST                         NUMBER
SM26273_BLACK_INK_CARTRIDGE                 NUMBER
MOUSE_PAD                                   NUMBER
MULTIMEDIA_SPEAKERS_3INCH                   NUMBER
OS_DOC_SET_ENGLISH                          NUMBER
SIMM_16MB_PCMCIAII_CARD                     NUMBER
STANDARD_MOUSE                              NUMBER
```

Customer Data for Text Mining

The text mining demos use the same customer data from tables in `SH`, but they include either an extra text column or a collection type column. The collection type is a nested table of type `DM_NESTED_NUMERICALS`.

You can list these tables with the following SQL statements.

```
SQL>CONNECT dmuser
Enter password: password
```

```
SQL>SELECT table_name FROM user_tables WHERE table_name LIKE '%MINING%';
```

The text mining tables are listed in Table 8–5.

Table 8–5   Tables Used by the Text Mining Sample Programs

Table Name	Description
MINING_APPLY_NESTED_TEXT	Apply table with COMMENTS column as DM_NESTED_NUMERICALS
MINING_BUILD_NESTED_TEXT	Build table with COMMENTS column as DM_NESTED_NUMERICALS
MINING_TEST_NESTED_TEXT	Test table with COMMENTS column as DM_NESTED_NUMERICALS
MINING_APPLY_TEXT	Apply table with COMMENTS column as VARCHAR2(4000)
MINING_BUILD_TEXT	Build table with COMMENTS column as VARCHAR2(4000)
MINING_TEST_TEXT	Test table with COMMENTS column as VARCHAR2(4000)

In the MINING_BUILD_TEXT, MINING_TEST_TEXT, and MINING_APPLY_TEXT tables, the COMMENTS column is of type VARCHAR2(4000).

```
SQL> DESCRIBE mining_build_text
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUST_ID                                   NOT NULL NUMBER
 CUST_GENDER                               NOT NULL CHAR(1)
 AGE                                                NUMBER
 CUST_MARITAL_STATUS                                VARCHAR2(20)
 COUNTRY_NAME                              NOT NULL VARCHAR2(40)
 CUST_INCOME_LEVEL                                  VARCHAR2(30)
 EDUCATION                                          VARCHAR2(21)
 OCCUPATION                                         VARCHAR2(21)
 HOUSEHOLD_SIZE                                     VARCHAR2(21)
 YRS_RESIDENCE                                      NUMBER
 AFFINITY_CARD                                      NUMBER(10)
 BULK_PACK_DISKETTES                                NUMBER(10)
 FLAT_PANEL_MONITOR                                 NUMBER(10)
 HOME_THEATER_PACKAGE                               NUMBER(10)
 BOOKKEEPING_APPLICATION                            NUMBER(10)
 PRINTER_SUPPLIES                                   NUMBER(10)
 Y_BOX_GAMES                                        NUMBER(10)
 OS_DOC_SET_KANJI                                   NUMBER(10)
 COMMENTS                                           VARCHAR2(4000)
```

In the MINING_*_NESTED_TEXT tables, the COMMENTS column is of type DM_NESTED_NUMERICALS.

```
SQL> DESCRIBE mining_build_nested_text
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUST_ID                                   NOT NULL NUMBER
 CUST_GENDER                               NOT NULL CHAR(1)
 AGE                                                NUMBER
 CUST_MARITAL_STATUS                                VARCHAR2(20)
 COUNTRY_NAME                              NOT NULL VARCHAR2(40)
 CUST_INCOME_LEVEL                                  VARCHAR2(30)
 EDUCATION                                          VARCHAR2(21)
 OCCUPATION                                         VARCHAR2(21)
 HOUSEHOLD_SIZE                                     VARCHAR2(21)
 YRS_RESIDENCE                                      NUMBER
 AFFINITY_CARD                                      NUMBER(10)
```

```
BULK_PACK_DISKETTES                             NUMBER(10)
FLAT_PANEL_MONITOR                              NUMBER(10)
HOME_THEATER_PACKAGE                            NUMBER(10)
BOOKKEEPING_APPLICATION                         NUMBER(10)
PRINTER_SUPPLIES                                NUMBER(10)
Y_BOX_GAMES                                     NUMBER(10)
OS_DOC_SET_KANJI                                NUMBER(10)
COMMENTS                                        DM_NESTED_NUMERICALS
```

The process of extracting terms from a text column into a nested table column is described in "Text Mining Programs" on page 8-4 and in *Oracle Data Mining Application Developer's Guide*.

Index

S

sample programs,　1-6, 2-11, 8-1, 8-3
　　data used by,　8-7
　　Java,　1-11, 1-12, 8-3
　　PL/SQL,　1-10, 1-12, 8-1
sample schemas,　1-1, 2-11
scoping,　xii
scoring,　viii, 8-2
Scoring Engine,　xii
security,　xii, 4-1
SELECT ANY MINING MODEL,　4-2
SELECT MINING MODEL,　4-3
settings,　7-3
SGA_TARGET,　viii
SH schema,　2-11, 8-7
shared cursor,　viii
sparse data,　xii
Spreadsheet Add-In
　　See Oracle Spreadsheet Add-In for Predictive
　　　　Analytics
SQL AUDIT,　7-4
SQL COMMENT,　7-3
SQL Data Mining functions,　8-2
SQL Developer,　2-2
SQL*Plus,　2-2
Support Vector Machine,　1-11, 8-3
SVM
　　See Also One-Class SVM
　　See Support Vector Machine
SYS,　xi, 3-1, 3-2
system privileges,　4-2

T

text mining,　1-11, 5-1, 8-3, 8-4, 8-8
transformations,　1-11
tuning,　viii, 5-1

U

upgrading,　3-1
user ID,　1-9, 2-1, 2-11, 4-1, 5-2, 6-1, 7-6
USER_AUDIT_OBJECT,　7-5
USER_MINING_MODEL_ATTRIBUTES,　7-1
USER_MINING_MODEL_SETTINGS,　7-1
USER_MINING_MODELS,　1-13, 7-1
USER_OBJ_AUDIT_OPTS,　7-5
USER_OBJECTS,　xii
users,　7-6