

# Oracle9i Data Mining

Administrator's Guide

Release 2 (9.2)

March 2002

Part No. A95959-01

## 1 Introduction

This document describes how to install the Oracle9i Data Mining (ODM) software and how to perform other administrative functions common to all ODM administration on both UNIX and Windows platforms.

### 1.1 Intended Audience

This administrator's guide is intended for anyone planning to install and run Oracle9i Data Mining — either a database administrator or a system administrator.

### 1.2 Structure

This guide contains the following sections:

- **Section 2, "Overview"**: Briefly describes Oracle9i Data Mining release 2.
- **Section 3, "Oracle9i Data Mining Installation"**: Describes the generic installation steps (platform-specific information is in the platform-specific release notes).
- **Section 4, "Oracle9i Data Mining Administration"**: Describes topics of interest to administrators, including improving ODM performance, starting and stopping the ODM task monitor, detecting errors, etc.

### 1.3 Where to Find Further Information

The documentation set for Oracle9i Data Mining is part of the *Oracle9i Database Documentation Library*; the ODM document set consists of the following documents:

**ORACLE®**

Copyright © 2002, Oracle Corporation.  
All Rights Reserved.

Oracle is a registered trademark, and Oracle9i, SQL\*Net, and SQL\*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

- *Oracle9i Data Mining Administrator's Guide*, Release 2 (9.2) (this document, which includes installation instructions that are the same across all platforms).
- *Oracle9i Data Mining Concepts*, Release 2 (9.2).

For detailed information about the ODM API, see the ODM Javadoc in the directory `$ORACLE_HOME/dm/doc` on any system where ODM is installed. To prepare the Javadoc for user access, see [Section 4.3](#).

### 1.3.1 Related Manuals

For more information about the Oracle9i database, see:

- *Oracle9i Administrator's Guide*, Release 2 (9.2)
- Release notes for your platform.
- *Oracle Universal Installer Concepts Guide*
- *Oracle9i Database Migration*

## 1.4 Conventions

In this manual, Windows refers to the Windows NT, Windows 2000, and Windows XP operating systems.

The SQL interface to Oracle9i is referred to as SQL. This interface is the Oracle9i implementation of the SQL standard ANSI X3.135-1992, ISO 9075:1992, commonly referred to as the ANSI/ISO SQL standard or SQL92. In examples, an implied carriage return occurs at the end of each line, unless otherwise noted. You must press the Return key at the end of a line of input.

## 2 Overview

Oracle9i Data Mining (ODM) allows application programmers to perform data mining in the Oracle9i database. All model-building and scoring functions are accessible through a Java-based API. The Oracle9i database provides the infrastructure for application developers to build integrated applications, with complete programmatic control of data mining functions to deliver data mining within the database.

ODM release 2 has many new features; for details, see *Oracle9i Data Mining Concepts*.

ODM release 2 runs on Real Application Clusters.

## 3 Oracle9i Data Mining Installation

This section contains generic ODM requirements, an installation overview, and detailed installation steps.

### 3.1 ODM Requirements

ODM is an option to Oracle9i Enterprise Edition.

ODM has the following general software requirements:

- The ODM API requires Java 1.3.1\_01
- ODM uses the `java.sql` package that is included in JDK 1.2 known as the JDBC 2.0 API

#### 3.1.1 ODM on a Real Application Cluster (RAC)

If you plan to run ODM on a RAC, the ODM tablespace must be at least 250MB. (In a RAC environment, tablespaces are built on a raw device; they are not growable as on UFS.)

### 3.2 Installation Overview

This document provides the generic instructions for installing Oracle9i Data Mining.

Before you install ODM, confirm that your system satisfies the software and hardware requirements for Oracle9i Enterprise Edition, as described in the release notes for your platform. You should also ensure that your system contains enough space for the tables that you plan to use during data mining.

There are three common cases for installing ODM:

- Oracle9i and ODM are not installed on your system ([Section 3.2.1](#)).
- Oracle9i release 1 (or earlier) is installed on your system ([Section 3.2.2](#))
- Oracle9i release 2 is installed on your system ([Section 3.2.2](#))

To install ODM on an Oracle9i Real Application Cluster, see [Section 3.3](#).

#### 3.2.1 Oracle9i Not Installed

If you are installing ODM on a system where the database is not installed, there are two basic ways to install Oracle9i Enterprise Edition:

1. Create a database using the seed database ([Section 3.2.1.1](#)).
2. Create a database but do not use the seed database, that is, create a custom database ([Section 3.2.1.2](#)).

**3.2.1.1 ODM Installation With a Seed Database** Oracle provides a seed or preconfigured database that automatically includes features that result in a highly effective and easier to manage database.

Follow these steps to install Oracle9i and ODM:

1. Start Oracle Universal Installer (OUI). For details, see the *Oracle Universal Installer Concepts Guide*.
2. Select the Enterprise Edition of Oracle9i. Select "Create General Purpose Database". If you do not wish to create a general purpose database, see [Section 3.2.1.2](#) for information about installing ODM with a custom database.
3. When the installation completes successfully, unlock the ODM and ODM\_MTR accounts and change the default passwords using the following commands:

```
alter user odm identified by new_ODM_password account unlock;  
alter user odm_mtr identified by new_MTR_password account unlock;
```

where *new\_ODM\_password* is the new password for the ODM account and *new\_MTR\_password* is the new password for the ODM\_MTR account.

4. Log in to the database as ODM user and start the ODM Monitor with the following SQL\*Plus command:

```
exec DM_START_MONITOR
```

After successful installation, all ODM software is located in the \$ORACLE\_HOME/dm directory.

**3.2.1.2 ODM Installation With a Custom Database** Creating a custom database takes longer than creating a seed one, but gives you full control to specify and change all database parameters.

These are the major steps required to install ODM without using a seed database:

1. Install Oracle9i Enterprise Edition and create a custom database. See [Section 3.2.3](#) for information about database parameters required by ODM.
2. Run the Oracle Database Configuration Assistant (DBCA) tool to install the ODM option. DBCA is described in the *Oracle9i Database Administrator's Guide*.

Follow these steps to install ODM after you have created your custom database:

1. Start DBCA. Pick the ODM option.

2. Log in to the database as ODM user and start the ODM Monitor with the following SQL\*Plus command:

```
exec DM_START_MONITOR
```

After successful installation, all ODM software is located in the \$ORACLE\_HOME/dm directory.

### 3.2.2 Oracle9i or Earlier Installed

If Oracle9i release 1 or earlier is installed on your system, follow these steps:

1. Upgrade the database to Oracle9i Release 2. If ODM 9.0.1 is installed on your system, it is upgraded to ODM release 2 when the database is upgraded.
2. If ODM is not installed on your system, install and configure ODM release 2 as described in [Section 3.2.1](#).

For information about upgrading the database, see *Oracle9i Database Migration*. For information about upgrading ODM, see [Section 3.4](#).

### 3.2.3 ODM Database Parameters

The default value of initial parameters in an Oracle preconfigured (seed) database is generally sufficient for running ODM. If you are creating a custom database, the following parameter settings can be used as a general guideline for the database. The ODM-specific parameters must be set exactly as shown. We recommend that you tune other parameters based on your hardware resource capacity, volume of your input datasets, and other characteristics of your system:

```
#####
# Cache and I/O
#####
db_block_size=8192
db_cache_size=67108864
db_file_multiblock_read_count=16

#####
# Cursors and Library Cache
#####
open_cursors=300

#####
# Miscellaneous
#####
compatible=9.2.0.0.0
```

```

#####
# Pools
#####
java_pool_size=67108864
large_pool_size=10M
shared_pool_size=67108864

#####
# Optimizer
#####
hash_join_enabled=TRUE

#####
# Processes and Sessions
#####
processes=150

#####
# Sort, Hash Joins, Bitmap Indexes
#####
sort_area_size=5242880
sort_area_retained_size=2097152

#####
# ODM Specific
#####
aq_tm_processes = 1
job_queue_processes =10

#####
# Parallel setting, adjust according to CPU number
#####
parallel_max_servers = 32
parallel_min_servers = 2

```

### 3.3 ODM Installation on a Real Application Cluster

ODM installation on a Real Application Cluster (RAC) is similar to ODM installation on a non-RAC system. If you use Oracle Universal Installer to create the preconfigured database on RAC, ODM will be installed in this database just as it is in a non-RAC environment.

If you choose to create a custom database on your RAC and install ODM there, we recommend that you configure the ODM tablespace with a raw device partition of at least 250 MB.

## 3.4 Upgrading ODM

ODM upgrade is part of the Oracle database upgrade process. If your ODM 9.0.1 release is installed in an Oracle9i release 1 database and your ODM schema name is ODM, the Database Upgrade Assistant (DBUA) will upgrade ODM from 9.0.1 to 9.2.0 at the same time that the database is upgraded. All ODM 9.2.0 related files will be located in the \$ORACLE\_HOME/dm directory. When the database upgrade completes, ODM will be at the 9.2.0 release level.

---

---

**Note:** For release 2 of ODM, ODM upgrade has the following restrictions:

- You cannot downgrade from ODM release 2.
  - User-created bin boundaries tables are not upgraded.
- 
- 

For detailed information about upgrading an Oracle database, see the *Oracle9i Database Migration* manual.

The ODM upgrade has two parts:

- ODM schema upgrade
- ODM Java object upgrade

### 3.4.1 ODM Schema Upgrade

When you upgrade ODM from release 1 to release 2, the ODM schema repository is upgraded from 9.0.1 to 9.2.0. There are many changes in the ODM schema definition for 9.2.0.

After the upgrade, all ODM 9.0.1 models will continue to function in the ODM 9.2.0 environment. Association Rules and Naive Bayes models are fully migrated to ODM 9.2.0.

The database upgrade utility calls ODM upgrade scripts during the database upgrade process; the ODM scripts perform the following actions:

- Upgrade ODM 9.0.1 table definition and migrate 9.0.1 ODM model representation to 9.2.0 ODM models
- Remove ODM 9.0.1 tables and dependent objects
- Remove ODM 9.0.1 Java objects
- Upgrade all Java objects to 9.2.0
- Upgrade ODM 9.0.1 Java objects resided in ODM schema (BLOB)
- Install ODM 9.2.0 new schema objects

- Install ODM\_MTR 9.2.0 schema objects

All upgrade scripts are in the `$ORACLE_HOME/dm/admin` directory. The top-level upgrade script for ODM is

```
$ORACLE_HOME/rdbms/admin/odmdbmig.sql
```

### 3.4.2 ODM Java Object Upgrade

All mining models recorded in schema will be migrated to 9.2.0 format; that is, all Naive Bayes and Association Rules models are upgraded.

You cannot downgrade ODM Java objects at this release.

## 3.5 Deinstalling ODM

If you want to preserve existing mining models, you should not deinstall ODM. To install a newer version of ODM, upgrade as described in *Oracle9i Database Migration* and in [Section 3.4](#).

If you wish to deinstall ODM, you should stop the ODM Task Monitor as described in [Section 4.8](#). Then you can deinstall ODM using OUI, just as you would any other database component.

## 4 Oracle9i Data Mining Administration

This section contains information of interest to ODM administrators.

For information about administering an Oracle9i database, see the *Oracle9i Database Administrator's Guide*.

### 4.1 Improving ODM Performance

To improve ODM performance, enable parallelism by setting the database initialization parameters `PARALLEL_MAX_SERVERS` and `PARALLEL_MIN_SERVERS` based on the characteristics of your system.

### 4.2 Changing Default ODM Passwords

You should change the ODM default passwords after installation completes. You change ODM passwords just as you change any other database passwords.

### 4.3 ODM API Documentation

Documentation for the ODM API, created using Javadoc, is in the file `$ORACLE_HOME/dm/doc/odmjdoc.tar`. You should untar this file so that users can display it in a browser.



## 4.4 ODM Configuration Parameters

The following ODM configuration parameters reside in the ODM\_CONFIGURATION table. These parameters may require modification for your environment.

### **ABN\_ALG\_SETTING\_NF\_DEPTH**

Data type is int; default is 10. Specifies the maximum depth of any Network Feature for ABN setting.

### **ABN\_ALG\_SETTING\_NUM\_NF**

Data type is int; default is 10. Specifies the maximum number of Network Features for ABN setting.

### **ABN\_ALG\_SETTING\_NUM\_PRUNED\_NF**

Data type is int; default is 5. Specifies maximum number of consecutive pruned Network Features for ABN setting.

### **AI\_BUILD\_SEQ\_PER\_PARTITION**

Data type is int; default is 50000.

### **AUTO\_BIN\_CATEGORICAL\_NUM**

Data type is int; default is 5. Specifies the number of bins used by automated binning for categorical attributes. This value should be  $\geq 2$ .

### **AUTO\_BIN\_CATEGORICAL\_OTHER**

Data type is STRING; default is OTHER. Specifies the name of the "Other" bin generated during Top- $n$  categorical binning.

### **AUTO\_BIN\_CL\_NUMERICAL\_NUM**

Data type is int; default is 100. Specifies the maximum number of bins allowed for numerical attributes for clustering. Useful values are between 2 and 100. This parameter is used in conjunction with CL\_ALG\_SETTING\_KM\_BIN\_FACTOR and CL\_ALG\_SETTING\_OC\_BIN\_FACTOR.

### **AUTO\_BIN\_NUMERICAL\_NUM**

Data type is int; default is 5. Specifies the number of bins used by automated binning for numerical attributes. This value should be  $\geq 2$ .

### **CLASSIFICATION\_APPLY\_SEQ\_PER\_PARTITION**

Data type is int; default is 50000. Specifies the maximum number of unique sequence IDs per partition used by clustering apply.

### **CLASSIFICATION\_BUILD\_SEQ\_PER\_PARTITION**

Data type is int; default is 50000. Keeps the computations constrained to memory-sized chunks and determines the size of the random sample used for MDL computations (scoring within build). There is no maximum value; this value should not be smaller than 1000.

### **CLUSTERING\_APPLY\_SEQ\_PER\_PARTITION**

Data type is int; default is 50000. Constrains the scoring to memory-sized chunks of the data and loop through such chunks. The value for this parameter depends on the sorting area (SA) and the number of clusters. The larger the SA, the larger this parameter can be. A rough formula for this parameter is

$$\text{CLUSTERING\_APPLY\_SEQ\_PER\_PARTITION} = \text{SA} / (100 * \text{Num\_Clusters})$$

where "SA" is the size of the sorting area and "Num\_Clusters" is the number of clusters.

### **CL\_ALG\_SETTING\_CHI2\_LOW**

Data type is NUMBER; default is 1.353. Controls the level of statistical significance for O-Cluster to determine if more data is necessary to refine a model.

### **CL\_ALG\_SETTING\_KM\_BIN\_FACTOR**

Data type is NUMBER; default is 2. Factor used in automatic bin number computation for the *k*-means algorithm. Increasing this value will increase resolution by increasing the number of bins. However, the number of bins is also capped by AUTO\_BIN\_CL\_NUMERICAL\_NUM.

### **CL\_ALG\_SETTING\_KM\_BUFFER**

Data type is int; default is 10000. Number of rows used by the in-memory buffer used by *k*-means. For an installation with limited memory, this number should be smaller than the default data size. Summarization is activated for datasets larger than the buffer size.

### **CL\_ALG\_SETTING\_KM\_FACTOR**

Data type is NUMBER; default is 20. Controls the number of points produced by data summarization for *k*-means. The larger the value, the more points. The formula for the number of points is:

$$\text{Number of Points} = \text{CL\_ALG\_SETTING\_KM\_FACTOR} * \text{Num\_Attributes} * \text{Num\_Clusters}$$

where "Num\_Attributes" is the number of attributes and "Num\_Clusters" is the number of clusters.

The number of points must be  $\leq 1000$ . This parameter can be any positive value; however, a small number of summarization points can produce poor accuracy.

### **CL\_ALG\_SETTING\_MIN\_CHI2\_POINTS**

Data type is int; default is 10. Controls the minimum number of rows required by O-Cluster to find a cluster. For data tables with a very small number of rows, this number should be set to a value between 2 and 10.

### **CL\_ALG\_SETTING\_OC\_BIN\_FACTOR**

Data type is NUMBER; default is 0.9. Factor used in automatic bin number computation for the O-Cluster algorithm. Increasing this value will increase the number of bins. However, increasing the number of bins may have a negative effect on the statistical significance of the model.

### **CL\_ALG\_SETTING\_OC\_BUFFER**

Data type is int; default is 50000. Specifies the number of rows used by the in-memory buffer used by O-Cluster. For an installation with limited memory, this number should be smaller than the default size.

### **CL\_ALG\_SETTING\_TREE\_GROWTH**

Data type is int; default is 1. Must be 1 or 2. 1 specifies that *k*-means is a balanced tree; 2 specifies that *k*-means is an unbalanced tree.

### **LOG\_LEVEL**

Data type is int; default is 2. Must be 0, 1, 2, or 3. Specifies the type of messages written to the LOG\_FILE. 0 means no logging; 1 means write Internal Error, Error, and Warning messages to LOG\_FILE; 2 means write all messages for 1 plus Notifications; 3 means write all messages for 2 plus trace information.

### **ODM\_CLIENT\_TRACE**

Data type is int; default is 0. Must be 0, 1, 2, or 3. Enables trace for the ODM client. 0 indicates no trace; 1 indicates low; 2 indicates moderate; 3 indicates high.

### **ODM\_SCHEMA**

Data type is STRING; default is ODM. Specifies the owner of the ODM schema repository.

### **ODM\_SERVER\_JAVA\_TRACE**

Data type is int; default is 0. Must be 0, 1, 2, or 3. Enables trace for the ODM client. 0 indicates no trace; 1 indicates low; 2 indicates moderate; 3 indicates high.

### **ODM\_SERVER\_SQL\_TRACE**

Data type is int; default is 0. Must be 0, 1, 2, or 3. Enables trace for the ODM client. 0 indicates no trace; 1 indicates low; 2 indicates moderate; 3 indicates high.

## **4.5 Increasing the Speed of ODM Clustering**

You can speed up clustering package (dmguh, dmguh, dmkmh, dmkmh, dmoch, dmoch) procedures by compiling them into native code residing in shared libraries. The procedures are translated into C code, then compiled with your usual C compiler and linked into the Oracle process. For details

on how to compile PL/SQL procedures into native code, see the *PL/SQL User's Guide and Reference*.

## 4.6 Verifying ODM Installation

Oracle9i Data Mining is an option to the Oracle9i Enterprise Edition. If ODM is part of your installation, the following query should return a value

TRUE:

```
SELECT value
FROM v$option
WHERE parameter = 'Oracle Data Mining';
```

This query is usually run by the DBA logged in as system/manager.

## 4.7 Need for Compatible Character Sets

When using ODM in a shared JVM environment, such as when integrated with a servlet application, all connections made to an ODM server (also known as a DMS) must be based on databases with compatible character sets. Otherwise, string length tests conducted in the JVM may not recognize these differences, allowing data to pass to the database, which could result in server-side failures.

## 4.8 Starting and Stopping the ODM Task Monitor

To start ODM, log in as ODM user and start the ODM Monitor with the following SQL\*Plus command:

```
exec DM_START_MONITOR
```

To stop ODM, log in as ODM user and stop the ODM Monitor with the following SQL\*Plus command:

```
exec DM_STOP_MONITOR
```

## 4.9 ODM Errors

Executing an ODM method results in the execution of a PL/SQL program in an Oracle9i database. Errors can occur at the Java level or at the PL/SQL level. There are two error tables that you should consult when errors occur in programs that use ODM classes and methods:

- ODM\_MESSAGE\_LOG for errors captured at the Java level
- ODM\_ERROR\_TABLE for errors captured at the PL/SQL level

## 5 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. Standards will continue to evolve over time, and Oracle Corporation 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 additional information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

### 5.1 Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, 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, JAWS may not always read a line of text that consists solely of a bracket or brace.

