Oracle Financial Services Behavior Detection Platform
Installation Guide - Stage 1

Release 1.1
September 2012

Oracle Financial Services Behavior Detection Platform
Installation Guide - Stage 1

Release 1.1
September 2012

Document Control Number: 9MN16-0005
Document Number: IG-12-CTR-0005-6.1-1.1-01-Stage1

Oracle Financial Services Software, Inc.
1900 Oracle Way
Reston, VA 20190-4602

Document Number: IG-12-CTR-0005-6.1-1.1-01-Stage1
First Edition (September 2012)

Oracle Financial Services Software, Inc.
1900 Oracle Way
Reston, VA 20190-4602
Phone: (703)478-9000
Fax: (703)318-6340
Internet: www.oracle.com/financialservices

Revision History

Table 1 describes the revision history of *Oracle Financial Services Behavior Detection Platform Installation Guide*.

Table 1. Revision History

Date	Edition	Description
September 2012	First edition.	The first release of *Oracle Financial Services Behavior Detection Platform Installation Guide - Stage 1*.

Contents

List of Tables

About this Guide

This guide provides comprehensive instructions for installing and configuring the *Oracle Financial Services Behavior Detection Platform*.

This chapter focuses on the following topics:

- Who Should Use this Guide
- Scope of this Guide
- How this Guide is Organized
- Where to Find More Information
- Conventions Used in this Guide

Note: This document is one of the three documents needed for installation. This document contains information about the Stage 1 installation.

Who Should Use this Guide

The *Oracle Financial Services Behavior Detection Platform Installation Guide - Stage 1* is designed for use by the Application Installers and System Administrators. Their roles and responsibilities include the following:

- **Application Installer:** This user installs and configures the Oracle Financial Services Applications and the client-specific solution sets at a deployment site. This user also installs upgrades, and additional solution sets. It requires access to deployment-specific configuration information (For example, machine names, and port numbers).
- **System Administrator:** This user configures, maintains, and adjusts the system and is usually an employee of a specific Oracle Financial Services client. The System Administrator maintains user accounts and roles, archives data, and loads data feeds.

Scope of this Guide

This guide provides instructions for installing and configuring the *Oracle Financial Services Behavior Detection Platform* system, subsystem components, and the related third-party software needed to operate the system.

How this Guide is Organized

The *Oracle Financial Services Behavior Detection Platform Installation Guide Stage 1* includes following chapters:

- Chapter 1, *Introduction,* discusses activities that occur prior to deployment, typical installation configuration, and identifies all third-party software necessary to run the Oracle Financial Services Behavior Detection Platform.
- *Chapter 2, Installation Overview,* explains the installation process, describes the silent properties file, and provides an installation checklist for a typical deployment.
- Chapter 3, *Installing the Application Server,* explains how to install and configure the necessary components on the Database server in a typical deployment configuration.
- Appendix A, *Variables Used in the Silent Properties File,* lists and defines all variables in the sample properties files used to silently install the application.
- Appendix B, *Environment Variables (.cshrc) File,* provides a list of environment variables along with a description and an example of each for you to use as a guide in setting your system's environment variables.
- Appendix C, *Oracle Financial Services and Business Data Model Variables,* lists and defines all Oracle variables in the `db_variables.cfg` file needed to install the *Oracle Financial Services Behavior Detection Platform* and Business data models properly.
- Appendix D, *List of Acronyms and Abbreviations,* defines all of the acronyms and abbreviations that this guide uses.

Where to Find More Information

For more information about the Oracle Financial Services Behavior Detection Platform, refer to the following documents:

- *Oracle Financial Services Currency Transaction Reporting Data Interface Specification Guide, Release 1.1*
- *Oracle Financial Services Currency Transaction Reporting Administration Guide, Release 1.1*
- *Oracle Financial Services Currency Transaction Reporting Configuration Guide, Release 1.1*
- *Oracle Financial Services Currency Transaction Reporting User Guide, Release 1.1*
- *Oracle Financial Services Currency Transaction Reporting Release Notes, Release 1.1*

To find more information about Oracle Financial Services application and our complete product line, visit our web site at www.oracle.com/financialservices.

Conventions Used in this Guide

Table 2 lists the conventions used in this guide.

Table 2. Conventions Used in this Guide

Convention	Meaning
Italics	● Names of books, chapters, and sections as references ● Emphasis
Bold	● Object of an action (menu names, field names, options, button names) in a step-by-step procedure ● Commands typed at a prompt ● User input
`Monospace`	● Directories and subdirectories ● File names and extensions ● Process names ● Code sample, including keywords and variables within text and as separate paragraphs, and user-defined program elements within text
`<Variable>`	● Substitute input value

CHAPTER 1 Introduction

This chapter includes the following topics:

- Predeployment Information
- Deployment Environments
- Deployment Configuration
- Prerequisites

Predeployment Information

Before the deployment, workshops are held that identify the manner in which your organization conducts business and the type of information you need to capture. Information gleaned from these workshops helps Oracle Financial Services Software engineers determine the hardware configuration that best supports your business needs. Subsequently, a pre-installation checklist is generated that details this configuration. After this checklist is approved, the Oracle Financial Services Behavior Detection Installer can begin the deployment.

Deployment Environments

Typically, Oracle Financial Services Software clients have a test environment and a production environment for running Oracle Financial Services Behavior Detection software. Both environments include required servers and software components necessary to run the Oracle Financial Services Behavior Detection application.

The test environment not only serves as a backup to the production environment, but is also used to install and test new Oracle Financial Services Software and third-party software before installing it in the production environment. Troubleshooting installation issues in this environment prevent any interruptions in your daily operations.

Follow the instructions in this guide to install and configure in the test environment before installing the software in the production environment.

Deployment Configuration

As mentioned in the previous section, the approved pre-installation checklist determines the exact configuration for your deployment. This guide uses a typical deployment configuration to explain how the different servers, subsystems, and components interact.

System Hardware and Components

The hardware involved in an installation includes the following:

- Database server
- Data Ingestion/Behavior Detection (Application) server

The Oracle Financial Services Behavior Detection application runs with any subsystem installed on any one of these servers, or all subsystems can run on a single server.

Table 3 lists the Server Deployment of Server Configuration.

Table 3. Server Deployment Configuration

Server/Workstation	Subsystems and Components
Database server	● Financial Services Schema ● Oracle Financial Services Behavior Detection Metadata ● Case Management Schema (Objects are created during Stage 1 Installation, but populated at a later time) ● Business and Market Schemas ● Configuration Schema **Note:** *Oracle Financial Services Behavior Detection Platform* does not need to install software on the database server. The database schemas may be created by running DDL scripts from another server.
Data Ingestion/Behavior Detection (Application) server	● Scenarios ● Database Tools ● Ingestion Manager ● Behavior Detection Algorithms

Prerequisites

This section lists, by server, the prerequisite third-party products you must have installed to run Oracle Financial Services Behavior Detection. Any prerequisites that require Oracle Financial Services Behavior Detection-specific instructions for installation and configuration are noted in the Installation Details column in Table 4.

Database Server

Table 4 lists the software prerequisites required for the Database Server.

Table 4. Prerequisites for the Database Server

Category	Software	Installation Details
Database Software	Oracle 11gR2 Enterprise Edition with Partitioning	General instructions: ● Install and create a database instance ● Use guidelines in Appendix B, *Variables Used in the Silent Properties File*, on page 49, to configure the database

Database Configuration

The database instance must be established and started before the installation can proceed. Oracle Financial Services Behavior Detection provides scripts to create tablespaces and database files; however, to comply with your firm's database configuration standards, your Database Administrator may choose to create the tablespaces before running the Oracle Financial Services Behavior Detection data model creation scripts.

Oracle Database Parameters

Table 5 lists the Oracle database parameters that need to be sized individually for each customer installation.

Review the Oracle documentation for guidelines in setting the `SGA_TARGET`, `SGA_MAX_SIZE`, and `PGA_AGGREGATE_TARGET` parameters. The values for these memory parameters can vary significantly based on database server specifications and estimated data volume.

Table 5. Tunable Database Parameters

Tunable Database Parameters		Parameter Values		
Category	**Parameter Name**	**Type**	**Default**	**Oracle Financial Services Behavior Detection Recommended**

Table 5. Tunable Database Parameters

Parameters affecting database creation (not tunable through the init.ora file)	CHARACTER SET	string	AL32UTF8	AL32UTF8
	NLS_LENGTH_SEMANTICS	string	byte	char
	NLS_SORT	binary	binary	binary
	MAXDATAFILES	integer	254	
	MAXINSTANCES	integer	1	
	MAXLOGFILES	integer	32	
	MAXLOGHISTORY	integer	24794	
	MAXLOGMEMBERS	integer	2	4
	REDO LOG SIZE	integer	10M	500M
Parameters affecting I/O operation	DB_BLOCK_SIZE	integer	2048	8192
	DB_FILE_MULTIBLOCK_READ_COUNT	integer	8	32
	DB_FILES	integer	200	
	DISK_ASYNCH_IO	boolean	TRUE	
	TAPE_ASYNCH_IO	boolean	TRUE	
	DB_WRITER_PROCESSES	integer	1	4
Parameters affecting resource consumption and parallel operations	FAST_START_PARALLEL_ROLLBACK	string	LOW	HIGH
	LOG_BUFFER	integer	7M	10000000
	LOG_CHECKPOINT_INTERVAL	integer	0	10000
	LOG_CHECKPOINT_TIMEOUT	integer	1800	0
	OPEN_CURSORS	integer	50	4096
	PARALLEL_EXECUTION_MESSAGE_SIZE	integer	2148	8192
	PARALLEL_MAX_SERVERS	integer	10 * No of CPUs	Do not set or change
	PARALLEL_MIN_SERVERS	integer	0	Do not set or change
	PROCESSES	integer	150	600
	LARGE_POOL_SIZE	integer	0	512M
	PARALLEL_MIN_PERCENT	integer	0	Do not set or change
	PARALLEL_THREADS_PER_CPU	integer	2	
	SHARED_POOL_SIZE	integer	8M(32 Bit)/ 64M(64 Bit)	512M

Table 5. Tunable Database Parameters

	SHARED_POOL_RESERVED_SIZE	integer	5% of SHARED_POOL_SIZE	32M
Additional needed parameters	OPTIMIZER_MODE	string		CHOOSE
	COMPATIBLE	string		11.2.0 (for Oracle 11gR2)
	CURSOR_SPACE_FOR_TIME	string	FALSE	TRUE
	GLOBAL_NAMES	string	FALSE	TRUE
	PRE_PAGE_SGA	string	FALSE	TRUE
	UNDO_MANAGEMENT	string	AUTO	AUTO
	UNDO_TABLESPACE	string		Set as Per Site Values
	UNDO_RETENTION	integer	900	3600
	TIMED_STATISTICS	boolean	TRUE	TRUE
	OPTIMIZER_INDEX_CACHING	integer	0	
	OPTIMIZER_INDEX_COST_ADJ	integer	100	30

Data Ingestion/Behavior Detection Server

Table 6 lists the software prerequisites that the Data Ingestion server requires.

Table 6. Prerequisites for the Data Ingestion/Behavior Detection Server

Category	Software	Installation Details
Operating System	RHEL 5.3/5.5 or Sun Solaris 10	Install and configure.
Database Client Software	Oracle 11gR2 Client	Configure TNS names (in the `tnsnames.ora` file) to reach the database instance. **Note:** Verify installation of all Oracle utilities (that is, sqlldr).
Other Software	Sun Java Runtime Environment (JRE) jdk1.6 for use by Ingestion Manager and Database Tools	Install and configure OS-appropriate version.

General Environment

You must set your locale to UTF-8 locale. Specifying a locale depends on your data and the operating system installed on your system.

For example, Solaris OS: `setenv LANG en_US.UTF-8`

For example, Linux OS: `setenv LANG en_US.UTF-8`

You can determine the locale on your system using the `locale -a` command.

For an Oracle installation, set your Oracle `NLS_LANG` environment variable to an appropriate UTF-8 character set. For example, `setenv NLS_LANG AMERICAN_AMERICA.AL32UTF8`

Note: Before running the installation, check all the paths defined in the environment file (`.cshrc`) to ensure that they exist and are correct. Refer to Appendix C, *Oracle Financial Services and Business Data Model Variables*, on page 37, for a sample file.

CHAPTER 2 Installation Overview

This chapter explains the installation process, describes the silent properties file, and provides an installation checklist for a typical deployment. This chapter covers the following topics:

- Understanding the Installation Process
- Copying the Installation Files
- Running a Silent Installation
- Installation Checklist

Understanding the Installation Process

Use the following installation process for installing on one or multiple host machines:

1. Copy the installation files to a host machine (refer to *Copying the Installation Files*, on page 7, for more information).
2. Modify the sample silent properties file with information from your environment and save the file with the name of the host machine (refer to *Understanding the Silent Properties File*, on page 9, for more information).
3. Run the installation program to unpack the components and subsystems specified in the properties file for that host machine (refer to *Running a Silent Installation*, on page 9, for more information).
4. Configure the components and subsystems, as needed.

Repeat this process for each deployed host machine.

Copying the Installation Files

Before you can install the software, you must copy the appropriate software files from the installation CD to a working directory on a specific host machine. From this directory, run the installation program to create the Oracle Finance installation directory (referred to in this guide as `<Product Installed Directory>`) on the UNIX server.

This section discusses the following topics:

- Installation Files to be Copied
- Copying Methods for the Installation Files

Installation Files to be Copied

The installation CD contains directories which support different operating systems. For each host machine, find the directory appropriate for your environment (Linux, Solaris) and copy the following files to a working directory:

- `install.bin`: Installation program that contains all Oracle Financial Services Softwares.
- `installStage1.properties.sample`: Text file that supplies answers to variables that the installation program contains, when you are installing on an Application server. This file contains information from the Oracle Finance test environment and needs to be modified for your environment.

Copying Methods for the Installation Files

You can use one of the following methods to copy files:

- Mount the installation CD and use the copy command.
- Use File Transfer Protocol (FTP) to copy the file from a Windows machine to your UNIX server.

Note: If you use the File Transfer Protocol (FTP) method to copy the files, upon completion verify that all file sizes correspond to those provided with the installation CD.

Copying the Installation files

To copy files to a UNIX server by mounting the CD, follow these steps:

1. Mount the installation CD using the appropriate command.
2. Use the copy command (cp) to copy the files to a working directory.

Transfering Installation Files Using FTP

To transfer the files onto a UNIX server using FTP, follow these steps:

1. Copy the installation files to a directory on a Windows workstation. Copy the BIN files in binary mode and the sample property files in ASCII mode.
2. Use FTP to transfer the files from your workstation to the desired location.

Running a Silent Installation

When you run the Oracle Finance Installation Program, it silently queries the properties file for answers to variables embedded in its code that identifies what to install and where to install it. The Oracle Finance installation program unpacks the necessary files and lays down the directory structure for the specified subsystem and components.

You can install any subsystem or component on any server or you can install all items on a single server. Refer to *Understanding the Silent Properties File*, on page 9, for more information.

After the silent installation has finished, execute the following command:

```
<Product_Installed_Directory>/changePasswords.sh all
```

This prompts for the passwords of the required application users. The passwords entered are not output to the screen and the same password must be re-entered in order to be accepted. All passwords must be entered; it is not be possible to skip a password.

Understanding the Silent Properties File

The Oracle Finance installation program contains tokens that map to the variables that the silent properties file contains. During installation, the Oracle Finance installation program looks at the silent properties file to detect which tokens to be replaced with which values. This provides the flexibility to install any component or subsystem on any host machine.

The silent properties file contains the following sections:

- Variables that Define which Components to Install
- Variables Common to Multiple Components
- Variables for Individual Components

Refer to Appendix A, *Variables Used in the Silent Properties File*, on page 23 for the entire list of variables used in the silent properties file, along with definitions and examples of each.

Note: Please ignore all informatica installation variables in *Data Ingest Variables* section of Silent Properties file. Also ignore all installation variables in *Services Variables* section of Silent Properties file.

Variables that Define which Components to Install

The first section of the silent properties file identifies which components you want to install. Values for items within this group can be true or false. The Oracle Finance installation program reviews other sections of the file for answers to variables for those components marked as true. The program ignores variables for items marked as false. The following is a sample of code from this section of the properties file.

```
##############################################
### Select Components to Install
##############################################
```

```
(The following have values of "true" or "false")

DATABASE=true
DB_TOOLS=true
GOLDEN_DATA=false

BEHAVIOR_DETECTION=true
INGESTION_MANAGER=true
PATCH_INSTALLER=true
SERVICES=false
```

You should set up the properties file for each host machine ahead of time. Because the program ignores variables for components marked as false, you can fill out all the variables in one file, duplicate it for each host machine, and set false values for those components not necessary for a particular host machine.

To create multiple properties files, follow these steps:

1. Fill in the `installStage1.properties.sample` file (all variables) completely.
2. Save the file with the name of a host machine (for example, `installStage1.properties.sample.appserver` for the Application server).
3. Repeat Step 2 for each host machine.
4. Open each properties file and set the appropriate `true or false` values.

Variables Common to Multiple Components

This section of the silent properties file lists variables that are more than one subsystem uses. The following is a sample of code from this section of the properties file:

```
##############################################
# Common Variables
# Variables used in several subsystems.
##############################################

db_type=Oracle

base_country=US
data_ingest_user=INGEST_USER
db_inst=T5O9S10
db_home_dir=/kds/sparc-sun-solaris10/pkg/oracle/product/10.2.0
jdk_home=/kds/sparc-sun-solaris10/pkg/jdk1.6

kdd_name=KDD
market_schema_owner=MARKET
tns_admin=/kds/oracle/net
oracle_sid=T1O9S8
default_jurisdiction=AMEA
```

Variables for Individual Components

The last section of silent properties file lists each component and the variables it uses, refer Appendix C, *Oracle Financial Services and Business Data Model Variables*, on page 37, for variable names). The Oracle Finance installation program uses these variables only if the specific component has its value set to true; otherwise, the program ignores them.

Installation Checklist

Table 7 provides a checklist that guides you through the installation process and provides the page numbers for the location of each step within this guide. Perform the listed tasks, in order, to complete the process successfully. Print the checklist to use as a reference during the installation process.

Table 7. Installation Checklist

#	Task	Done
1.	Verify your configuration; refer to *System Hardware and Components*, on page 2, for more information.	☐
2.	Verify that all prerequisite softwares have been installed; refer to *Prerequisites*, on page 3, for more information.	☐
3.	Start copying files; refer to *Copying the Installation Files*, on page 14, for more information.	☐
4.	**Note:** Before running the Silent Install, check all the paths defined in the environment file (.cshrc) to ensure that they exist and are correct. Refer to Appendix C, *Oracle Financial Services and Business Data Model Variables*, on page 37, for variable definitions. Start the silent installation; read about running the silent properties file in *Running a Silent Installation*, on page 9, then proceed with the silent installation following the appropriate instructions in the body of this document.	☐
5.	Execute the Password Manager Utility. **Note:** If you are installing for the first time, select **All Options**.	☐
6.	Install the data model; refer to *Installing the Data Model*, on page 16, for more information.	☐
7.	Configure the Data Ingestion subsystem; refer to *Configuring the Data Ingestion Subsystem*, on page 19, for more information.	☐
8.	Execute the Password Manager Utility. **Note:** If you are installing for the first time, select **All Options**.	☐
9.	(Optional) Install multiple instances of Data Ingestion; refer to *Installing Multiple Instances of the Data Ingestion Subsystem*, on page 20, for more information.	☐

CHAPTER 3 Installation Activities

This chapter includes following topics:

- Installing the Application Server
- Installing Components
- Running the Installation Program on the Application Server
- Installing the Data Model
- Configuring the Data Ingestion Subsystem
- Installing Multiple Instances of the Data Ingestion Subsystem

This chapter includes the examples for a typical configuration. The following sections provide a high-level list of tasks that you need to perform.

Installing the Application Server

Installing the Application Server involves the following procedures:

- **Installing Software**
  - Installing Components
  - Running the Installation Program on the Application Server
- **Setting up the Database** - Installing the Data Model
- **Setting up Ingestion Manager**
  - Configuring the Data Ingestion Subsystem
  - Installing Multiple Instances of the Data Ingestion Subsystem

The following sections describe these procedures.

Installing Components

This section lists the high-level tasks you need to perform, in addition to a reference to the specific section and page where the tasks are explained.

1. Run the Installation Program on the Application Server (refer to *Running the Installation Program on the Application Server*, on page 14, for information).
2. Install the data models (refer to *Installing the Data Model*, on page 16, for information).
3. Configure the Data Ingestion subsystem (refer to *Configuring the Data Ingestion Subsystem*, on page 19, for information).
4. Install multiple instances of the Data Ingestion subsystem to improve performance (refer to *Installing Multiple Instances of the Data Ingestion Subsystem*, on page 20, for information).

Running the Installation Program on the Application Server

Run the Installation Program to create the installation directory on the Application Server, and unpack the database files that you need to configure and run the database.

Use the following procedures to run the Installation Program:

1. Copying the Installation Files
2. Creating the Silent Properties File
3. Running the Silent Installation

Copying the Installation Files

You must run the Installation Program from the host machine where you want the installation directory to reside. Copy the following installation files from the installation CD or media pack to a working directory on the Application Server:

- `install.bin`
- `installStage1.properties.sample`

Refer to *Copying the Installation Files*, on page 14, for information about these files and for help in copying them.

Creating the Silent Properties File

Create the silent properties file to inform the Installation Program about your deployment environment, and to identify the database components you want to install.

Before you run the installation program, ensure the following:

- Your path environment variable includes the `<Product Installed Directory>`.
- Your path environment variable contains the current directory (“.”).

  Note: You must include the period in the path environment variable to populate the `behavior_detection/algorithms` subdirectory tree successfully.
- You have created a user account under which the Oracle Financial Services application Processes run.

Refer to *Understanding the Installation Process*, on page 7, for more information about the silent properties file.

To create the Application Server properties file, follow these steps:

1. Copy the `installStage1.properties.sample` file with an appropriate name for the Application Server properties.

   For example,

   ```
   cp installStage1.properties.sample install.properties.appserver
   ```
2. Open the install.properties.appserver file for editing.

3. Scroll down to the **Select Components to Install** section of the file and modify the variable values, as shown in Table 8.

Table 8. Application Server Components to Install

Variable	Value
`USER_INSTALL_DIR`	`Set as appropriate for your environment`
`DATABASE`	`true`
`DB_TOOLS`	`true`
`GOLDEN_DATA`	`false`
`BEHAVIOR_DETECTION`	`true`
`INGESTION_MANAGER`	`true`
`PATCH_INSTALLER`	`true`
`SERVICES`	`false`

4. Enter variable values in the following sections of the `install.properties.appserver` file:
   - General Installation Variables (includes several small sections)
   - Common Variables
   - Database Variables
   - Algorithm Variables
   - Data Ingest Variables
   - Logging Variables
   - Behavior Detection Variables
   - Services Variables

   Refer to *Appendix B, Environment Variables (.cshrc) File*, on page 35, for detailed information about each variable.
5. Save and close the `install.properties.appserver` file.

Running the Silent Installation

After you copy the necessary files to the Application Server and create the silent properties file, you can run the silent installation.

To run the silent installation, follow these steps:

1. Change the directory to the location of the Installation Program.
2. Type the following command to run the silent installation:
   `install.bin -f <path_to_file>/install.properties.appserver`

   Where:

   `<path_to_file>/install.properties.appserver` is the directory path and file name for the properties file you edited in the *Creating the Silent Properties File*, on page 14.
3. Run the following command:
   `<Product_Installed_Directory>/changePasswords.sh all`

   Then enter the passwords for the users as prompted by the Password Manager Utility.

The Silent Installation ensures that the Oracle Financial Service software is installed on your system.

Installing the Data Model

Installing Data Model is a two-step process. Each step consists of running the Database Builder Utility and providing a configuration file, which identifies the specific scripts to be run.

To install Data model, follow these steps:

1. Installs the necessary system objects (tablespaces, roles, and users) needed for Oracle Financial Services application. Refer to the note below if the tablespaces are built manually by the Database Administrator prior to installing the Oracle Financial Services application Data Model.
2. Installs all the Oracle Financial Services application Data Model objects for the various Oracle Financial Services application users.

   Note: Before installing the Data Model on the Application Server, verify that you have:

   - Created the Oracle instance for the Data Mart
   - Mounted and opened the database to users
   - Started the Oracle Listener
   - Created the necessary `ORACLE_SID` in the `tnsnames.ora` file

   To install Data Model, you need `db_variables.cfg properties` file. This file describes your database environment and resides in the `<Product Installed Directory>/database/db_tools/mantas_cfg` subdirectory.

Note: By default, the installation process creates tablespaces. If the database administrator has created tablespaces manually, you must modify the `<Product Installed Directory>/database/db_tools/mantas_cfg/mts_system_install.cfg` file and comment out the lines (that is, insert a pound sign [#] at the start of the line) that begin with `<Product Installed Directory>/database/mantas_schema/ddl/pfm_create_tablespaces.sql` and `<Product Installed Directory>/database/bus_mkt_schema/ddl/or_create_tablespaces.sql` before using the following procedure.

For database schema passwords, certain characters are restricted. The following special characters are allowed:. `"@", "-", "_", "\", "/", ":", "."`

Installing the System Objects

To install the System Objects, follow these steps:

1. Change to the `<Product Installed Directory>/database/db_tools/mantas_cfg directory`.
2. Edit the `db_variables.cfg` file.

   Refer *Table 11 on page 37,* lists all the variables, definitions, and examples.

   Note: If the database is created using DBCA (Database Configuration Assistant) then USERS table space will be created by default.

   Follow these steps before executing Database Builder Utility:

   a. Open `<Product Installed Directory>/database/mantas_schema/ddl/pfm_create_tablespaces.sql`

   a. Remove below SQL

   ```
   CREATE TABLESPACE USERS DATAFILE '&&user_data_file' SIZE
   &&user_data_size EXTENT MANAGEMENT LOCAL SEGMENT SPACE
   MANAGEMENT AUTO ONLINE

   ;
   ```

3. Run the Database Builder Utility to create system objects using the following command:

   ```
   ../bin/run_dbbuilder_utility.sh mts_system_install.cfg
   ```

   The system prompts you for the following information:

   - Username for the 'system' user (either a system user or a DBA user is required
   - Password for the 'system' user
   - Re-enter the password for the 'system' user

     The Database Builder Utility now parses the database installation scripts to substitute any variables in the scripts with the installation specific values.

The system prompts you to enter and confirm the password values for the following variables:

- `mantas_schema_pwd`
- `config_pwd`
- `kdd_schema_pwd`
- `web_pwd`
- `server_pwd`
- `tools_pwd`
- `altio_pwd`
- `report_pwd`
- `business_schema_pwd`
- `market_schema_pwd`
- `ingest_user_pwd`
- `db_util_pwd`
- `case_schema_pwd`

The values provided are used as passwords for the users when they are created. As a security measure, the values entered are not displayed back on the console. If the password value and confirmation value are not the same, you are re-prompted to enter the password.

The list of scripts being executed and the completion status for each are displayed on the console. After all the scripts have finished executing, the console displays a message indicating that the Database Builder Utility has run successfully.

Note: If a script fails, the error information is captured in the log file `<Product Installed Directory>/database/db_tools/logs/db_builder.log`.

When the system objects have been successfully installed, you can install the Oracle Financial Services application Data Model objects.

Installing the Data Model Objects

To install the Data Model objects, follow these steps:

1. Change to `<Product Installed Directory>/database/db_tools/mantas_cfg subdirectory`.
2. Run the Database Builder Utility to create the Oracle Financial Services application Data Model Objects using the following command:

   ```
   ../bin/run_dbbuilder_utility.sh mts_pdm_install.cfg
   ```

   The system prompts you for the following information:

   - Password for the 'BUSINESS' user
   - Username for the 'system' user (either a system or a DBA user is required)
   - Password for the 'system' user
   - Username for the 'kdd' user

- Password for the 'kdd' user
- Password for the 'MANTAS' user
- Password for the 'MARKET' user
- Password for the 'CASE' user

The list of scripts being executed and the completion status is displayed on the console. After all the scripts have completed, the console displays a message indicating that the Database Builder Utility has run successfully.

If a script fails, the error information is captured in the log file `<Product Installed Directory>/database/db_tools/logs/db_builder.log`

Configuring the Data Ingestion Subsystem

You may also choose to modify other ingestion configuration files, however, it is not required to run the system.

Optional: Modifying Additional Configuration Files

You can modify the following additional configuration files (although it is not a requirement that you modify them to run the system):

- **`DataIngest.properties`**: The `DataIngest.properties` file (located in the `ingestion_manager/config` subdirectory) contains the variable values you specified in the silent properties file.
- **`DataIngest.xml`**: The `DataIngest.xml` file (which resides in the `ingestion_manager/config` subdirectory) contains configuration settings that are required to configure each Ingestion Management runtime component, for example, setting up and configuring the number of threads used by each component. For more information on configuring this file, refer to the *Oracle Financial Services Currency Transaction Reporting Administration Guide*.

Installing Multiple Instances of the Data Ingestion Subsystem

Performance of the server can adversely affect the amount of time required to process large quantities of business or market data. To improve server performance, you can use this optional procedure to install the software on multiple servers (no need to install multiple instances on the same server). Attempt this procedure only if it has been demonstrated that the performance on a single server is inadequate.

This procedure can be used specifically for the Market Data Server (MDS), which processes Inside Quote, Market Center Quote, and Reported Market Sales data files, and the Firm Data Transformer (FDT), which processes Order, Trade Execution, and Open Order files. Refer to the *Oracle Financial Services Currency Transaction Reporting Administration Guide* for more information about these processes.

All previously specified files are referred to collectively as Trading Data for the rest of this section.

Understanding the Process

A single *primary* instance is installed to process all reference data files (files not referred to as Trading Data). Multiple *secondary* instances are then installed to evenly distribute the processing of the Trading Data files. Each instance is installed on a separate server. The *primary* instance is used to run all the data ingestion processes, while the *secondary* instance is used to run only those processes that are relevant to the Trading Data files.

After all instances have been installed, the Trading Data files must split into non-overlapping subsets by the Security Short Name (stock symbol). These subsets should be as evenly distributed as possible; each subset is delivered to an instance's inbox directory. It is important that a security being processed by a given instance continue to be processed by that instance for as long as the instance is in use. In other words, once data for AMZN is sent to instance A, it must always be sent to instance A. Then the system is prepared to begin Data Ingestion. For any assistance for this process, contact Oracle Financial Services application Engagement Representative.

To create multiple ingestion instances, follow these steps:

1. Determine how many additional servers are required to meet performance requirements.
2. Create a primary instance by running a silent installation on the Data Ingestion server.
3. Repeat Step 2 to create as many secondary instances as needed, making sure to change the `USER_INSTALL_DIR` variable within the silent properties file to a different value for each instance, if required.
4. Create an algorithm to evenly and consistently split the Trading Data files among the instances.

Installing multiple instances does not apply to the Informatica components of Ingestion Manager. To save space, you can delete the contents of the `<Product Installed Directory>/ingestion_manager/informatica` subdirectory in your second installation.

You have completed the installation and configuration of multiple instances of the Data Ingestion subsystem. To begin data ingestion, refer to the *Oracle Financial Services Currency Transaction Reporting Administration Guide*.

APPENDIX A Variables Used in the Silent Properties File

This appendix identifies and defines all product variables within the properties files that Oracle Financial Services installation program uses to perform the silent installation.

Files that the Installation CD Contains

The installation CD contains the `installStage1.properties.sample` properties file that is installed on a Web Application server. This file is prepopulated with default values that are used in the Oracle Financial Services test environment.

Use this appendix as a reference to populate the file.

Table 9. Silent Mode Installation Variables

Installation Variable	
General Installer Parameters	
`INSTALLER_UI`	Specifies whether the installer prompts the user for field values or reads them from file. The values should be read only from the file. **Value:** `INSTALLER_UI=silent`
Locale	
`USER_LOCALE`	Sets the system locale. **Default:** `USER_LOCALE=en`
Product Installation Directory	
`USER_INSTALL_DIR`	Identifies the directory in which the Oracle Financial Services software is installed. If the directory does not exist, the installer creates it.
Select Components to Install	

`BEHAVIOR_DETECTION`	Installs the Behavior Detection subsystem if set to *true*.
	Valid values:
	`true`
	`false`

`DATABASE`	Installs the database subsystem and the scripts to create the data model if set to *true*.
	Valid values:
	`true`
	`false`

`DB_TOOLS`	Installs the database tools if set to *true*.
	Valid values:
	`true`
	`false`

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
GOLDEN_DATA	Installs the Oracle Financial Services test data if set to *true*. **Valid values:** true false
INGESTION_MANAGER	Installs the Data Ingestion components if set to *true*. On UNIX, it installs the scripts and parameter files, as well as Informatica repository files and scripts that Informatica uses directly. **Valid values:** true false
PATCH_INSTALLER	Installs the Patch utility if set to *true*. **Valid values:** true false
SERVICES	Installs the Services if set to *true*. **Valid values:** true false
Common Variables	
base_country	ISO country code to use during data ingestion to record institution-derived geography risk on parties on transactions that are internal to the Oracle Financial Services client. **Example:** base_country=US
data_ingest_user	Name of the data ingest user. **Example:** data_ingest_user=INGEST_USER
db_home_dir	Full path to the Oracle software installation. **Example:** db_home_dir=/kds/sparc-sun-solaris10/pkg/oracle/product/10.2.0
db_inst	Name of the database instance to which the software points is the same across all subsystems. **Example:** db_inst=T5O9S10
db_type	Type of database used. **Example:** db_type=Oracle
default_jurisdiction	Jurisdiction to assign the derived entities and derived addresses. **Example:** default_jurisdiction=AMEA

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`jdk_home`	Installation directory path of the Java Development Kit (JDK). All components except Alert Management use this variable. This variable can point to either the JDK or the JRE, as needed. **Example:** `jdk_home=/kds/sparc-sun-solaris10/pkg/jdk1.6.0` **Note:** Use the `alert_management_jdk_home` variable to identify the installation directory path of the JDK version that the Alert Management subsystem is to use.
`kdd_name`	Name of the KDD Schema owner. **Example:** `kdd_name=KDD`
`market_schema_owner`	Schema where market data resides. **Example:** `market_schema_owner=MARKET`
`managing_tp_from_ui`	Its default value is N. In this case, the Informatica workflow for trusted pair risk review is not executed. If the value is set to Y Informatica workflow for trusted pair risk review is also executed. This variable should not be set to Y when trusted pairs are loaded through the DIS file. **Note:** Oracle Financial Services application supports only one method of managing trusted pairs per installation. Clients may elect to create and manage trusted pairs through the loading of trusted pairs via a DIS file OR utilize the Oracle Financial Services application User Interface for creation and management of trusted pairs. However, both the methods should not be utilized concurrently.
`oracle_sid`	Database connect string (SID) for the Oracle instance. **Example:** `oracle_sid=T1O9S8`
`tns_admin`	Full path to the directory where the `tnsnames.ora` file resides. Typically, it is in the `network/admin` subdirectory of the Oracle installation. **Example:** `tns_admin=/kds/oracle/net`
Database Variables	
`actions_foureyes_approval`	Indicator of whether Oracle Financial Services application installs the database meta data for alerts to accommodate *standard* or *four eyes* actions. **Valid Value** **Description** Y Installs the four-eyes action versions. N Installs the standard action versions. **Example:** `actions_foureyes_approval=N`
`analyst_pool_max_conn`	Maximum number of connections for Analyst Pool.
`business_schema_owner`	Schema where business data resides. **Example:** `business_schema_owner=BUSINESS`

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`case_schema_owner`	Account that the Web Application uses to connect to the database. **Example:** `case_schema_owner=CMREVMAN`
`config_user`	Account that the Web Application uses to connect to the database. **Example:** `config_user=REVELEUS`
`data_loader_role`	Role that has privileges to load data into the Business and Market schemas. **Example:** `data_loader_role=DATA_LOADER`
`data_reader_role`	Role that has privileges to read data into the Business and Market schemas. **Example:** `data_reader_role=DATA_READER`
`db_port`	Port on which the database is listening. Oracle's default port is 1521. **Example:** `db_port=1521`
`db_server`	Hostname of the server on which the database is running. **Example:** `db_server=dbhost`
`db_utils_db_user`	Database user that the database utilities uses to connect to the database to run the functions such as loading scenarios, starting a batch, and setting the system date. **Example:** `db_utils_db_user=DB_UTIL_USER`
`extract_db_connect`	Connect string (SID) for the database from which scenarios are extracted (for the scenario migration utility). This is an optional installation parameter; if supplied, it typically points to a development or QA environment. **Example:** `extract_db_connect=Financial Services_DEV`
`jdbc_driver`	JDBC driver use `Oracle.jdbc.OracleDriver.`
`jdbc_url`	JDBC URL to database. You can use either: ● OCI (`jdbc:oracle:oci:@<DBNAME>`) or ● Thin (`jdbc:oracle:thin:@<server>:<port>:<database name>`)
`kdd_algorithm_role`	Database role to which the `kdd_algorithm_user` is assigned. Rights to objects in the database are granted to the role, as opposed to the user. **Example:** `kdd_algorithm_role=KDD_ALGORITHM`

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`kdd_algorithm_user`	Database user that the Behavior Detection subsystem uses to connect to the database to run Behavior Detection jobs. **Example:** `kdd_algorithm_user=KDD_ALG`
`kdd_altio_user`	Database user that Altio applications in Oracle Financial Services application Active Pages use to connect to the database. **Example:** `kdd_altio_user=KDD_ALTIO`
`kdd_analyst_role`	Database role to which the `kdd_web_user` is assigned. Rights to objects in the database are granted to the role, as opposed to the user. **Example:** `kdd_analyst_role=KDD_ANALYST`
`kdd_loader_role`	Role that has privileges to load data into the KDD schemas. **Example:** `kdd_loader_role=KDD_LOADER`
`kdd_mnr_role`	Database role to which the `kdd_mnr_user` is assigned. Rights to objects in the database are granted to the role, as opposed to the user. **Example:** `kdd_mnr_role=KDD_MINER`
`kdd_mnr_user`	Data miner account used to connect to the database. **Example:** `kdd_mnr_user=KDD_MNR`
`kdd_reader_role`	Role that has privileges to read data in the KDD schemas. **Example:** `kdd_reader_role=KDD_READER`
`kdd_schema_owner`	Database schema that Oracle Financial Services uses to store basic reference metadata for the operation of the Behavior Detection algorithms. No process or user logs into this schema directly. **Example:** `kdd_schema_owner=KDD`
`kdd_web_user`	Database user that the Web Application uses to create connection pools to the database. **Example:** `kdd_web_user=KDD_WEB`
`mantas_loader_role`	Role that has privileges to load data into the Oracle Financial Services schemas. **Example:** `mantas_loader_role=MANTAS_LOADER`
`mantas_reader_role`	Role that has privileges to read data in the Oracle Financial services schemas. **Example:** `mantas_reader_role=MANTAS_READER`

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`mantas_schema_owner`	Schema where Oracle Financial Services data resides. **Example:** `mantas_schema_owner=MANTAS`
`miner_pool_max_conn`	Maximum number of connections for Miner Pool
`report_user`	Schema provided to users to support reporting applications built on top of Oracle Financial Services. **Example:** `report_user=KDD_REPORT`
`smtp_host`	Hostname of the e-mail gateway to be used by the database for e-mail notifications. **Example:** `smtp_host=mailhost.domain.com`
`config_user`	Account that the Web Application uses to connect to the database. **Example:** `config_user=REVELEUS`
Services Variables	
`services_jdbc_url`	JDBC URL to database. **Example:** `jdbc:oracle:thin:@benji.mantas.com:1521:D1O10N93`
`alert_inheritance`	Variable that determines if a user inherits alerts owned by a pool when they take action or view details on that alert. **Valid values:** **Description** `true` User does not inherit alerts that a pool owns `false` User inherits alerts that a pool owns
`active_pages_server_url`	Root URL for the Active Pages installation. **Example:** `active_pages_server_url=https://10.184.134.133:9450/altio515`

`base_alert_default_configuration_set`	Default configuration set to be available on the preferences tab for alerts.
	Valid values:
	`Anti-Money Laundering`
	`Broker Compliance`
	`Standard`
	`Trading Compliance`

`case_inheritance`	When Case Management is enabled, this variable determines if a user inherits cases owned by a pool when they take action or view details on that case. **Valid values:** **Description** `true` User does not inherit cases that a pool owns `false` User inherits cases that a pool owns

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`case_management_enabled`	Variable determines whether Case Management is enabled or not. **Valid values:** **Description** `true` Case Management is disabled `false` Case Management is enabled
`case_prefix`	When Case Management is enabled, this variable is used as the prefix for the case ID. **Example:** `case_prefix=CA`
`mts_service_http_port`	Variable that defines the port on which the Alert Processing Service is running. **Example:** `mts_service_http_port=8070`
`mts_service_shutdown_port`	Variable that defines the shutdown port of the Alert Processing Service. **Example:** `mts_service_shutdown_port=8071`
`post_alert_default_owner`	The default owner for a case that falls into one of the following categories: ● The default owner for a case that resulted from a promote-to-case action of the Correlate Alert processing step requested in a PostAlert service operation call if the Assign Case processing step is not explicitly requested ● The default owner to supply to the Assign Alert processing step if it is requested but no DefaultOwner parameter is passed in with the request **Example:** `post_alert_default_owner=MANTAS`
`post_alert_default_case_owner`	The default owner for a case that falls into one of the following categories: ● The default owner for a case that resulted from a promote-to-case action of the Correlate Alert processing step requested in a PostAlert service operation call if the Assign Case processing step is not explicitly requested ● The default owner to supply to the Assign Case processing step if it is requested but no DefaultOwner parameter is passed in with the request. **Example:** `post_alert_default_case_owner=MANTAS`
`post_alert_correlate_alert_to_alerts`	Default flag indicating whether or not to perform the Alert-to-Alert correlation functionality of the Correlate Alert processing step requested in a Post Alert service operation call. This parameter is used if no CorrelateAlertToAlerts parameter is passed in with the request. **Example:** `post_alert_correlate_alert_to_alerts=Y`

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`post_alert_default_score_correlations`	Default flag indicating whether or not to perform the scoring action on a correlation that resulted from the Alert-to-Alert correlation functionality of the Correlate Alert processing step requested in a Post Alert service operation call if no ScoreCorrelations parameter is passed in with the request. **Example:** `post_alert_default_score_correlations=Y`
`post_alert_default_promote_to_case`	Default flag indicating whether or not to perform the case-promotion action on a correlation that resulted from the Alert-to-Alert correlation functionality of the Correlate Alert processing step requested in a Post Alert service operation call if no PromoteCorrelationsToCases parameter is passed in with the request. **Example:** `post_alert_default_promote_to_case=N`

`post_alert_default_promote_single_alert`	Parameter used to specify whether the default behavior of the Alert Management Service should be to allow or disallow for the promotion of single alerts to cases (rather than multiple alerts that are grouped as part of a correlation).
	Valid values:
	`Y`
	`N`
	Example:
	`post_alert_default_promote_single_alert=Y`

`user_initiated_alert_default_score`	The default score to be assigned when the user creates a user-defined alert using the Research Workflow from the Oracle Financial Services user interface. This value can be up to 100. **Example:** `user_initiated_alert_default_score=10`
Behavior Detection Variables	
`alg_job`	Number of concurrent detection jobs that can be run. The algorithms engine queues jobs if more requests than this number occur. **Example:** `alg_job=20`
`NLS_LANG`	Language of the base country. **Example:** `NLS_LANG=AMERICAN_AMERICA.AL32UTF8`
Data Ingestion Variables	
`acct_trust_from_cust`	Flag that tells the Data Ingestion process whether the account risk should be exempt or trusted based on the exempt or trusted status of the customer's risk. The value Y indicates that the Oracle Financial Services mapping will consider the customer exempt and trust risk in calculating the account risk. The value N indicates that the Oracle Financial Services mapping will NOT consider the customer exempt and trust risk in calculating the account risk. **Valid Value** — **Description** `Y` — Consider the customer exempt and trust risk in calculating the account risk. `N` — Do not consider the customer exempt and trust risk in calculating the account risk.

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`fuzzy_log_dir`	Directory where the Fuzzy Matcher utility places the log files that it generates. This utility is packaged with the Data Ingestion server. **Example:** `fuzzy_log_dir=/opt/software/informatica/pc/SessLogs`
`inf_log_dir`	Directory where data ingestion stores log files. This must be set to the same directory to which the Informatica `$PMSessionLogDir` server variable refers. **Example:** `inf_log_dir=/opt/software/informatica/pc/SessLogs`
`inf_param_dir`	Directory where Informatica stores parameter files. This must be the same as `$PMRootDir/ParamFiles`. **Example:** `inf_param_dir=/opt/software/informatica/pc/ParamFiles`
`inf_port`	Port on which the pmserver process for the repository runs. This value is set for the repository from the workflow manager tool. **Example:** `inf_port=4001`
`inf_script_dir`	Directory where Informatica stores script files. This must be the same as `$PMRootDir/Scripts`. **Example:** `inf_script_dir=/opt/software/informatica/pc/Scripts`
`inf_server`	Fully qualified host name on which Informatica is running. **Example:** `inf_server=dbhost.domain.com`
`inf_src_dir`	Directory where Informatica source files reside. Set this variable to the same directory to which the Informatica `$PMSourceFileDir` server variable refers. **Example:** `inf_src_dir=/opt/software/informatica/pc/SrcFiles`
`inf_tmp_dir`	Directory where Informatica stores temporary files. This must be the same as `$PMRootDir/Temp`. **Example:** `inf_tmp_dir=/opt/software/informatica/pc/Temp`
`inf_user`	User that is defined within the repository with privileges to run the sessions and workflows. **Example:** `inf_user=administrator`
`infa_bin_dir`	Installation directory of Informatica. This directory must contain the `pmserver` executable. **Example:** `infa_bin_dir=/opt/software/informatica/pc/server/bin`
`infile_dir`	Directory where Informatica target files reside. Set this variable to the same directory to which the Informatica `$PMTargetFileDir` server variable refers. **Example:** `infile_dir=/opt/software/informatica/pc/TgtFiles`

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`informatica_domain`	The Informatica domain where the Oracle Financial Services repository is running. This is an Informatica 8 concept. Refer to the Informatica documentation for more information. **Example:** `informatica_domain=DOM_<ServerName>`
`informatica_service`	The Informatica service that supports the Oracle Financial Services repository. This service executes workflows. This is an Informatica 8 concept equivalent to the Informatica 7's server. Refer to the Informatica documentation for more information. **Example:** `informatica_service=Financial Services60_SRVC`
`num_split_files`	Number of files into which a single input file should be divided for processing. This value cannot exceed 10. **Example:** `num_split_files=5`
`percent_diff`	Indicator of how much a security must move by the end of the day to be considered whether a *win* or *loss*. If the security moves by less than the specified percentage, the system does not count it either way. If it moves by this percentage or more, it counts as *a win* or *a loss*, depending on whether the movement was beneficial to the account that made the trade. **Example:** `percent_diff=5`
`process_bank_to_bank`	Flag that tells the Data Ingestion process whether it should populate the bank to bank transfer field in the Front Office transaction table or if it should leave it blank for the firm to populate. The value **Y** indicates that the Oracle Financial Services mapping runs. The value **N** indicates that the firm calculates this field and provides the value during Data Ingestion. **Valid Value** — **Description** Y — Run the Oracle Financial Services mappings N — Use the provided value
`process_foreign_flag`	Flag that tells the Data Ingestion process whether it should populate the foreign field in the Front Office transaction table, or if it should leave it blank for the firm to populate. The value **Y** indicates that the Oracle Financial Services mapping runs. The value **N** indicates that the firm calculates this field and provides the value during Data Ingestion. **Valid Value** — **Description** Y — Run the Oracle Financial Services mappings N — Use the provided value
`process_pass_thru`	Flag that identifies whether the Data Ingestion process determines whether a transaction is pass-through or if the customer is performing this determination. The value **Y** indicates that the Oracle Financial Services mapping runs. The value **N** indicates that the firm calculates this field and provides the value during Data Ingestion. **Valid Value** — **Description** Y — Run the Oracle Financial Services mappings N — Use the provided value

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`process_secondary_names`	Flag that tells the Data Ingestion process whether it should populate the secondary originator and secondary beneficiary name fields in the Front Office transaction table or if it should leave it blank for the firm to populate. The value **Y** indicates that the Oracle Financial Services mapping runs. The value **N** indicates that the firm calculates this field and provides the value during Data Ingestion. **Valid Value** — **Description** Y — Run the Oracle Financial Services mappings N — Use the provided value
`process_trxn_xref`	Flag that tells the Data Ingestion process whether it should populate `BUSINESS.TRXN_PARTY_XREF` table or leave it blank. **Valid Value** — **Description** Y — Populate the `BUSINESS.TRXN_PARTY_XREF` table. N — Do not populate the `BUSINESS.TRXN_PARTY_XREF` table.
`rv_daemon`	RV daemon parameter (used in TIBCO environments). **Example:** `rv_daemon=tcp:7602`
`rv_network`	RV network parameter (used in TIBCO environments). **Example:** `rv_network=eri0`
`rv_service`	RV service parameter (used in TIBCO environments). **Example:** `rv_service=7602`
`ssaname3_home`	Specifies the path to the SSA Name 3 installation. **Example:** `ssaname3_home=/software/nm3_2704_k022_linux_amd64`
`tibco_config_dir`	TIBCO configuration directory. **Example:** `tibco_config_dir=/usr/tek/rel/local/config`
`tibco_inst_dir`	TIBCO RV installation directory. **Example:** `tibco_inst_dir=/usr/tek/rv`
`use_tax_id`	Instruction to Ingestion Manager when to use tax identifiers (both account and customer) in determining unrelated journals. If a substitute tax identifier is used when the account's or customer's tax identifier is unknown, then unrelated accounts may appear to be related when tax identifiers are considered. This variable enables the Ingestion Manager to ignore substitute tax identifiers. **Valid Value** — **Description** Y — Ingestion Manager is to use tax identifiers N — Ingestion Manager is not to use tax identifiers
`wls_shutdown_port`	The port on which the Watch List Service will run. **Example:** `wls_http_port=8087`

Table 9. Silent Mode Installation Variables (Continued)

Installation Variable	
`wls_http_port`	The port from which the Watch List Service can be stopped. **Example:** `wls_shutdown_port=8004`
Logging Variables	
`log_tz`	Time zone in which the log times should be recorded. The format of the value should follow the UNIX time zone standards. For Solaris, time zones are the directories under /usr/share/lib/zoneinfo. **Example:** `log_tz=US/Eastern`

APPENDIX B Environment Variables (.cshrc) File

This appendix provides a list of Oracle Financial Services application environment variables that must be set in the `.cshrc` file.

Environment Variables for the `.cshrc` File

Table 10 provides a list of environment variables along with a description and an example of each for you to use as a guide in setting your system's environment variables:

Table 10. Environment Variables for the `.cshrc` File

Variable Name	Description	Example
`PATH`	A shell variable that specifies the location of the command you typed. In the example `.cshrc` file, the path variable is set to `"/usr/bin:/usr/local:/usr/local/bin:/usr/bin/X11:/usr/ucb: /usr/opt/bin".` This setting tells the shell to look first in the `/usr/bin` directory, then in the `/usr/local` directory, next in the `/usr/local/bin` directory, and so on until the file has been found or all directories have been examined.	`setenv PATH ${PATH}:.:/kds/sparc-sun-solaris10/pkg/jdk1.6.0/bin:/software/pc8.6.1/server/bin`
`LD_LIBRARY_PATH`	Provides the run-time shared library loader (`ld.so`) an extra set of directories to look for when searching for shared libraries. Multiple directories can be listed, separated by a colon (:). This list is prepended to the existing list of compiled-in loader paths for a given executable, and any system default loader paths.	`setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:/kds/sparc-sun-solaris10/pkg/oracle/product/10.2.0/lib32:/kds/sparc-sun-solaris10/pkg/jdk1.6.0/jre/lib/sparcv9/server`
`LANG`	Required, along with `NLS_LANG`, to support double byte characters.	For Solaris OS: `setenv LANG en_US.UTF-8`
`NLS_LANG`	Required along with `LANG`, to support double byte characters.	`setenv NLS_LANG AMERICAN_AMERICA.AL32UTF8`

Table 10. Environment Variables for the `.cshrc` File (Continued)

Variable Name	Description	Example
`INFA_HOME`	Specifies the Informatica PowerCenter Services installation directory. If you modify the PowerCenter Services directory structure, you must configure the environment variable to the location of the PowerCenter Services installation directory or the directory where the installed PowerCenter Services files are located.	`setenv INFA_HOME /software/pc8.6.1`
`INFA_JAVA_OPTS`	Specifies the amount of system memory used by PowerCenter Services. Informatica PowerCenter uses the Java Runtime Environment (JRE) to start and run PowerCenter Services. By default, PowerCenter Services uses a maximum of 512 MB of system memory. If you set `INFA_JAVA_OPTS` after starting PowerCenter Services, you must restart the node for the changes to take effect.	`setenv INFA_JAVA_OPTS -Xmx1024m`
`INFA_DOMAINS_FILE`	Specifies the file that contains the connectivity information for the gateway nodes in a domain. The connectivity information includes the domain names, domain host names, and domain host port numbers. You can configure this variable (`INFA_DOMAINS_FILE`) before or after installation.	`setenv INFA_DOMAINS_FILE /software/ /pc8.6.1/domainFile`
`JAVA_HOME`	Sets the Java installed directory.	`setenv JAVA_HOME /kds/ sparc-sun-solaris10/pkg/jdk1.6.0`

APPENDIX C Oracle Financial Services and Business Data Model Variables

This appendix identifies and defines the variables used in the `db_variables.cfg` file to install the Business data model.

Variables in the `db_variables.cfg` File

This appendix lists and defines the variables in the `db_variables.cfg` file.

Table 11. Variables in the `db_variables.cfg` File

Variable Name	Description	Example
Installation Flag Variables		
`four_eyes_flag`	Determines whether Oracle Financial Services installs the database alert metadata to accommodate "standard" or *four eyes* actions. If the flag is set to **Y**, then Oracle Financial Services loads alert metadata to support the *four eyes* approval processing.	`N`
Oracle Financial Services Schema Owners		
`mantas_schema_owner`	Mantas schema user.	`MANTAS`
`kdd_schema_owner`	Schema where base metadata will reside. Use KDD.	`KDD`
`business_schema_owner`	BUSINESS schema user.	`BUSINESS`
`market_schema_owner`	MARKET schema user.	`MARKET`
`web_user`	Web Application user.	`KDD_WEB`
`mantas_user`	Database user with permissions on Mantas schema objects for loading.	`KDD_USER`
`server_user`	Database user used to run Behavior Detection algorithms.	`KDD_ALG`
`tools_user`	Database user used to run Scenario Manager.	`KDD_MNR`
`ingest_user_name`	INGEST user.	`INGEST_USER`
`db_util_user`	Database user used to run database utilities	`DB_UTIL_USER`
`altio_user`	Altio user.	`KDD_ALTIO_USER`

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
`report_user`	Database user created to support reporting applications built on top of Oracle Financial Services.	`KDD_REPORT`
`case_schema_owner`	Database user with permissions on Case schema objects for loading.	`CMREVMAN`
`config_user`	Configuration Schema User	`REVELEUS`
Oracle Financial Services Roles		
`kdd_algorithm_role`	Database role with permissions for running Behavior Detection algorithms.	`KDD_ALGORITHM`
`kdd_analyst_role`	Database role with permissions to view alerts.	`KDD_ANALYST`
`kdd_miner_role`	Database role with permissions to run Scenario Manager.	`KDD_MINER`
`kdd_reader_role`	Database role with read permissions on all KDD schema objects.	`KDD_READER`
`kdd_loader_role`	Database role with permissions on all KDD schema objects for loading.	`KDD_LOADER`
`mantas_reader_role`	Database role with Read privileges on Mantas schema objects.	`MANTAS_READER`
`mantas_loader_role`	Database role with privileges on Mantas schema objects for loading.	`MANTAS_LOADER`
`data_reader_role`	Database role with read permissions on FIRM and MARKET schema objects.	`DATA_READER`
`data_loader_role`	Database role with permissions on FIRM and MARKET schema objects for loading.	`DATA_LOADER`
Oracle Financial Services Product Tablespaces		
`kdd_data_tablespace`	Tablespace used for the data in the KDD schema.	`KDD_DATA`
`kdd_index_tablespace`	Tablespace used for the indexes in the KDD schema.	`KDD_INDEX`
`mantas_data_tablespace`	Tablespace used for data in the Mantas schema.	`MANTAS_DATA`
`mantas_index_tablespace`	Tablespace used for indexes in the Mantas schema.	`MANTAS_INDEX`
`alert_data_tablespace`	Tablespace used for data in `KDD_*` tables in the TCM schema.	`ALERT_DATA`
`alert_index_tablespace`	Tablespace used for indexes on the `KDD_*` tables in the TCM schema.	`ALERT_INDEX`
`miner_data_tablespace`	Tablespace where Miner tables reside.	`MINER_DATA`
`miner_index_tablespace`	Tablespace where Miner indexes reside.	`MINER_INDEX`
`user_data_tablespace`	Default tablespace for user creation.	`KDD_DATA`
`temp_tablespace`	Temporary tablespace.	`TEMP`

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
Physical Tablespaces Created During Installation		
DATA_BUS1_TBSP	Database tablespaces for Business and Market data, and indexes.	DATA_BUS1_TBSP
DATA_BUS2_TBSP		DATA_BUS2_TBSP
DATA_BUS3_TBSP		DATA_BUS3_TBSP
DATA_BUS4_TBSP		DATA_BUS4_TBSP
DATA_BUS5_TBSP		DATA_BUS5_TBSP
DATA_BUS6_TBSP		DATA_BUS6_TBSP
DATA_BUS7_TBSP		DATA_BUS7_TBSP
DATA_BUS8_TBSP		DATA_BUS8_TBSP
DATA_MKT1_TBSP		DATA_MKT1_TBSP
DATA_MKT2_TBSP		DATA_MKT2_TBSP
DATA_MKT3_TBSP		DATA_MKT3_TBSP
DATA_MKT4_TBSP		DATA_MKT4_TBSP
IDX_BUS1_TBSP		IDX_BUS1_TBSP
IDX_BUS2_TBSP		IDX_BUS2_TBSP
IDX_BUS3_TBSP		IDX_BUS3_TBSP
IDX_BUS4_TBSP		IDX_BUS4_TBSP
IDX_BUS5_TBSP		IDX_BUS5_TBSP
IDX_BUS6_TBSP		IDX_BUS6_TBSP
IDX_BUS7_TBSP		IDX_BUS7_TBSP
IDX_BUS8_TBSP		IDX_BUS8_TBSP
IDX_MKT1_TBSP		IDX_MKT1_TBSP
IDX_MKT2_TBSP	Database tablespaces for business and market data and indexes.	IDX_MKT2_TBSP
IDX_MKT3_TBSP		IDX_MKT3_TBSP
IDX_MKT4_TBSP		IDX_MKT4_TBSP

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
Logical Tablespace References Mapped to Physical Tablespaces		
acct_data_tablespace	Mapping of logical tablespaces used during object creation to physical tablespaces.	DATA_BUS8_TBSP
bal_data_tablespace		DATA_BUS5_TBSP
bbo_data_tablespace		DATA_MKT1_TBSP
bo_trxn_data_tablespace		DATA_BUS1_TBSP
bus_arc_data_tablespace		DATA_BUS6_TBSP
bus_data_tablespace		DATA_BUS2_TBSP
cash_trxn_data_tablespace		DATA_BUS2_TBSP
cust_data_tablespace		DATA_BUS6_TBSP
emp_data_tablespace		DATA_BUS8_TBSP
mi_trxn_data_tablespace		DATA_BUS3_TBSP
mkt_arc_data_tablespace		DATA_MKT4_TBSP
mkt_data_tablespace		DATA_MKT4_TBSP
order_data_tablespace		DATA_BUS3_TBSP
posn_data_tablespace		DATA_BUS6_TBSP
quote_data_tablespace		DATA_MKT2_TBSP
reported_sale_data_ tablespace		DATA_MKT3_TBSP
scrty_data_tablespace		DATA_BUS5_TBSP
smry_data_tablespace		DATA_BUS7_TBSP
staging_data_tablespace		DATA_BUS1_TBSP
trade_data_tablespace		DATA_BUS4_TBSP
wire_trxn_data_tablespace		DATA_BUS4_TBSP
acct_idx_tablespace		IDX_BUS8_TBSP
bal_idx_tablespace		IDX_BUS5_TBSP
bbo_idx_tablespace		IDX_MKT1_TBSP

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
bo_trxn_idx_tablespace	Mapping of logical tablespaces used during object creation to physical tablespaces.	IDX_BUS1_TBSP
bus_arc_idx_tablespace		IDX_BUS6_TBSP
bus_idx_tablespace		IDX_BUS2_TBSP
cash_trxn_idx_tablespace		IDX_BUS2_TBSP
cust_idx_tablespace		IDX_BUS6_TBSP
emp_idx_tablespace		IDX_BUS8_TBSP
mi_trxn_idx_tablespace		IDX_BUS3_TBSP
mkt_arc_idx_tablespace		IDX_MKT4_TBSP
mkt_idx_tablespace		IDX_MKT4_TBSP
order_idx_tablespace		IDX_BUS3_TBSP
posn_idx_tablespace		IDX_BUS6_TBSP
quote_idx_tablespace		IDX_MKT2_TBSP
reported_sale_idx_ tablespace		IDX_MKT3_TBSP
scrty_idx_tablespace		IDX_BUS5_TBSP
smry_idx_tablespace		IDX_BUS7_TBSP
staging_idx_tablespace		IDX_BUS1_TBSP
trade_idx_tablespace		IDX_BUS4_TBSP
wire_trxn_idx_tablespace		IDX_BUS4_TBSP
case_data_tablespace		CASE_DATA
case_data_tablespace		CASE_INDEX
Data File Names		
kdd_data_file	File used for KDD tables.	/CHANGE_ME/kdd_data_01.dbf
kdd_index_file	File used for KDD indexes.	/CHANGE_ME/kdd_idx_01.dbf
mantas_data_file	File used for Oracle Financial Services tables.	/CHANGE_ME/mantas_data_01.dbf
mantas_index_file	File used for Oracle Financial Services indexes.	/CHANGE_ME/mantas_idx_01.dbf
alert_data_file	File used for Alert tables.	/CHANGE_ME/alert_data_01.dbf
alert_index_file	File used for Alert indexes.	/CHANGE_ME/alert_idx_01.dbf
miner_data_file	File used for Miner tables.	/CHANGE_ME/miner_data_01.dbf
miner_index_file	File used for Miner indexes.	/CHANGE_ME/miner_idx_01.dbf

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
data_bus1_filename	File name of each tablespace.	/CHANGE_ME/bus1_data_file01.dbf
data_bus2_filename		/CHANGE_ME/bus2_data_file01.dbf
data_bus3_filename		/CHANGE_ME/bus3_data_file01.dbf
data_bus4_filename		/CHANGE_ME/bus4_data_file01.dbf
data_bus5_filename		/CHANGE_ME/bus5_data_file01.dbf
data_bus6_filename		/CHANGE_ME/bus6_data_file01.dbf
data_bus7_filename		/CHANGE_ME/bus7_data_file01.dbf
data_bus8_filename		/CHANGE_ME/bus8_data_file01.dbf
data_mkt1_filename		/CHANGE_ME/mkt1_data_file01.dbf
data_mkt2_filename		/CHANGE_ME/mkt2_data_file01.dbf
data_mkt3_filename		/CHANGE_ME/mkt3_data_file01.dbf
data_mkt4_filename		/CHANGE_ME/mkt4_data_file01.dbf
idx_bus1_filename		/CHANGE_ME/bus1_idx_file01.dbf

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
idx_bus2_filename	File name of each tablespace.	/CHANGE_ME/bus2_idx_file01.dbf
idx_bus3_filename		/CHANGE_ME/bus3_idx_file01.dbf
idx_bus4_filename		/CHANGE_ME/bus4_idx_file01.dbf
idx_bus5_filename		/CHANGE_ME/bus5_idx_file01.dbf
idx_bus6_filename		/CHANGE_ME/bus6_idx_file01.dbf
idx_bus7_filename		/CHANGE_ME/bus7_idx_file01.dbf
idx_bus8_filename		/CHANGE_ME/bus8_idx_file01.dbf
idx_mkt1_filename		/CHANGE_ME/mkt1_idx_file01.dbf
idx_mkt2_filename		/CHANGE_ME/mkt2_idx_file01.dbf
idx_mkt3_filename		/CHANGE_ME/mkt3_idx_file01.dbf
idx_mkt4_filename		/CHANGE_ME/mkt4_idx_file01.dbf
informatica_mapping_filename		/CHANGE_ME/infa_data_file.dbf
case_data_file		/CHANGE_ME/case_data_01.dbf
case_index_file		/CHANGE_ME/case_index_01.dbf

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
Data File Sizes		
data_bus1_size	Data file size for each data file name identified in *Data File Names* above.	256M
data_bus2_size		256M
data_bus3_size		256M
data_bus4_size		256M
data_bus5_size		256M
data_bus6_size		256M
data_bus7_size		256M
data_bus8_size		256M
data_mkt1_size		512M
data_mkt2_size		512M
data_mkt3_size		512M
data_mkt4_size		512M
idx_bus1_size		256M
idx_bus2_size		256M
idx_bus3_size		256M
idx_bus4_size		256M
idx_bus5_size		256M
idx_bus6_size		256M
idx_bus7_size		256M
idx_bus8_size	Data file size for each data file name identified in *Data File Names* above.	256M
idx_mkt1_size		256M
idx_mkt2_size		256M
idx_mkt3_size		256M
idx_mkt4_size		256M
case_data_size		512M
case_index_size		512M
Daily Partition Names		
Note: Daily Partition Names (Format is PYYYYMMDD. Example: P20070405 is partition to hold data for April 5th, 2007). DataDumpDt_minus_X_name is the name for the partition X business days in the past.		
DataDumpDt_minus_8_name	Name of the partition for the business day eight days prior to the current business day that data is loaded.	P20091130
DataDumpDt_minus_7_name	Name of the partition for the business day seven days prior to the current business day that data is loaded.	P20091201
DataDumpDt_minus_6_name	Name of the partition for the business day six days prior to the current business day that data is loaded.	P20091202
DataDumpDt_minus_5_name	Name of the partition for the business day five days prior to the current business day that data is loaded.	P20091203

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
DataDumpDt_minus_4_name	Name of the partition for the business day four days prior to the current business day that data is loaded.	P20091204
DataDumpDt_minus_3_name	Name of the partition for the business day three days prior to the current business day that data is loaded.	P20091207
DataDumpDt_minus_2_name	Name of the partition for the business day two days prior to the current business day that data is loaded.	P20091208
DataDumpDt_minus_1_name	Name of the partition for the previous business day or initial business day that data is loaded.	P20091209
DataDumpDt_minus_0_name	Name of current business day or initial business day of data to be loaded.	P20091210
DataDumpDt_plus_1_name	Name of the partition for the next business day after current business day is loaded.	P20091211
LastDayLastMnth_name	Name of the partition for the last calendar day of the previous month.	P20091130
Daily Partition Dates		
Note: Daily Partition Dates (Format is 'DD-MON-YYYY'). DataDumpDt_minus_X is the date X business days in the past.		
DataDumpDt_minus_8	Date, eight business days in the past from the current business day.	‘30-NOV-2009’
DataDumpDt_minus_7	Date, seven business days in the past from the current business day.	‘01-DEC-2009’
DataDumpDt_minus_6	Date, six business days in the past from the current business day.	‘02-DEC-2009’
DataDumpDt_minus_5	Date, five business days in the past from the current business day.	‘03-DEC-2009’
DataDumpDt_minus_4	Date, four business days in the past from the current business day.	‘04-DEC-2009’
DataDumpDt_minus_3	Date, three business days in the past from the current business day.	‘07-DEC-2009’
DataDumpDt_minus_2	Date, two business days in the past from the current business day.	‘08-DEC-2009’
DataDumpDt_minus_1	Date, one business day in the past from the current business day.	‘09-DEC-2009’
DataDumpDt_minus_0	Date of the current business day or initial business day that data is loaded.	‘10-DEC-2009’
DataDumpDt_plus_1	Date of the next business day after current business day.	‘11-DEC-2009’
Weekly Partition Names		
Weekly Partition Names (Format is PYYYYMMDD. Example: P20070402 is partition to hold data for the week ending April 2nd, 2007). The weekly partitions should always end on a Friday, regardless of the Start of Week, End of Week settings. The partition names should be set to the actual Friday date.		
EndThisWeek_minus_01_name	Name for the previous business week.	P20091211

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
`EndThisWeek_minus_00_name`	Name of current business week or initial business week of data to be loaded.	`P20091218`
Weekly Partition Dates		
Weekly Partition Dates (Format is 'DD-MON-YYYY'). The partition dates must be set to the Saturday *after* the Friday identified in the Partition Name.		
`EndThisWeek_minus_01`	Date of the Friday for the previous business week.	`'12-DEC-2009'`
`EndThisWeek_minus_00`	Date of the Friday of current business week or initial business week of data to be loaded.	`'19-DEC-2009'`
Monthly Partition Names		
Monthly Partition Names (Format is PYYYYMM. Example: P200704 is partition to hold data for April 2007).		
`StartNextMnth_plus_01_name`	Name of the partition for the first day of the month after the next month.	`P201001`
`StartNextMnth_minus_00_name`	Name of the partition for the current month.	`P200912`
`StartNextMnth_minus_01_name`	Name of the partition for the previous month.	`P200911`
`StartNextMnth_minus_02_name`	Name of the partition for the month two months prior to the current month.	`P200910`
`StartNextMnth_minus_03_name`	Name of the partition for the month three months prior to the current month.	`P200909`
`StartNextMnth_minus_04_name`	Name of the partition for the month four months prior to the current month.	`P200908`
`StartNextMnth_minus_05_name`	Name of the partition for the month five months prior to the current month.	`P200907`
`StartNextMnth_minus_06_name`	Name of the partition for the month six months prior to the current month.	`P200906`
`StartNextMnth_minus_07_name`	Name of the partition for the month seven months prior to the current month.	`P200905`
`StartNextMnth_minus_08_name`	Name of the partition for the month eight months prior to the current month.	`P200904`
`StartNextMnth_minus_09_name`	Name of the partition for the month nine months prior to the current month.	`P200903`
`StartNextMnth_minus_10_name`	Name of the partition for the month 10 months prior to the current month.	`P200902`
`StartNextMnth_minus_11_name`	Name of the partition for the month 11 months prior to the current month.	`P200901`
`StartNextMnth_minus_12_name`	Name of the partition for the month 12 months prior to the current month.	`P200912`

Table 11. Variables in the `db_variables.cfg` File (Continued)

Variable Name	Description	Example
Monthly Partition Dates		
`StartNextMnth_plus_01`	Date of the first day of the month after the next month.	`'01-FEB-2009'`
`StartNextMnth_minus_00`	Date of the first day of the current month.	`'01-JAN-2009'`
`StartNextMnth_minus_01`	Date of the first day of the previous month.	`'01-DEC-2009'`
`StartNextMnth_minus_02`	Date of the first day of the month two months prior to the current month.	`'01-NOV-2009'`
`StartNextMnth_minus_03`	Date of the first day of the month three months prior to the current month.	`'01-OCT-2009'`
`StartNextMnth_minus_04`	Date of the first day of the month four months prior to the current month.	`'01-SEP-2009'`
`StartNextMnth_minus_05`	Date of the first day of the month five months prior to the current month.	`'01-AUG-2009'`
`StartNextMnth_minus_06`	Date of the first day of the month six months prior to the current month.	`'01-JUL-2009'`
`StartNextMnth_minus_07`	Date of the first day of the month seven months prior to the current month.	`'01-JUN-2009'`
`StartNextMnth_minus_08`	Date of the first day of the month eight months prior to the current month.	`'01-MAY-2009'`
`StartNextMnth_minus_09`	Date of the first day of the month nine months prior to the current month.	`'01-APR-2009'`
`StartNextMnth_minus_10`	Date of the first day of the month 10 months prior to the current month.	`'01-MAR-2009'`
`StartNextMnth_minus_11`	Date of the first day of the month 11 months prior to the current month.	`'01-FEB-2009'`
`StartNextMnth_minus_12`	Date of the first day of the month 12 months prior to the current month.	`'01-JAN-2009'`
`partition_date_format`	Format of the date used in specifying partition dates.	`'DD-MON-YYYY'`

APPENDIX D

List of Acronyms and Abbreviations

This appendix defines acronyms and abbreviations used in this guide.

DBA	Database Administrator
DDL	Database Definition Language
DIS	Data Interface Specification
DN	Distinguished Name
FDT	Firm Data Transformer
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
LDIF	Lightweight Directory Interchange Format
MDS	Market Data Server
SQL	Structured Query Language
SSL	Secure Socket Layer
TNS	Transparent Network Substrate
UI	User Interface
URL	Uniform Resource Locator
XML	Extensible Markup Language
WAR	Web Application Archive

ORACLE