

Patch Installer Guide

Release 1.0

July 2013



Patch Installer Guide

Release 1.0
July 2013

Document Control Number: 9MN12-0001
Document Number: PIG-13-OFS-0001-1.0-01

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

Document Number: PIG-13-OFS-0001-1.0-01
First Edition (July 2013)

Copyright © 1996-2013, Oracle and/or its affiliates. All rights reserved.

Printed in U.S.A. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior written permission.

Trademarks

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

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

Contents

List of Figures	ix
List of Tables	xi
About this Guide	xiii
Who Should Use this Guide	xiii
How this Guide is Organized	xiii
Conventions Used in this Guide	xiv
CHAPTER 1 <i>About Patch Installer</i>	1
Patch Installer Overview	1
Deployment Environments	1
Deployment Configuration	2
System Hardware and Components	2
Pre-requisites	3
Database Server	3
<i>Database Configuration</i>	3
<i>Oracle Database Parameters</i>	3
Patch Installer Server	5
General Environment	5
CHAPTER 2 <i>Installation Overview</i>	7
Steps in Installation Process	7
Copying the Installation Files	7
Copying Installation Files	8
Copying Methods for the Installation Files	8
<i>Copying the Installation Files Using Copy Command</i>	8
<i>Transferring Installation Files Using FTP</i>	8
Running a Silent Installation	8
Understanding the Silent Properties File	8
<i>Variables that Define which Components to Install</i>	9
<i>Variables Common to Multiple Components</i>	9
<i>Variables for Individual Components</i>	10
Installation Checklist	10
CHAPTER 3 <i>Installing the Patch Installer</i>	13
Running the Patch Installer Installation Program	13
Copying the Installation Files	13

Creating the Silent Properties File	13
Running the Silent Installation	14
Setting All Passwords	14
Installing the Data Model	16
Installing the System Objects	17
Installing the Patch Data Model Objects.....	17
Installing Multiple Instances of the Patch Installer	18
Understanding the Process	18
CHAPTER 4 <i>Using the Patch Installer</i>	19
Instance Registration.....	19
Validations Performed During Instance Registration.....	20
Log Files and Registration History.....	20
Patch Installation	20
Validations Performed During Patch Installation	21
Patches Which Apply to Multiple Subsystems	21
Backup Processing	21
Manual Steps	21
Handling Customizations.....	21
Log Files and Installation History	22
Patch Un-installation.....	22
Validations Performed During Patch Un-installation.....	22
Manual Steps.....	22
Log Files and Un-Installation History	23
Instance Un-Registration.....	23
Validations Performed During Instance Un-Registration.....	23
Log Files and Un-Registration History.....	24
Patch Installer Usage.....	25
Patch Installer Configuration	25
<i>Common Resources</i>	25
Patch Package.....	36
Patch Information.....	36
Patch History	36
APPENDIX A <i>List of Acronyms and Abbreviations</i>	37
APPENDIX B <i>Variables Used in the Silent Properties File</i>	39
Files that the Installation CD Contains.....	39
APPENDIX C <i>Environment Variables File</i>	47
Environment Variables	47

APPENDIX D	<i>Oracle Financial Services and Business Data Model</i>	
Variables	49
Variables in the db_variables.cfg File.....		49

List of Figures

Figure 1. Sample install.cfg.....35

List of Tables

Table 1. Conventions Used in this Guide	xiv
Table 2. Server Deployment Configuration.....	2
Table 3. Prerequisites for the Database Server.....	3
Table 4. Tunable Database Parameters.....	4
Table 5. Prerequisites for the Patch Installer	5
Table 6. Installation Checklist	10
Table 7. Components to Install	14
Table 8. Setting Password	15
Table 9. Silent Mode Installation Variables	39
Table 10. Environment Variables	47
Table 11. Variables in the db_variables.cfg File.....	49

About this Guide

This guide provides comprehensive instructions for installing and configuring the Patch Installer (PI).

This chapter focuses on the following topics:

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

Who Should Use this Guide

The *Patch Installer Guide* is designed for use by the Installers and Administrators. Their roles and responsibilities include the following:

- **Installer:** This user installs and configures Patch Installer at a deployment site.
- **Administrator:** This user maintains and adjusts the system and is usually an employee of a client. The Oracle Administrator archives data, and loads data feeds.

How this Guide is Organized

The *Patch Installer Guide* includes following chapters:

- Chapter 1, *About Patch Installer* discusses activities that occur prior to deployment, typical installation configuration, and identifies all third-party software necessary to run the Patch Installer.
- 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 Patch Installer* explains how to install and configure the necessary components on the Database server in a typical deployment configuration.
- Chapter 4, *Using the Patch Installer* explains about Instance Registration, Patch Installation, Patch Un-installation, Instance Un-Registration, Patch Utility Usage.
- Appendix A, *List of Acronyms and Abbreviations*, defines all of the acronyms and abbreviations that this guide uses.
- Appendix B, *Variables Used in the Silent Properties File*, lists and defines all variables in the silent properties file used to silently install the application.
- Appendix C, *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 D, *Oracle Financial Services and Business Data Model Variables*, lists and defines all Patch Installer variables in the `db_variables.cfg` file needed to install the Patch data model properly.

Conventions Used in this Guide

Table 1 lists the conventions used in this guide.

Table 1. Conventions Used in this Guide

This convention . . .	Stands for . . .
<i>Italics</i>	<ul style="list-style-type: none">● Names of books, chapters, and sections as references● Emphasis
Bold	<ul style="list-style-type: none">● 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	<ul style="list-style-type: none">● 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

The Oracle Financial Services Financial Crime and Compliance Management Patch Installer is used to apply software updates to the 6.x releases of the Financial Crime and Compliance Management. The installer can also be used to un-install updates that were previously installed using the Installer, and to display information about currently installed updates. A history of all installation and un-installation updates is also available.

This chapter includes the following topics:

- Patch Installer Overview
- Deployment Environments
- Deployment Configuration
- Pre-requisites

Patch Installer Overview

Updates to the Oracle Financial Services (OFS) Financial Crime and Compliance Management platform are delivered as patches or hotfixes are managed through the Patch Installer. The Patch Installer records details of installed patches in a database and leverages that data for reporting and dependency checking. Additionally, you must install the Patch Installer on each server regardless of whether other Stage 1 components are installed on this server or only Stage 3 components are installed. Each of these copies of the Patch Installer may point to its own database to record patch details, or multiple may share a single database. When deciding on an approach, consider that a disaster recovery (DR) setup can cause issues. In a typical disaster recovery environment, the primary production database is replicated to the backup (DR) server constantly to keep them in sync. During the replication process, the DR database is locked to all access other than the replication process itself. The software on the DR server must be installed separately; it cannot be restored from a backup of the production server due to differences in configuration parameters, and so on. Patches must be installed directly to the DR server. If the Patch Installer involves updating tables in the database being replicated, the updates will fail due to the replication locking. The Patch Installer resolves this issue by storing patch information in a database separate from the application database, so the tables holding the patch information are not subject to replication locking. This will prevent you from installing patches in DR when Production is down, but this should not be a regular occurrence. Before installing or un-installing patches, customers must register the application in the Patch Installer.

Deployment Environments

Typically, clients have a test environment and a production environment for running Patch Installer. Both environments include required servers and software components necessary to run the Patch Installer.

The test environment not only serves as a backup to the production environment, but is also used to install and test new Oracle 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

An 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
- Patch Installer server

Patch Installer runs with any subsystem installed on any one of these servers, or all subsystems can run on a single server.

Table 2 lists the Server Deployment of the Server Configuration.

Table 2. Server Deployment Configuration

Server/Workstation	Subsystems and Components
Database server	<ul style="list-style-type: none">● Patch Schema Objects (created during Patch Installer installation)
Patch Installer server	<ul style="list-style-type: none">● Database Tools <p>Note: Patch Installer must be installed on all the physical servers where any of the components from the Oracle FCCM Suite of Products is installed.</p>

Pre-requisites

This section lists the prerequisite third-party products you must have installed to run Patch Installer. Any pre-requisites that require Patch Installer specific instructions for installation and configuration are noted in the Installation Details column in Table 3.

This section includes the following topics:

- Database Server
- Patch Installer Server
- General Environment

Database Server

Table 3 lists the software prerequisites required for the Database Server.

Table 3. Prerequisites for the Database Server

Category	Software	Installation Details
Third-Party: Database Software	Oracle 11gR2 Enterprise Edition with Partitioning	General instructions: <ul style="list-style-type: none"> ● Install and create a database instance ● Use guidelines in Appendix B, <i>Variables Used in the Silent Properties File</i>, on page 39, to configure the database

Database Configuration

The database instance must be established and started before the installation can proceed. Patch 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 Patch data model creation scripts.

Oracle Database Parameters

Table 4 lists the Oracle database parameters that need to be sized individually for each customer installation.

Note: Review the Oracle recommended 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 4. Tunable Database Parameters

Tunable Database Parameters		Parameter Values		
Category	Parameter Name	Type	Default	Oracle Recommended
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	3G	
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	10M
	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
SHARED_POOL_RESERVED_SIZE	integer	5% of SHARED_PO OL_SIZE	32M	

Table 4. Tunable Database Parameters (Continued)

Tunable Database Parameters		Parameter Values		
Category	Parameter Name	Type	Default	Oracle Recommended
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	

Note: To avoid *signaled in parallel query server P003* error for Scenario *AM-UnfairAllocDist* do the following: In `init.ora` file, add `_optimizer_distinct_agg_transform=FALSE` and restart the database.

Patch Installer Server

Table 5 lists the software prerequisites that the Patch Installer server requires.

Table 5. Prerequisites for the Patch Installer

Category	Software	Installation Details
Operating System	Red Hat Linux 5/Sun Solaris 10/AIX 6.1	Install and configure.
Third Party: Database Software	Oracle 11gR2 Client	Configure TNS names (in the <code>tnsnames.ora</code> file) to reach the database instance. Note: Verify installation of all Oracle utilities (that is, <code>sqlldr</code>).
Third Party: Other Software	Sun Java Runtime Environment (JRE) jdk1.6 for use by Patch Installer 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,

- For Solaris OS: `setenv LANG en_US.UTF-8`
- For Linux OS: `setenv LANG en_US.utf8`
- For AIX: `setenv LANG EN_US.UTF-8`

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

For an Oracle Database 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` or `.profile`) to ensure that they exist and are correct. Refer to Appendix C, *Environment Variables File*, on page 47, for a sample file.

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:

- Steps in Installation Process
- Copying the Installation Files
- Running a Silent Installation
- Installation Checklist

Steps in Installation Process

Use the following installation process for installing Patch Installer 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 8, 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 8, for more information).
4. Configure the components and subsystems, as needed.

Note: 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 or media pack to a working directory on a specific host machine. From this working directory (herein called as <Installer directory>), run the installation program to create the Patch Installer installation directory (referred as <PI Installed Directory>) on the UNIX server.

This section discusses the following topics:

- Copying Installation Files
- Copying Methods for the Installation Files

Copying Installation Files

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

- `installUtilities.bin`: Installation program that contains PI components.
- `utilities.properties.sample`: Text file that supplies values to variables that the installation program contains, when you are installing Patch Installer. This file contains information from the Patch Installer test environment and needs to be manually 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 or media pack 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 or media pack.

Copying the Installation Files Using Copy Command

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

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

Transferring 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 file in binary mode and the sample property file 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 Patch Installer, it *silently* queries the properties file for answers to variables embedded in its code that identifies where and what to install. The Patch Installer installation program unpacks the necessary files and lays down the directory structure for the specified subsystem and components in <PI Installed Directory>.

You can install any subsystem or component on any servers or you can install all items on a single server. Refer to *Understanding the Silent Properties File* on page 8, for more information.

Understanding the Silent Properties File

The Patch Installer installation program contains tokens that map to the variables that the silent properties file contains. During installation, the Patch Installer 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 B, *Variables Used in the Silent Properties File*, on page 39 for the entire list of variables used in the silent properties file, along with definitions and examples of each.

Variables that Define which Components to Install

The first section of the silent properties file identifies the components to install. Values for items within this group can be true or false. The Patch Installer 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")

```
PATCH_INSTALLER=true
```

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

Note: You may install the Database component on only one host machine. This is required only once for creation of the Patch Database. Additionally, if you already have an existing Patch Database (which you might have configured for Patch Installer embedded with an earlier version of OFSBDF product, you may choose not to create a fresh Patch database at all. You may simply point the Patch Installer to the existing database.

Variables Common to Multiple Components

This section of the silent properties file lists variables that 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.
```

#####

```
ORACLE_HOME=/scratch/oracle/app/oracle/product/11.2.0/client_1
DB_TNS_ADMIN=/scratch/oracle/app/oracle/product/11.2.0/dbhome_2/network/admin
INGEST_JAVA_HOME=/scratch/oracle/jdk1.6.0_37
SQL_DRIVER_JDBC=oracle.jdbc.OracleDriver
PDB_URL=jdbc:oracle:thin:@ofss220074.in.oracle.com:1521:Ti1011L56
PDB_UTILS_DB_USER=ECM62_B05_PATCH2
PATCH_SCHEMA_OWNER=ECM62_B05_PATCH2
```

Variables for Individual Components

The last section of silent properties file lists each component and the variables it uses, refer *Appendix D, Oracle Financial Services and Business Data Model Variables*, on page 49, for variable names). The Patch Installer installation program uses these variables only if the specific component has its value set to true; otherwise, the program ignores them.

Installation Checklist

Table 6 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 6. Installation Checklist

S.No	Task	Completed
1.	Verify your configuration. Refer to <i>System Hardware and Components</i> , on page 2, for more information.	<input type="checkbox"/>
2.	Verify that all prerequisite softwares have been installed. Refer to <i>Pre-requisites</i> , on page 3, for more information.	<input type="checkbox"/>
3.	Copy all files. Refer to <i>Copying the Installation Files</i> , on page 7, for more information.	<input type="checkbox"/>
4.	Set the <i>Silent Properties File</i> with all the values. Refer to <i>Understanding the Silent Properties File</i> , on page 8 and <i>Environment Variables File</i> , on page 47, for more information.	<input type="checkbox"/>
5.	Check all the paths defined in the environment file (.cshrc or .profile) to ensure that they exist and are correct. Refer to Appendix C, <i>Environment Variables File</i> , on page 47, for variable definitions.	<input type="checkbox"/>
6.	Execute the silent installation. Refer how to run the silent properties file in <i>Running a Silent Installation</i> , on page 8, and then proceed with the silent installation following the appropriate instructions.	<input type="checkbox"/>

Table 6. Installation Checklist (Continued)

S.No	Task	Completed
7.	Execute the Password Manager Utility. Refer to <i>Setting All Passwords</i> , on page 14 for information. Note: If you are installing for the first time, select All Options .	<input type="checkbox"/>
8.	Install the patch data model. Refer to <i>Installing the Data Model</i> , on page 16, for more information.	<input type="checkbox"/>

This chapter explains step by step instruction to install Patch Installer and includes examples for a typical configuration. The following sections provide a high-level list of tasks that you need to perform.

Installing the Patch Installer involves the following procedures:

- Running the Patch Installer Installation Program
- Setting All Passwords
- Installing the Data Model
- Installing Multiple Instances of the Patch Installer

Running the Patch Installer Installation Program

Run the Patch Installer installation program to create Patch Installer directory, and unpack the database files that you need to configure and run the database.

To run the Patch Installer installation program, follow these steps:

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

Copying the Installation Files

You must run the Patch Installer 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 the <Installer directory> on the host machine:

- `installUtilities.bin`
- `utilities.properties.sample`

Refer to *Copying the Installation Files Using Copy Command* section on page 8, for information about these files.

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

Note: The PI must be run using the same user account under which the OFSFCCM suite of products have been installed.

To create the Patch Installer properties file, follow these steps:

1. Copy the `utilities.properties.sample` file with an appropriate name for the Patch Installer properties. For example,

```
cp utilities.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 7.

Table 7. Components to Install

Variable	Value
DATABASE	true
PATCH_INSTALLER	true

4. Enter variable values in the following sections of the `install.properties.appserver` file:
 - Product Installation Directory
 - Select Components to install
 - Common Variables
 - Patch Utility Variables

Refer to *Appendix B Variables Used in the Silent Properties File*, on page 39, 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 host machine 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 Patch Installer Installation Program.
2. Type the following command to run the silent installation:

```
installUtilities.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* section on page 13.

The Silent Installation ensures that the Patch Installer software is installed on your system at directory structure `<PI Installed Directory>`.

Setting All Passwords

After completing the silent installation, execute the following command:

<PI Installed Directory>/changePasswords.sh all

This prompts for the passwords of the required application users as shown below. 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 possible to skip a password..

Table 8. Setting Password

S.No	Description	Commands
1	Enter password for Patch Database User	<enter password for (patch_util_user)>
2	Enter password for Database Utility User	<enter password for (db_util_user)>
3	Enter password for Data Miner User	<enter password for (tools_user)>
4	Enter password for Data Ingest User	< enter password for (ingest_user_name)>

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

Note: For details of users refer to <PI Installed Directory>/patch_installer/mantas_cfg/db_variables.cfg file for schema names written against the variables provided in brackets of respective schema.

Installing the Data Model

Installing the 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.

Note: Before installing the Data Model, increase the log trace level to the maximum by setting the `log.trace=true` in `<PI Installed Directory>/database/db_tools/mantas_cfg/install.cfg`. This helps in capturing any errors that might occur while installing the Data Model.

This step is required only once for creation of the Patch Database. Additionally, if you already have an existing Patch Database (which you might have configured for Patch Installer embedded with the an earlier version of OFSBDF product, you may choose not to create a fresh Patch database at all. You may simply skip this step and point the Patch Installer to the existing database.

To install the Data Model, follow these steps:

1. Install the system objects (tablespaces, roles, and users) needed for Patch Installer application. Refer to the note below if the tablespaces are built manually by the Database Administrator prior to installing the Patch Data Model.
2. Install Patch Data Model objects for the patch user.

Note: Install the Patch Data Model before installing any patches, also 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 Patch Data Model, you need `db_variables.cfg` configuration file. This file describes your database environment and resides in the `<PI Installed Directory>/database/db_tools/mantas_cfg` directory.

Note: By default, the installation process creates tablespaces. If the database administrator has created tablespaces manually, you must modify the `<PI Installed Directory>/database/patch_schema/CR_13696933.system.sql` file and comment the following line (that is, insert two hyphens "--" at start of the line) before using the following procedure.

```
CREATE TABLESPACE &&patch_data_tablespace DATAFILE '&&patch_data_filename' SIZE
&&patch_data_size
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO;
```

```
CREATE TABLESPACE &&patch_index_tablespace DATAFILE '&&patch_index_filename' SIZE
&&patch_index_size
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO;
```

Installing the System Objects

To install the System Objects, follow these steps:

1. Navigate to `<PI Installed Directory>/database/db_tools/mantas_cfg` directory.
2. Edit the `db_variables.cfg` file to replace `<CHANGE_ME>` to datafile file path.

Table 11 on page 49 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.

3. Change to `<PI Installed Directory>/database/patch_schema` directory and run the Database Builder Utility to create system objects using the following command:

```
../db_tools/bin/run_dbbuilder_utility_pdb.sh CR_13696933.system.cfg
```

The system prompts 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 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 `patch_util_pwd`. The value provided is used as password for the patch schema user when it is created. As a security measure, the value entered is 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.

Note: Refer to `<PI Installed Directory>/database/db_tools/mantas_cfg/db_variables.cfg` file for schema names written against the variables provided in brackets of respective schema.

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 following message: *Database Builder Utility has run successfully*. The database schema users should not contain any special characters and must comprise of only of letters and numbers. For information on database schema passwords refer to *Oracle Database Password Policy*.

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

When the system objects have been successfully installed, you can install the Patch Data Model objects.

Installing the Patch Data Model Objects

To install the Patch Data Model objects, follow these steps:

1. Change to following sub-directory: `<PI Installed Directory>/database/patch_schema`.
2. Run the Database Builder Utility to create the Patch Data Model Objects using following commands:

```
../db_tools/bin/run_dbbuilder_utility_pdb.sh CR_13696933.cfg
```

```
../db_tools/bin/run_dbbuilder_utility_pdb.sh PR_14189718.cfg
```

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*.

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

Once Patch Data Model installation is complete, you may decrease the log trace level to default setting `log.trace=false` in <PI Installed Directory>/database/db_tools/mantas_cfg/install.cfg.

Installing Multiple Instances of the Patch Installer

The Patch Installer can install patches only on the physical server where it is installed. In case you have any of the OFSFCCM product components installed on different physical servers use this procedure to install the Patch Installer software on multiple servers.

Understanding the Process

To create multiple Patch Installer instances, follow these steps:

1. Determine how many additional servers are required to meet Patch Installation requirements.
2. Create a primary instance by running a silent installation on the Patch Installation 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.

Once you have completed the installation and configuration of multiple instances of the Patch Installation, you can begin data installing patches, refer to the *Chapter 4 - Using the Patch Installer*, for more information.

This chapter contains the following sections:

- Instance Registration
- Patch Installation
- Patch Un-installation
- Instance Un-Registration
- Patch Installer Usage

Instance Registration

All new customers must register any application instances for which the Patch Installer is used. An application instance is identified by a server name and an instance name as configured in the `install.cfg` file for the Patch Installer. The pairing of server name and instance name must be unique. The registry information will help validate product details for literal errors during patch installation.

Customers who wish to install patches for a product must register the product with the Patch Installer. This must be done from each physical server on which the product is installed to allow the Patch Installer to record information about each server and subsystem.

To register an application, follow these steps:

1. Navigate to the following directory.

```
cd <PI Installed Directory>/patch_installer
```

2. Execute the `patchUtil.sh` script file as follows.

```
patchUtil.sh register
```

After the execution of the above command, the Patch Installer will ask for the Product Name, Product Version, and Product Installed directory as shown below:

For Stage 1 (Oracle Financial Services Behavior Detection Framework Hotfixes), register as shown below:

```
<PI Installed Directory>/patch_installer patchUtil.sh register
```

```
Enter Product Name: BDF
```

```
Enter Product Version: 6.2
```

```
Enter the Product Install Directory: <OFSBDF Installed Directory>
```

```
Product Registered Successfully.
```

For Stage 3 (OFSFCCM Hotfixes), register as shown below:

Using the Patch Utility

```
<PI Installed Directory>/patch_installer patchUtil.sh register
```

Enter Product Name: FCCM

Enter Product Version: 6.2

Enter the Product Install Directory (Press 'Enter' to register only 'ftpshare') : <FIC_HOME>

Enter the Install Directory for 'ftpshare' (Press 'Enter' to skip): <Absolute path of parent directory of ftpshare>

Enter Folder Name for 'ftpshare' (Press 'Enter' for 'ftpshare'):

Product Registered Successfully.

For multi-tier architecture, each server should be registered individually, by keeping same Patch Instance Name as found in <PI Installed Directory>/patch_installer/mantas_cfg/install.cfg.

Validations Performed During Instance Registration

During registration, the Patch Installer checks whether the product is already registered for the same instance on the same server or machine. If it was previously registered, then an error message displays and cancels the registration process. Based on the application path specified during installation, the Patch Installer will pick up valid subsystems defined for the product in the `install.cfg` file.

Log Files and Registration History

The Patch Installer produces a log file that includes messages indicating the progress of the product registration process. The log files should be examined for error messages, and should be preserved to review if a subsequent problem occurs.

Patch Installation

OFS Financial Crime and Compliance Management software updates are provided in a package that includes the updated software components and a manifest file, which contains information about the update. Using the information in the manifest, the Patch Installer performs various validations to ensure that the patch is installed properly.

Note: Unless the Application has been registered in the Patch Installer, you may not install or un-install patches.

To install a patch, follow these steps:

1. Navigate to the following directory.

```
cd <PI Installed Directory>/patch_installer
```

2. Execute the `patchUtil.sh` script file as follows.

```
patchUtil.sh install
```

Validations Performed During Patch Installation

The patch is validated to ensure that it is not corrupted or modified. The Patch Installer ensures that the patch being applied is compatible with the correct OFS Financial Crime and Compliance Management software release installed on your machine.

Some patches require the prior installation of another patch or patches. The Patch Installer ensures that these prior patches, or dependencies, have already been applied to the OFS Financial Crime and Compliance Management software.

The manifest also includes the version identifier for each of the updated software components. The Patch Installer compares the versions of the updated components to previously installed components in order to prevent installation of older versions.

The Patch Installer validates registry details based on product details present in the `KDD_SYS_INFO` table. The data is loaded into the `KDD_SYS_INFO` table during the instance registration process.

Patches Which Apply to Multiple Subsystems

Some patches include components for several subsystems that may be installed on multiple servers. The Patch Installer installs the updated components for each of the affected subsystems on a server, and informs the user when additional installation steps are required on additional servers to complete the patch installation.

Backup Processing

The Patch Installer makes backup copies of components being updated. If an error occurs during the installation, then backup copies of the original patch components are re-instated. Backup copies can also facilitate the un-installation of future patches, if necessary.

Manual Steps

A patch may require manual steps for a successful installation. For example, updates to configuration files may require manual editing of the configuration files to prevent loss of site-specific configuration information. If manual steps are required to complete a patch installation, then the Patch Installer displays a warning message.

Handling Customizations

OFS Financial Crime and Compliance Management software can be customized in various ways. For example, certain User Interface (UI) components can be customized by placing site-modified files in the `custom` directory. If a patch includes updates to a file that has been customized by the site, then Patch Installer displays a warning message to inform the user that customizations exist for the indicated patch component. Because the Patch Installer does not install components in the `custom` directory, the customized version of the UI files override the updated versions of those files provided in the patch. To maximize the benefits of a patch update, the UI customizations should be re-applied to the updated versions of the files provided in the patch.

OFS Financial Crime and Compliance Management scenarios can also be customized. When an updated scenario provided in a patch is loaded into the database, the customized version of that scenario are overwritten, losing the

affect of the customization. Because the Patch Installer cannot detect that an installed scenario has been customized, care should be taken when loading updated scenario versions. Keep a list of the customized scenarios, and the details of each customization, to prevent loss of customized functionality. Keep backup copies of customized scenarios, and contact Oracle Professional Services for assistance in re-applying customizations to updated scenarios.

Log Files and Installation History

The Patch Installer produces a log file that includes log messages indicating the progress of the patch installation. The log files should be examined for error messages, and should be preserved for later review in case a subsequent problem occurs.

The Patch Installer maintains a history of all patch installations and can be used to display this history information, as well as the current status of the installed patches.

Patch Un-installation

It may be necessary to un-install a previously installed patch. The Patch Installer can be used to un-install patches. The Patch Installer performs various validations during the un-installation process to ensure the integrity of the software installation.

To un-install a patch, follow these steps.

1. Navigate to the following directory.

```
cd <PI Installed Directory>/patch_installer
```
2. Execute the `patchUtil.sh` script file as follows.

```
patchUtil.sh uninstall
```

Validations Performed During Patch Un-installation

During patch installation, the Patch Installer validates that the original files have not been modified or corrupted. The Patch Installer ensures that there are no patches that are dependent upon the patch being un-installed. Warning messages are displayed if dependent patches exist. The dependent patches must be un-installed before the un-installation of any other files.

Manual Steps

A patch may require manual steps to complete the successful un-installation of the patch. Some manual steps performed during the installation of a patch cannot be easily reversed. For example, database updates applied during patch installation may be impossible to undo; scenarios that were reloaded during a patch installation will have to be manually reloaded from a backup copy to undo the updates applied; updates made to configuration files during patch installation may require manual editing of the configuration files to reverse the affects of the update.

Log Files and Un-Installation History

The Patch Installer produces a log file that includes log messages indicating the progress of the patch un-installation process. The log files should be examined for error messages, and should be preserved for later review in case a subsequent problem occurs.

The Patch Installer maintains a history of all patch un-installations. The Patch Installer can be used to display this history information, as well as the current status of the installed patches.

Instance Un-Registration

Any applications which have been registered in the Patch Installer can be un-registered. This will not allow further patch installation or un-installation.

To un-register an application, follow these steps:

1. Navigate to the following directory.
`cd <PI Installed Directory>/patch_installer`
2. Execute the `patchUtil.sh` script file as follows.
`patchUtil.sh unregister`

Once the application has been un-registered, the Patch Installer will ask for the Product Name and Product Version as shown below.

For Stage 1 (OFSBDF Hotfixes):

```
<PI Installed Directory>/patch_installer patchUtil.sh unregister
```

Enter Product Name: BDF

Enter Product Version: 6.2

Product Un-Registered Successfully.

For Stage 3 (OFSFCCM Hotfixes):

```
<PI Installed Directory>/patch_installer patchUtil.sh unregister
```

Enter Product Name: FCCM

Enter Product Version: 6.2

Product Un-Registered Successfully !!

Validations Performed During Instance Un-Registration

During un-registration, the Patch Installer checks whether the product is already registered for the same instance on the same server or machine. If it is registered then the utility will un-register the application. If the product is not registered, then an error message displays and cancels the un-registration process.

Log Files and Un-Registration History

The Patch Installer produces a log file that includes messages indicating the progress of the un-registration process. The log files should be examined for error messages, and should be preserved to review if a subsequent problem occurs.

Note: Registration and un-registration are one-time activities. If a user wants to install patches on a new application, then the application must be registered. After registration, patches may be applied to the application.

Patch Installer Usage

The following sub-sections describe how the Patch Installer is used, and the configuration steps that must be performed prior to running the Patch Installer.

Patch Installer Configuration

The `KDD_SYS_INFO` table is populated by the Patch Installer when the application is registered. Also, whenever an update to the table is needed (for example, when a new subsystem is installed on an existing server), the Patch Installer takes care of the subsystem for an instance by using product details present in the `KDD_SYS_INFO` table.

Note: Each time the Patch Installer starts, it synchronizes the `KDD_SYS_INFO` table with all the subsystems installed on the server. It updates the `STATUS_CD` column with the status of all the subsystems. If a subsystem is present then its value becomes I (denotes installed or registered). If a subsystem is removed then its value becomes U (denotes un-installed). If an application for an instance is un-registered then its value becomes UR (denotes un-registered). For all newly added subsystems, a fresh entry is done in the `KDD_SYS_INFO` table and the value of the `STATUS_CD` column becomes I.

The Patch Installer should point to a separate database for easy migration and refresh activities for Disaster Recovery environments. Often clients refresh their non-production environments from production. The Patch Installer should point to a separate database (Patch database) to perform these tasks without affecting the patch related information pertinent to the environment.

Common Resources

The configuration files are used to configure common resources such as database configuration, directing output messages, setting up logging activities, and so forth. Common resource files include the following:

- `install.cfg` file
- `categories.cfg` file
- `mantas_patchutil_message_lib_en.dat` file
- `Manual_Instructions.xml` file

install.cfg File

Configuration information resides in the `<PI_Installed Directory>/patch_installer/mantas_cfg/install.cfg` file. The configuration file contains instructions that can be modified for Oracle database drivers. The configuration file provides information that the Patch Installer needs for installing or un-installing a patch, as well as patch query information. It also provides how to set the user name and password that you need to connect to the database. In this file, you can modify values of specific installer parameters, change the locations of output files, and specify database details. Passwords are set by the Password Manager Utility. For more information about the Password Manager Utility, refer to the *Administration Guide*

The `install.cfg` file contains information unique to Patch Installer and common configuration parameters; headings in the file clearly identify the installer's parameters. You can also modify the current logging configuration (for example, activate or deactivate particular logging levels and specify locations for logging entries).

Using the Patch Utility

You can also remove the backup directory while un-installing the patch by configuring the `uninstall.DeleteBackupFiles` parameter in the `install.cfg` file. If the value of `uninstall.DeleteBackupFiles` parameter is set as **false**, the Patch Installer does not delete file from backup folder. If the value of `uninstall.DeleteBackupFiles` parameter is set as **true**, the Patch Installer deletes file from backup folder on un-installing the patch.

The following pages contain a sample `install.cfg` file with common and utility-specific information Figure 1. Logging information appears at the end of the file.

```
# The file contains Properties used by the Patch Utility for accomplishing tasks
# like Instance Registration/Un-registration, Patch Installation, Patch
un-Installation,
# View Patch information/History and Patch Data Migration

##### INFORMATION PERTAINING TO DATABASE CONNECTION #####

# database driver for connecting to PDB
# Example:
# database.driverName=oracle.jdbc.OracleDriver
database.driverName=oracle.jdbc.OracleDriver

# database URL for PDB (MANDATORY Parameter)
# Example:
# pdb.database.urlName=jdbc:oracle:thin:@dbserver:1521:compdb
pdb.database.urlName=jdbc:oracle:thin:@ofss220074.in.oracle.com:1521:Ti1011L56

# database login user for PDB (MANDATORY Parameter)
# Example:
# pdb.database.username=PATCH_UTL
pdb.database.username=ECM62_B08_PATCH

# PDB password for the login user mentioned above (MANDATORY Parameter)
pdb.schema.password=EszReAekX7Bj5pYU2omHy++63tBN1MCBHRxPNuE96Uc=

# Database Schema Owner for PDB (MANDATORY Parameter)
```

(Continued from last page)

```
# Example:
# schema.patch.owner=PATCH_UTL
schema.patch.owner=ECM62_B08_PATCH

# File paths for installation and uninstallation (MANDATORY Parameter)

#Specify the full path of the directory where the backups for the replaced
# components would be maintained (MANDATORY Parameter)
# Example:
# patch.backup.dir=/software/product/backup
patch.backup.dir=/scratch/ofsaapp/ECM6.2/ECM6.2.0.1.2/PI62012/backup

# patch temp directory - path where patch archive will be extracted (MANDATORY
Parameter)
# Example:
# patch.temp.dir=/users/product/tmp
patch.temp.dir=/scratch/ofsaapp/ECM6.2/ECM6.2.0.1.2/PI62012/temp

# server name - name of the server machine where patch utility is installed (MANDATORY
Parameter)
# Example:
# patch.server.name=prodsrvr
patch.server.name=ofss220101.in.oracle.com

# Instance name - Instance name of product (MANDATORY Parameter)
# Example:
# patch.instance.name=product_engine_prod
patch.instance.name=PI62012

# Delete Backup Directory : Setting the below flag to true denotes that the backup
directory for a patch
# shall be deleted once it has been un-installed; By default, the value is set to
false, specifying that
# the backup directory shall not be deleted
uninstall.DeleteBackupFiles = false
```

(Continued from last page)

```
#Date format
patch.date.format = yyyy-MM-dd HH:mm:ss

# Display formats for Patch Info and Patch History
# Do not Edit this section of the file
patch.header.format = %1$-25s: %2$-50s
component.header.format = %1$-50s %2$-26s %3$-12s
component.display.format = \n %1$-15s: %2$-50s
action.display.format = %1$-22s %2$-12s %3$-15s %4$-50s

# Product Subsystems: The values specified will be used to validate the product
subsystems while creating patch.
# This ensures that the patch can have only defined set of subsystems for the product.
# Do not edit this section of the file
product.subsystem.MANTAS=behavior_detection,alert_management,ingestion_manager,service
s,patch_installer,golden_data,active_pages,data_ingest,scenarios,database,bdf
product.subsystem.FCCM=ficweb,ficapp,ficdb,ftpshare,obi,OBIEE,AM,alert_management

# Specify the special consideration subsystems which will have different install home
compared to the product
# product.subsystem.spcons=<subsystems separated by comma>
# Do not edit this section of the product
product.subsystem.spcons=ftpshare

# Infodom standards for the subsystem
# Do not edit this section of the file

subsystem.infodom.FTPSHARE=RRS,KYC,OR,AM,CM,FSDF,CTR,FATCA

#### DATA MODEL CONFIFURATION #####

#Data Model Upload directory name
patch.subsystem.datamodel.directory=ERWIN

#Library path of ficapp
datamodel.library=ficapp/common/FICServer/lib/
```

(Continued from last page)

```
#Data Model user name to upload
datamodel.username = sysadmn
```

```
#Data Model type like AM - incremental, N - New, or R - Rebuild codes
datamodel.uploadmode = AM
```

```
#Directory datamodel refers to for upload in Data Model Upload directory present in
infodom
datamodel.filecopy.path = erwinXML
```

```
#### PATCH UTILITY MIGRATION SPECIFIC CONFIFURATION
#####
```

```
# database connection URL for extracting patch data from old Mantas schema
# Example:
#extract.conn.url=jdbc:oracle:thin@dbserver:1521:compdb
extract.conn.url=@PATCH_EXTRACT_DB_URL@
```

```
#database user name for extracting the patch data from old Mantas schema
# Example:
# extract.database.username=DB_UTIL_USER
patch.extract.username=@PATCH_EXTRACT_USER@
```

```
#database user password for extraction
patch.extract.password=
```

```
# Mantas schema name for Extracting the Patch data to XML file(s)
# Example:
# extract.schema.mantas=MANTAS
extract.schema.mantas=@PATCH_EXTRACT_SCHEMA@
```

```
# Path to the directory where the XML files would be stored after extracting data from
the Mantas Schema
# Example:
# extract.dirname=/software/mantas/patchmig
extract.dirname=@PATCH_EXTRACT_DIR_PATH@
```

(Continued from last page)

```
# File name prefix for extracted data for the XML file created after extracting the patch data
```

```
# Example:
```

```
# extract.file.prefix=BDP_PATCHMIG_
```

```
extract.file.prefix=BDP_PATCHMIG_
```

```
# database connection URL for loading data into PDB
```

```
# Example:
```

```
#load.conn.url=jdbc:oracle:thin@dbserver:1521:compdb
```

```
load.conn.url=jdbc:oracle:thin:@ofss220074.in.oracle.com:1521:T1011L56
```

```
#database schema name for load
```

```
# Example:
```

```
# load.schema.pdb=PATCH_UTL
```

```
load.schema.pdb=ECM62_B08_PATCH
```

```
# PDB database user name for loading patch data
```

```
# Example:
```

```
# patch.load.username=PATCH_UTL
```

```
patch.load.username=ECM62_B08_PATCH
```

```
# PDB database user password for loading
```

```
patch.load.password=
```

```
# Path to the directory from where the XML files containing Patch information
```

```
# (extracted from Mantas Schema) are stored. The Utility would read XML files
```

```
# from this location for loading the Patch information into the PDB
```

```
# Example:
```

```
# load.dirname=/software/mantas/patchmig
```

```
load.dirname=@PATCH_LOAD_DIR_PATH@
```

(Continued from last page)

```
##### Database Builder Utility Configuration #####
#
# File containing tokens and their value
db_tools.tokenfile=mantas_cfg/db_variables.cfg

Oracle.DuplicateRow=1
Oracle.ObjectExists=955,2260,2275,1430,1442,1451,957,1408,2261
Oracle.ObjectDoesNotExist=942,1418,1434,2441,904,4043,1927,2443

dbscript.execution.users=(system|business|mantas|market|miner|ingest|report|kdd|algorithms|case|config|fatca|ctr|kyc|fsdf)

schema.mantas.owner=ECM62_B08_MANTAS
utils.miner.user=ECM62_B08_KDD_MNR
utils.miner.password=AC70a0/6XtbqRC/rd0qTXYD/80/HyV9PowFUOSL3niI=
schema.business.owner=ECM62_B08_BUSINESS
schema.market.owner=ECM62_B08_MARKET
ingest.user=ECM62_B08_INGEST_USER
ingest.password=8vVxOMt3I+b5OJY2TOVsbrAL6YLlhqmCH5jrv9mwM4g=

schema.kdd.owner=ECM62_B08_KDD
casemng.schema.owner=ECM62_B08_CASE

utils.database.urlName=${load.conn.url}
utils.database.username=ECM62_B08_DB_UTIL_USER
utils.database.password=Vu6QbmI8Lx08QK69n5lsgG1+msubiKxXCJBYvH//Lbk=

#### GENERAL SCENARIO MIGRATION SETTINGS

#Specify the flags for whether scoring rules and wrapper datasets need to be extracted
or loaded
score.include=N
wrapper.include=N

#Specify the Use Code for the scenario. Possible values are 'BRK' or 'EXP'
load.scnro.use=BRK
```

(Continued from last page)

```
#If custom patterns exist for a product scenario, set to 'Y' when loading a scenario
hotfix.
#This should normally be set to 'N'.
load.ignore.custom.patterns=N

#Specify the full path of depfile and name of fixfile used for extraction and loading
#Note : fixfile need not be specified in case of loading
sm.depfile=mantas_cfg/dep.cfg

sm.release=5.7.1

#### LOAD

# Specify the jdbc driver details for connecting to the target database
load.conn.driver=oracle.jdbc.OracleDriver
load.conn.url=jdbc:oracle:thin:@ofss220074.in.oracle.com:1521:Tl1011L56

#Target System ID
load.system.id=Tl1011L56

# Specify the schema names for Load
load.schema.mantas=ECM62_B08_MANTAS
load.schema.business=ECM62_B08_BUSINESS
load.schema.market=ECM62_B08_MARKET
load.user.miner=ECM62_B08_KDD_MNR
load.miner.password=${utils.miner.password}

#Directory where scenario migration files reside for loading
load.dirname=data

# Specify whether threshold can be updated
load.threshold.update=Y

# Specify whether or not to verify the target environment on load
verify.target.system=N
```

(Continued from last page)

```
#### GENERAL PATCH UTILITY SETTINGS #####
#### LOG CONFIGURATION #####

# Specify which priorities are enabled in a hierarchical fashion, i.e., if
# DIAGNOSTIC priority is enabled, NOTICE, WARN, and FATAL are also enabled,
# but TRACE is not.
# Uncomment the desired log level to turn on appropriate level(s).
# Note, DIAGNOSTIC logging is used to log database statements and will slow
# down performance. Only turn on if you need to see the SQL statements being
# executed.
# TRACE logging is used for debugging during development. Also only turn on
# TRACE if needed.

log.fatal=true
log.warning=true
log.notice=true
log.diagnostic=false
log.trace=false
log.time.zone=US/Eastern

# Specify whether logging for a particular level should be performed
# synchronously or asynchronously.

log.fatal.synchronous=false
log.warning.synchronous=false
log.notice.synchronous=false
log.diagnostic.synchronous=false
log.trace.synchronous=false

# Specify the format of the log output. Can be modified according to the format
# specifications at:
# http://logging.apache.org/log4j/docs/api/org/apache/log4j/PatternLayout.html
# NOTE: Because of the nature of asynchronous logging, detailed information
# (class name, line number, etc.) cannot be obtained when logging
```

(Continued from last page)

```
# asynchronously. Therefore, if this information is desired (i.e. specified
# below), the above synchronous properties must be set accordingly (for the
# levels for which this detailed information is desired). Also note that this
# type of detailed information can only be obtained for Java code.
```

```
log.format=%d [%t] %p %m%n
```

```
# Specify the full path and filename of the message library.
```

```
#log.message.library=mantas_cfg/etc/mantas_patchutil_message_lib_en.dat
```

```
log.message.library=mantas_cfg/etc/mantas_patchutil_message_lib_en.dat
```

```
# Specify the full path to the categories.cfg file
```

```
#log.categories.file.path=mantas_cfg/
```

```
log.categories.file.path=mantas_cfg/
```

```
# Specify where a message should get logged for a category for which there is
# no location property listed above.
```

```
# This is also the logging location of the default product category unless
# otherwise specified above.
```

```
# Note that if this property is not specified, logging will go to the console.
```

```
log.default.location=logs/PatchUtil.log
```

```
# Specify the maxfile size of a logfile before the log messages get rolled to
# a new file (measured in MBs).
```

```
# If this property is not specified, the default of 10 MB will be used.
```

```
log.max.size=
```

(Continued from last page)

```
#NOTE: The values for the following variables need not be changed
# Specify the ID range for wrapper datasets
dataset.wrapper.range.min=113000001
dataset.wrapper.range.max=114000000
product.id.range.min=113000000
product.id.range.max=200000000
```

Figure 1. Sample install.cfg

categories.cfg File

In the <PI Installed Directory>/patch_installer/mantas_cfg/categories.cfg file, you can modify the default location for direct logging output of each function within the Patch Installer. The entries that you make require a specific format.

mantas_patchutil_message_lib_en.dat File

The Common Logging component provides a centralized mechanism for logging OFS Financial Crime and Compliance Management messages, in which the system places all log messages in a single message library file.

The message template repository resides in a flat text file and contains messages in the format <message id 1> <message text>.

The *log.message.library* in the Patch Installer's `install.cfg` file contains the full path to the message library file.

Patch Package

The Patch Archive file contains updated components, manifest, and manual instruction information. A separate ReadMe file is shipped with the patch detailing the components delivered as a part of the patch.

Patch Information

To view information about an installed patch, follow these steps:

1. Navigate to the following directory.
`cd <PI Installed Directory>/patch_installer`
2. Execute the `patchUtil.sh` script file as follows.
`patchUtil.sh info`

Note: You may choose to view information for a particular installed patch or for all installed patches.

Patch History

To view Patch Installation History, follow these steps:

1. Navigate to the following directory.
`cd <PI Installed Directory>/patch_installer`
2. Execute the `patchUtil.sh` script file as follows.
`patchUtil.sh hist`

Note: You may choose to view installation history for a particular patch or for all patches.

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
FTP	File Transfer Protocol
FSDF	Financial Services Data Foundation
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
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

Variables Used in the Silent Properties File

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

Files that the Installation CD Contains

The installation CD contains the `utilities.properties.sample` file that is installed on a Patch Installer server. This file is prepopulated with default values that are used in the Patch Installer test environment.

Use this appendix as a reference to populate the file.

Note: The schema name must be entered only in UPPER case and without a trailing space.

Table 9. Silent Mode Installation Variables

Installation Variable	
General Installer Parameters	
INSTALLER_UI	<p>Specifies whether the installer prompts the user for field values or reads them from file. The values should be read only from the file.</p> <p>Value:</p> <p>INSTALLER_UI=silent</p>
Locale	
USER_LOCALE	<p>Sets the system locale.</p> <p>Default:</p> <p>USER_LOCALE=en</p>
Product Installation Directory	
USER_INSTALL_DIR	<p>Identifies the directory in which the Patch Installer software is installed. If the directory does not exist, the installer creates it.</p> <p>Its referred in the document as <PI Installed Directory></p>
Select Components to Install	
DATABASE	<p>Installs the database subsystem and the scripts to create the data model if set to <i>true</i>.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ● true ● false

Table 9. Silent Mode Installation Variables

PATCH_INSTALLER	<p>Installs the patch installer if set to <i>true</i>.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ● true ● false
Common Variables	
SQL_DRIVER_JDBC	JDBC driver use oracle.jdbc.OracleDriver.
ORACLE_HOME	<p>Full path to the Oracle software installation. Contact your Database Administrator for database home directory information.</p> <p>For example:</p> <p>ORACLE_HOME =/kds/sparc-sun-solaris10/pkg/oracle/product/10.2.0</p>
INGEST_JAVA_HOME	<p>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.</p> <p>For example:</p> <p>INGEST_JAVA_HOME =/kds/sparc-sun-solaris10/pkg/jdk1.6.0</p>
DB_TNS_ADMIN	<p>Full path to the directory where the tnsnames.ora file resides. Typically, it is in the network/admin sub-directory of the Oracle installation.</p> <p>For example:</p> <p>DB_TNS_ADMIN=/kds/oracle/net</p>
Patch Installer Variables	
PDB_UTILS_DB_USER	<p>Patch database user that the patch installer uses to connect to the database to keep track of hotfixes installed.</p> <p>For example:</p> <p>PDB_UTILS_DB_USER=PDB_UTILS_DB_USER</p>
PATCH_SCHEMA_OWNER	<p>Schema where patch data resides.</p> <p>For example:</p> <p>PATCH_SCHEMA_OWNER=pdb</p>

Table 9. Silent Mode Installation Variables

PATCH_BACKUP	The installer creates a backup of the files being replaced as a result of patch installation. These files are reverted back to the installed location in case a patch is un-installed.
PATCH_TEMP	Directory where the installer extracts the contents of the patch, and uses it as a working directory for installation/uninstallation of patches.
SERVER_NAME	Hostname of the server on which the patch installer is running. For example: SERVER_NAME=<dbhost>
INSTANCE_NAME	Product instance name of the server on which the patch installer is running. For example: INSTANCE_NAME= UAT
DB_URL	JDBC URL to database. OFSDBF application recommends Thin driver in connecting to the database. jdbc:oracle:thin:@<db_server>:<db_port>:<oracle_sid> > For example: jdbc:oracle:thin:@ofss220074.in.oracle.com:1521:Ti1O11L56
PDB_URL	JDBC URL to patch database. Patch Installer application recommends Thin driver in connecting to the database. For example: jdbc:oracle:thin:@ofss220074.in.oracle.com:1521:Ti1O11L56
db_inst	Name of the database instance to which the software points is the same across all subsystems. For example: db_inst=T509S10
Users	

Table 9. Silent Mode Installation Variables

BUSINESS_SCHEMA_OWNER	<p>Schema where business data resides.</p> <p>For example:</p> <p>BUSINESS_SCHEMA_OWNER=BUSINESS</p>
DB_UTILS_DB_USER	<p>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.</p> <p>For example:</p> <p>DB_UTILS_DB_USER=DB_UTIL_USER</p>
KDD_ALGORITHM_USER	<p>Database user that the Behavior Detection subsystem uses to connect to the database to run Behavior Detection jobs.</p> <p>For example:</p> <p>KDD_ALGORITHM_USER=KDD_ALG</p>
KDD_MINER_USER	<p>Data miner account used to connect to the database for performing scenario related operations.</p> <p>For example:</p> <p>KDD_MINER_USER=KDD_MNR</p>
KDD_SCHEMA_OWNER	<p>Database schema that Patch Installer uses to store basic reference metadata for the operation of the Behavior Detection algorithms. No process or user logs into this schema directly.</p> <p>For example:</p> <p>KDD_SCHEMA_OWNER=KDD</p>
MANTAS_SCHEMA_OWNER	<p>Schema where Patch Installer Mantas data resides.</p> <p>For example:</p> <p>MANTAS_SCHEMA_OWNER=MANTAS</p> <p>This is the Alert Management database User which is one of the four Atomic schemas of Financial Crime and Compliance Management - Stage 3.</p>
WEB_DBUSER	<p>This value should be same as value for kdd_web_user.</p> <p>For example:</p> <p>WEB_DBUSER=KDD_WEB</p>
REPORT_USER	<p>Schema provided to users to support reporting applications built on top of Patch Installer.</p> <p>For example:</p> <p>REPORT_USER=KDD_REPORT</p>

Table 9. Silent Mode Installation Variables

<p>CONFIG_USER</p>	<p>Account that the OFSAAI uses to connect to the database.</p> <p>Note: Make sure to keep maximum length of this user upto 20 characters</p> <p>For example:</p> <p>CONFIG_USER=REVELEUS</p> <p>In case for OFSAAI has already been installed using separate user then provide SAME user id for this parameter.</p>
<p>PURGE_UTIL_DB_USER</p>	<p>Database user that the Alert/Case Purge installer uses to connect to the database to run the Alert/Case purge operations.</p> <p>For example:</p> <p>PURGE_UTIL_DB_USER=PURGE_UTIL_USER</p>
<p>FSDF_STG_SCHEMA_OWNER</p>	<p>Schema where FSDF staging slice reside.</p> <p>For example:</p> <p>FSDF_STG</p> <p>This is the Case Management database User which is one of the four Atomic schemas of Financial Crime and Compliance Management - Stage 3.</p>
<p>CASE_SCHEMA_OWNER</p>	<p>Account that the Web Application uses to connect to the database.</p> <p>For example:</p> <p>CASE_SCHEMA_OWNER=CMREVMAN</p> <p>This is the Case Management database User which is one of the four Atomic schemas of Financial Crime and Compliance Management - Stage 3.</p>
<p>KYC_SCHEMA_OWNER</p>	<p>Account that the Web Application uses to connect to the database.</p> <p>For example:</p> <p>KYC_SCHEMA_OWNER= KYC</p> <p>This is the KYC database User which is one of the four Atomic schemas of Financial Crime and Compliance Management - Stage 3.</p>
<p>DATA_INGEST_USER</p>	<p>Name of the data ingest user.</p> <p>For example:</p> <p>DATA_INGEST_USER=INGEST_USER</p>
<p>MARKET_SCHEMA_OWNER</p>	<p>Schema where market data resides.</p> <p>For example:</p> <p>MARKET_SCHEMA_OWNER=MARKET</p>

Table 9. Silent Mode Installation Variables

FATCA_USER	FATCA user only if FATCA is installed. For example: FATCA_USER=FATCA
CTR_USER	CTR user only if CTR is installed. For example: CTR_USER=CTR
KYC_USER	KYC user only if KYC is installed. For example: KYC_USER=KYC
REV_USER	Reveleus user only if Reveleus is installed. For example: REV_USER=REV
AM_INFODOM	AM infodom configured for the product
CM_INFODOM	CM infodom configured for the product
KYC_INFODOM	KYC infodom configured for the product
CTR_INFODOM	CTR infodom configured for the product
FATCA_INFODOM	FATCA infodom configured for the product
AM_SEGMENTAM	AM infodom segment configured for the product
CM_SEGMENT	CM infodom segment configured for the product
CTR_SEGMENT	CTR infodom segment configured for the product
FATCA_SEGMENT	FATCA infodom segment configured for the product

Table 9. Silent Mode Installation Variables

CONTEXT_NAME	Context Name for the product.
Roles	
DATA_LOADER_ROLE	<p>Role that has privileges to load data into the Business and Market schemas.</p> <p>For example:</p> <p>DATA_LOADER_ROLE=DATA_LOADER</p>
DATA_READER_ROLE	<p>Role that has privileges to read data into the Business and Market schemas.</p> <p>For example:</p> <p>DATA_READER_ROLE=DATA_READER</p>
KDD_ALGORITHM_ROLE	<p>Database role to which the <code>kdd_algorithm_user</code> is assigned. Rights to objects in the database are granted to the role, as opposed to the user.</p> <p>For example:</p> <p>KDD_ALGORITHM_ROLE=KDD_ALGORITHM</p>
KDD_ANALYST_ROLE	<p>Database role to which the <code>kdd_web_user</code> is assigned. Rights to objects in the database are granted to the role, as opposed to the user.</p> <p>For example:</p> <p>KDD_ANALYST_ROLE=KDD_ANALYST</p>
KDD_LOADER_ROLE	<p>Role that has privileges to load data into the KDD schemas.</p> <p>For example:</p> <p>KDD_LOADER_ROLE=KDD_LOADER</p>
KDD_MNR_ROLE	<p>Database role to which the <code>kdd_mnr_user</code> is assigned. Rights to objects in the database are granted to the role, as opposed to the user.</p> <p>For example:</p> <p>KDD_MNR_ROLE=KDD_MINER</p>
KDD_READER_ROLE	<p>Role that has privileges to read data in the KDD schemas.</p> <p>For example:</p> <p>KDD_READER_ROLE=KDD_READER</p>

Table 9. Silent Mode Installation Variables

MANTAS_LOADER_ROLE	Role that has privileges to load data into the Patch schemas. For example: MANTAS_LOADER_ROLE=MANTAS_LOADER
MANTAS_READER_ROLE	Role that has privileges to read data in the Patch schemas. For example: MANTAS_READER_ROLE=MANTAS_READER

This appendix provides a list of Patch Installer application environment variables that must be set in the `.cshrc` or `.profile` file.

Environment Variables

Table 12 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

Variable Name	Description	Example
PATH	A shell variable that specifies the location of the command you typed. In the example <code>.cshrc</code> or <code>.profile</code> file, the path variable is set to <code>"/usr/bin:/usr/local:/usr/local/bin:/usr/bin/X11:/usr/ucb:/usr/opt/bin"</code> . This setting tells the shell to look first in the <code>/usr/bin</code> directory, then in the <code>/usr/local</code> directory, next in the <code>/usr/local/bin</code> directory, and so on until the file has been found or all directories have been examined.	<code>.cshrc</code> or <code>.profile</code> file: <code>setenv PATH</code> <code>\${PATH}:/kds/sparc-sun-solaris10/pkg/jdk1.6.0/bin</code> <code>.profile</code> file: <code>export</code> <code>PATH=\${PATH}:/kds/sparc-sun-solaris10/pkg/jdk1.6.0/bin</code>
LD_LIBRARY_PATH	Provides the run-time shared library loader (<code>ld.so</code>) 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.	<code>.cshrc</code> or <code>.profile</code> file: <code>setenv LD_LIBRARY_PATH</code> <code>\${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</code> <code>.profile</code> file: <code>export</code> <code>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</code>

Table 10. Environment Variables

LANG	Required, along with NLS_LANG, to support double byte characters.	<p>.cshrc or .profile file:</p> <p>For Solaris OS: <code>setenv LANG en_US.UTF-8</code> For Linux OS: <code>setenv LANG en_US.utf8</code> For AIX: <code>setenv LANG EN_US.UTF-8</code></p> <p>.profile file:</p> <p>For Solaris OS: <code>export LANG=en_US.UTF-8</code> For Linux OS: <code>export LANG=en_US.utf8</code> For AIX: <code>export LANG=EN_US.UTF-8</code></p>
NLS_LANG	Required along with LANG, to support double byte characters.	<p>.cshrc or .profile file:</p> <p><code>setenv NLS_LANG AMERICAN_AMERICA.AL32UTF8</code></p> <p>.profile file: <code>export NLS_LANG=AMERICAN_AMERICA.AL32UTF8</code></p>
JAVA_HOME	Sets the Java installed directory.	<p>.cshrc or .profile file:</p> <p><code>setenv JAVA_HOME /kds/ sparc-sun-solaris10/pkg/jdk1.6.0</code></p> <p>.profile file: <code>export JAVA_HOME=/kds/ sparc-sun-solaris10/pkg/jdk1.6.0</code></p>

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 Patch data model.

Variables in the `db_variables.cfg` File

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

<CHANGE_ME> to be replaced by the `EXISTING` path in the Database server where the `*.dbf` files to be created while creating table space. For more details contact your Database Administrator.

The schema name must be entered only in upper case.

The below variables are distributed between the `db_variables.cfg` files present under:

<PI Installed Directory>/database/db_tools/mantas_cfg/

And

<PI Installed Directory>/patch_installer/mantas_cfg/

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

Variable Name	Description	Example
OFSFCCM Schema Owners		
<code>mantas_schema_owner</code>	Alert Management Schema User	MANTAS
<code>kdd_schema_owner</code>	Schema where base metadata will reside. Use KDD.	KDD
<code>business_schema_owner</code>	BUSINESS schema user.	BUSINESS
<code>market_schema_owner</code>	MARKET schema user.	MARKET
<code>web_user</code>	Web Application user.	KDD_WEB
<code>server_user</code>	Database user used to run Behavior Detection algorithms.	KDD_ALG
<code>tools_user</code>	Database user used to run Scenario Manager.	KDD_MNR
<code>ingest_user_name</code>	INGEST user.	INGEST_USER
<code>db_util_user</code>	Database user used to run database utilities	DB_UTIL_USER
<code>report_user</code>	Database user created to support reporting applications built on top of OFSPI.	KDD_REPORT

Table 11. Variables in the db_variables.cfg File

case_schema_owner	Database user with permissions on Case schema objects for loading	CMREVMAN
config_user	Configuration Schema User to be used for OFSAAI installation. OFSAAI can be installed on separate user also optionally. Note: In case OFSAAI has already been installed using separate user, then provide SAME User ID for this parameter. Make sure to keep maximum length of this user upto 20 characters	REVELEUS
kyc_schema_owner	KYC user to be used by Stage3 as 3 rd Atomic user	KYC
fsdf_stg_schema_owner	Schema where FSDF staging slice reside.	FSDF_STG
purge_util_user	Purge Utility user	PURGE
patch_util_user	Patch Installer user	PATCH
OFSFCCM 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
OFSFCCM 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 Alert Management Schema.	MANTAS_DATA

Table 11. Variables in the db_variables.cfg File

mantas_index_tablespace	Tablespace used for indexes in the Alert Management schema.	MANTAS_INDEX
business_data_tablespace	Tablespace used for data in the Business Schema.	DATA_BUS1_TBSP
business_index_tablespace	Tablespace used for indexes in the Business schema.	IDX_BUS1_TBSP
market_data_tablespace	Tablespace used for data in the Market Schema.	DATA_MKT1_TBSP
market_index_tablespace	Tablespace used for indexes in the Market schema.	IDX_MKT1_TBSP
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
case_data_tablespace	Tablespace used for data in the Case Schema.	CASE_DATA
case_index_tablespace	Tablespace used for indexes in the Case schema.	CASE_INDEX
kyc_data_tablespace	Tablespace used for data in the KYC Schema.	KYC_DATA
kyc_index_tablespace	Tablespace used for indexes in the KYC schema.	KYC_INDEX
fsdf_data_tablespace	Tablespace used for data in the FSDF schema	DATA_FSDF1_TBSP
fsdf_index_tablespace	Tablespace used for indexes in the FSDF schema	IDX_FSDF1_TBSP
patch_data_tablespace	Tablespace used for data in the Patch schema	PDB_DATA_TBSP
patch_index_tablespace	Tablespace used for indexes in the Patch schema	PDB_INDEX_TBSP
alert_index_tablespace	Tablespace used for indexes on the KDD_* tables in the TCM schema	ALERT_INDEX
alert_data_tablespace	Tablespace used for data in KDD_* tables in the TCM schema	ALERT_DATA
Physical Tablespaces Created During Installation		

Table 11. Variables in the db_variables.cfg File

DATA_BUS1_TBSP	Database tablespaces for Business data, Market data,FSDF data, and Patch data	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		IDX_MKT2_TBSP	
IDX_MKT3_TBSP		IDX_MKT3_TBSP	
IDX_MKT4_TBSP		IDX_MKT4_TBSP	
DATA_FSDF1_TBSP		DATA_FSDF1_TBSP	
IDX_FSDF1_TBSP		IDX_FSDF1_TBSP	
PATCH_DATA_TBSP		PDB_DATA_TBSP	
PATCH_INDEX_TBSP		PDB_INDEX_TBSP	
Logical Tablespace References Mapped to Physical Tablespaces			

Table 11. Variables in the db_variables.cfg File

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

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	
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 OFSPI tables.	/CHANGE_ME/mantas_data_01.dbf
mantas_index_file	File used for OFSPI 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
case_data_file	File used for Case tables.	/CHANGE_ME/case_data_01.dbf
case_index_file	File used for Case indexes.	/CHANGE_ME/case_idx_01.dbf
user_data_file	File used for Miner tables.	/CHANGE_ME/users_data_01.dbf
kyc_data_file	File used for KYC tables.	/CHANGE_ME/kyc_data_01.dbf

Table 11. Variables in the db_variables.cfg File

kyc_index_file	File used for KYC indexes.	/CHANGE_ME/kyc_idx_01.dbf
patch_data_filename	File used for Patch tables.	/CHANGE_ME/pdb_data_file01.dbf
patch_index_filename	File used for Patch indexes.	/CHANGE_ME/pdb_idx_file01.dbf
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

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
data_fsdf1_filename		/CHANGE_ME/fsdf1_data_file01.dbf
idx_fsdf1_filename		/CHANGE_ME/fsdf1_idx_file01.dbf
Data File Sizes		

Table 11. Variables in the db_variables.cfg File

kdd_data_size	Data file size for each data file name identified in <i>Data File Names</i> above.	64M
kdd_index_size		64M
mantas_data_size		512M
mantas_index_size		512M
alert_data_size		512M
alert_index_size		512M
miner_data_size		256M
miner_index_size		256M
case_data_size		512M
case_index_size		512M
user_data_size		512M
kyc_data_size		512M
kyc_index_size		512M
data_bus1_size		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
patch_data_size		256M
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
patch_index_size		256M
idx_bus8_size		256M
idx_mkt1_size		256M
idx_mkt2_size		256M
idx_mkt3_size		256M
idx_mkt4_size		256M
data_fsdf1_size		256M
idx_fsdf1_size	256M	

Table 11. Variables in the db_variables.cfg File

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
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'

Table 11. Variables in the db_variables.cfg File

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
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 <i>after</i> 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

Table 11. Variables in the db_variables.cfg File

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
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'

