Oracle® Application Server Personalization

Administrator's Guide

10*g* Release 2 (10.1.2)

B14050-01

July 2005

ORACLE®

Oracle Application Server Personalization Administrator's Guide, 10g Release 2 (10.1.2)

B14050-01

Contents

5 Configuring OracleAS Personalization

6 JSP Tag Configuration

7 Initial Data Collection

8 Stopping and Starting OracleAS Personalization

Index

Preface

This manual describes the installation and administrative functions that need to be performed for Oracle Application Server Personalization (OracleAS Personalization). 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 Oracle Application Server 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. 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, seven days a week. For TTY support, call 800.446.2398.

Where to Find More Information

Documentation for OracleAS Personalization at the current release consists of the following documents:

- *Oracle Application Server Release Notes,* 10*g* Release 2 (10.1.2), which contains a chapter for each component of Oracle Application Server. The chapter for the OracleAS Personalization component contains platform-specific information, a bug report, and informatioin about any late-breaking changes.

- *Oracle Application Server Personalization User's Guide,* release 10*g* Release 2 (10.1.2).

- *Oracle Application Server Personalization Administrator's Guide,* release 10*g* Release 2 (10.1.2) (this document).

- *Oracle Application Server Personalization Programmer's Guide,* release 10*g* Release 2 (10.1.2). 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 (Oracle Application Server Personalization API Reference), updated for the current release.

Related Manuals

OracleAS Personalization documentation is a component of the Oracle Application Server 10*g* Release 2 (10.1.2) Documentation Library. See especially:

- *Oracle Application Server Concepts*

- *Oracle Application Server Administrator's Guide*

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

Documentation Formats

Documentation for OracleAS 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 OracleAS Personalization Administrative UI includes extensive online help that can be summoned from a list of contents and from Help buttons.

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.

Convention	Meaning
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.

1

OracleAS Personalization Administration Overview

Oracle Application Server Personalization (OracleAS Personalization) permits applications to collect and store customer data in an Oracle database, build predictive models, and make recommendations on the basis of the models. For more information about OracleAS Personalization and a description of how it works, see the Oracle Application Server Personalization User's Guide.

This manual provides the generic instructions for administering OracleAS Personalization and provides notes regarding installation. For more information about installing OracleAS Personalization, see the Oracle Application Server Installation Guide and Oracle Application Server Release Notes.

This manual is organized as follows:

- Chapter 1: Overview

- Chapter 2: Security and Internationalization

- Chapter 3: Installation Notes

- Chapter 4: OracleAS Personalization Schemas

- Chapter 5: OracleAS Personalization Configuration

- Chapter 6: JSP Tags

- Chapter 7: Initial Data Collection

- Chapter 8: Stopping and Starting OracleAS Personalization

1.1 Installation

The installation procedures are described in Chapter 3 and in the Oracle Application Server Installation Guide.

1.2 Configuration Requirements

The Administrative UI requires SSL. See Section 2.2 for details.

2

OracleAS Personalization Security and Internationalization

This chapter describes issues associated with Oracle Application Server Personalization (OracleAS Personalization) security and internationalization.

2.1 OracleAS Personalization Security

OracleAS Personalization follows the Oracle database server security policy. OracleAS Personalization 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 OracleAS Personalization components.

- The database administrator (DBA) grants access to qualified users.

- The OracleAS Personalization Administrative user interface (UI) uses secure SSL technology to access the Oracle HTTP Server.

2.1.1 Classes of Users and Their Privileges

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

The OracleAS Personalization user is required to log in with a valid username and password over SSL to the remote customer database. For the OracleAS Personalization Schema Creation Wizard, the SSL encryption services are leveraged.

The Oracle Application Server application administrator needs to know the following in order to access the OracleAS Personalization component:

- the customer database SYS password

- the connect string information (TNS, Alias)

- JDBC connection information (SID, HOST, PORT)

OracleAS Personalization controls user access by providing two database roles:

- OP_ADMIN: This role allows the user to access the OracleAS Personalization product as an OracleAS Personalization Administrator. This means having access to the entire OracleAS Personalization product, including the Message Viewer.

■ OP_REPORT: This role allows the user to view OracleAS Personalization reports and access the Report Viewer URL (`https://<hostname>:<port>/OP/Admin/ReportsPage`).

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

For new OracleAS Personalization users, the DBA can grant either full privileges (OP_ADMIN) or more limited privileges (OP_REPORT, access only to the OracleAS Personalization 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 the OracleAS Personalization product at the appropriate level.

2.2 Enabling SSL in the Oracle Application Server 10*g*

By default, Oracle Application Server 10*g* does not enable SSL connections.

To enable SSL, you must edit the `ORACLE_HOME/opmn/conf/opmn.xml` file and change `ssl-disabled` to `ssl-enabled`.

Then restart the `opmn` managed processes using the commands

```
opmnctl stopall
opmnctl startall
```

2.3 Internationalization

This section describes some issues associated with running OracleAS Personalization in an environment that supports more than one language or character set.

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

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

You can specify a value in NLS_LANGUAGE. OracleAS Personalization 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, OracleAS Personalization defaults to AMERICAN. This value also specifies the language used for email notification.

Supported languages and their character sets are as shown in Table 2–1, below:

Table 2–1   Supported Languages and Their Character Sets

NLS Language	HTML_CHARSET	
AMERICAN	ISO-8859-1	US7ASCII

Table 2–1   (Cont.) Supported Languages and Their Character Sets

NLS Language	HTML_CHARSET	
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 Oracle Application Server Personalization (OracleAS Personalization).

The installation of OracleAS Personalization is integrated with Oracle Application Server standards.

The installation is done in two parts. The first part is done by an Oracle Application Server administrator and the second part is done by the database administrator.

3.1 Part 1 Installation

Note the following:

- Before installing OracleAS Personalization, install the Oracle Application Server Infrastructure database.

- To install OracleAS Personalization, you must select **Business Intelligence and Forms** as the type of installation.

The first part of the installation happens automatically as part of installing Oracle Application Server Business Intelligence on your system, which copies the software onto the Oracle Application Server `ORACLE_HOME/mp` directory and configures the OracleAS Personalization Admin UI servlet in Oracle Application Server Containers for J2EE (OC4J).

When the Oracle Application Server installation completes, the OracleAS Personalization Admin UI will be available at this location:

`https://<hostname>:<SSLPORT>/OP/Admin`

(You will find the actual names in `$ORACLE_Home/ Apache/Apache/setupinfo.txt.` Two paths are shown; use the `https:` path.)

When the Oracle Application Server installation completes, you will be able to view the OracleAS Personalization login screen, but you cannot log in until you have completed the second part of the installation. (If you try to log in before completing the second part, login fails with "Authentication Errors" because the MOR schema required for authentication is not available.)

3.2 Part 2 Installation

In the second part of the installation, your DBA configures the required schemas in the Oracle database and Oracle Application Server Containers for J2EE (OC4J) is reloaded to read the new configuration settings. See the *Oracle Application Server Containers for J2EE User's Guide* for information on reloading OC4J.

The schemas are configured using the OracleAS Personalization Schema Creation Wizard, which is a Java application that creates these schemas over the network to your Oracle database.

3.2.1 Before Running the OracleAS Personalization Schema Creation Wizard

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

Documentation on updating the database `init.ora` parameters can be found in the Database Administrator Authentication section of the *Database Administrator's Guide* (`http://download.oracle.com/otndoc/oracle9i/901_doc/server.901/a90117/dba.htm#10431`). <<<find the lastest equivalent link>>>

Remote SYS access can be tested using the following command on the system where you installed Oracle Application Server 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 collect the following information for the Mining Object Repository (MOR), the Recommendation Engine Repository (RE), and the Mining Table Repository (MTR) 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 Oracle 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

   - Host

   - Port

   - 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 Repository)

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

   - Host

   - Port

   - 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

3.2.2 Running the OracleAS Personalization Creation Wizard

After collecting the required information, log in to the Oracle Application Server system where you installed Oracle Application Server Business Intelligence and run the following command:

```
(UNIX)
prompt> $ORACLE_HOME/mp/admin/opconfig.sh

(Windows)
prompt> cd %ORACLE_HOME%\mp\admin
prompt> opconfig.bat
```

3.2.3 After Running OracleAS Personalization Schema Creation Wizard

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

Use the Oracle Enterprise Manager 10*g* to restart OC4J, Business Intelligence, and Forms. The OracleAS Personalization Admin UI will not allow you to log in until you have restarted OC4J.

3.3 Data Collection Form

This section displays the data that will be collected while you are running the OracleAS Personalization Schema Creation Wizard.

Table 3–1   Data to Be Collected for the MOR

Item	Collected Data	Default
SYS password	******	change_on_install
JDBC connect info	Host	None
	Port	None
	SID	None

Table 3–1   (Cont.) Data to Be Collected for the MOR

Item	Collected Data	Default
Schema name		OPMOR
Password		MORPW
Fully qualified file names of the schema tablespace data file		
Fully qualified file names of the schema tempspace data file		
Email address for replies from notification messages		Your.Email@domain.com

Table 3–2   Data to Be Collected for the RE

Item	Collected Data	Default
SYS password	******	change_on_install
JDBC connect info	Host	None
	Port	
	SID	
Schema name for the RE		RE
Password for the RE		REPW
Fully qualified file name of the schema tablespace data time		None

Table 3–3   Data to Be Collected for the MTR

Item	Collected Data	Default
SYS password	******	change_on_install
JDBC connect info	Host	None
	Port	
	SID	
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

OracleAS Personalization Schemas

Oracle Application Server Personalization (OracleAS Personalization) uses several database schemas, as follows:

- **Mining Object Repository (MOR)**: The MOR controls the operation of OracleAS Personalization and contains mining objects, such as packages, reports, and schedule items.

- **Mining Table Repository (MTR)**: Contains customer profile data tables plus other information such as bin boundaries, hot picks, and taxonomy.

- **RE Schema**. The RE schema is where recommendations are generated from a predictive model package and profile data is collected and staged for syncing with the MTR.

All OracleAS Personalization schemas reside on the systems where Oracle databases are 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; 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 OracleAS Personalization. You can examine the schema of the unpopulated MTR and populate it with your own data.

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

Before you can obtain recommendations or collect data, you must create and deploy a package. You cannot create a package until data exists in the MTR. There are three ways to initially populate an MTR:

- Populate with external data, that is, map existing historical data that was collected by your application and stored in an Oracle database.

- Populate with the seed data, which enables a dummy package to be built and deployed.

- Once a package is built and deployed, the application can collect data using the REProxyRT method `addItem` or `addItems` (see Chapter 7), or the application can collect data separately and populate the MTR outside of OracleAS Personalization.

4.1 Mining Table Repository

The OracleAS Personalization MTR consists of the tables and views listed in Table 4–1, below. Certain of these tables must be populated with data specific to your application in accordance with the MTR schema. Other tables, such as the tables associated with sessions and recommendations, are automatically populated by OracleAS Personalization. The third column indicates whether the table is to be populated by the user, by OracleAS Personalization, or by either.

Table 4–1   MTR Tables and Views

Table Name	Table or View	Populated by
MTR_ATTR_ID_BIN_BOUNDARY	VIEW	OP[1]
MTR_ATTR_NAME_TO_ID_MAP	VIEW	OP
MTR_BIN_BOUNDARY	TABLE	User
MTR_CATEGORY (optional)	TABLE	User
MTR_CONFIGURATION	TABLE	Either
MTR_CUSTOMER	TABLE	Either
MTR_CUSTOMER_NAV_DETAIL	TABLE	Either
MTR_CUSTOMER_RATING_DETAIL	TABLE	Either
MTR_HOTPICK (optional)	TABLE	User
MTR_HOTPICK_GROUP (optional)	TABLE	User
MTR_INTERNAL_CONFIGURATION	TABLE	OP
MTR_ITEM	TABLE	User
MTR_NAVIGATION_DETAIL	VIEW	OP
MTR_PROFILE_DATA	VIEW	OP
MTR_PROXY	TABLE	User
MTR_PURCHASING_DETAIL	TABLE	User
MTR_RATING_DETAIL	VIEW	OP
MTR_RECOMMENDATION_DETAIL	TABLE	OP
MTR_SCHEMA_VERSION	VIEW	OP
MTR_SESSION	TABLE	OP
MTR_TAXONOMY (optional)	TABLE	User
MTR_TAXONOMY_CATEGORY (optional)	TABLE	User
MTR_TAXONOMY_CATEGORY_ITEM (optional)	TABLE	User
MTR_VISITOR_NAV_DETAIL	TABLE	Either
MTR_VISITOR_RATING_DETAIL	TABLE	Either

[1] OP - OracleAS Personalization

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

4.1.1 Item Table

The item table (MTR_ITEM) contains a list of all the individual items that the application deals with. When OracleAS Personalization returns a recommendation, it returns an item's ID and ITEM_TYPE. The ID and ITEM_TYPE together uniquely identify an item. Different item types may use the same ID. The item table is usually mapped to the catalog tables in the application database. The schema for MTR_ITEM has four fields; they are listed in Table 4–2 with their data types. The LABEL column can be used for any locally-defined purpose; for example, it could contain a catalog identifier used in the local database.

Table 4–2   Item Table

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

4.1.2 Bin Boundaries

The model-building algorithms in OracleAS Personalization employ counting techniques to calculate probabilities. The data manipulated by the algorithms must be discrete, that is, not continuous. Hence, numeric data must be divided into bins to present discrete values to OracleAS Personalization. The process of dividing data into bins is called binning.

In OracleAS Personalization, binning is performed in a transformation step before model build. The value ranges for binning. The bin boundaries must be specified in the bin boundaries table for OracleAS Personalization to bin the values.

Categorical data should be mapped to numbers. This can be a one-to-one mapping if the cardinality should be preserved. In the case of high cardinality, a many-to-one mapping can be used to reduce the cardinality.

In summary, OracleAS Personalization requires all numerical data to be binned, and categorical data to be mapped.

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.

> **Note:** OracleAS Personalization requires ratings and demographic data to be binned. Purchasing and navigation data is not binned.

The table MTR_BIN_BOUNDARY has seven fields; they are listed in Table 4–3, in order, with their data types.

Table 4–3   Bin Boundary Fields

Field	Data Type
DATA_SOURCE_TYPE	NUMBER(3)
ITEM_TYPE	VARCHAR2(30)
ATTRIBUTE_NAME	VARCHAR2(30)

Table 4–3   (Cont.) Bin Boundary Fields

Field	Data Type
LOWER_VALUE	NUMBER
UPPER_VALUE	NUMBER
STRING_VALUE	VARCHAR2(60)
BIN_NUMBER	NUMBER(15)

> **Note:**   DATA_SOURCE_TYPE values are 1 for demographic data, 2 for purchasing data, 3 for ratings data, and 4 for navigational data.

Examples of Specifying Bin Boundaries

The following examples illustrate how to specify bin boundaries.

Consider movie rating data on a scale of 1 to 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.

4.1.3 Taxonomy

> **Note:**   Refer to the *Oracle Application Server Personalization User's Guide* for an explanation of taxonomy in OracleAS Personalization.

Taxonomies are implemented by using a group of tables. Taxonomies are specified by the OracleAS Personalization user at MTR setup time. These tables are:

■ MTR_TAXONOMY: Lists the different taxonomies used by an application. Each taxonomy has a unique ID, name, and description. The schema for this table has three fields; they are listed in Table 4–4 with their data types:

Table 4–4   MTR_TAXONOMY Table Fields

Field	Data Type
ID	NUMBER PK
NAME	VARCHAR2 (150)
DESCRIPTION	VARCHAR2 (4000)

■ MTR_CATEGORY: Specifies the different categories to be used in the taxonomy. The schema for this table has three fields: the unique identifier of the category, its name, and a description. They are listed in Table 4–5 with their data types:

Table 4–5   MTR_CATEGORY Table Fields

Field	Data Type
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 in Table 4–6 with their data types. Each row in this table describes an "edge" of the taxonomy graph, that is, a line connecting two categories. The TAXONOMY_LEVEL is the level of the parent category. PARENT_ID and CHILD_ID are values from the MTR CATEGORY table.

Table 4–6   MTR_TAXONOMY_CATEGORY Table Fields

Field	Data Type
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. In other words, it lists each item as a member of a particular category in a particular taxonomy. If an item belongs to more than one category, the item is listed once for each category. The schema for this table has four fields; they are listed in Table 4–7 with their data types:

Table 4–7   MTR_TAXONOMY_CATEGORY_ITEM Table Fields

Field	Data Type
CATEGORY_ID	NUMBER PK
TAXONOMY_ID	NUMBER PK
ITEM_ID	NUMBER PK

Table 4–7   (Cont.) MTR_TAXONOMY_CATEGORY_ITEM Table Fields

Field	Data Type
ITEM_TYPE	VARCHAR2 (30) PK

Samples of the MTR Taxonomy Tables

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

4.1.4 Customer Table

The MTR_CUSTOMER table contains demographic information about an application's registered customers. Some customer attributes are common to all OracleAS Personalization 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.

Fifty generic attributes are available in the table that can be defined specifically for your application. The first 25 of these attributes are of type VARCHAR2, and the second 25 are of type NUMBER.

All customer attributes are locally defined and are mapped from the application database or collected through application registration procedures.

The schema of the MTR_CUSTOMER table has the following fields. They are listed in Table 4–8 with their data types.

Table 4–8   MTR_CUSTOMER Table Fields and Data Types

Field	Data Type
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)
. . .	. . .
ATTRIBUTE25	VARCHAR2 (150)
ATTRIBUTE26	NUMBER
ATTRIBUTE27	NUMBER
. . .	. . .

Table 4–8   (Cont.) MTR_CUSTOMER Table Fields and Data Types

Field	Data Type
ATTRIBUTE50	NUMBER

4.1.5 Hot Picks

Hot picks are used by some Web sites to force recommendations from a particular group of items. For example, items that are daily specials may 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 unique group ID, the group name (LABEL), and a brief description of the group. The schema for this table has three fields; they are listed in Table 4–9.

Table 4–9   MTR_HOTPICK_GROUP Fields

Field	Data Type
ID	NUMBER PK
LABEL	VARCHAR2 (150)
DESCRIPTION	VARCHAR2 (400)

- MTR_HOTPICK lists the items in each hot pick group, arranged according to group ID (corresponding to the ID field in MTR_HOTPICK_GROUP). 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 in Table 4–10. 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.

Table 4–10   MTR_HOTPICK Fields

Field	Data Type
ITEM_ID	NUMBER
ITEM_TYPE	VARCHAR2 (30)
GROUP_ID	NUMBER

4.1.6 MTR_PROXY

MTR_PROXY is used to set up proxies for new items. When a new item is introduced, there will likely not be any mention of this item in any customer profile. As such, the predictive models will not be able to identify any rules for it, and no recommendations can be made for that item. Hence, OracleAS Personalization uses data about a similar existing product, called a *proxy*. The fields in the MTR_PROXY table are listed in Table 4–11.

Table 4–11   MTR_PROXY Fields

Field	Data Type
PROXY_ID	NUMBER
PROXY_TYPE	VARCHAR2 (30)
ITEM_ID	NUMBER PK
ITEM_TYPE	VARCHAR2 (30) PK

4.1.7 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 stores the navigational data corresponding to a user session. Each record contains a SESSION_ID which is linked to the CUSTOMER_ ID through the session table. In case there is no SESSION_ID collected, CUSTOMER_ID is used directly. One of the two must be not null. The ID and type of the item, and the time of the activity are also stored. The MTR_CUSTOMER_ NAV_DETAIL table is populated with data collected in the RE using the data collection APIs. If the application is already collecting this data, the table could be a view defined on the application's tables. The fields of the table are listed in Table 4–12.

Table 4–12   MTR_CUSTOMER_NAV_DETAIL Fields

Field	Data Type
SESSION_ID	NUMBER
TIMESTAMP	DATE
NAVIGATIONAL_VALUE	NUMBER
ITEM_ID	NUMBER
ITEM_TYPE	VARCHAR2 (30)
CUSTOMER_ID	VARCHAR2 (32)

- MTR_CUSTOMER_RATING_DETAIL stores rating data for customers. The RATING_VALUE attribute stores the value of the rating. This table is populated using the data collected in the RE through the data collection APIs. In cases where the rating data is already collected by the application, this table could be a view defined on the application's tables. The fields in this table are listed in Table 4–13.

Table 4–13   MTR_CUSTOMER_RATING_DETAIL Fields

Field	Data Type
TIMESTAMP	DATE
RATING_VALUE	NUMBER
CUSTOMER_ID	VARCHAR2 (32)
ITEM_ID	NUMBER
ITEM_TYPE	VARCHAR2(30)

- MTR_PURCHASING_DETAIL stores purchasing data on a per-session basis. Typically this data is collected by the application. CUSTOMER_ID provides navigability when SESSION_ID is not collectible. The table fields are listed in Table 4–14.

Table 4–14   MTR_PURCHASING_DETAIL Fields

Field	Data Type
ORDER_ID	NUMBER
SESSION_ID	NUMBER
TIMESTAMP	DATE

Table 4–14   (Cont.) MTR_PURCHASING_DETAIL Fields

Field	Data Type
PURCHASING_VALUE	NUMBER
ITEM_ID	NUMBER
ITEM_TYPE	VARCHAR2 (30)
CUSTOMER_ID	VARCHAR2 (32)

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

- MTR_VISITOR_NAV_DETAIL stores the navigation data corresponding to a visitor session. Each record contains a SESSION_ID which is linked to the VISITOR_ID through the session table. In case there is no SESSION_ID collected, VISITOR_ID is used directly. One of the two must be not null. The ID and type of the item, and the time of the activity are also stored. This table is populated with data collected in the RE using the data collection APIs. If the application is already collecting this data, this table could be a view defined on the application's tables. The fields in the MTR_VISITOR_NAV_DETAIL table are listed in Table 4–15.

Table 4–15   MTR_VISITOR_NAV_DETAIL Fields

Field	Data Type
SESSION_ID	NUMBER
VISITOR_ID	VARCHAR2 (32)
NAVIGATIONAL_VALUE	NUMBER
TIMESTAMP	DATE
ITEM_ID	NUMBER
ITEM_TYPE	VARCHAR2 (30)

- MTR_VISITOR_RATING_DETAIL stores rating data for visitors. The RATING_ VALUE attribute stores the value of the rating. This table is populated using the data collected in the RE using the data collection APIs. In cases where the rating data is already collected by the application, this table could be a view defined on those tables. The table's fields are listed in Table 4–16.

Table 4–16   MTR_VISITOR_RATING_DETAIL Fields

Field	Data Type
TIMESTAMP	DATE
RATING_VALUE	NUMBER
VISITOR_ID	VARCHAR2 (32)
ITEM_ID	NUMBER
ITEM_TYPE	VARCHAR2 (30)

If the application is not collecting data through the RE, these tables can be mapped to other tables in the application schema where such information is maintained.

4.1.8 Miscellaneous MTR Tables

The following tables are used internally by OracleAS Personalization:

- MTR_ATTR_NAME_TO_ID_MAP is used to speed up package building. It links an attribute name to an ID number.

- MTR_CONFIGURATION and MTR_INTERNAL_CONFIGURATION stores configuration information.

Table 4–17   MTR_CONFIGURATION Fields

Field	Data Type
DATA_TYPE	VARCHAR2 (32)
NAME	VARCHAR2 (128) PK
VALUE	VARCHAR2 (128)
DESCRIPTION	VARCHAR2 (100)

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

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

4.2 Recommendation Engine

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

- Tables specific to the model rules tables that are currently deployed to the RE:

  ATTR_ID_BIN_BOUNDARY
  HOTPICK
  HOTPICK_GROUP
  I_I_ANTECEDENT
  I_I_RULE
  P_I_CATEGORY_RULES
  P_I_ITEM_RULES
  TAXONOMY_CATEGORY
  TAXONOMY_CATEGORY_ITEM
  TAXONOMY_TRANS_CLOSURE

- Internal tables:

  RE_DEPLOYABLE_PACKAGE
  RE_DEPLOYABLE_PKG_CONTENTS
  RE_ERROR_TABLE
  RE_INTERNAL_CONFIGURATION
  RE_LOG
  RE_PROFILE_DATA
  RE_SCHEMA_ACCESS
  RE_ACTIVE_USER

- Data tables:

  RE_CURRENT_SESSION_DATA
  RE_RECOMMENDATION_DETAIL

- Diagnostic table:

  RE_MESSAGE_LOG

- Configuration table:

  The RE_CONFIGURATION table stores configuration information for REs. Its fields are listed in Table 4–18.

Table 4–18   RE_CONFIGURATION Fields

Field	Data Type	Description
LOG_LEVEL	INT	1=Errors and warnings. 2=Errors, warnings and notifications. 3=All plus trace
RE_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE 3=HIGH
REAPIRT_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE 3=HIGH
REAPIDEMO_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE 3=HIGH
UTIL_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE 3=HIGH
REAPIBATCH_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE 3=HIGH
TimeoutInterval	INT	Session timeout interval (in seconds). Default is 1800. This is modified by the Administrative UI.
TimeoutFlag	INT	Session timeout indicator (1=TRUE, 0=FALSE). Default is 1. This is modified by the Administrative UI.
DataSyncInterval	INT	Interval on which to synchronize customer data (in seconds). Default is 1800. This is modified by the Administrative UI.
SyncCustomerNavigationalData	INT	Is customer navigational data synchronized (boolean) Default is T. This is modified by the Administrative UI.
SyncCustomerRatingData	INT	Is customer rating data synchronized (boolean). Default is T. This is modified by the Administrative UI.
SyncVisitorNavigationalData	INT	Is visitor navigational data synchronized (boolean). Default is T. This is modified by the Administrative UI.
SyncVisitorRatingData	INT	Is visitor rating data synchronized (boolean). Default is T. This is modified by the Administrative UI.
SyncPurchasingData	INT	Is customer purchasing data synchronized (boolean). Default is T. This is modified by the Administrative UI.
SyncDemographicData	INT	Is customer demographic data synchronized (boolean). Default is T. This is modified by the Administrative UI.
ConnectionPoolSize	INT	Maximum size of JDBC connection pool per REProxyRT instance. Default is 1500.
ConnectionTimeOut	INT	Maximum period allowed for pooled connections stay alive. Default is 0.

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.

4.3 Mining Object Repository

Much of the administrative work done by OracleAS Personalization uses MOR tables and views. The MOR includes the following tables (partial list). These tables cannot be configured or modified by the user.

```
MOR_CONFIGURATION (in Table 4–19)
MOR_VISITOR_TO_BROWSER_REPORT
MOR_CROSS_SOLD_ITEMS_REPORT
MOR_DEPLOYABLE_PACKAGE
MOR_EFFECTIVENESS_REPORT
MOR_EMAIL_ADDRESS
MOR_ERROR_TABLE
MOR_INTERNAL_CONFIGURATION
MOR_MESSAGE_LOG
MOR_MINING_MODEL
MOR_MINING_RESULT
MOR_MTR_CONNECTION
MOR_RECOMMENDATION_ENGINE
MOR_RECOMMENDATION_REPORT
MOR_RECOMMENDATION_STRATEGY
MOR_RE_FARM
MOR_SCHEDULE_EVENT
MOR_SCHEDULE_ITEM
MOR_SCHEMA_ACCESS
MOR_TAXONOMY_TRANS_CLOSURE
MOR_TRAN_SUPERVISED_RESULT
```

Table 4–19   MOR_CONFIGURATION Fields

Field	Data Type	Description
NLS_LANGUAGE	String	Specifies the language used for OP database messages.
NLS_TERRITORY	String	Specifies the territory used for OP database messages.
MOR_USERNAME	String	MOR user name.
MOR_PASSWORD	String	MOR schema password.
MOR_DBALIAS	String	MOR database alias name.

Table 4–19   (Cont.) MOR_CONFIGURATION Fields

Field	Data Type	Description
MOR_SCHEMA	String	MOR schema name.
MOR_HOST_URL	String	MOR host URL.
MOR_PORT	String	MOR port number.
MOR_SID	String	Oracle SID name of MOR schema.
MOR_VERSION	String	MOR version number.
MAIL_PREFERENCE	String	Indicates type of e-mail notifications sent: MAILTEXT or MAILHTML.
ADMIN_EMAIL_ADDRESS	String	Indicates the E-mail address of the Oracle Personalization administrator.
scheduleItemGracePeriod	INT	This value indicates the number of minutes past the next start time for overdue schedule items.
MAXNUMPURCHASINGSESS	INT	The maximum number of Purchasing Sessions reports to keep per recommendation engine farm.
MAXNUMRECEFFREP	INT	The maximum number of Recommendation Effectiveness reports to keep per recommendation engine farm.
MAXNUMITEMIZEDRECEFFREP	INT	The maximum number of Itemized Recommendation Effectiveness reports to keep per recommendation engine farm.
NUMOFITEMSINITEMIZEDRECEFFREPORT	INT	The number of top ranked items in Itemized Recommendation Effectiveness reports.
buildEvents	INT	The maximum number of Build events to keep in log.
deployEvents	INT	The maximum number of Deploy events to keep in log.
reportEvents	INT	The maximum number of Report events to keep in log.
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.
MAIL_SERVER_HOST	String	This value indicates the hostname of the SMTP mail server used by OP.
MAIL_SERVER_PORT	String	This value indicates the port of the SMTP mail server used by OP.
LOG_FILE	String	Log file name
LOG_LEVEL	INT	0 - OFF, 1 - Internal Error + Error + Warning, 2 - All types that are logged for log_level = 1 + Notification, 3 - All types that are logged for log_level = 2 + Trace
ALGS_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE, 3=HIGH
DMAPI_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE, 3=HIGH
PAR_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE, 3=HIGH
TNB_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE, 3=HIGH
UI_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE, 3=HIGH
UTIL_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE, 3=HIGH
WFJAVA_TRACE	INT	0=OFF, 1=LOW, 2=MODERATE, 3=HIGH

5

Configuring OracleAS Personalization

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

This chapter

- lists the external OracleAS Personalization configuration parameters and their default values

- indicates the parameters that you can change and tells you 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 OracleAS Personalization configuration parameters reside on the system where Oracle is installed.

5.1 Configuring the OracleAS Personalization Scheduler

OracleAS Personalization allows users to schedule package builds, deployments, and reports. The OracleAS Personalization 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 OracleAS Personalization administrator. This address is used as the "return" address for email notifications. For example, if a user of the OracleAS Personalization 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 OracleAS Personalization.

5.2 Changing OracleAS Personalization 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 OracleAS Personalization Administration UI, indicated by UI in the **Change** column in the summary tables.

5.2.1 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
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

Table 5–1 (Cont.) Recommendation Engine Configuration Parameters

Parameter Name	Data Type	Default Value	Description	Change
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 [(Requires restart of Oracle AS Personalization)]

5.2.2 MOR Configuration Parameters

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

If the value in the **Change** 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	<username>	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	<sid>	MOR system ID; change value on install	N
MOR_VERSION	string	9.0.4	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

Table 5–2   (Cont.) MOR Configuration Parameters

Parameter	Data Type	Value	Description	Change
MAXNUMPUCHASEINGSESS	int	20	The maximum number of purchasing sessions reports to keep per recommendation engine farm	
MAXNUMRECEFFREP	int	20	The maximum number of recommendation effectiveness reports to keep per recommendation engine farm	
MAXNUMITEMIZEDRECEFF REP	int	20	The maximum number of itemized recommendation effectiveness reports to keep per recommendation engine farm	
NUMOFITEMSINITEMIZEDR ECEFFREPORT	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	
LOG_FILE	string	morlog.txt	Log file name	
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

Table 5–2   (Cont.) MOR Configuration Parameters

Parameter	Data Type	Value	Description	Change
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
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 OracleAS Personalization 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 OracleAS Personalization installation is that this is empty (blank). However, if it is edited, OracleAS Personalization 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, OracleAS Personalization defaults to AMERICAN. This value also specifies the language used for email notifications.

³ The default value for NLS_TERRITORY after OracleAS Personalization installation is that this is empty (blank). However, if it is edited, OracleAS Personalization 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, OracleAS Personalization defaults to AMERICA. This value specifies also specifies the language used for email notifications.

5.2.3 MTR Configuration Parameters

Table 5–3 describes the configuration parameters for the OracleAS Personalization 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 OracleAS Personalization 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.

5.3 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.

5.3.1 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 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.

5.3.2 Required RE Parameters

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

JOB_QUEUE_PROCESSES=2

5.3.3 Suggested RE Database Parameters (All Load Levels)

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

BUFFER_POOL_KEEP   (50 buffers)
SORT_AREA_SIZE (819200 bytes)
SORT_AREA_RETAINED_SIZE (819200 bytes)

5.3.4 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   Suggested 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
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

5.3.5 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   Suggested RE Configuration 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

5.4 MTR Installation and Configuration

The Mining Table Repository (MTR) database holds customer (demographic, purchasing, ratings, navigational data) and product data. Predictive 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

5.5 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.

5.5.1 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 OracleAS Personalization.

You configure the MTR connection using the OracleAS Personalization 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/mp/jlib`; this location is unknown to Oracle Application Server Containers for J2EE (OC4J). The jar should be copied to other Oracle Application Server instances that do not have OracleAS Personalization installed but need to access this API either directly or via the JSP tags.

Applications that use REAPI directly or through OracleAS Personalization JSP tags need to make `opreapi_rt.jar` and `dmtutil.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 OracleAS Personalization 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 OracleAS 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 Oracle is installed.

7.1 Creating and Removing Seed Data

OracleAS Personalization 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/mp/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 OracleAS Personalization administrator can run these scripts when appropriate.

7.2 Preparing to Build Packages with Seed Data

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

1. Install OracleAS Personalization 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 OracleAS Personalization 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 OracleAS Personalization 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 OracleAS Personalization

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

8.1 Stopping OracleAS Personalization

Follow these steps to stop OracleAS Personalization:

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

   Use the OracleAS Personalization 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 OracleAS Personalization 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 filename>` looks like:

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

1. Stop the OracleAS Personalization 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.

2. 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.

8.2 Starting OracleAS Personalization

You start OracleAS Personalization 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 OracleAS Personalization 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 OracleAS Personalization 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 OracleAS Personalization.

   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.

A

Troubleshooting

This appendix describes common problems that you might encounter when administering, using, and developing for OracleAS Personalization. It explains how to solve them and contains the following topics:

- Section A.1, "Problems and Solutions"
- Section A.2, "Need More Help?"

A.1 Problems and Solutions

This section describes common problems and solutions. It contains the following topics:

- Section A.1.1, "Changes Made Through opconfig Wizard Do Not Take Effect"
- Section A.1.2, "No Recommendations are Returned from the REAPI"
- Section A.1.3, "Cross-sell Recommendations are Not Returned"
- Section A.1.4, "Unable to Determine the Rules Resulting from a Package Build"
- Section A.1.5, "JSP Tags Return Tags that are Not Useful"
- Section A.1.6, "Error 404 Encountered When Using Administration GUI"
- Section A.1.7, "Scheduled Build Does Not Start"

A.1.1 Changes Made Through opconfig Wizard Do Not Take Effect

Changes made using the opconfig wizard are not effected.

Problem

After using the opconfig wizard to change configuration settings, these settings do not take effect.

Solution

After running the opconfig wizard to create the database schemas, you must restart the OC4J instance named `OC4J_BI_Forms` in order for the configuration changes made to the `web.xml` file are recognized. You can restart `OC4J_BI_Forms` using the Application Server Control Console.

A.1.2 No Recommendations are Returned from the REAPI

If an application invokes recommendations through the REAPI, none may be returned.

Problem

There may be occasions when no recommendations are returned from the REAPI.

Solution

There are valid instances where no recommendations are returned by the REAPI. In such cases, you need to design your application accordingly.

A.1.3 Cross-sell Recommendations are Not Returned

Cross-sell recommendations are not returned.

Problem

Sometimes no cross-sell recommendations are returned.

Solution

When using cross-sell recommendations, the `DataSourceType` and `InterestDimension` tuning settings must have the same value. This value can only be "PURCHASING" or "NAVIGATION".

No cross-sell recommendations are returned if the `DataSourceType` is "ALL". This is because navigation events can only predict clicks, and purchase events can only predict purchases in the current predictive association rules model.

A.1.4 Unable to Determine the Rules Resulting from a Package Build

Rules cannot be determined from a package build.

Problem

Unable to determine rules that result from a package build.

Solution

The number of data mining rules that were generated when a package was last built can be seen by clicking the Result icon of the appropriate package in the Packages tab. The page that appears is the build results page, which shows the statistics for the aggregated model and cross-sell model builds. These statistics include the number of data mining rules generated for each model.

A.1.5 JSP Tags Return Tags that are Not Useful

JSP tags return tags that are not useful.

Problem

The JSP tags return the item ID JSP tags, which are not appropriate for rendering a page. The results from a JSP tag need to be wrapped in a scriptlet to do lookups of what to display. Also, Java `FOR` loops, which are used to iteratively return the tags, do not belong in a Web page.

Solution

The JSP tags for OracleAS Personalization in the JSP tag library were designed before the Model-View-Controller paradigm became popular and should be avoided. In the future, the REAPI will be integrated into the Oracle Application Development Framework Runtime Libraries.

A.1.6 Error 404 Encountered When Using Administration GUI

Error 404 encountered when using the OracleAS Personalization administration GUI.

Problem

HTTP error 404 is encountered when the browser's "Back" button is used or the browser is navigated away from an OracleAS Personalization administration UI page.

Solution

To avoid 404 errors when using the OracleAS Personalization administration UI, do not use the browser's "Back" button or navigate away from the administration UI Web pages unless you log out of your session. If you encounter 404 errors when returning to the administration UI, you must exit and restart your browser to create a new session.

A.1.7 Scheduled Build Does Not Start

A scheduled build does not start.

Problem

In the event that a scheduled build does not occur (as evidenced when the status of a package build does not change from "Not built" to "Running"), a possible cause could be due to the failure of the `dbms_job` process. This process checks the build schedule to see when build jobs are scheduled to run.

Solution

If a scheduled package build fails to run, you may need to have your DBA check the status of the `dbms_job` processes in the MOR and RE accounts. A `dbms_job` process sets a broken flag if it fails to run 16 times in a row. This failure can be caused by a lack of system resources such as shared memory or disk space. Once the underlying resource problem is resolved, a provided script can be run to assist in resetting the broken flag. The name of this script is:

```
<ORACLE_HOME>/mp/admin/runbrokenjobs.sql
```

A.2 Need More Help?

In case the information in the previous section is not sufficient, you can find more solutions on Oracle *MetaLink*, `http://metalink.oracle.com`. If you do not find a solution for your problem, log a service request.

> **See Also:**
>
> - *Oracle Application Server Release Notes*, available on the Oracle Technology Network:
>   `http://www.oracle.com/technology/documentation/index.html`

Index

W

Wizard
  Personalization Schema Creation, 3-2