Pharmacy Reengineering

VA Pharmacy Enterprise Customization System

PECS Installation Guide

November 2012

Department of Veterans Affairs
Office of Information and Technology (OIT)
Product Development (PD)

(This page included for two-sided copying)

Revision History

Date	Version	Description	Author
2/6/2013	3.15	Technical Updates to various sections(for PECS 2.2)	Sachin Sharma
11/26/2012	3.14	Updated to include adjustments identified during the FDB Comparison Report updates.	Alicia Ray
11/07/2012	3.13	Updated Revision History Numbering, updated dates	Brian Holihan
10/16/2012	3.12	Updated links, minor text updates	Brian Holihan
10/15/2012	3.11	Updated various section(Log4j template updated, help deployment section updated)	Sachin Sharma
09/14/2012	3.10	Added additional configure.exportfile.properties	Brian Holihan, Gary Duff
09/11/2012	3.9	Updated various sections to deploy PECS Help Build file	Sachin Sharma
09/10/2012	3.8	Formatting updates	Brian Holihan
08/17/2012	3.7	Modified the import data production steps to disable database constraints when loading data and re-enable them once complete.	Alicia Ray
07/20/2012	3.6	Formatting updates; update TOC, Footers, etc.	Brian Holihan
07/19/2012	3.5	Added Appendix E – Rollback Process	Gary Duff
07/10/2012	3.4	Updated formatting and pagination; updated date references; minor spelling and grammatical edits; updated Revision History	Brian Holihan
07/09/2012	3.3	Updated version references; added instructions regarding the data listener; added step to the post-migration procedure	Alicia Ray
07/05/2012	3.2	Changed component deployment order	Gary Duff
06/21/2012	3.1	Minor updates (table captions, step numbering, etc.)	Brian Holihan
6/14/2012	3.0	Updated various Sections for database changes for v2.2.	Alicia Ray
1/20/2012	2.6	Updated pagination, edits, TOC, and release date (back to November 2011)	Marella Colyvas

Date	Version	Description	Author
1/12/2012	2.5	Updated database configuration section to reflect lessons learned during the initial load by testing services	Alicia Ray
12/08/2011	2.4	Updated TOC and formatting	Marella Colyvas
12/07/2011	2.3	Updated database configuration section to include procedure for importing production data and database rollback procedures	Alicia Ray
11/29/2011	2.2	Updated document version number	Marella Colyvas
11/28/2011	2.1	Updated database configuration procedures to clarify installation and migration steps.	Alicia Ray
11/10/2011	2.0	Updated various Sections for configuration changes for v2.1. Updated formatting, acronym table, footers, and TOC.	Sachin Sharma, Gina Scorca
9/27/2011	1.18	Updated database configuration steps to reflect current Oracle environment and PECS installation requirements	Alicia Ray
07/19/2011	1.17	Updated formatting, edits.	Marella Colyvas
07/11/2011	1.16	Updated various Sections. (removed section 5.2- JMS configuration and C4 as no longer needed)	Sachin Sharma
05/29/2011	1.15	Updated JMS Section	Sachin Sharma
04/22/2011	1.14	Edit formatting, apply template	Susan Heiress
04/17/2011	1.13	Updated for PECS 2.0, added Logical Deployment section	Sachin Sharma
04/17/2011	1.12	Added WebLogic JTA section	Sachin Sharma
10/13/2010	1.11	Adjustments to JMS configurations due to a database user change.	PECS Team
08/04/2010	1.10	Added section on configuring JMS related to CCR 2902.	PECS Team
03/31/2010	1.9	Added the creation of the exportfile.properties to support the sending of custom update files to an FTP server.	PECS Team
02/10/2010	1.8	Updated Various Sections as per AITC Input	PECS Team- (Database Section Sreedhar, WebLogic Section Sachin Sharma.)

Date	Version	Description	Author
12/14/2009	1.7	Updated Various Sections	PECS Team
12/7/2009	1.6	Updated Various sections	SwRI
9/30/2009	1.5	Removed content for installation of WebLogic and KAAJEE. Updated Appendix A to contain log4j configuration	PECS Team
08/26/2009	1.4	Added Appendix B: Custom Update File Installation	PECS Team
08/12/2009	1.3	Removed jdbc.properties reference, as this file will be accessed on the server. Removed Post-Installation section, as no longer needed. Added section “4.5 Add CT_VERSION table to FDB schema”.	PECS Team
01/06/2009	1.2	Updated database datasource driver in WebLogic setup section.	PECS Team
12/02/2008	1.1	Added log4j.properties reference. Added KAAJEE jdbc.properties reference. Added kaajeeConfig.xml reference. Added user_staged_config WebLogic KAAJEE library directory reference.	PECS Team
11/24/2008	1.0	Initial version	PECS Team

1. Introduction

This document describes the process used to install the Department of Veterans Affairs (VA) Pharmacy Enterprise Customization System (PECS) application on an instance of a WebLogic server. The PECS software is a Web-based application, packaged as a J2EE standardized Enterprise Application Archive (EAR) file, which is then deployed on the WebLogic server using the server's standard deployment process. The installation described in this document also outlines the steps necessary to install and configure the application's database. This includes the installation of the database schema on an Oracle server, and loading data into configuration tables. The document outlines the configuration of two datasources, and the deployment of the EAR file on the WebLogic server. The installation of the PECS application assumes that the servers necessary to execute the software are configured and running as per any applicable VA standards.

In order to understand the installation and verification process, the installer should be familiar with the WebLogic 10.3 console administration and Oracle 11g Database configuration.

1.1 Assumptions

For successful deployment of the Pharmacy Reengineering (PRE) PECS software at a site, the following assumptions must be met:

- Red Hat Enterprise Linux 5 operating system is properly installed.
- The WebLogic Server 10.3.2 is configured and running.
- Access to the WebLogic console is by means of a valid administrative user name and password.
- Oracle 11g Database Server is configured and running.
- Java JDK version used is 1.6.0_14.
- FDB (First Data Bank) DIF (Drug Information Framework) database is installed. The FDB-DIF install instructions for database setup and FDB Updater tool are provided by FDB-DIF. They can be referenced from PRE SharePoint site.
    - The Install document(s) and Scripts to configure FDB-DIF are on the PRE SharePoint at http://go.va.gov/xjqr
    - The Install document(s) for install/use of FDB-DIF Tool to load FDB-DIF database is at http://go.va.gov/xjqr

Note:	Cut and paste any of the links shown to your browser. This document is unable to support live links. If you are unable to locate a referenced document, please contact the PRE Configuration Manager who should be identified on the project's Technical Services Project Repository (TSPR) site.

- Kernel Authentication & Authorization for J2EE (KAAJEE) security Application Program Interface (API) setup and configured on the WebLogic Server.
    - The KAAJEE(v1.1.0.007) installation documents can be reference at KAAJEE website: *http://vista.med.va.gov/kernel/kaajee/download_9-10.asp*
    - The KAAJEE(v1.1.0.007) Software/API can be downloaded from KAAJEE website : *http://vista.med.va.gov/kernel/kaajee/download_9-10.asp*
    - Please note that KAAJEE installation includes the KAAJEE security provider System Design Specifications (SDS) datasource on the target WebLogic server. The KAAJEE and SDS database should be configured as specified in the KAAJEE and SDS install guides respectively.
    - For Configuring SDS Datasource, please contact the Technical PRE-PECS point of contact at Austin Information Technology Center (AITC). (The Uniform Resource Locator (URL), username, password will be provided).
    - The above links are provided as reference; the Install Guides and Documentation are maintained by respective projects – KAAJEE, VistALink, SDS. If you are not able to reach the link (or any issue with documentation), please contact the respective group. (PRE-PECS team can also help to co-ordinate with above groups if required).
- The installation instructions are followed in the order that the sections are presented within this Installation Guide.

1.2 Scope

Installation steps in scope include:

- Installation of the PECS database staging schema on an existing Oracle server, and a data load into configuration tables.
- Configuration of database datasources on an existing WebLogic application server.
- Deployment of the PECS application EAR file on a configured WebLogic application server.

Processes out of scope include:

- Installation and configuration of server environments, including the operating system, database server, and application server, and/or any other network component as may be required to host the PECS application on the VA network.
- KAAJEE security API setup and configuration on the WebLogic server.
- FDB-DIF database installation or update process.
- Process to add or configure users in the VistA application for authentication and authorization to the PECS application.
- Process to check out the PECS codebase from the ClearCase repository and/or the build process.
- Installation details of the Java Runtime environment.
- Initial load of Pharmacy Benefits Management (PBM) customized order checks.

1.3 Definitions, Acronyms and Abbreviations

Here is a list of terms and acronyms and their definitions.

1.3.1 Definitions

Term	Definition
%DATAFILE_LOCATION%	The directory location where the PECS database schema file will be located.
Data Definition Language	A computer language for describing the records, fields, and "sets" making up a database
Datasource	Database connection definition, including connection pool on an application server.
Deployment Archive	A compressed file organized in the J2EE deployment standard.

1.3.2 Acronyms

Term	Definition
AITC	Austin Information Technology Center
API	Application Program Interface
CT	Custom Table
DBA	Database Administrator
DDL	Data Definition Language..
EAR	J2EE Enterprise Application Archive file.
FDB-DIF	First DataBank Drug Information Framework database
FTP	File Transfer Protocol
GUI	Graphical User Interface
J2EE	Java 2 Enterprise Edition
JMS	Java Messaging Service
KAAJEE	Kernel Authentication and Authorization for J2EE
PECS	Pharmacy Enterprise Customization System
PBM	Pharmacy Benefits Management
PRE	Pharmacy Reengineering
RDBMS	Relational Database Management System
SDS	System Development Support
SQL	Structured Query Language

Term	Definition
SSPI	Security Service Provider Interface
TSPR	Technical Services Project Repository
URL	Uniform Resource Locator
VA	Department of Veterans Affairs

1.4 Overview

The steps necessary to install and configure the components required by the PECS application are outlined in the following pages. The order that the components appear in the outlined steps is the suggested installation order. Installation Prerequisites should be installed or verified on the build environment first, followed by the installation of the database schema, application server configuration, and the deployment of the PECS application.

2. Installation Prerequisites

- Installation and configuration of server environments, including the operating system, database server, and application server, and/or any other network component as may be required to host the PECS application on the VA network.
- Target production VistA implementation must have PECS users and their security keys installed.
- KAAJEE Security Service Provider Interface (SSPI) Software, VistALink Software, and the KAAJEE Security Provider installed on the WebLogic application server.
- The target production FDB-DIF database is available.

3. Database Tier Installation

This section describes the operating system and software for the PRE PECS V.2.2 Database Tier installation and configuration. Initially, install and configure the operating system and software according to the manufacturer's specifications.

3.1 Oracle Database

The Custom Table (CT) staging schema or PECS Database is designed to be operating system independent. The only constraint is that Oracle 11g Database Enterprise Edition Release 11.2.0.2.0 – Production must be properly installed and configured. The following sections describe the installation, features, user creation, and configuration for the Oracle database.

Note:	Please note that the PECS staging database user should be configured as "CTSTAGING" (CTSTAGING schema) and the FDB-DIF database user should be configured as "FDB_DIF" (FDB_DIF schema).

3.1.1 Oracle Installation

Proper installation of the Oracle Relational Database Management System (RDBMS) is one in which the Oracle Universal Installer was used to perform an error-free installation and a general purpose instance was created. A properly configured Oracle RDBMS is one in which the associated Oracle application development and configuration tools, namely Structured Query Language (SQL)*Plus and Oracle Enterprise Manager, can be used to connect to the instance through a Transparent Network Substrate alias.

3.1.2 Oracle Configuration

The CT staging schema or PECS Database is the primary data repository for the PECS application. The database should be installed and configured appropriately for the PECS operating environment.

Two schemas must be created for the PECS Environment: FDB_DIF and CTSTAGING. Prior to creation of the schemas, logical and physical environment structures must be set up for storage of the schemas database objects: tablespaces and data files. For the PECS database configuration, data and index storage are separated for each schema. Separating indexes and table data is considered an Oracle best practice and provides improved run-time performance, reporting/monitoring, and manageability.

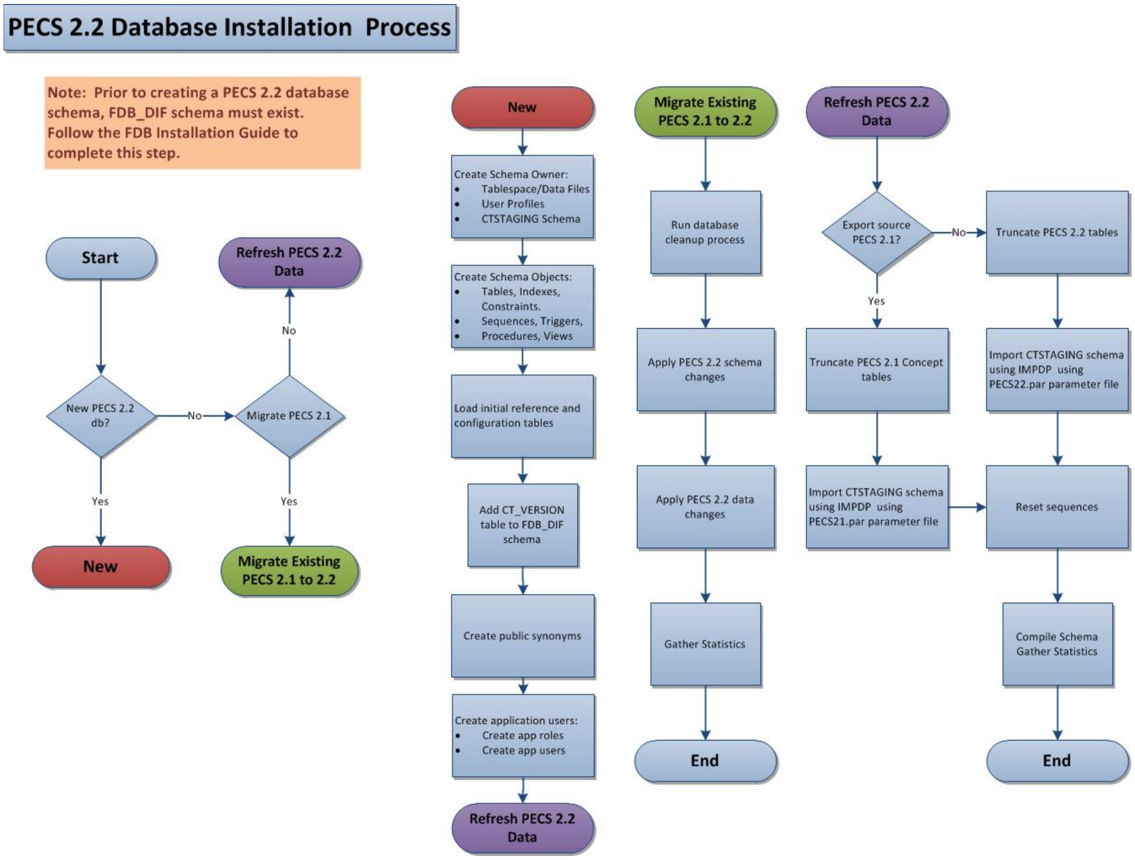
As stated in the assumptions of this document, the FDB (First Data Bank) DIF (Drug Information Framework) database must be installed prior to proceeding with the PECS database installation. The FDB-DIF install instructions for database setup and FDB Updater tool are provided by FDB-DIF. They can be referenced from PRE SharePoint site.

Note:	Please contact the PRE Configuration Manager, who should be identified on the project’s TSPR site to get the following from the PRE SharePoint: • The Install document(s) and Scripts to configure FDB-DIF • The Install document(s) for FDB-DIF Tool to load FDB-DIF database

The CTSTAGING schema must also be created in the database instance. The CTSTAGING table creation and population scripts are run from a separate process.

3.2 CTSTAGING Installation Instructions

This section describes the database scripts necessary for the installation of the PECS CTSTAGING database, and the order in which they should be executed. It is highly recommended that the PECS staging database user be configured as “CTSTAGING” and the FDB-DIF database user be configured as “FDB_DIF” as that is the usernames that are used throughout the remainder of the PECS installation documentation. **Executing steps 3.2.1 – 3.2.6 in this section will result in the creation of a PECS 2.2 database. Executing step 3.2.7 will migrate an existing PECS v2.1 database to PECS 2.2 compatibility.** If you are migrating from an existing PECS 2.1 schema with production data, skip to 3.2.7 to migrate to PECS 2.2 compatibility. The complete PECS Database Installation Process is graphically depicted below and in Appendix E – The PECS Database Installation Process.


PECS 2.2 Database Installation Process
Start
New PECS 2.2 db?
No
Yes
New
Migrate PECS 2.1
No
Yes
Refresh PECS 2.2 Data
Migrate Existing PECS 2.1 to 2.2
New
Create Schema Owner:
Tablespace/Data Files
User Profiles
CTSTAGING Schema
Create Schema Objects:
Tables, Indexes, Constraints.
Sequences, Triggers, Procedures, Views
Load initial reference and configuration tables
Add CT_VERSION table to FDB_DIF schema
Create public synonyms
Create application users:
Create app roles
Create app users
Refresh PECS 2.2 Data
Migrate Existing PECS 2.1 to 2.2
Run database cleanup process
Apply PECS 2.2 schema changes
Apply PECS 2.2 data changes
Gather Statistics
End
Refresh PECS 2.2 Data
Export source PECS 2.1?
No
Yes
Truncate PECS 2.1 Concept tables
Import CTSTAGING schema using IMPDP using PECS21.par parameter file
Truncate PECS 2.2 tables
Import CTSTAGING schema using IMPDP using PECS22.par parameter file
Reset sequences
Compile Schema Gather Statistics
End

Note: Prior to creating a PECS 2.2 database schema, FDB_DIF schema must exist. Follow the FDB Installation Guide to complete this step.

Note:	To migrate an existing PECS 2.1 database schema, skip to Section 3.2.7 – PECS 2.1 Database Migration.

Prior to executing the following sections, the Oracle 11g database needs to be installed and a Database Administrator login generated with sys_dba privileges is generated. The DBA login is necessary to run the first database script to create the tablespaces and user accounts for the remainder of the installation.

Note:	To get Install Scripts, please contact the PRE Configuration Manager, who should be identified on the project's TSPR site.

3.2.1 Create the Users

Prior to creation of the schemas, logical and physical environment structures must be setup for storage of the schemas database objects: tablespaces and data files. For the PECS Database configuration data and index storage are separated for each schema. For the CTSTAGING schema two tablespaces must be created:

- CTSTAGING_DATA
- CTSTAGING_INDEX
- LOB_DATA
- LOB_INDEX

In addition, user profiles are used to standardize resource limits for PECS schemas. There are 2 user profiles that have to be created:

- SERVICE_ACCOUNT
- USER_ACCOUNT

Before the user profiles can be created the script **utlpwdmg.sql** has to be executed. The script is located in the RDBMS\ADMIN directory within your installation home. Consult Oracle installation manual for the full directory path for the proposed environment.

To create the users in the database for the PECS application, the database administrator will need to execute the **pecs22_creation_pkg1.sql** script as **SYSTEM**. This script will execute other scripts that will create the tablespaces, user profiles and create the CT Staging User:

- `PECS22_Create_CTSTAGING_Tablespaces.sql`
- `pecs22_create_user_profiles_ddl.sql`
- `pecs22_create_user_modified.sql`

Prior to running the scripts, modifications should be made to tailor for the current installation environment. The following steps should be followed:

1. Open a text editor and open the PECS22_Create_CTSTAGING_Tablespaces.sql script. `Replace %DATAFILE_LOCATION% with the data file directory` the directory entered should already exist on the database server.

Note:	Note: If you are creating a development environment, use PECS22_Create_CTSTAGING_Tablespaces_Dev.sql instead .

2. Login to the SQL client using a database account that has sys_dba privileges
3. Execute the “`pecs22_creation_pkg1.sql`” script.
4. Open the “`pecs22_creation_pkg1.log`” file and search the log file for any errors.
5. This process creates the temporary file `fdb_dif2ctstaging.sql.` Open this file and scroll to the bottom and verify the following entry at the bottom of the file ‘`GRANT SELECT ON FDB_DIF.FDB_VERSION TO CTSTAGING`’. This will ensure that all necessary privileges were granted to the FDB tables that the CTSTAGING user needs to access.

3.2.2 Create Staging Tables and Database Objects

To create the CTSTAGING database for the PECS application, the administrator will need to execute the **pecs22_creation_pkg2.sql** script. This script will execute thirteen other scripts that create the CTSTAGING tables and populate those tables with some initial data values. The following steps should be followed:

1. Login to the SQL client using the CTSTAGING user account.
2. Execute the “pecs22_creation_pkg2.sql” script.
3. Open the pecs22_creation_pkg2.log file and search the log file for any errors.

3.2.3 Modification of the FDB_DIF Database

To modify the FDB-DIF data repository to work with the PECS application, the administrator will need to execute the **fdb_modification_pkg3.sql** script. This script will create a new table in the FDB-DIF data repository and modify one of the existing tables to change the constraints add an index.

1. Login to the sql client using the FDB_DIF user account.
2. Execute the “fdb_modification_pkg3.sql” script.
3. Open the fdb_modification_pkg3.log file and search the log file for any errors.

3.2.4 Create Public Synonyms

The PECS application access spans both FDB_DIF and CTSTAGING schema objects. Public synonyms are utilized to provide seamless application access across PECS application components. To create the public synonyms, the administrator will need to execute the **pecs22_create_public_synonyms.sql** script. This scripts executes two scripts: **pecs22_create_FDB_synonyms.sql**, **pecs22_create_CTSTAGING_synonyms.sql**. The following steps should be followed:

1. Login to the SQL client using the SYSTEM account.

2. Execute the “pecs22_create_public_synonyms.sql” script.
3. Open the pecs22_create_ public_synonyms.log file and search the log file for any errors.

3.2.5 PECS Application Users

The PECS database schemas have been devised to provide separation of ownership and CRUD data access levels thru the use of user/schemas and access roles assigned. Schemas/Roles that are required by the application are depicted in the cross-reference table listed below:

Table 1 - Database Users and Roles

User	Schema	Access Level	Assigned Role
FDB_DIF	FDB_DIF	Schema Owner	
FDB_DIF_APP_USER	FDB_DIF	Read Only user	FDB_DIF_READ_ONLY_ROLE
FDB_DIF_UPDATE_USER	FDB_DIF	CRUD user	FDB_DIF_UPDATE_USER_ROLE
CTSTAGING	CTSTAGING	Schema Owner	
CTSTAGING_READ_ONLY	CTSTAGING	Read Only user	CTSTAGING_READ_ONLY_ROLE
CTSTAGING_UPDATE_USER	CTSTAGING	CRUD user	CTSTAGING_UPDATE_USER_ROLE FDB_DIF_READ_ONLY_ROLE
PECSJMS	PECSJMS	Schema Owner	
PECSJMS_APP_USER	PECSJMS	CRUD user	PECSJMS_APP_USER_ROLE

Both FDB_DIF and CTSTAGING schema owners have been create prior to this step, however, additional users are required by the application. To create the PECS application user roles and users, the administrator will need to execute the **pecs22_create_application_roles_users.sql** script. This script will execute scripts that create the required PECS user roles and application users. Additionally, the script will create the PECSJMS schema objects that are required by the PECS application by executing **pecs_create_jms_process.sql** script.

Prior to running the driver script, **pecs22_create_application_roles_users.sql,** modifications should be made to **CreateTablespacePECSJMS.sql** to tailor for the current installation environment. .

The following steps should be followed:

1. Open a text editor and open the CreateTablespacePECSJMS.sql script. Replace %DATAFILE_LOCATION% with the data file directory for the current installation environment. The directory entered should already exist on the database server.
2. Login to the SQL client using the SYSTEM account.
3. Execute the “pecs22_create_application_roles_users.sql” script.
4. Open the “`pecs22_create_application_roles_users.log`” file and search the log file for any errors.

VA Standard Data Services (SDS) has created and maintains standardized tables in an Oracle database (e.g., VA Institutions). These tables must be accessible to PECS as a Web-based application. Please refer to the SDS Database Installation Guide for the information necessary to install the SDS Data Service database tables, indexes and data.

KAAJEE makes internal API calls to the SDS Database/Tables located on an Oracle database. PECS is KAAJEE-enabled. The KAAJEE user ID, schema, and SSPI tables must be accessible to PECS as a Web-based application. Please refer to the KAAJEE Database Installation Guide for the information necessary to install the KAAJEE database tables, indexes and data.

A complete listing of the PECS Schema Creation SQL Scripts invoked from the driver scripts are listed below.

Table 2. List of PECS Schema Creation SQL Scripts (TBD)

Script Description	File Name
A Master script to create the tablespace and user package script.	pecs_creation_pkg1.sql
Master script to create the ctstaging tables and database objects package script.	pecs_creation_pkg2.sql
Master script to modify the FDB schema package script.	fdb_modification_pkg3.sql

3.2.6 Import PECS Production Data

Production data from a PECS 2.1 database or development data from a PECS 2.2 database can be loaded into a newly created PECS 2.2 database using the following procedures. Once loaded, the PECS 2.2 database is fully functional and no further installation steps are necessary. Section 3.2.7 of this document is not required for this installation path.

To load Production Data from a PECS 2.1 database:

1. Download dump file from PRE share and place in the DPDUMP directory within your environment: \\vhaisppres3\PRE\PECS\Database\OracleExport
2. Modified the parameter file, PECS21.par, replace <dump_filename.dmp> and <dump_filename.log> with the dump file downloaded in the prior step.
3. Disable constraints prior to the load by running the following script file as SYSTEM, Build_Script_to_Disable_Constraints.sql.
4. Truncate the concept tables from the CTSTAGING schema by running the following script file as SYSTEM, Truncate_PECS21_Concept_Tables.sql.
5. From the command prompt, issue the following command to load PECS 2.1 data from the import file: impdp parfile=PECS21.par. Enter SYSTEM username/password when prompted.
6. After the load has been completed, enable constraints by running the following script file as SYSTEM, Build_Script_to_Enable_Constraints.sql. Check the log Build_Script_to_Enable_Constraints.log for errors, if any constraints failed then re-run this step after the objects have been recompiled in Step 8.

7. Reset PECS sequences by running the command file, PECS22_Rebuild_Sequences_After_Load.sql. Run as CTSTAGING user.
8. Create new PECS 2.2 reference table entries for Dose_Route and Field_Metadata tables by running the command file, Add_Reference_Tables_Entries_After_Load.sql. Run as CTSTAGING user.
9. Run script to compile all schema objects and gather fresh statistics as SYSTEM, Recompile_Schema.sql.

Note:	To get the Data Import Guide and access to an export file, please contact the PRE Configuration Manager, who should be identified on the project's TSPR site.

3.2.7 PECS v2.1 Database Migration

Prior to migrating PECS v2.1 database schema to PECS v2.2 compatibility, a backup of the database should be performed either using RMAN or Oracle 11g DataPump export utility. Securing a backup of the database is integral to the database rollback procedures in the event that the upgrade/migration needs to revert back to the prior version. Oracle DataPump utilities provide more granularity to backup specific schemas. PECS v2.1 consists of two database schemas: CTSTAGING, FDB_DIF. To backup the PECS v2.1 database using Oracle DataPump utility, issue the following command logged in as a USER with DBA privileges:

- **expdp DUMPFILE**=<dumpfilename.dmp> **SCHEMAS**=CTSTAGING,FDB_DIF **CONTENT**=ALL **LOGFILE**=<logfilename.log>

When prompted, enter the SYSTEM userid and password to complete the export and note the dump and log files for future use.

Prior to performing the steps needed to migrate a PECS v2.1 database to PECS v2.1 compatibility, the Oracle listener for the PECS database instance should be brought down to ensure consistency and limit access during the conversion efforts. As an Oracle Administrator, the following command can be issued from the LINUX command prompt to stop the listener for the current instance: **lsnrctl stop**.

To migrate PECS v2.1 database schema to PECS v2.2 compatibility, the database administrator will need to execute the following database scripts as the USER specified below. Each of these scripts acts as a driver script to initiate and log migration activities. At the completion of each of the steps. check the log file for any errors or anomalies in processing the required transactions.

Table 3: List of PECS 2.2 Driver SQL Scripts

List of PECS 2.2 Driver SQL Scripts			
Script Description	File Name	User	Log File
PECS Preparation Migration Script	PECS22_pre_migration_setup.sql	SYSTEM	PECS22_pre_migration_setup.log

List of PECS 2.2 Driver SQL Scripts			
Script Description	File Name	User	Log File
PECS Migration Driver script	PECS22_migration.sql	CTSTAGING	PECS22_migration.log
PECS Migration Cleanup Script	PECS22_post_migration_cleanup.sql	SYSTEM	PECS22_post_migration_cleanup.log

Step by Step procedure to accomplish the migration is as follows:

1. Prior to run the PECS22_pre_migration_setup.sql, the file Create_LOB_Tablespaces.sql must be edited to point to the location within your environment to create the LOB tablespaces. Once modified, proceed to the next step.
2. Login to the sql client using the SYSTEM user account.
3. Execute the "PECS22_pre_migration_setup.sql" script.
4. Open the "PECS22_pre_migration_setup.log" file and search the log file for any errors.

If all commands completed successfully, then proceed:

1. Login to the sql client using the CTSTAGING user account.
2. Execute the "PECS22_migration.sql" script.
3. Open the "PECS22_migration.log" file and search the log file for any errors.

Since migration scripts can be run multiple times to migrate data from 2.1 base tables to 2.2 schema, Expected errors include Drop commands on entities that aren't in existence.

1. Login to the sql client using the SYSTEM user account.
2. Execute the "PECS22_post_migration_cleanup.sql" script.
3. Open the "PECS22_post_migration_cleanup.log" file and search the log file for any errors.

After all the migration steps have been completed without error, the Oracle listener for the PECS database instance should be restarted. As an Oracle Administrator, the following command can be issued from the LINUX command prompt to start the listener for the current instance: **lsnrctl start**.

A complete listing of the scripts invoked from the driver scripts are listed below.

Table 4: List of PECS 2.2 SQL Scripts

PECS 2.2 Driver Scripts	Description	Purpose
PECS22_pre_migration_setup.sql	Driver Script to run pre-migration setup activities	Database Migration Driver Scripts

PECS22_migration.sql	Driver Script to migrate from PECS 2.1 to PECS 2.2 schema	Database Migration Driver Scripts
PECS22_post_migration_cleanup.sql	Driver Script to run post migration activities	Database Migration Driver Scripts
DB Cleanup Scripts	**Description**	**Purpose**
Build_Script_to_Move_Indices_Tablespace.sql	Script to dynamically build and run the commands to move all CTSTAGING indexes to the correct tablespace	Database cleanup
Build_Script_to_Rebuild_Indices.sql	Script to dynamically build and run the commands to rebuild all CTSTAGING indexes	Database cleanup
Build_Script_to_Enable_Constraints.sql	Script to dynamically build and run the commands to enable all CTSTAGING constraints	Database cleanup
Build_Script_to_Rename_Unnamed_Constraints.sql	Script to dynamically build and run the commands to rename unnamed CTSTAGING constraints	Database cleanup
Build_Script_to_Rename_Unnamed_Indices.sql	Script to dynamically build and run the commands to rename unnamed CTSTAGING indexes	Database cleanup
Create_CTSTAGING_Synonyms.sql	Script to create CTSTAGING Synonyms for all schema objects	Database cleanup
Drop_Unused_DB_Procedures.sql	Script to drop unused CTSTAGING procedures	Database cleanup
Drop_Unused_DB_Sequences.sql	script to drop unused CTSTAGING sequences	Database cleanup
Drop_Unused_DB_Synonyms.sql	script to drop unused CTSTAGING synonyms	Database cleanup
Drop_Unused_DB_Tables.sql	script to drop unused CTSTAGING tables	Database cleanup
Recompile_Schema.sql	script to recompile schema objects and gather fresh statistics for schema objects	Database cleanup

PECS2_db_cleanup_process.sql	Driver script for database cleanup efforts	Database cleanup
New PECS 2.2 Components	**Definition**	**Purpose**
Create_LOB_Tablespaces.sql	Create tablespace to store Large Binary Objects (LOBs)	New in PECS 2.2
CCR5159_ModifyDoseRoute_DATA.sql	Add two dose routes per CCR5159	New in PECS 2.2
CCR5443_AddCTSTAGINGIndexes_ConceptType_ID.sql	Add Dose Range Indexes on Concept Type and ID on CTSTAGING.	New in PECS 2.2
CCR5443_AddFDBIndex_ConceptType_ID.sql	Add Dose Range Indexes on Concept Type and ID on FDB_DIF schema	New in PECS 2.2
DIFWK33_Tables_Create_GenericRoutedDrug.sql	Create new table to store Generic Routed Drug descriptions after deletion from FDB	New in PECS 2.2
FDB_Comparison_Report_DATA.sql	Populate new report concepts for FDB comparison reports	New in PECS 2.2
FDB_Comparison_Report_DDL.sql	Create new table to store FDB Comparison reports	New in PECS 2.2
FDB_Comparison_Report_Synonyms_DDL.sql	Create synonyms for new FDB Comparison report objects	New in PECS 2.2
FieldMetadata_DATA.sql	Add additional field metadata for PECS 2.2 fields	New in PECS 2.2
Forward_Reverse_Mono_Migrate.sql	Update forward/reverse monographs to migrate updates for subsequent DATUP processing	New in PECS 2.2
ForwardReverseMonographs_DATA.sql	Populate forward/reverse monograph table with preset values	New in PECS 2.2
ForwardReverseMonographs_DDL.sql	Create synonyms for new Forward/Reverse monograph objects	New in PECS 2.2
ForwardReverseMonographs_Synonyms_DDL.sql	Create new table forward/reverse monographs	New in PECS 2.2
Record_Locking.sql	Create new table for PECS record locking	New in PECS 2.2
Data Load Processing	**Description**	**Purpose**
PECS22_Rebuild_Sequences_After_Load.sql	Rebuild sequences after data load	Data load
pecs22.par	PECS 2.2 data migration parameter file to be used to import data from an existing PECS 2.2 database export	Data load

pecs21.par	PECS 2.2 data migration parameter file to be used to import data from an existing PECS 2.1 database export	Data load

3.2.8 PECS v2.2 Database Migration Rollback

Prior to migrating PECS v2.1 database schema to PECS v2.2 compatibility, a backup of the database was performed to ensure rollback capability. This section addresses the steps needed to rollback to PECS v2.1 using the secured backup.

To restore the PECS v2.1 schema from the backup taken prior to the migration, follow the procedures outlined in the Data Import Guide for platform specific instructions (Unix, Windows).

Procedures for restoring/loading production data include the following steps regardless of platform:

- Prepare database for restoring production data
  - Drop existing schema objects (tables, sequences) for each schema
- Import each schema by issuing the following commands logged in as a USER with DBA privileges preferably SYSTEM:

  -
    ```
    impdp DUMPFILE=<dumpfilename.dmp> SCHEMAS=FDB_DIF
    LOGFILE=<logfilename.log> CONTENT=ALL
    TABLE_EXISTS_ACTION=REPLACE
    ```

  -
    ```
    impdp DUMPFILE=<dumpfilename.dmp> SCHEMAS=CTSTAGING
    LOGFILE=<logfilename.log> CONTENT=ALL
    TABLE_EXISTS_ACTION=REPLACE
    ```

  When prompted, enter the SYSTEM userid and password to complete the import. Review log files for each import to verify the successful completion of the rollback.

Note:	To get the Data Import Guide, please contact the PRE Configuration Manager, who should be identified on the project's TSPR site.

4. Users

PECS uses the KAAJEE framework for user authorization. KAAJEE authenticates users against the Local VistA. Access to PECS is limited to the users in the PECS VistA that are configured to have the PECS security keys. When a new users need to be added, contact an experienced Local VistA administrator. Provide the administrator with a list of users that will be needed along with their required security keys. PECS security keys are discussed in Appendix C.

5. WebLogic Application Server Configuration

The WebLogic server configuration assumes that there is an existing WebLogic server installed and domain configured for use by the PECS application. Configuration steps to set up datasources will depend on the version of the WebLogic server. Furthermore, it is assumed that the installation of the WebLogic server and domain follows existing standards for a production environment installation. The configuration steps detailed below include the configuration of two datasources and the deployment of the PECS EAR archive.

5.1 Dependency Installation

VistALink Version 1.6.0.028 and KAAJEE Version 1.1.0.007 software packages must be installed prior to deployment of PECS on the WebLogic server. Follow the respective installation guides supplied by the VA for this software prior to continuing with this installation.

Please read Appendix C and ensure the administrative KAAJEE user is installed prior to installing the PECS EAR file.

Note:	Please note that prior to the PECS EAR file deployment, the KAAJEE station ID configuration information must be updated to refer to the target VistA server. This information is updated in the <station-number> section of the WEB-INF\kaajeeConfig.xml file that is in the EAR deployment archive. Example steps to perform this process are outlined below (*NIX based): Explode the CT_EAR.ear file, explode CT_WEB.war inside the exploded CT_EAR.file, then edit CT_EAR.ear/CT_WEB.war/WEB-INF/ kaajeeConfig.xml to set the institution IDs.

The steps described above would literally translate to the following Linux commands:

```
Edit the file:

cp CT_EAR.ear /tmp
cd /tmp
mkdir CT_EAR
cd CT_EAR
jar -xvf ../CT_EAR.ear
jar -xvf CT_WEB.war WEB-INF/kaajeeConfig.xml
vi WEB-INF/kaajeeConfig.xml

Save and restore the modified EAR file:

jar -uvf CT_WEB.war WEB-INF/kaajeeConfig.xml
rm -rf WEB-INF/
mv ../CT_EAR.ear ../CT_EAR.ear.orig
jar -cvf ../CT_EAR.ear *
cd ..

rm -rf CT_EAR
```

5.2 Configure WebLogic Datasources

There are three datasources that need to be configured on the WebLogic administration server for the PECS application. Configuration values for the URL, Username, and Password will be dependent on where the FDB and STAGING databases have been installed. The configuration for each datasource is summarized below:

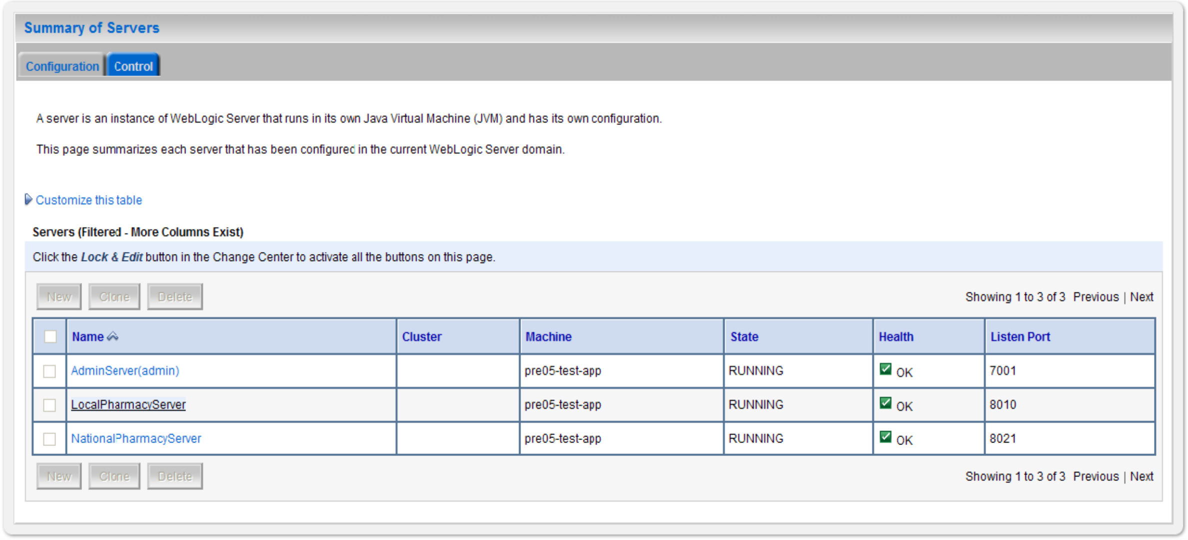
Note:	Contact the DBA for the HOST_SERVER, DATABASE_SID and passwords used below. These items are bolded surrounded by percent signs below. When entering the information, do not enter the percent signs.

```
Name: CTFdbDataSource
JNDI Name: jdbc/CTFdbDataSource
URL: jdbc:oracle:thin:@%HOST_SERVER%:1521:%DATABASE_SID%
Driver: oracle.jdbc.xa.client.OracleXADataSource
Username: FDB_DIF_APP_USER
Password: %FDB_DIF_APP_USER_PASSWORD%

Name: CTStagingDataSource
JNDI Name: jdbc/CTStagingDataSource
URL: jdbc:oracle:thin:@%HOST_SERVER%:1521:%DATABASE_SID%
Driver: oracle.jdbc.xa.client.OracleXADataSource
Username: CTSTAGING_UPDATE_USER
Password: %CTSTAGING_UPDATE_USER_PASSWORD%

Name: PECSJMS Data Source
JNDI Name: jdbc/PecsJmsDataSource
URL: jdbc:oracle:thin:@%HOST_SERVER%:1521:%DATABASE_SID%
Driver: oracle.jdbc.OracleDriver
Username: PECSJMS_APP_USER
Password: %PECSJMS_APP_USER_PASSWORD
```

5.3 WebLogic Server Startup Configuration

PECS requires additional arguments added to the WebLogic Server's Server Start properties. This section details the steps to add the arguments to the server .

1. Open and log into the WebLogic console, using an administrative user name and password. The WebLogic console is located at: `http://<Deployment Machine>:7001/console`.
2. Click on the server name corresponding to the deployment server in the **Summary of Servers** panel found in the right column of the WebLogic console. **For reference only**, see Figure 5-1.



A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.

This page summarizes each server that has been configured in the current WebLogic Server domain.

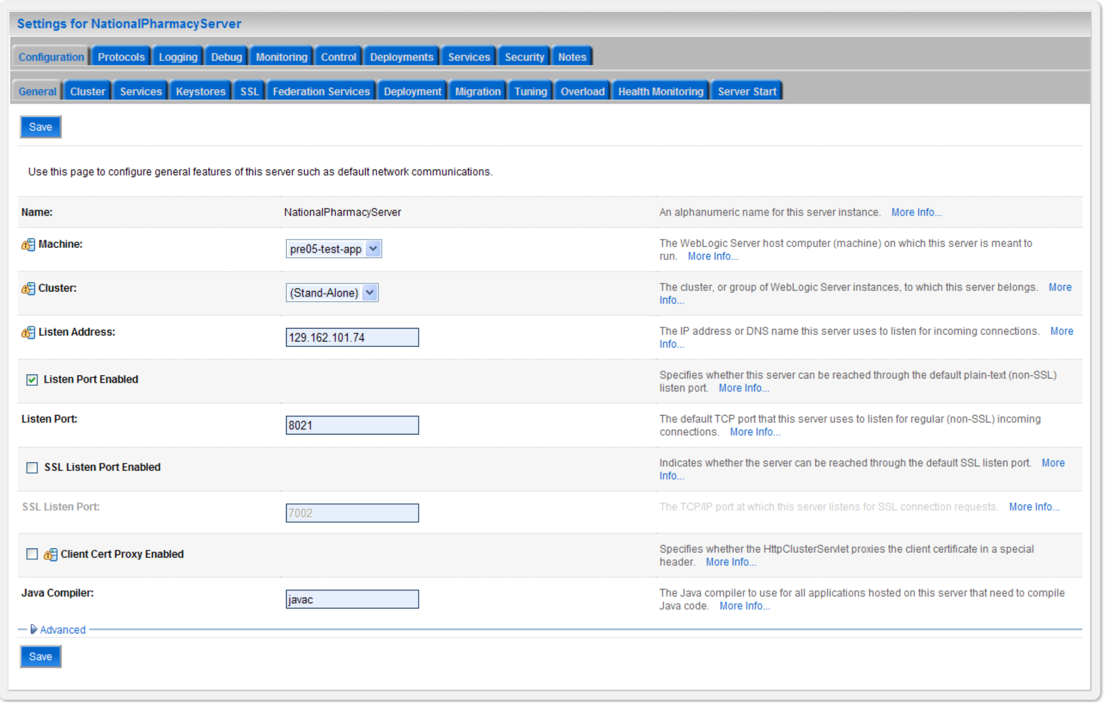
Servers (Filtered - More Columns Exist)

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

□	Name	Cluster	Machine	State	Health	Listen Port
□	AdminServer(admin)		pre05-test-app	RUNNING	☑ OK	7001
□	LocalPharmacyServer		pre05-test-app	RUNNING	☑ OK	8010
□	NationalPharmacyServer		pre05-test-app	RUNNING	☑ OK	8021

Figure 5-1. Summary of Servers

3. WebLogic will now display the panel **Settings for Deployment Server** in the right column of the console, where configuration of the Deployment Server are set. For reference, see Figure 5-2.



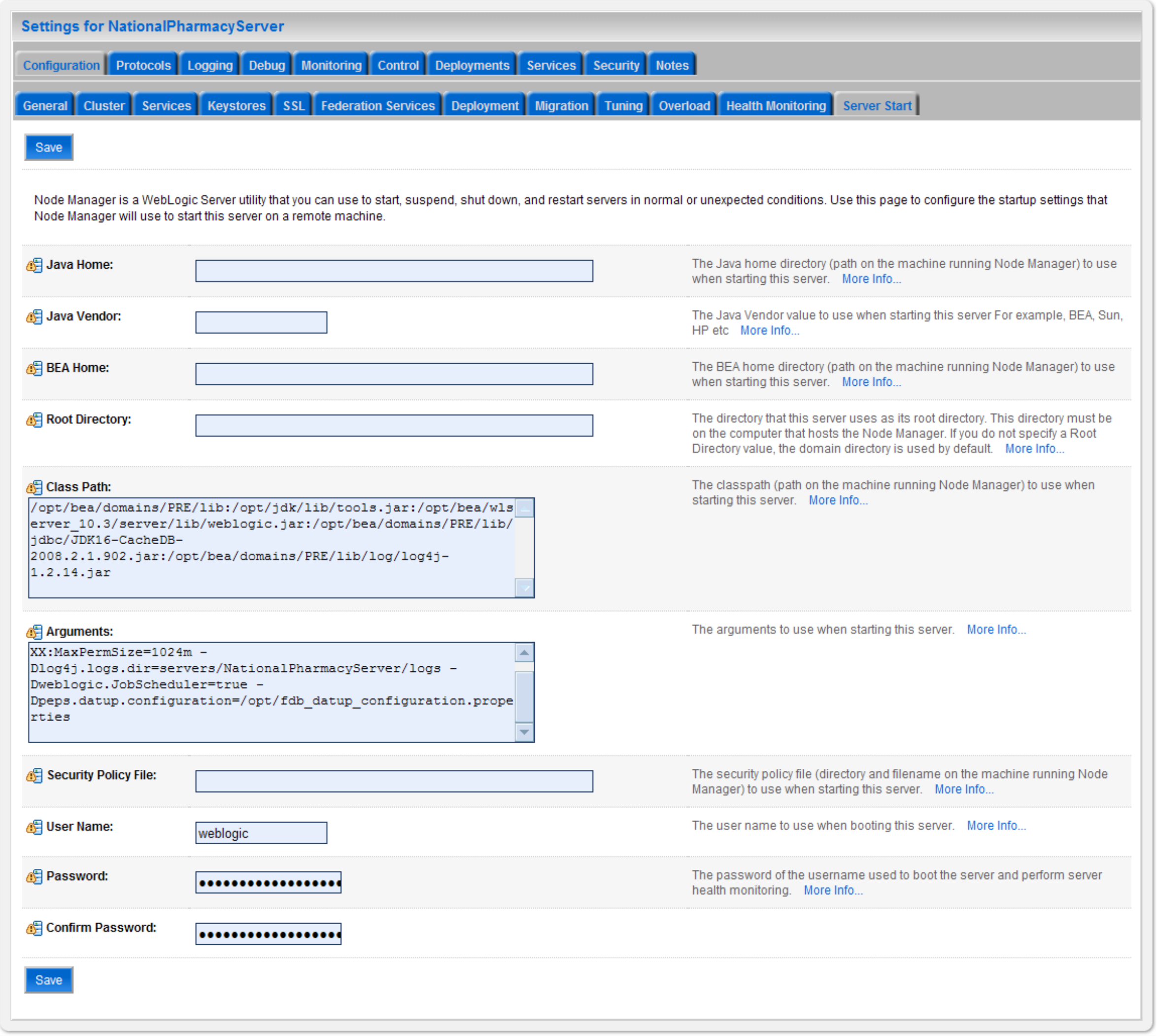
Settings for NationalPharmacyServer

Use this page to configure general features of this server such as default network communications.

Name:	NationalPharmacyServer	An alphanumeric name for this server instance. More Info...
Machine:	pre05-test-app	The WebLogic Server host computer (machine) on which this server is meant to run. More Info...
Cluster:	(Stand-Alone)	The cluster, or group of WebLogic Server instances, to which this server belongs. More Info...
Listen Address:	129.162.101.74	The IP address or DNS name this server uses to listen for incoming connections. More Info...
☑ Listen Port Enabled		Specifies whether this server can be reached through the default plain-text (non-SSL) listen port. More Info...
Listen Port:	8021	The default TCP port that this server uses to listen for regular (non-SSL) incoming connections. More Info...
☐ SSL Listen Port Enabled		Indicates whether the server can be reached through the default SSL listen port. More Info...
SSL Listen Port:	7002	The TCP/IP port at which this server listens for SSL connection requests. More Info...
☐ Client Cert Proxy Enabled		Specifies whether the HttpClusterServlet proxies the client certificate in a special header. More Info...
Java Compiler:	javac	The Java compiler to use for all applications hosted on this server that need to compile Java code. More Info...

Figure 5-2. Settings for Deployment Server

4. Click on the **Server Start** tab.
5. WebLogic will now display the panel Server Start tab in the **Settings for Deployment Server** in the right column of the console, where configuration of the Deployment Server is set. For reference, see Figure 5-3.



Settings for NationalPharmacyServer

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to configure the startup settings that Node Manager will use to start this server on a remote machine.

Java Home: — The Java home directory (path on the machine running Node Manager) to use when starting this server. More Info...

Java Vendor: — The Java Vendor value to use when starting this server For example, BEA, Sun, HP etc More Info...

BEA Home: — The BEA home directory (path on the machine running Node Manager) to use when starting this server. More Info...

Root Directory: — The directory that this server uses as its root directory. This directory must be on the computer that hosts the Node Manager. If you do not specify a Root Directory value, the domain directory is used by default. More Info...

Class Path: — The classpath (path on the machine running Node Manager) to use when starting this server. More Info...

```
/opt/bea/domains/PRE/lib:/opt/jdk/lib/tools.jar:/opt/bea/wls
erver_10.3/server/lib/weblogic.jar:/opt/bea/domains/PRE/lib/
jdbc/JDK16-CacheDB-
2008.2.1.902.jar:/opt/bea/domains/PRE/lib/log/log4j-
1.2.14.jar
```

Arguments: — The arguments to use when starting this server. More Info...

```
XX:MaxPermSize=1024m -
Dlog4j.logs.dir=servers/NationalPharmacyServer/logs -
Dweblogic.JobScheduler=true -
Dpeps.datup.configuration=/opt/fdb_datup_configuration.prope
rties
```

Security Policy File: — The security policy file (directory and filename on the machine running Node Manager) to use when starting this server. More Info...

User Name: weblogic — The user name to use when booting this server. More Info...

Password: •••••••••••••••••• — The password of the username used to boot the server and perform server health monitoring. More Info...

Confirm Password: ••••••••••••••••••

Figure 5-3. Server Start Tab

6. Insert the following text in the **Arguments** box:

7. **-d64 -server -Xms768m -Xmx4096m -XX:PermSize=256m -XX:MaxPermSize=512m -Djava.awt.headless=true –**

   Also add a argument for Log4j file. (see for reference below, modify path per your server configuration) :-

   Dlog4j.configuration=file:/u01/app/user_projects/domains/sqa_PECS/log4j.xml

8. Click **Save**.

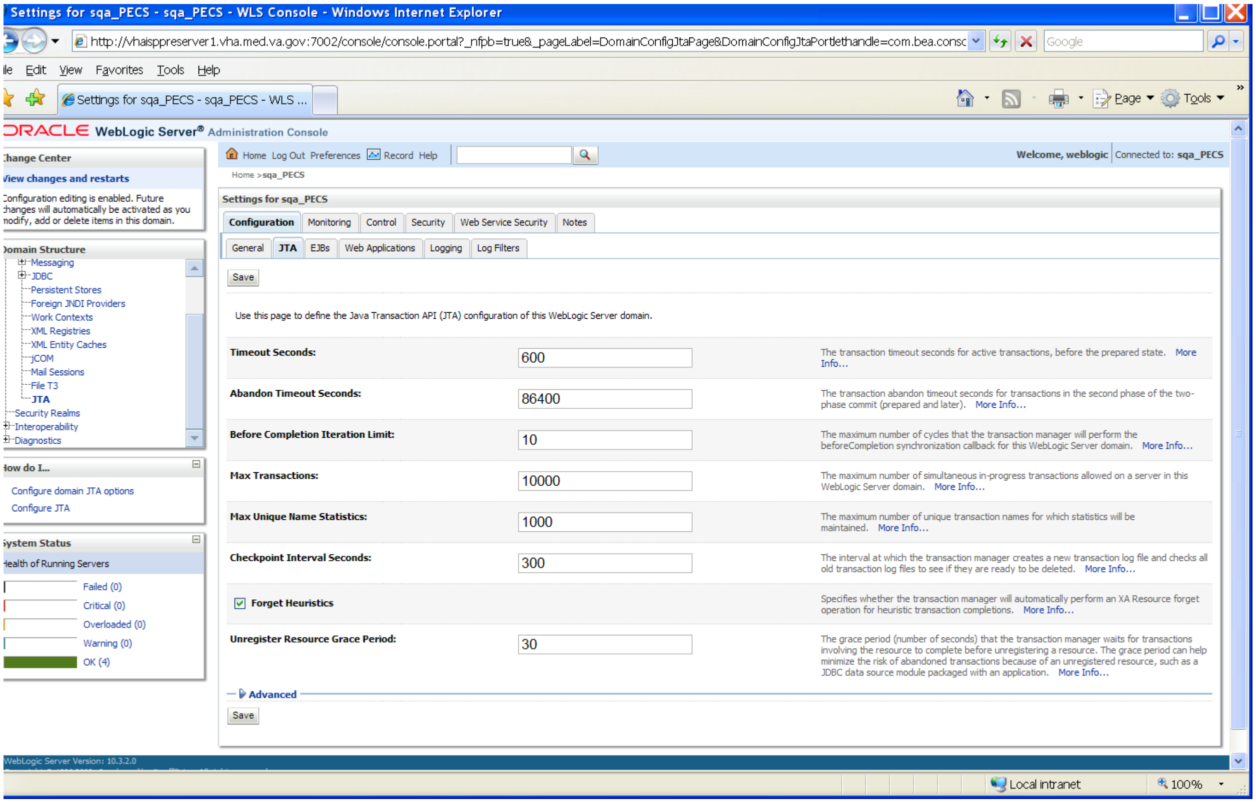
5.4 Configure WebLogic JTA

The application requires the Setting the JTA Transaction Timeout for processing of reports.

Step 1: Configure JTA

1. In the WebLogic Administration Console, expand **Services**.
2. Click on **JTA**.
3. On the Configuration tab, for “Timeout Seconds”, change the value to **600**.(see below Console screen)
4. Click the **Save** button.

The WebLogic Administration Console screen should look similar to the following:



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Use this page to define the Java Transaction API (JTA) configuration of this WebLogic Server domain.

Timeout Seconds:	600	The transaction timeout seconds for active transactions, before the prepared state. More Info...
Abandon Timeout Seconds:	86400	The transaction abandon timeout seconds for transactions in the second phase of the two-phase commit (prepared and later). More Info...
Before Completion Iteration Limit:	10	The maximum number of cycles that the transaction manager will perform the beforeCompletion synchronization callback for this WebLogic Server domain. More Info...
Max Transactions:	10000	The maximum number of simultaneous in-progress transactions allowed on a server in this WebLogic Server domain. More Info...
Max Unique Name Statistics:	1000	The maximum number of unique transaction names for which statistics will be maintained. More Info...
Checkpoint Interval Seconds:	300	The interval at which the transaction manager creates a new transaction log file and checks all old transaction log files to see if they are ready to be deleted. More Info...
☑ **Forget Heuristics**		Specifies whether the transaction manager will automatically perform an XA Resource forget operation for heuristic transaction completions. More Info...
Unregister Resource Grace Period:	30	The grace period (number of seconds) that the transaction manager waits for transactions involving the resource to complete before unregistering a resource. The grace period can help minimize the risk of abandoned transactions because of an unregistered resource, such as a JDBC data source module packaged with an application. More Info...

WebLogic Server Version: 10.3.2.0

WebLogic JTA Configuration is complete.

5.5 Configure exportfile.properties

One functional piece of PECS allows a Release Manager to export data from the Oracle database so that it can be imported at various sites to support the Order Check process. The export file can be downloaded to the user's desktop, but a copy needs to be sent to an File Transfer Protocol (FTP) server so that it can be utilized in other server processes. To know where to place the file, a property file named exportfile.properties needs to be created. This file should reside in the DOMAIN_HOME/user_staged_config directory. The properties that need to be entered in to this file are(see below for reference. Also please configure with or without leading "/" depending on relative or absolute path):

```
export.file.server=10.3.29.201

export.file.dir=/pharmacy/fdb_dif

export.user.name=PECS

fdb.file.dir=/pharmacy

scheduled.time=2230
```

The exportfile.properties file will need to be readable by the user who runs the WebLogic application server.

5.6 Application Deployment

5.6.1 PECS Application Deployment

Specific deployment steps will vary depending on the version of the WebLogic server the PECS application will be deployed on. The PECS application is a J2EE application packaged in a standard EAR file format. The application should be deployed following the recommended process for deploying EAR files for the WebLogic server version platform. Use default values to deploy the Ear file and associate it with domain/server as per WebLogic install for PECS.

Note:	Please note that you must associate the application with the target server, and activate the application after deployment, before it can service any requests.

5.6.2 PECS Help Application Deployment

Specific deployment steps will vary depending on the version of the WebLogic server the PECS Help application will be deployed on. The PECS Help application is a RoboHelp application packaged in a standard EAR file format. The Help application should be deployed following the recommended process for deploying EAR files for the WebLogic server version platform. Use default values to deploy the Ear file and associate it with domain/server as per WebLogic install for PECS. Some recommended pointers for install of pecs-hlp.xxx.ear file:

a. Install the deployment as an application. (PECS Help application is accessible at the context root "pecsHelp")

b. On deployment targets page, select the PECS managed server.

c. On Optional Settings page, name the deployment – "pecs-Help"

Note:	Please note that you must associate the Help application with the target server, and activate the application after deployment, before it can service any requests.

5.7 Configure log4j.properties

The PECS application uses log4j loggers to create and write log information to application event logs. The logging properties for the PECS application are included in Appendix A. Logger and appender configuration is included for the PECS application, and optionally the Hibernate API. Update logging properties as appropriate to the host server:

- Set logging level to "info" for production mode
- Set "File" properties to the identified log directory on the server.
- Set "ConversionPattern" to the standard VA pattern.

The properties in Appendix A should be inserted into the existing log4j properties file that exists at the beginning of the WebLogic server classpath. (please use log4j.xml for reference from Apendix A).

6. Post-Installation Notes

The entrance URL for the application is: *http://%SERVER%:%PORT%/ct/public/Welcome.html*.

This is a generic URL for PECS. You need to replace the %SERVER% and %PORT% with the server name and port number assigned to your deployment.

For example, the entrance URL for the AITC SQA server is as follows:
http://vaww.oed.portal.va.gov/projects/pre/PRE_IPT_Rev/PRE_IPT_Rev_PECS2-1/default.aspx

(This page included for two-sided copying.)

Appendix A: log4j Properties

```
 <?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

      <appender name="STDOUT" class="org.apache.log4j.ConsoleAppender">
            <layout class="org.apache.log4j.PatternLayout">
                  <param name="ConversionPattern" value="%d %-5p [%t] %C{2}
(%F:%L) - %m%n"/>
            </layout>
      </appender>

      <appender name="FileAppender"
class="org.apache.log4j.RollingFileAppender">
        <param name="File" value="PECSLogs/server.log"/>
        <param name="Append" value="false"/>
        <param name="MaxBackupIndex" value="10"/>
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{dd MMM yyyy hh:mm:ss
a} %-5p [%c:%M] %m%n"/>
        </layout>
    </appender>

    <appender name="HibernateAppender"
class="org.apache.log4j.RollingFileAppender">
        <param name="File" value="PECSLogs/hibernate.log"/>
        <param name="Append" value="false"/>
        <param name="MaxBackupIndex" value="10"/>
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{dd MMM yyyy hh:mm:ss
a} %-5p [%c:%M] %m%n"/>
        </layout>
    </appender>

   <appender  name="PepsAppender"  class="org.apache.log4j.RollingFileAppender">
      <param name="File" value="PECSLogs/peps.log"/>
      <param name="Append" value="false"/>
      <param name="MaxBackupIndex" value="10"/>
      <layout class="org.apache.log4j.PatternLayout">
         <param name="ConversionPattern" value="%d{dd MMM yyyy hh:mm:ss a} %-
5p [%c:%M] %m%n"/>
      </layout>
   </appender>



    <appender name="SpringAppender"
class="org.apache.log4j.RollingFileAppender">
        <param name="File" value="PECSLogs/spring.log"/>
        <param name="Append" value="false"/>
        <param name="MaxBackupIndex" value="10"/>
        <layout class="org.apache.log4j.PatternLayout">
```

```
            <param name="ConversionPattern" value="%d{dd MMM yyyy hh:mm:ss
a} %-5p [%c:%M] %m%n"/>
        </layout>
    </appender>

    <appender name="StrutsAppender"
class="org.apache.log4j.RollingFileAppender">
        <param name="File" value="PECSLogs/struts.log"/>
        <param name="Append" value="false"/>
        <param name="MaxBackupIndex" value="10"/>
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{dd MMM yyyy hh:mm:ss
a} %-5p [%c:%M] %m%n"/>
        </layout>
    </appender>

      <appender name="CT" class="org.apache.log4j.RollingFileAppender">
            <param name="file" value="PECSLogs/ct_prod.log"/>
                  <param name="MaxFileSize" value="10000KB"/>
                  <param name="MaxBackupIndex" value="10"/>
            <layout class="org.apache.log4j.PatternLayout">
              <param name="ConversionPattern" value="%d %5p %l - %m%n"/>
            </layout>
      </appender>

      <appender name="PECS" class="org.apache.log4j.RollingFileAppender">
            <param name="file" value="PECSLogs/pecs_prod.log"/>
                  <param name="MaxFileSize" value="10000KB"/>
                  <param name="MaxBackupIndex" value="10"/>
            <layout class="org.apache.log4j.PatternLayout">
              <param name="ConversionPattern" value="%d %5p %l - %m%n"/>
            </layout>
      </appender>

      <logger name="gov.va.med.pharmacy.ct" additivity="false">
        <level value="debug"/>
        <appender-ref ref="CT" />
      </logger>

      <logger name="gov.va.med.pharmacy.pecs" additivity="false">
       <level value="debug"/>
       <appender-ref ref="PECS" />
      </logger>

      <logger name="gov.va.med.pharmacy.ct.web" additivity="false">
       <level value="debug"/>
       <appender-ref ref="CT" />
      </logger>


  <!-- INFO-level logger: turn on to record timing audit information -->

      <logger name="gov.va.med.monitor.time.AuditTimer" additivity="false" >
                  <level value="info" />
                  <appender-ref ref="FileAppender"/>
      </logger>
      <logger
name="org.apache.beehive.netui.pageflow.internal.AdapterManager"
additivity="false" >
                  <level value="warn" />
                  <appender-ref ref="FileAppender"/>
      </logger>

    <logger name="org.apache.log4j">
        <level value="info" />
    </logger>

    <logger name="org.hibernate" additivity="false">
        <level value="info" />
        <appender-ref ref="HibernateAppender"/>
    </logger>

    <logger name="org.hibernate.type" additivity="false">
        <level value="warn" />
        <appender-ref ref="HibernateAppender"/>
    </logger>

    <logger name="org.hibernate.loader" additivity="false">
        <level value="warn" />
        <appender-ref ref="HibernateAppender"/>
    </logger>

    <logger name="org.hibernate.impl" additivity="false">
        <level value="warn" />
        <appender-ref ref="HibernateAppender"/>
    </logger>

    <logger name="org.springframework" additivity="false">
        <level value="error" />
        <appender-ref ref="SpringAppender"/>
    </logger>

    <logger name="org.apache.struts2" additivity="false">
        <level value="error" />
        <appender-ref ref="StrutsAppender" />
    </logger>

    <logger name="com.opensymphony.xwork2" additivity="false">
        <level value="error" />
        <appender-ref ref="StrutsAppender" />
    </logger>

    <logger name="org.apache.commons.digester" additivity="false">
        <level value="error" />
        <appender-ref ref="StrutsAppender" />
    </logger>

    <logger name="freemarker.cache" additivity="false">
        <level value="error" />
        <appender-ref ref="StrutsAppender" />
    </logger>

    <logger name="org.apache.tiles" additivity="false">
```

```
        <level value="error" />
        <appender-ref ref="StrutsAppender" />
    </logger>

    <logger name="net.sf.navigator" additivity="false">
        <level value="error" />
        <appender-ref ref="StrutsAppender" />
    </logger>

    <logger name="org.displaytag" additivity="false">
        <level value="error" />
        <appender-ref ref="StrutsAppender" />
    </logger>

    <logger name="org.apache.commons">
        <level value="warn" />
    </logger>

   <logger name="gov.va.med.pharmacy.peps" additivity="false">
       <level value="error" />
       <appender-ref ref="PepsAppender"/>
   </logger>


    <root>
        <priority value="info" />
        <appender-ref ref="FileAppender"/>
        <appender-ref ref="HibernateAppender"/>
        <appender-ref ref="SpringAppender"/>
        <appender-ref ref="StrutsAppender"/>
        <appender-ref ref="STDOUT"/>
        <appender-ref ref="CT"/>
        <appender-ref ref="PECS"/>
    </root>

</log4j:configuration>
```

(This page included for two-sided copying.)

Appendix B: Custom Update File Installation

B.1 Introduction

This appendix describes the process to load the FDB-DIF and PECS update files into an existing FDB-DIF Oracle database using the FDB Updater Tool. The FDB-DIF update file is received on a schedule from FDB every two weeks and must be loaded in sequence. The PECS update file can be generated from the PECS application interface at any time by a user in the Release Manager role, and loaded with the FDB Update Tool.

B.2 Scope

This appendix will include the process to update the FDB-DIF Oracle database with the FDB and PECS Update files using the FDB Update Tool

Processes in scope will include:

- The process to load the FDB-DIF file using the FDB Update Tool.
- The process to load the PECS update file using the FDB Update Tool.
- The process to recover the FDB-DIF FDB_CUSTOM_* tables if a Custom Tables update file load failure should occur.

Processes out of scope will include:

- The process to receive the FDB-DIF update file from FDB.
- The process to generate the PECS update file from the Custom Tables application.
- The process to recover the FDB-DIF database if a FDB-DIF update file load failure should occur.

B.3 Update Process Prerequisites

- The FDB Updater Tool is installed on the machine performing the update process.
- The FDB and PECS update files are available and/or have been generated.
- The FDB-DIF update file must be loaded in sequence. The version number of the new file must be the next in sequence.
- The user performing Oracle operations must have sufficient rights to delete data, drop, and create tables.
- The PECS preparation and recovery SQL scripts are available.
- An Oracle tool such as SQLPlus, must be installed on the machine executing the FDB Updater tool.

B.4 Apply FDB-DIF Update File

This section lists the steps necessary to apply the FDB+DIF Update file.

B.4.1 Execute FDB Update Tool

Here are the steps required to execute the FDB update tool (four steps):

B.4.1.1 Obtain the FDB-DIF Update File

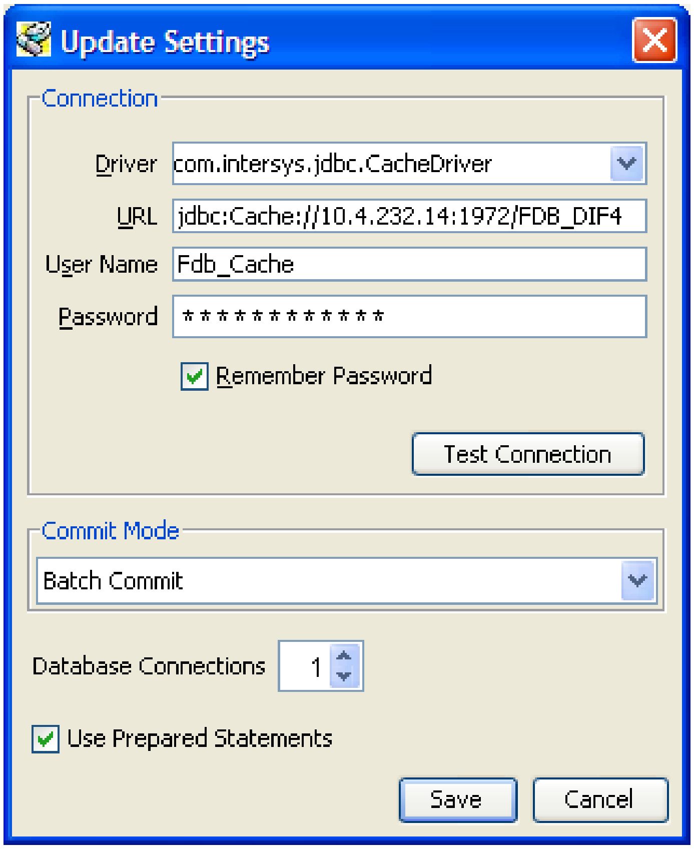
Either download the update file from the FTP directory, or insert the update CD into drive.

B.4.1.2 Start FDB Update Tool GUI

Navigate to where the FDB Update Tool has been installed, and click on the GUI.bat file.

B.4.1.3 Configure Connection

Select the View -> Setting menu option on the GUI and input the connection data relevant to your location and click the "Save" button. A sample screen is shown.



Driver: com.intersys.jdbc.CacheDriver

URL: jdbc:Cache://10.4.232.14:1972/FDB_DIF4

User Name: Fdb_Cache

Password: ************

☑ Remember Password

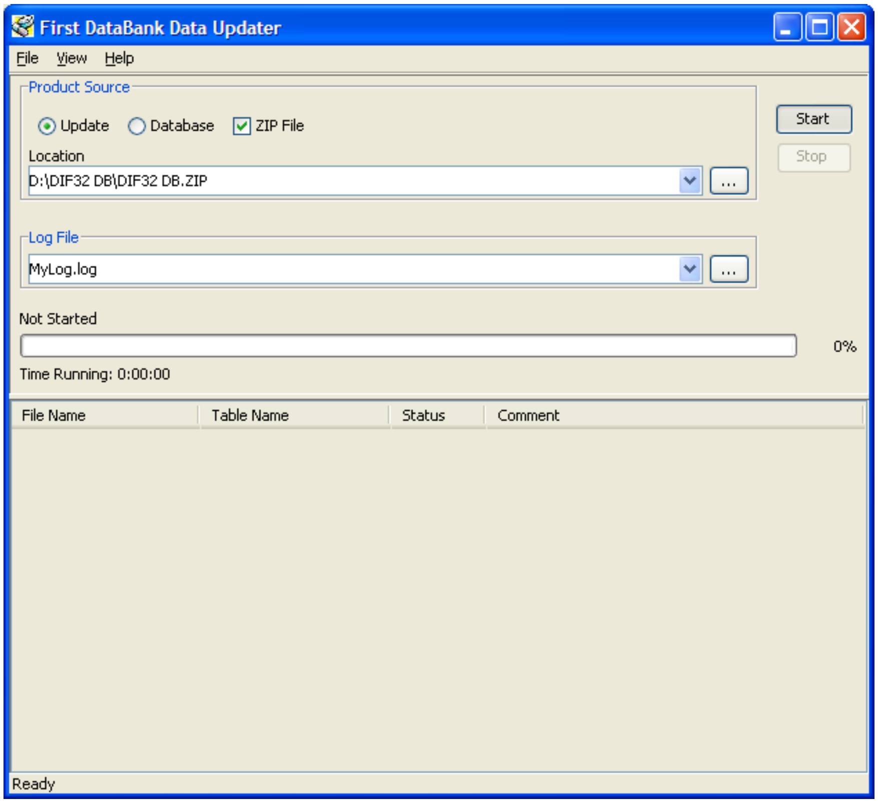
Commit Mode: Batch Commit

Database Connections: 1

☑ Use Prepared Statements

B.4.1.4 Provide File Paths

Enter the path to the update and log files relevant to your location. Select whether the update is incremental or complete. Click the “Start” button. A sample screen is shown:



Product Source

● Update ○ Database ☑ ZIP File

Location

D:\DIF32 DB\DIF32 DB.ZIP

Log File

MyLog.log

Not Started

0%

Time Running: 0:00:00

File Name	Table Name	Status	Comment

Ready

B.5 Apply Custom Tables Update File

This section describes the two major steps necessary to apply the Custom Tables Update File.

B.5.1 Verify CT_VERSION Table

The CT_VERSION table is an additional table added to the FDB schema (as recommended by FDB) to track the PECS update file version. If the table does not exist, execute the following DDL:

```
CREATE TABLE FDB.CT_VERSION
(
    VERSIONKEY NUMBER(6) NOT NULL,
    DBVERSION VARCHAR2(5) NULL,
    BUILDVERSION VARCHAR2(5) NULL,
    FREQUENCY VARCHAR2(1) NULL,
    ISSUEDATE VARCHAR2(8) NULL,
    VERSIONCOMMENT VARCHAR2(80) NULL,
    DBTYPE VARCHAR2(10) NULL
)

CREATE UNIQUE INDEX PKCTVERSION ON FDB.CT_VERSION(VERSIONKEY)
```

B.5.2 Execute FDB Update Tool

The steps to apply the Custom Tables Update file are the same steps as outlined in Apply FDB DIF Update File. Instead of entering the path to the FDB-DIF update file, enter the path to the Custom Tables Update file, relevant to your location. Select whether the update is incremental or complete. Click the “Start” button.

B.6 Recover FDB-DIF Custom Tables from Load Failure

The recover process may be necessary if a failure has occurred during the application of the PECS Update file (see Apply Custom Tables Update File step). The recovery process involves the execution of a SQL script, and verification that the data has been recovered.

B.6.1 Execute Recovery

The recovery entails the deletion of any data that may have been loaded to the FDB_CUSTOM_* tables during the execution of the update process.

B.6.2 Verify Data Recovery

Verify that the data in the FDB_CUSTOM_* tables has been deleted.

B.6.3 Generate Full PECS Update File

After logging into the PECS application, a user in the Release Manager role will navigate to the “Custom Update” tab, and click the “Download New Full Update” button. This will generate a PECS update file with all currently approved order check customizations.

Appendix C: KAAJEE

PECS uses the KAAJEE framework for user authorization and authentication. KAAJEE authenticates users against the Local VistA. Access to PECS is limited to known users with the security role. If new users need to be added, contact an experienced Local VistA administrator. Provide the administrator with a list of users that will be needed along with their required security keys. PECS security keys are discussed in the next section.

C.1 Security Keys

After a user is authenticated, KAAJEE retrieves his or her security keys from VistA and maps them to WebLogic security roles. The PECS application is secured so that only users running with the PECS security roles may access the PECS application.

PECS relies on the following four security keys, which must be added to VistA:

- PSS_CUSTOM_TABLES_ADMIN
- PSS_CUSTOM_TABLES_APPROVER
- PSS_CUSTOM_TABLES_REL_MAN
- PSS_CUSTOM_TABLES_REQUESTOR

PRE will rollout VISTA Patch, PSS*1*147 which exports the Security Keys, and those Security Keys are technically in the PSS namespace. These security keys have to be associated with the user accounts that will be set up on a VistA M server. The process for setting up the user accounts and the security keys is a part of the VistALink API setup mentioned in the VistALink Installation Manual available with the API software at: *http://vista.med.va.gov/kernel/kaajee/download_9-10.asp#vista_m_server*

C.2 Administrator User Role

PECS is configured to use the KAAJEE administrator user role by default. This requires the creation of a KAAJEE administrator account in WebLogic if one does not exist.

C.3 Resource Adapter

Use Resource Adapter with the supporting jars included in it and do not deploy jars as library files. The deployment order for the Resource Adapter must be at a lower number than the deployment order for PECS. This is to make sure the Resource Adapter is loaded and that the classes in the Resource Adapter are available when PECS is started. It is recommended to set the deployment order for the Resource Adapter to 95 or 99. This is because the default deployment order is 100.

(This page included for two-sided copying.)

Appendix D: PECS Logical Deployment Architecture

D.1 Logical Deployment Design – PECS

Application Server:

The WebLogic Application Server 10.3 will host PECS and its business services.

Database Server:

The Database Server- Oracle 11g will have Red Hat Linux Enterprise version RHEL5 as it OS. It will host the Custom Table Staging database and FDB-DIF database.

Failover Server:

There will be a Failover server. It will host both BEA WebLogic Application Server and Oracle Database Server to provide redundancy.

Legacy Interface:

There will be an existing VistA server which will host legacy KAAJEE and VistALink interface.

The figure below shows the overview of Logical Deployment Design for the PRE PECS Application.

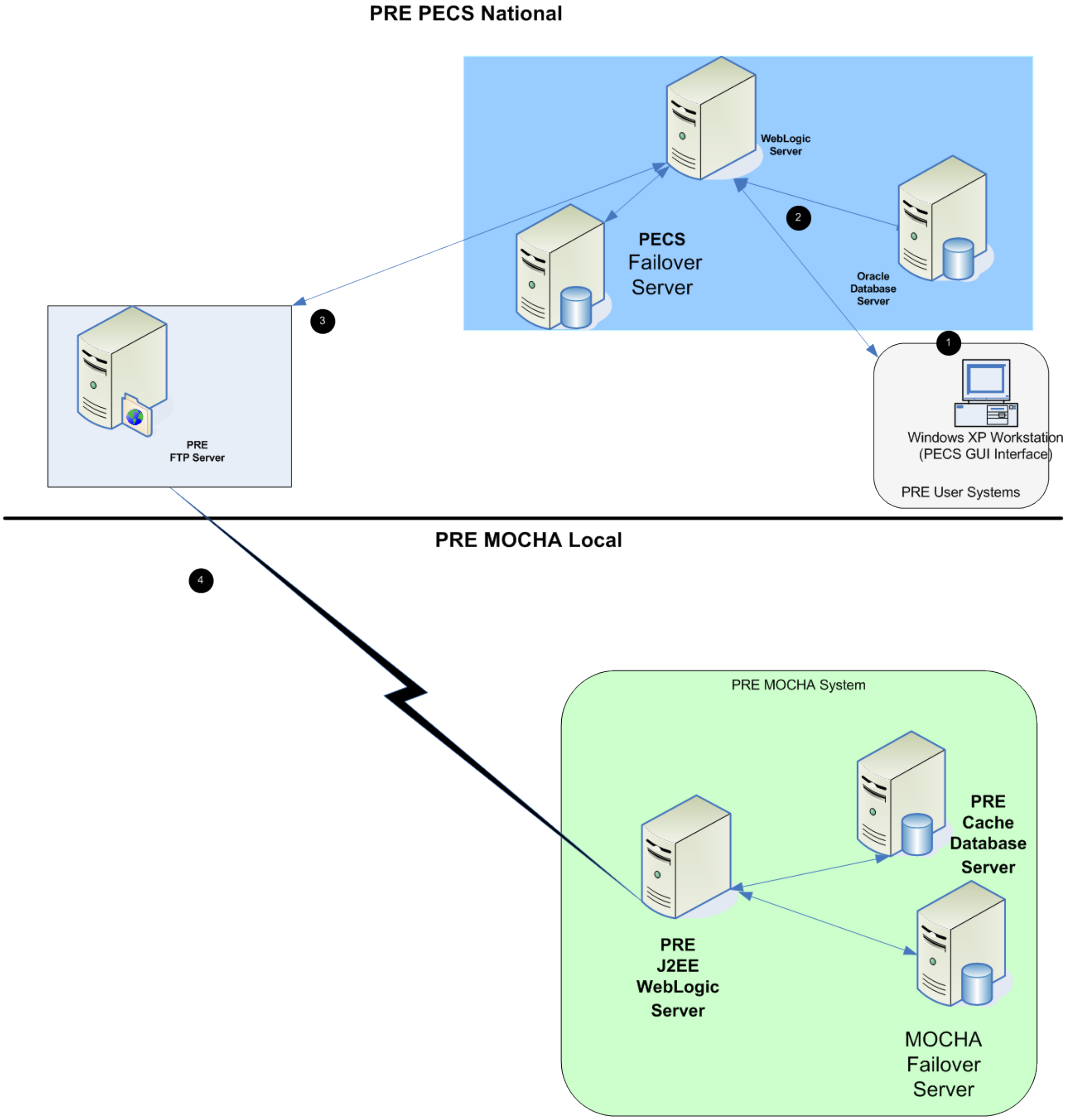
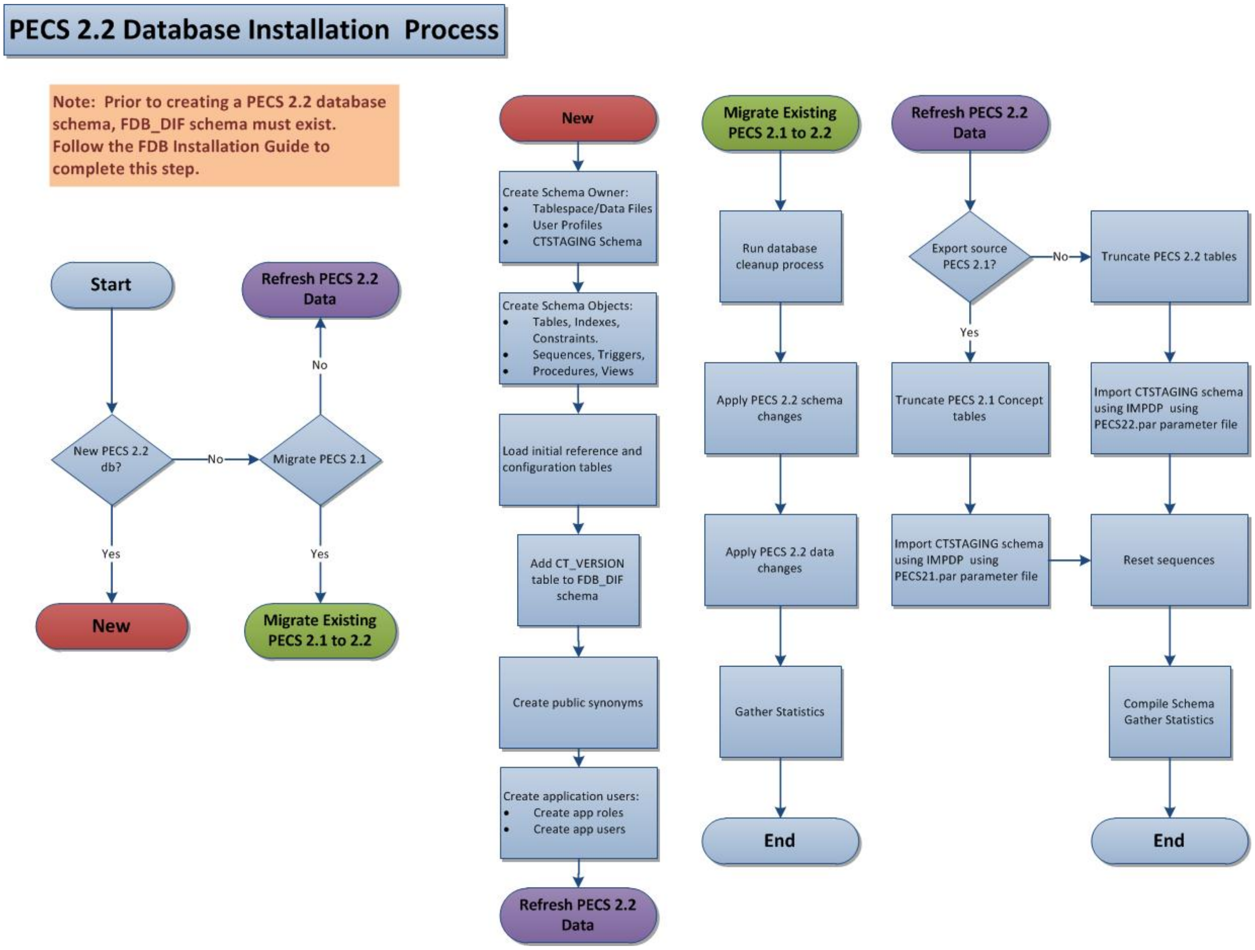


Figure 4: PECS Deployment

Appendix E: PECS Database Installation Process

E.1 Database Installation Process Flow – PECS


PECS 2.2 Database Installation Process
Note: Prior to creating a PECS 2.2 database schema, FDB_DIF schema must exist. Follow the FDB Installation Guide to complete this step.
Start
New PECS 2.2 db?
No
Migrate PECS 2.1
No
Refresh PECS 2.2 Data
Yes
New
Yes
Migrate Existing PECS 2.1 to 2.2
New
Create Schema Owner: • Tablespace/Data Files • User Profiles • CTSTAGING Schema
Create Schema Objects: • Tables, Indexes, Constraints. • Sequences, Triggers, • Procedures, Views
Load initial reference and configuration tables
Add CT_VERSION table to FDB_DIF schema
Create public synonyms
Create application users: • Create app roles • Create app users
Refresh PECS 2.2 Data
Migrate Existing PECS 2.1 to 2.2
Run database cleanup process
Apply PECS 2.2 schema changes
Apply PECS 2.2 data changes
Gather Statistics
End
Refresh PECS 2.2 Data
Export source PECS 2.1?
No
Truncate PECS 2.2 tables
Yes
Truncate PECS 2.1 Concept tables
Import CTSTAGING schema using IMPDP using PECS21.par parameter file
Import CTSTAGING schema using IMPDP using PECS22.par parameter file
Reset sequences
Compile Schema Gather Statistics
End

Appendix F: Rollback Process

If the installation process must be stopped when updating an environment from a previous version of PECS, use the following to determine and follow the steps outlined in order to rollback the application.

If both the database and the application have been deployed…

1. Shutdown the WebLogic domain.
2. Follow the instructions in 3.2.8 PECS v2.2 Database Migration Rollback.
3. Start the WebLogic domain.
4. Deploy the prior version of PECS using the instructions in 5.6 Application Deployment.

If only the database has been deployed

1. Shutdown the WebLogic domain.
2. Follow the instructions in 3.2.8 PECS v2.2 Database Migration Rollback.
3. Start the WebLogic domain.

(This page included for two-sided copying.)