Oracle9*i*AS Personalization

Administrator's Guide

Release 2 (v9.0.2)

May 2002
Part No. A95243-03

ORACLE®

Oracle9*i*AS Personalization Administrator's Guide, Release 2 (v9.0.2)

Part No. A95243-03

Contents

1 OP Administration Overview

2 OP Security and Internationalization

3 Installation Notes

4 OP Schemas

5 Configuring Oracle9*i*AS Personalization

6 JSP Tag Configuration

7 Initial Data Collection

8 Stopping and Starting OP

Index

Send Us Your Comments

Oracle9*i*AS Personalization Administrator's Guide, Release 2 (v9.0.2)

Part No. A95243-03

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- DARWINDOC@us.oracle.com
- FAX: 781-238-9893 Attn: Oracle9*i*AS Personalization Documentation
- Postal service:
  Oracle Corporation
  Oracle9*i*AS Personalization Documentation
  10 Van de Graaff Drive
  Burlington, Massachusetts 01803
  U.S.A.

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This manual describes the installation and administrative functions that need to be performed for Oracle9*i*AS Personalization (OP). See the release notes for platform-specific details and any late-breaking information.

Intended Audience

This manual is intended for anyone planning to install and run Oracle9*i*AS Personalization— either a database administrator or a system administrator.

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/`.

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.

Structure

This manual contains the following chapters:

Where to Find More Information

Documentation for Oracle9*i*AS Personalization at the current release consists of the following documents:

- *Oracle9iAS Personalization Release Notes,* Release 9.0.2 (contains platform-specific information, a bug report, and information about any late-breaking changes).

- *Oracle9iAS Personalization User's Guide,* Release 9.0.2.

- *Oracle9iAS Personalization Administrator's Guide,* Release 9.0.2 (this document).

- *Oracle9iAS Personalization Recommendation Engine API Programmer's Guide,* Release 9.0.2. A programmer's manual for programming the recommendation engines in real time and for obtaining bulk recommendations. The API classes and methods are also described in Javadoc.

Related Manuals

Oracle9*i*AS Personalization documentation is a component of the Oracle9*i* Application Server Documentation Library. See especially:

- *Oracle9i Application Server*

- *Oracle9i Administrator's Guide*

- *Oracle9i Application Server Installation Guide* (the appropriate version for your operating system).

Documentation Formats

Documentation for Oracle9*i*AS Personalization is provided in PDF and HTML formats.

To view the PDF files, you will need

- Adobe Acrobat Reader 3.0 or later, which you can download from `http://www.adobe.com`.

To view the HTML files, you will need

- Netscape 4.x or later, or

- Internet Explorer 4.x or later

Online Help

The OP Administrative UI includes extensive online help that can be summoned from a list of contents and from Help buttons.

Conventions

In this manual, Windows refers to the Windows 95, Windows 98, Windows NT, and the Windows XP operating systems.

The SQL interface to Oracle9*i* is referred to as SQL. This interface is the Oracle9*i* 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.

The table below shows the conventions followed in this manual and their meanings:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
boldface	Commands, menu names, menu items, names of dialogs.
italics	Argument names and placeholders in command formats

Convention	Meaning
< >	Angle brackets enclose user-supplied names.
[ ]	Brackets enclose optional clauses from which you can choose one or none.
% **user input** system output	In interactive examples, user input is shown in bold typewriter, and system output is shown in regular typewriter.

1

OP Administration Overview

Oracle9*i*AS Personalization (OP) permits Web applications to collect and store customer data in an Oracle database, build models, and make recommendations on the basis of the models. For more information about OP and how it works, see the *Oracle 9iAS Personalization User's Guide*.

This manual provides the generic instructions for administering Oracle9*i*AS Personalization and provides notes regarding installation. For more information about installing OP, see the *Oracle9i Application Server Installation Guide* and *Oracle9iAS Personalization Release Notes.*

This manual is organized as follows:

- Overview (Chapter 1)
- Security and Internationalization (Chapter 2)
- Installation Notes (Chapter 3)
- OP Schemas (Chapter 4)
- OP Configuration (Chapter 5)
- JSP Tags (Chapter 6)
- Initial Data Collection (Chapter 7)
- Stopping and Starting OP (Chapter 8)

Changes in Functionality

OP release 9.0.2 includes the following changes in functionality that administrators need to know about:

- Changes in installation
- Changes to scheduling OP tasks

- Changes in configuration parameters
- Changes to report names

Installation

The changes in installation procedures are described in Chapter 3, "Installation Notes" and in the *Oracle9i Application Service Installation Guide.*

Scheduling OP Tasks

The way OP tasks are scheduled has changed. The first release of Oracle9*i*AS Personalization used Oracle Workflow to perform scheduling of package builds, package deployments, and reports. Workflow also provided email notifications when these events completed. The scheduling tasks and email notification now use a different underlying mechanism, but provide the same functionality. See "Configuring the OP Scheduler" in Chapter 5 for details about configuring scheduling in OP.

Changed Configuration Parameters

The following configuration parameter no longer exists:

- WF_SCHEMA in MOR_CONFIGURATION

The following configuration parameter is new for release 9.0.2:

- ADMIN_EMAIL_ADDRESS in MOR_CONFIGURATION to hold email address of the OP administrator.

Changed Report Names

The "Visitor to Customer Conversion Report" is now called the "Purchasing Session Report."

The "Recommendation Effectiveness Report" is still called the "Recommendation Effectiveness Report."

The "Most Active Cross-Sold Products Report" is now called the "Itemized Recommendation Effectiveness Repport."

2

OP Security and Internationalization

This chapter describes issues associated with OP security and internationalization.

OP Security

Oracle9*i*AS Personalization follows the Oracle Data Server security policy. OP implements the following measures:

- Any data that needs to be protected is encrypted using standard Oracle security tools.
- Transmission of sensitive data is encrypted between various OP components.
- The DBA grants access to qualified users.
- OP uses secure SSL technology to access Oracle HTTP Server.

User Access

The DBA is responsible for creating database users and setting up proper privileges for them so that they can access OP.

OP controls user access by providing two database roles:

- OP_ADMIN: This role allows the user to access the OP product as an OP Administrator. This means having access to the entire OP product, including the Message Viewer.
- OP_REPORT: This role allows the user to view OP reports and access the Report Viewer URL (`http://<hostname>/OP/Admin/ReportsPage`).

By default, the MOR schema owner (the user named MOR) is the OP administrator, and thus has the full privileges to access the OP product.

For new OP users, the DBA can grant either full privileges (OP_ADMIN) or more limited privileges (OP_REPORT, access only to the OP reports) as follows:

```
SQL> grant OP_ADMIN to <new_user>;

SQL> grant OP_REPORT to <new_user>;
```

For existing database users, the DBA must grant one of the two roles to allow the user access to OP product at the appropriate level.

Internationalization

Oracle9iAS Personalization is designed to support many languages and character sets. In addition, the OP UI, the online help for the UI, and all OP messages are translated by Oracle into several European and Asian languages.

When OP is installed, no default value is specified for NLS_LANGUAGE, an MOR configuration parameter. By default, OP uses the same character set as the database character set (for databases of native character sets).

You can specify a value in NLS_LANGUAGE. OP verifies that the value specified is supported and then uses it instead of the database default language for the server code. If neither this value nor the database default indicates one of the supported languages, OP defaults to AMERICAN. This value also specifies the language used for email notification.

Supported languages and their character sets are as follows:

NLS_LANGUAGE	HTML_CHARSET	DB_CHARSET
AMERICAN	ISO-8859-1	US7ASCII
FRENCH	ISO-8859-1	WE8IS08859P1
GERMAN	ISO-8859-1	WE8IS08859P1
ITALIAN	ISO-8859-1	WE8IS08859P1
SPANISH	ISO-8859-1	WE8IS08859P1
BRAZILIAN PORTUGESE	ISO-8859-1	WE8IS08859P1
JAPANESE	SHIFT_JIS	JA16EUC
KOREAN	EUC-KR	KO16KSC5601
SIMPLIFIED CHINESE	GB2312	ZHS16CGB231280

3

Installation Notes

This chapter presents information that will be helpful to anyone installing Oracle9*i*AS Personalization.

Background

The installation of Oracle9*i*AS Personalization has been redesigned to integrate into the new Oracle9*i*AS standards.

The installation is done in two parts. The first part is done by an Oracle9*i* Application Server administrator; the second part is to be done by the database administrator.

Part 1 Installation

The first part happens automatically as part of the installation of *i*AS2 Business Intelligence or Unified Messaging on your system, which copies the software onto Oracle9*i*AS `ORACLE_HOME/dmt` directory and configures the OP Admin UI servlet in OC4J. The Admin UI will be available at `https://localhost:4443/OP/Admin` once the *i*AS installation completes. You will be able to view the login screen, but login will fail with "Authentication Errors" until the required database schemas are configured by your DBA in the second part of the installation.

Note the following:

- Before installing OP, you must first install the Oracle9*i*AS Infrastructure database. If you are installing the Oracle9*i*AS Infrastructure database, you can check the default values during the installation.

- To install Oracle Personalization, you must select **Business Inteligence and Forms** as the type of *i*AS installion.

Part 2 Installation

The second part of the installation configures the required schemas in the Oracle9*i* database. The schemas are configured using the Oracle Personalization Configuration Wizard — a Java application — that creates these schemas over the network to your Oracle9*i* database.

Before Running the Personalization Schema Creation Wizard

The database administrator will need to allow remote SYS access for the duration of the schema creation process.

Documentation on how to update the database `init.ora` parameters can be found in the Database Administrator Authentication section of the *Oracle9i Database Administrator's Guide*, Release 1 (9.0.1), part number A90117-01 (`http://download.oracle.com/otndoc/oracle9i/901_ doc/server.901/a90117/dba.htm#10431`).

Remote SYS access can be tested using the following command on the system where you installed *i*AS Business Intelligence:

```
sqlplus "SYS/<password@GlobalDatabaseName> AS SYSDBA"
```

where `password` is the SYS password on the remote database and `GlobalDatabaseName` is the database alias in the `tnsnames.ora` file.

The DBA will need to determine the following before starting the Java application to configure the required database schemas.

Note the following: When you specify the data file names for the tablespaces, the names must be the fully qualified names for the data files on the local file system where the Oracle9*i* database is running. You must replace the <full path> prompt with the directory name for the data file.

MOR (Mining Object Repository)

1. SYS password for the database
2. JDBC connect information
   a. Host
   b. Port
   c. SID
3. Schema name for the MOR (default: OPMOR)

4. Password for the MOR

5. Fully qualified file name of the tablespace data file for the MOR

6. Fully qualified file name of the tempspace data file for the MOR

7. Email address for replies from notification messages

RE (Recommendation Engine)

1. SYS password for the database

2. JDBC connect information

    a. Host

    b. Port

    c. SID

3. Schema Name for the RE (default: RE)

4. Password for the RE

5. Fully qualified file name of the tablespace data file for the RE

MTR (Mining Table Repository)

1. SYS password for the database

2. JDBC connect information

    a. Host

    b. Port

    c. SID

3. Schema name for the MTR (default: MTR); importing the movie data forces the name DEMO_MTR

4. Password for the MTR

5. Fully qualified file name of the tablespace data file for the MTR

Running the Personalization Configuration Wizard

After collecting the required information, log in to the *i*AS system where you installed the Oracle9*i*AS Business Intelligence as the *oracle* account and run the following command:

   (UNIX) `$ORACLE_HOME/dmt/admin/opconfig.sh`

   (Windows) `%ORACLE_HOME%/dmt/admin/opconfig.bat`

After Running the Wizard

Once the schemas have been successfully created, remote SYS access can be safely disabled. However, it will need to be re-established if there is a need to configure more REs.

Data Collection Form

MOR

	Collected Data	Default
SYS password	******	change_on_install
JDBC connect info	Host Port SID	None
Schema name		OPMOR
Password		MORPW
Fully qualified file name of the schema tablespace data file		
Fully qualified file name of the schema tempspace data file		
Email address for replies from notification messages		Your.Email@domain.com

RE

	Collected Data	Default
SYS password	******	change_on_install
JDBC connect info	Host Port SID	None
Schema name for the RE		RE
Password for the RE		REPW

	Collected Data	Default
Fully qualified file name of the schema tablespace data file		None

MTR

	Collected Data	Default
SYS password	******	change_on_install
JDBC connect info	Host Port SID	None
Schema name for the MTR; importing the movie data forces the name DEMO_MTR		DEMO_MTR or MTR
Password for the MTR		MTRPW
Fully qualified file name of the schema tablespace data file		None

4

OP Schemas

Oracle9*i*AS Personalization (OP) uses several database schemas, as follows:

- **Mining Object Repository (MOR)**: The MOR controls the operation of OP and contains mining objects (deployable packages, reports, schedule items, etc., as well as package build results such as rule tables, etc.)

- **Mining Table Repository (MTR)**: Contains mining tables plus other important information such as bin boundaries, hot picks, taxonomy, etc.).

- **RE Schema**. The RE schema is where recommendations are held, where model package information is held, is where deployable packages are held in the Oracle9*i*AS server. RE schemas are tables that support the recommendation engine — they contain deployable packages used to generate real-time recommendations based on current and past session data.

All OP schemas reside on the system where Oracle9*i* is installed.

To see a small example of the MOR, MTR, and RE schemas that are correctly populated, install the REAPI Demo. (During installation, you are prompted with a question asking whether you want to install the demo data (movie data); if you answer yes, the demo data and the REAPI Demo are installed.) If you installed the REAPI Demo, you can examine the tables there. Alternatively, you can install an unpopulated MTR when you install OP. You can examine the schema of the unpopulated MTR and populate it with your own data.

Note that OP uses a fixed schema for the MTR. By "fixed," we mean that the MTR must be populated with tables matching OP table and column names.

Before you can obtain recommendations, you must create and deploy a package. You cannot create a package until there is some data available in the MTR. There are two ways to populate an MTR:

- Collect data using the REProxyRT method `addItem` or `addItems`.

■ Populate with existing data, i.e., map existing historical data that was collected by your Web application and stored in an Oracle database.

Mining Table Repository

The OP MTR consists of the tables and views listed below.

```
MTR_ATTR_ID_BIN_BOUNDARY                       TABLE
MTR_ATTR_NAME_TO_ID_MAP                        TABLE
MTR_BIN_BOUNDARY                               TABLE
MTR_CATEGORY                                   TABLE
MTR_CONFIGURATION                              TABLE
MTR_CUSTOMER                                   TABLE
MTR_CUSTOMER_NAV_DETAIL                        TABLE
MTR_CUSTOMER_RATING_DETAIL                     TABLE
MTR_HOTPICK                                    TABLE
MTR_HOTPICK_GROUP                              TABLE
MTR_INTERNAL_CONFIGURATION                     TABLE
MTR_ITEM                                       TABLE
MTR_NAVIGATION_DETAIL                          VIEW
MTR_PROFILE_DATA                               VIEW
MTR_PROXY                                      TABLE
MTR_PURCHASING_DETAIL                          TABLE
MTR_RATING_DETAIL                              VIEW
MTR_RECOMMENDATION_DETAIL                      TABLE
MTR_SCHEMA_VERSION                             VIEW
MTR_SESSION                                    TABLE
MTR_TAXONOMY (optional)                        TABLE
MTR_TAXONOMY_CATEGORY  (optional)              TABLE
MTR_TAXONOMY_CATEGORY_ITEM  (optional)         TABLE
MTR_VISITOR_NAV_DETAIL                         TABLE
MTR_VISITOR_RATING_DETAIL                      TABLE
```

Certain tables that make up the OP MTR must be populated with data specific to your Web site in accordance with the MTR schema. Other tables, such as the tables associated with sessions and recommendations, are automatically populated by OP.

The rest of this section describes the schemas for the MTR tables; tables that you must populate are described in detail.

Item Table

The item table contains a list of all the individual items that a Web site deals with. When OP returns a recommendation, it returns the ID and ITEM_TYPE; the item table provides more information. The item table is usually mapped to the catalog tables in the site database. The schema for MTR_ITEM has four fields; they are listed below, in order, with their data types.

```
ID                    NUMBER PK
ITEM_TYPE             VARCHAR2(30) PK
LABEL                 VARCHAR2(150)
DESCRIPTION           VARCHAR2(4000)
```

Bin Boundaries

The model-building algorithms in Oracle9*i*AS Personalization require discrete data. All numerical data must be discretized (binned) before the data is used to build a model. In OP, binning is performed in a transformation step before model build. The value ranges for binning (the bin boundaries) must be specified in order for OP to bin the values.

In the current release of OP, the bin boundaries must be specified in the bin boundaries table.

Binning is performed by joining the input data and the bin boundaries table.

Categorical data should also be binned in situations where there are many distinct values. If you bin categorical data, bin boundaries must be specified as for numerical data

In summary, OP requires all numerical data to be binned, and high cardinality categories should also be binned.

When you create bins of numeric values, specify the bounds (upper and lower values) for each bin. When you create bins of categorical data, specify the items in each bin. To map several values to the same bin, use several records with the same bin numbering.

The table MTR_BIN_BOUNDARY has seven fields; they are listed below, in order, with their data types:

```
DATA_SOURCE_TYPE              NUMBER(3)
ITEM_TYPE                     VARCHAR2(30)
ATTRIBUTE_NAME                VARCHAR2(30)
LOWER_VALUE                   NUMBER
UPPER_VALUE                   NUMBER
STRING_VALUE                  VARCHAR2(60)
BIN_NUMBER                    NUMBER(15)
```

Examples of Specifying Bin Boundaries

The following examples illustrate how to specify bin boundaries.

Consider movie rating data on a scale of 1 - 5. Suppose that you want to bin ratings as follows:

- 1 and 2 are in bin number 1

- 3 is in bin number 2

- 4 and 5 are bin number 3

You should enter the following into the bin boundaries table:

```
(3, 'MOVIE', 'VALUE', 1, 2.1, NULL, 1),
(3, 'MOVIE', 'VALUE', 3, 3.1, NULL, 2),
(3, 'MOVIE', 'VALUE', 4, 5.1, NULL, 3).
```

The range of the bin includes all values that are greater than or equal to the lower value and strictly less than the upper value. The data source type for rating is 3 and string value is set to NULL for numeric data.

The following entries in a bin boundary table bin marital status, a categorical attribute:

```
(1, 'NONE', 'MARITAL_STATUS', NULL, NULL, 'Single', 1),
(1, 'NONE', 'MARITAL_STATUS', NULL, NULL, 'Divorced', 2),
(1, 'NONE', 'MARITAL_STATUS', NULL, NULL, 'Separated', 2),
(1, 'NONE', 'MARITAL_STATUS', NULL, NULL, 'Married', 3),
(1, 'NONE', 'MARITAL_STATUS', NULL, NULL, 'Widowed', 4)
```

The data source type is 1 and the item type is NONE for demographic data. Lower and upper values are NULL for categorical data.

Taxonomy

In OP, taxonomy refers to the structural organization of items in a company's catalog or site. Typically the catalog and/or the site has a hierarchical structure, with the most general category at the base (for example, "clothing"), and branching to increasingly specific categories (for example, from "clothing" to "shoes" to "sneakers" to "tennis shoes").

Items can belong to more than one category and to different levels of the structure (for example, "tennis shoes" can be a category in "clothing" and also a category in "sports equipment").

The structure of the OP taxonomy is a DAG (direct acyclic graph), which can contain multiple top-level nodes. The different portions of the taxonomy can be disconnected too. Any node can connect to any other node but there cannot be a path that connects a node's child back to the node itself.

OP also supports multiple taxonomies (different ways of organizing the items).

The taxonomy is implemented using a group of tables; they are specified by the customer at installation time:

- MTR_TAXONOMY: Lists the different taxonomies used by the site. The schema for this table has three fields; they are listed below, in order, with their data types:

```
ID                 NUMBER PK
NAME               VARCHAR2(150)
DESCRIPTION        VARCHAR2(4000)
```

- MTR_TAXONOMY_CATEGORY: Specifies which categories belong to the different taxonomies. (A category can belong to multiple taxonomies; however, for a given taxonomy, there can be only one instance of any category.) The schema for this table has four fields; they are listed below, in order, with their data types:

```
TAXONOMY_LEVEL     NUMBER PK
TAXONOMY_ID        NUMBER PK
PARENT_ID          NUMBER PK
CHILD_ID           NUMBER PK
```

- MTR_TAXONOMY_CATEGORY_ITEM: Specifies which items go with a given taxonomy-category pair. The schema for this table has four fields; they are listed below, in order, with their data types:

```
CATEGORY_ID            NUMBER PK
TAXONOMY_ID            NUMBER PK
ITEM_ID                NUMBER PK
ITEM_TYPE              VARCHAR2(30) PK
```

- MTR_CATEGORY: Specifies the different categories used by the site. The schema for this table has three fields; they are listed below, in order, with their data types:

```
ID                     NUMBER PK
NAME                   VARCHAR2(150)
DESCRIPTION            VARCHAR2(4000)
```

Samples of the MTR Taxonomy Tables

The REAPI Demo includes a taxonomy; you can examine the demo MTR to see examples of all these tables.

Customer Table

The MTR_CUSTOMER table contains demographic information about the customer. Some customer attributes are common to all OP applications and some can be tailored to your application. The common attributes are customer ID, name, creation date, gender, age, marital status, personal income, whether or not the customer is the head of household, household income, household size, and whether the customer rents or owns.

You can specify up to 50 attributes specific to your Web application. These variable attributes are all strings.

The schema of the MTR_CUSTOMER table has the following fields. They are listed below, in order, with their data types.

```
ID                     VARCHAR2(32)
NAME                   VARCHAR2(80)
CREATION_DATE          DATE
GENDER                 VARCHAR2(10)
AGE                    NUMBER(3)
```

```
MARITAL_STATUS          VARCHAR2(20)
PERSONAL_INCOME         NUMBER
IS_HEAD_OF_HOUSEHOLD    CHAR(1)
HOUSEHOLD_INCOME        NUMBER
HOUSEHOLD_SIZE          NUMBER(2)
RENT_OWN_INDICATOR      VARCHAR2(30)
ATTRIBUTE1              VARCHAR2(150)
ATTRIBUTE2              VARCHAR2(150)
ATTRIBUTE3              VARCHAR2(150)
ATTRIBUTE4              VARCHAR2(150)
...
ATTRIBUTE49             VARCHAR2(150)
ATTRIBUTE50             VARCHAR2(150)
```

Hot Picks

Hot picks are used by some Web sites; for example, the daily specials might be hot picks. Information about hot picks is stored in two MTR tables, as follows:

- MTR_HOTPICK_GROUP lists the distinct hot pick groups used by the site. There is one record for each group. Each record contains a group ID, the group name (label), and a brief description of the group. The schema for this table has three fields; they are listed below, in order, with their data types.

```
ID              NUMBER PK
LABEL           VARCHAR2(150)
DESCRIPTION     VARCHAR2(400)
```

- MTR_HOTPICK lists the items in each hot pick group, arranged according to group ID. Each record consists of a group ID, an item ID, and an item type. The schema for this table has three fields; they are listed below, in order, with their data types.

```
ITEM_ID         NUMBER
ITEM_TYPE       VARCHAR2(30)
GROUP_ID        NUMBER
```

A hot pick group can also contain categories. In this case, the ITEM_TYPE is set to CATEGORY and item ID is set to the appropriate ID value in the MTR_CATEGORY table.

Detail Tables

Several tables in the MTR store the details of various activities.

- MTR_CUSTOMER_NAV_DETAIL stores the navigation data corresponding to a customer session. This table is populated with data collected in the RE.

- MTR_CUSTOMER_RATING_DETAIL stores rating data for customers. This table is populated using the data collected in the RE during data collection.

- MTR_PURCHASING_DETAIL stores purchasing data on a per-session basis. Typically this data is collected by the Web application.

- MTR_RECOMMENDATION_DETAIL stores the results of recommendation requests. The data stored in this table is used to generate reports on the performance of OP.

- MTR_VISITOR_NAV_DETAIL stores the navigation data corresponding to a visitor session. This table is populated with data collected in the RE.

- MTR_VISITOR_RATING_DETAIL stores rating data for visitors. This table is populated using the data collected in the RE during data collection.

Miscellaneous MTR Tables

The following tables are used internally by OP:

- MTR_ATTR_NAME_TO_ID_MAP is used to speed up package building.

- MTR_CONFIGURATION and MTR_INTERNAL_CONFIGURATION stores configuration information.

- The MTR_SESSION table stores information about the session that OP creates internally on behalf of the application.

- MTR_ATTR_ID_BIN_BOUNDARY is a materialized view of the join of the BIN_BOUNDARIES table and the ATTR_NAME_TO_ID_MAP table. It is used when transforming data during package builds.

- MTR_PROXY is used to set up proxies for new items. When a new item is introduced, there may not be enough detail information about it to build packages; OP uses data about a similar existing product.

Recommendation Engine

The RE schema stores current session data, package information, navigational information, as well as information about the taxonomy structure. RE data is sychronized back to the MTR automatically. The RE includes the following tables (partial list):

```
ATTR_ID_BIN_BOUNDARY          TABLE
HOTPICK                       TABLE
HOTPICK_GROUP                 TABLE
I_I_ANTECEDENT                TABLE
I_I_RULE                      TABLE
P_I_CATEGORY_RULES            TABLE
P_I_ITEM_RULES                TABLE
RE_ACTIVE_USER                TABLE
RE_CONFIGURATION              TABLE
RE_CURRENT_SESSION_DATA       TABLE
RE_DEPLOYABLE_PACKAGE         TABLE
RE_DEPLOYABLE_PKG_CONTENTS    TABLE
RE_ERROR_TABLE                TABLE
RE_INTERNAL_CONFIGURATION     TABLE
RE_LOG                        TABLE
RE_MESSAGE_LOG                TABLE
RE_PROFILE_DATA               TABLE
RE_RECOMMENDATION_DETAIL      TABLE
RE_SCHEMA_ACCESS              TABLE
TAXONOMY_CATEGORY             TABLE
TAXONOMY_CATEGORY_ITEM        TABLE
TAXONOMY_TRANS_CLOSURE        TABLE
```

HOT_PICKGROUP and HOTPICK are copies of the corresponding tables in the MTR.

RE_CURRENT_SESSION_DATA holds all the data collected using the data collection methods. This data is written back to the MTR using data synchronization.

RE_PROFILE_DATA stores the historical profiles of active users. When a user is detected, the profile of that user is loaded from the MTR to this table.

RE_RECOMMENDATION_DETAIL is the source of data for the corresponding table in the MTR. The data is synchronized back to the MTR.

ATTR_ID_BIN_BOUNDARY is a copy of the corresponding table in the MTR.

RE_CONFIGURATION and RE_INTERNAL_CONFIGURATION store the configuration parameters for the RE.

RE_DEPLOYABLE_PACKAGE keeps track of the deployable package that is currently deployed in the RE.

RE_LOG records events occurring in the RE.

RE_ACTIVE_USER stores information about all users who are currently active in the system. Data from this table is used to populate the session table in the MTR.

All other tables are used internally by the RE.

Mining Object Repository

Much of the administrative work done by OP uses MOR tables and views. The MOR includes the following tables (partial list):

```
MOR_VISITOR_TO_BROWSER_REPORT        TABLE
MOR_CONFIGURATION                    TABLE
MOR_CROSS_SOLD_ITEMS_REPORT          TABLE
MOR_DEPLOYABLE_PACKAGE               TABLE
MOR_EFFECTIVENESS_REPORT             TABLE
MOR_EMAIL_ADDRESS                    TABLE
MOR_ERROR_TABLE                      TABLE
MOR_INTERNAL_CONFIGURATION           TABLE
MOR_MESSAGE_LOG                      TABLE
MOR_MINING_MODEL                     TABLE
MOR_MINING_RESULT                    TABLE
MOR_MTR_CONNECTION                   TABLE
MOR_RECOMMENDATION_ENGINE            TABLE
MOR_RECOMMENDATION_REPORT            TABLE
MOR_RECOMMENDATION_STRATEGY          TABLE
MOR_RE_FARM                          TABLE
MOR_SCHEDULE_EVENT                   TABLE
MOR_SCHEDULE_ITEM                    TABLE
```

```
MOR_SCHEMA_ACCESS                    TABLE
MOR_TAXONOMY_TRANS_CLOSURE           TABLE
MOR_TRANS_SUPERVISED_RESULT          TABLE
```

5

Configuring Oracle9*i*AS Personalization

After you have installed OP and verified that the installation is correct, you can specify certain configuration parameters for OP.

This chapter

- lists the external OP configuration parameters and their default values
- indicates the parameters that you can change and how to change them
- describes RE configuration and data synchronization and how to configure it.
- describes how to determine appropriate parameter values for your installation.

All OP configuration parameters reside on the system where Oracle9*i* is installed.

Configuring the OP Scheduler

OP allows users to schedule package builds, deployments, and reports. The OP administrator can also specify that certain users get an email notification when a build, deploy, or report completes. The following configuration parameters in the MOR configuration table control scheduling and notification:

- The values for NLS_LANGUAGE and NLS_TERRITORY determine the languages used for the email notifications.
- The MAIL_PREFERENCE parameter specifies the formatting of email notifications, either plain text or HTML format. The default is MAILHTML, indicating HTML formatting.
- The ADMIN_EMAIL_ADDRESS parameter specifies the email address of the OP administrator. This address is used as the "return" address for email notifications. For example, if a user of the OP Administrative UI enters an

incorrect email address for notification, ADMIN_EMAIL_ADDRESS indicates the address used for warning about the incorrect information.

The default value of NAME - ADMIN_EMAIL_ADDRESS is the string Oracle.Personalization@oracle.com. Change this value when you install OP.

Changing OP Configuration Parameters

These parameters are divided into three categories:

- Values that are changed using SQL Plus, indicated by Y in the **Change** column in the summary tables.

- Values that you should *not* change, indicated by N in the **Change** column in the summary tables.

- Values that are changed using the OP Administration UI, indicated by UI in the **Change** column in the summary tables.

RE Configuration Parameters

Table 5–1 lists the RE configuration parameters, their data types, their default values, and a description for each. These parameters can be found in the RE_CONFIGURATION table.

Table 5–1   Recommendation Engine Configuration Parameters

Parameter Name	Data Type	Default Value	Description	Change
LOG_LEVEL	int	2	0=OFF, 1=INTERNAL ERROR plus Error and Warning, 2=All errors logged for 1 plus notifications, 3=All errors logged for 2 plus trace	Y
RE_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
REAPIRT_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
REAPIDEMO_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
UTIL_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y

Table 5–1   Recommendation Engine Configuration Parameters

Parameter Name	Data Type	Default Value	Description	Change
REAPIBATCH_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
TimeoutInterval	int	1800	Session timeout interval (in seconds)	UI
TimeoutFlag	int	1	Session timeout indicator (1=TRUE, 0=FALSE)	UI
DataSyncInterval	int	1800	Interval on which to synchronize customer data (in seconds)	UI
SyncCustomerNavigationalData	int	0	Is customer navigational data synchronized (boolean)	N
SyncCustomerRatingData	int	0	Is customer rating data synchronized (boolean)	N
SyncVisitorNavigationalData	int	0	Is visitor navigational data synchronized (boolean)	N
SyncVisitorRatingData	int	0	Is visitor rating data synchronized (boolean)	N
SyncPurchasingData	int	0	Is customer purchasing data synchronized (boolean)	N
SyncDemographicData	int	0	Is customer demographic data synchronized (boolean)	N
ConnectionPoolSize	int	128	Java connection pool limit per proxy.	Y[1]

[1]   Requires restart of OP.

MOR Configuration Parameters

Table 5–2 describes the configuration parameters for the OP Mining Object Repository (MOR). The table shows their data types, their default values, and a description for each. These parameters can be found in the MOR_CONFIGURATION table.

The table includes a column titled **Change**. If the value in this column is "N," do not change the parameter. If the value in this column is "Y," the value of the parameter

must be changed to a value suitable for your environment. The description of these parameters includes the instruction "change value on install."

Table 5–2   MOR Configuration Parameters

Parameter	Data Type	Value	Description	Change
MOR_USERNAME	string	<user name>	User name for Admin UI; change value on install	N
MOR_PASSWORD	string	<password>	Password for Admin UI; change value on install	N
MOR_DBALIAS	string	<alias>	Alias for the MOR database; change value on install	N
MOR_SCHEMA	string	< schema>	MOR schema name	N
MOR_HOST_URL	string	<hostname>	MOR hostname; change value on install	N
MOR_PORT	string	<port>	MOR port; change value on install	N
MOR_SID	string	isab900	MOR system ID; change value on install	N
MOR_VERSION	string	9.0.1	MOR version number	N
scheduleItemGracePeriod	int	60	Number of minutes a scheduled item must have been past due for it to cause an error	Y
buildEvents	int	20	Maximum number of events of this type to keep in log	UI
MAXNUMPUCHASEINGSESS	int	20	The maximum number of purchasing sessions reports to keep per recommendation engine farm	

Table 5–2   MOR Configuration Parameters

Parameter	Data Type	Value	Description	Change
MAXNUMRECEFFREP	int	20	The maximum number of recommendation effectiveness reports to keep per recommendation engine farm	
MAXNUMITEMIZEDRECEFFREP	int	20	The maximum number of itemized recommendation effectiveness reports to keep per recommendation engine farm	
NUMOFITEMSINITEMIZEDRECEFFREPORT	int	20	The number of top-ranked items in itemized recommendation effectiveness reports	
IAS_HOSTNAME	string		Parameter used by report workflow to construct a URL for email notification	
IAS_SERVLET	string		Parameter used by report workflow to construct a URL for email notification	
IAS_SERVLET_ZONE	string		Parameter used by report workflow to construct a URL for email notification	
IAS_PORT	string		Parameter used by report workflow to construct a URL for email notification	
IAS_SERVLET_MOR_CONN	string		Parameter used by report workflow to construct a URL for email notification	

Table 5–2   MOR Configuration Parameters

Parameter	Data Type	Value	Description	Change
deployEvents	int	20	Maximum number of events of this type to keep in log	UI
reportEvents	int	20	Maximum number of events of this type to keep in log	UI
LOG_LEVEL	int	2		Y
ALGS_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
DMAPI_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
PAR_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
TNB_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
UI_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
UTIL_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
WFJAVA_TRACE	int	0	0=OFF, 1=LOW (detail), 2=MODERATE, 3=HIGH	Y
MAIL_PREFERENCE[1]	string	MAILHTML	Format for email notifications; other possible value is MAILTEXT	Y

Table 5–2   MOR Configuration Parameters

Parameter	Data Type	Value	Description	Change
NLS_LANGUAGE[2]	string		Supported values: AMERICAN, FRENCH, GERMAN, ITALIAN, SPANISH, BRAZILIAN PORTUGUESE, JAPANESE, KOREAN, SIMPLIFIED CHINESE	Y
NLS_TERRITORY[3]	string		Supported values: AMERICA, FRANCE, GERMANY, ITALY, SPAIN, BRAZIL, JAPAN, KOREA, CHINA	Y
ADMIN_EMAIL_ADDRESS	string	Oracle. Personal- ization@ oracle.com	Email address of the Oracle Personalization administrator for email notifications	Y

[1] The email notifications sent by OP support either plain text or HTML format. The default setting in the MOR configuration table is MAILHTML, indicating HTML format. If the incoming mail server that you are using supports only plain text, set this parameter to MAILTEXT.

[2] The default value for NLS_LANGUAGE after OP installation is that this is empty (blank). However, if it is edited, OP verifies that the value is supported and uses it instead of the database default language for the server code (it does not affect PL/SQL messages however). If it is not specified, the database default language is used. If neither these values nor the database default indicates one of the supported languages, OP defaults to AMERICAN. This value also specifies the language used for email notifications.

[3] The default value for NLS_TERRITORY after OP installation is that this is empty (blank). However, if it is edited, OP verifies that the value is supported and uses it instead of the database default language for the server code (it does not affect PL/SQL messages however). If it is not specified, the database default language is used. If neither these values nor the database default indicates one of the supported languages, OP defaults to AMERICA. This value specifies also specifies the language used for email notifications.

MTR Configuration Parameters

Table 5–3 describes the configuration parameters for the OP Mining Table Repository (MTR). The table shows their data types, their default values, and a description for each. These parameters can be found in the MTR_CONFIGURATION table in MTR schema.

These parameters allow selecting different types of data to be synchronized to the MTR. At the end of an OP session, MTR synchronization adds data collected in the RE (during the session) to the data already stored in the MTR. In order for data synchronization to take place, the MTR must be configured to allow the various types of data to be synchronized.

Table 5–3   MTR Configuration Parameters[1]

Parameter	Data Type	Value	Description	Change
ALLOW_SYNC_DEMOGRAPHIC	boolean	T	Allows demographic data to be synchronized to MTR	Y
ALLOW_SYNC_NAVIGATION	boolean	T	Allows navigational data to be synchronized to MTR	Y
ALLOW_SYNC_PURCHASING	boolean	T	Allows purchasing data to be synchronized to MTR	Y
ALLOW_SYNC_RATING	boolean	T	Allows rating data to be synchronized to MTR	Y
ALLOW_MTR_SYNC_VISITOR_NAVIGATION	boolean	T	Allows visitor navigation data to be synchronized to MTR	Y
ALLOW_SYNC_VISITOR_RATING	boolean	T	Allows visitor rating data to be synchronized to MTR	Y

[1]   Change these values only when the MTR is not being used.

RE Installation and Configuration

Installation and configuration of a recommendation engine (RE) must be tailored to the expected number of active users that it will support. The RE in this context refers to a single engine in a single Recommendation Engine Farm on a single database instance. If multiple engines in one or more RE farms are installed on the same database instance, the configuration parameters require adjustment.

RE Optimization

Many factors go into the optimization of an RE. Some of these are set by the installation procedure, while others are techniques that may be used by the DBA. Configuration options fall into two broad categories:

- System availability settings
- Performance settings

System availability settings are those settings required by the RE to handle user load without failure. Performance settings are those settings that help maximize throughput.

System Availability

The system availability settings for RE configuration depend on the number of anticipated users. If you know the number of users, it is possible to estimate the approximate system resource requirements and make database configuration recommendations. Since the REAPI maintains a connection pool of user connections, which can be reused, the number of required connections depends on how well user requests are being satisfied by the RE. That is, if for some reason there is a slowdown in the RE causing connection links to be held longer in the REAPI connection pool, the number of connections will tend to increase. As the number of connections increases, the number of actual database sessions increases. Each connection in the REAPI connection pool represents a database session.

The maximum number of connections in the REAPI connection pool is a configurable parameter in each RE. If this limit is exceeded, it may indicate that there are performance issues that need to be addressed other than simply increasing the size of the connection pool.

Each Web application user's client session results in database activity in the RE schema. First, configure the database to handle the number of anticipated simultaneous users. Depending on the amount of available memory and CPUs in the system the RE database is installed on, it may be possible to support 50-100 users in a dedicated server environment. In this environment, each user connection to the database would require its own dedicated Oracle server for database access. As the number of users extends beyond 100, it may be more appropriate to use Oracle's Multi-Threaded Server environment where database connections are pooled and serviced by shared database servers. The DBA responsible for the RE must decide whether the dedicated or shared server environment is used.

Performance

REAPI performance may be affected by several factors. On the client side, the REAPI runs in the JServer environment. Sufficient memory and CPU must be available to the client to handle the throughput for the active users. Communication with the RE from the REAPI clients is implemented through JDBC connections over Oracle's SQL*Net network. As the number of users grows, so does the demand on the network.

Required RE Parameters

The recommendation engine requires certain database parameters to be set to a minimum value, as follows:

```
JOB_QUEUE_PROCESSES=2
(in ORACLEHOME/DBX directory)
ORACLEHOME/DBX/init_<SIDname>.ora
```

Suggested RE Database Parameters (All Load Levels)

The parameters and values listed below, while not necessary, are strongly recommended:

BUFFER_POOL_KEEP	50 @ 8192 bytes ea
SORT_AREA_SIZE	819200 bytes
SORT_AREA_RETAINED_SIZE	819200 bytes

Suggested RE Database Parameters (Variable Load Level)

Table 5–4 suggests guidelines for database configuration parameters based on number of projected users. The table shows, for a specified number of users, whether multi-threaded servers (MTS) are recommended, and recommended values for the number of MTS dispatchers, shared MTS servers, sessions, the size of large pool, and the size of shared pool:

Table 5–4   RE Database Parameters (Variable Load Level)

Users	MTS	MTS Dispatchers	MTS Shared Servers	Sessions	Large Pool	Shared Pool
100	No	N/A	N/A	100	Default	Default

Table 5–4   RE Database Parameters (Variable Load Level)

Users	MTS	MTS Dispatchers	MTS Shared Servers	Sessions	Large Pool	Shared Pool
1000	Yes	4	20	300	250M	100M
2000	Yes	8	30	600	500M	120M
3000	Yes	10	40	900	1000M	140M
4000	Yes	12	50	1200	1500M	160M

Suggested RE Configuration Parameters

The table settings in Table 5–5 are based on the number of estimated simultaneous sessions. These parameters are set in the RE schema table RE_CONFIGURATION. The table shows, for a specified number of simultaneous sessions, the recommended connection pool size and data synchronization interval:

Table 5–5   RE Configuratiuono Parameters

Sessions	Connection Pool Size (number of connections)	Data Synchronization Interval (in seconds)
100	128	300
1000	256	300
2000	512	300
3000	1024	180
4000	2048	180

MTR Installation and Configuration

The Mining Table Repository (MTR) database holds customer (demographic, purchasing, ratings, navigational data) and product data. Data mining models are built based on this data. Any data collected in the RE will be copied to the MTR at scheduled intervals. Customer profile data is also copied from the MTR into the RE when a customer begins a user session. All data transfer between the MTR and the RE is done using database links.

Table 5–6 offers guidelines for database configuration parameters based on the number of projected users. The table shows, for a specified number of users, whether multi-threaded servers (MTS) are recommended, and recommended values

for the number of MTS dispatchers, the number of MTS servers, the number of sessions, and the size of the large pool:

Table 5–6   Suggested Database Configuration Parameters

Users	MTS	MTS Dispatchers	MTS Servers	Database Sessions	Large Pool
100	No	N/A	N/A	100	Default
1000	Yes	2	10	100	25M
2000	Yes	3	20	200	50M
3000	Yes	4	20	300	100M
4000	Yes	50	30	400	150M

Data Synchronization

Data synchronization moves user-specific data that is collected in the RE during a session to permanent storage, that is, to the appropriate table in the mining table repository (MTR). Session and recommendation data are always synchronized; other kinds of data are synchronized according to the way the RE Farm and MTR are configured. See "Data to Synchronize", later in this chapter, for configuration instructions. Customer data and visitor data are copied to the appropriate MTR tables. (There is one set of MTR tables for customer data and a different set for visitor data.)

Data is synchronized every DataSyncInterval, which is a configuration parameter that is specified for an RE Farm. Data synchronization is performed only for users whose sessions are inactive. A session is *inactive* if there has been no activity for a specified period of time or if the session has been explicitly closed. Note that a user can have more than one session at any time. A customer ID is deleted from RE_PROFILE_DATA only when all the customer's sessions are inactive.

After the data is copied to the MTR, the data is purged (deleted) from the RE tables. Data that cannot be synchronized for some reason (for example, data that has an invalid item ID) is also purged.

Data is collected in the RE_CURRENT_SESSION_DATA table and the RE_RECOMMENDATION_DETAIL table. The data source type of the data determines the MTR table to which data is copied.

Table 5–7 shows the four data source types and the MTR table for each.

Table 5–7   Data Synchronization for RE_CURRENT_SESSION_DATA

DATA_SOURCE_TYPE	MTR Table
1 (demographic) for customers only	MTR_CUSTOMER
2 (purchasing) for customers only	MTR_PURCHASING_DETAIL
3 (rating) for visitors and customers	MTR_RATING_DETAIL
4 (navigational) for visitors and customers	MTR_NAVIGATIONAL_DATA

RE_RECOMMENDATION_DETAIL data is copied to the MTR_RECOMMENDATION_ DETAIL table and appropriate RE_ACTIVE_USER data is copied to MTR_SESSION table. RE_PROFILE_DATA is updated in the MTR_CUSTOMER table.

Configuring Data Synchronization

You specify two things: the synchronization interval for a Farm, and exactly what data to synchronized for a specific MTR connection:

Synchronization Interval

In the **Farms** tab of the Administrative UI, select a farm, select **Edit**, and then click the **Advanced Settings** button. Specify an appropriate data synchronization interval for the selected farm. (You can also specify the timeout interval here.)

The default synchronization interval is 300 seconds (5 minutes). The synchronization interval should be adjusted for the number of users of the application. If there are many users and the synchronization interval is long, the REs will fill with data.

Data to Synchronize

In order for data synchronization to take place, the MTR must allow that type of data to be synchronized. These rules are specified when you install OP.

You configure the MTR connection using the OP Administrative UI. At the top of the **Farms** tab, click **Options**, click **MTR database connections**, click **Edit**, and finally click the **Sync settings** button. The synchronization settings for this MTR are displayed. To change a setting click the appropriate checkbox.

The types of data that are allowed to be synchronized are indicated by a checkmark in the corresponding checkbox. If a selection is greyed out, the configuration of the MTR does not allow synchronization of that type of data.

By default, all four types of data are left unchecked, that is, no data is synchronized. You can choose to allow synchronization of any type of data for which the MTR allows synchronization. Any changes apply only to the current MTR connection.

6

JSP Tag Configuration

The location of `opreapi_rt.jar` as shipped is `$ORACLE_HOME/dmt/jlib`; this location is unknown to OC4J. The jar should be copied to other Oracle9*i* Application Server instances that do not have OP installed but need to access this API either directly or via the JSP tags.

Applications that use REAPI directly or through Oracle Personalization JSP tags need to make `opreapi_rt.jar` visible to their application class loader. There are several ways this can be achieved in OC4J. The simplest way to make a supporting JAR visible is to place it in the `WEB-INF/lib` directory of the application. If you wish to make REAPI visible to all applications in an OC4J instance, you need to place it in the `$J2EE_HOME/lib` directory. For more information about application deployment, consult OC4J documentation.

In a future release we expect to include the `opreapi_rt.jar` in the core installation so that these steps are not required.

The OP tags demo code is maintained by the JSP group. The tags' functionality is coded in the JSP VOB, and is also documented in the *Oracle JSP Utilities Guide*.

7

Initial Data Collection

Package building requires data. If you have data collected already, the data can be used to populate the MTR tables or the MTR schema can be mapped to the existing data via views. However, if you have no data, you can use the REAPI methods addItem and addItems to collect data. Data collection occurs in the Oracle9*i*AS Personalization Recommendation Engine (RE). For an RE to be up and running, there must be a package deployed in that RE. However, in order to build and deploy a package, you must have data in the MTR. To put it simply, you can't collect data unless you have enough data to build a package. You resolve this problem by populating the MTR with seed data and then using the seed data to build and deploy an initial package.

You create seed data on the system where Oracle9*i* is installed.

Creating and Removing Seed Data

OP includes a script to populate an MTR with seed data and another script to remove the seed data from the MTR once enough data is collected in the MTR. The two scripts are in the directory $ORACLE_HOME/dmt/admin/mtr:

- insertMTRSeedRecords.sql (populates an unpopulated MTR with seed data)

- removeMTRSeedRecords.sql (removes the seed data from the MTR once the initial package is built)

The OP administrator can run these scripts when appropriate.

Preparing to Build Packages with Seed Data

Follow these steps to populate the tables required for package building:

1. Install OP with an unpopulated MTR.

2. Configure synchronization settings in the MTR_CONFIGURATION table appropriately.

3. Create an MTR connection and allow synchronization for the data source types that you intend to collect.

4. Log in to SQL*Plus as the MTR account that you created using the OP Schema Creation Wizard.

5. Run `insertMTRSeedRecords.sql`.

6. Create a package using the default selections.

7. Create an RE Farm with at least one RE in it. Use the advanced settings to specify the customer data synchronization interval.

8. Build the package.

9. Deploy the package to the RE Farm that you created in step 7.

10. Log in to SQL*Plus using the MTR account that you created during installation; run `removeMTRSeedRecords.sql`.

11. Populate or map the following tables with data for your application:

    -   MTR_ITEM: This table is the site catalog. When the collected data is saved in the MTR, data corresponding to nonexistent items are thrown away as part of the validation check. So, if the MTR_ITEM table is empty, the collected data is thrown away.

    -   MTR_CUSTOMER: The customer-IDs are expected to be managed by the site. Hence the MTR_CUSTOMER table should be populated or mapped with at least the ID attribute before the data is written to the MTR. The data values for other attributes can be collected via the data collection API.

    -   MTR_ATTR_NAME_TO_ID_MAP and MTR_BIN_BOUNDARY: These tables should be populated with the mapping information that maps attribute names to numeric identifiers and the bin boundaries, respectively.

        You must specify bin boundaries for all data source types that you plan to collect.

12. After these tables are populated, the OP administrator must run the following script, which populates the MTR_ATTR_ID_BIN_BOUNDARY table:

```
populate_mtr_attr_id_bin_boundary.sql
```

Now collect data from your Web site. After you collect data for an appropriate amount of time, you can build packages based on the collected data.

8

Stopping and Starting OP

If you need to perform maintenance on the system or systems where OP resides, you should stop OP before you stop the operating system. After you perform any required maintenance, you must start OP. The procedures outlined in this chapter will ensure that you do not lose any data when you start and stop OP.

Stopping OP

Follow these steps to stop OP:

1. Stop all package builds, package deployments, and reports.

   Use the OP Administrative UI to ensure that all package builds, package deploys, and reports are stopped. If any of these are running, you can either cancel them or wait until they complete. You must also change the schedule for any builds, deploys, and reports to make sure that they do not take place while OP is stopped.

   Verify that nothing is running.

2. Stop all Recommendation Engines using the script `REmaint.sh` (in `$ORACLE_HOME/dmt/admin`).

   `REmaint.sh` stops (sets the status to offline) all REs known to the MOR and forces the REs to save any cached information to the MTR, using data synchronization. After the script completes, any application that attempts to use any of the stopped REs receives an RE_OFFLINE exception.

   The scripts are:

   ```
   % REmaint.sh <property filename> or
   % REmaint.sh <MOR username<MOR password><MOR JDBC connect string>
   ```

where property file looks like:

```
dbUrl=<JDBC connection string to MOR database>
dbuserName=<MOR username>
dbpassWord=<MOR password>
```

3. Stop the OP 9.0.2 scheduling engine with the following command:

```
SQL> execute OP_STOP_ENGINE
```

OP_STOP_ENGINE is an SQL script that resides in the MOR schema.

4. Stop the OP Save Items Archiver and OP Data Synchronization. For each RE, log in to the RE, and start SQL*Plus.

- Issue the following command to stop the archiver:

```
SQL> exec re_loader.stop_save_items_archiver_job;
```

- Issue the following command to stop synchronization:

```
SQL> exec re_loader.stop_data_sync_job;
```

At this point, you can perform any required maintenance except for changing MOR and MTR characteristics.

Starting OP

You start OP by reversing the steps that you followed to stop it:

1. Start the OP Save Items Archiver and OP Data Synchronization. For each RE, log in to the RE, and start SQL*Plus.

- Issue the following command to start the archiver:

```
SQL> exec re_loader.start_save_items_archiver_job;
```

- Issue the following command to start synchronization:

```
SQL> declare
    2> v_job number;
    3> begin
    4> v_job := re_loader.start_data_sync_job;
    5> end;
    6> /
```

2. Start the OP 9.0.2 scheduling engine with the following command:

```
SQL> execute OP_START_ENGINE
```

OP_START_ENGINE is an SQL script that resides in the MOR schema.

3. Start all package builds, package deployments, and reports.

   Use the OP Administrative UI to start or restart all package builds, package deploys, and reports that are stopped. You may have to change back the schedule for any builds, deploys, and reports that you changed when you stopped OP.

   Verify that all required events are running.

   After packages are deployed to them, the REs automatically start up, that is, they are no longer offline.

Index