Veterans Personal Finance System (VPFS)

Installation Guide

Version 1.2.0

July 2020

Department of Veterans Affairs (VA)
Office of Information and Technology (OIT)
Enterprise Program Management Office (EPMO)

Document Approval Signatures

Title: Project Manager	Signed: [signature on file] Typed: redacted	Date:
Title: SQA Manager	Signed: [signature on file] Typed: redacted	Date:

History of Revisions

Version	Section Changed	Description of Change	Effective Date	Contacts
1.0		Initial release of document	12/1/06	Project Manager: redacted Developers: redacted
1.1.2	2.3	Dependency services upgrade These changes are associated with the VPFS patch release 1.1.2	06/24/09	redacted

1.1.3	2.3 Dependencies 6.4.1 Before You Begin	Added note about SDS 18 upgrade Updated VPFS dependency components version. Updated Footer v 1.1.3 Step 7. Added notes about context-root These changes are associated with the VPFS patch release 1.1.3	06/28/2010 12/21/2010 04/12/2011	redacted
1.2.0		Incorporating 2FA Functionality and TRM Upgrades	1/7/2019 – 02/05/2020	redacted redacted redacted redacted redacted redacted

Table of Contents

About This Document **1**
Purpose 1
Audience 1
Conventions Used in this Document 1
1. Overview **2**
1.1 Application Modernization Project 2
2. Pre-Requisites/Conditions **3**
2.1 Target Installation Environment 3
2.2 Software Tools 3
2.3 Dependencies 4
3. Upgrading a Prior Installation **6**
4. Dependency Installation Notes ....
7 4.1 Interface Installation Notes 7
4.1.1 VistALink 7
4.1.2 Standard Data Service (SDS) 8
4.1.3 KAAJEE 8
4.1.4 Person Service Lookup (PSL) / Person Service Construct (PSC) 8
4.1.5 Electronic Signature 9
5. Database Setup **10**
Using an Installation Log 10
5.1 Preparing the Installation Scripts 10
5.2 Creating VistAMigrate Schema 11
5.3 Creating Objects for VistAMigrate User 11
5.4 Creating VPFS Schema 11
5.5 Granting Access to SDS Objects for VPFS 12
5.6 Creating Objects for VPFS User 12
5.7 Creating Jobs for the Stored Procedures 13
5.8 Creating VPFS Project in VistAMigrate 14
6. VPFS Application Installation **15**
6.1 Signing into the WebLogic Server Administration Console 15
6.2 Creating a JDBC Connection Pool for VPFS 16
6.3 Creating a Data Source for the Connection Pool 22
6.4 Deploying the VPFS Application 25
6.4.1 Before You Begin 25
6.4.2 Steps to Deploy 26
6.5 Logging 27
7. VPFS VistAMigrate Installation **29**
7.1 Logging in to the WebLogic Server Administration Console 29
7.2 Creating a JDBC Connection Pool for VistAMigrate 30
7.3 Creating a Data Source for the Connection Pool 36

7.4 Root and Code Folders Setup 38
7.5 Deploying the VistAMigrate Application 39
7.5.1 Before You Begin 39
7.5.2 Steps to Deploy 40
7.6 Logging 41

8. Post-Installation Steps 42
8.1 Post Interface Installation Steps 42
8.1.1 Connecting to VistA Sites. 42

9. Testing the VPFS Application Installation 43
9.1 Testing the Installation of the VPFS Application 43
9.2 Testing the Installation of the VistAMigrate Application 43
9.3 Testing the SMTP Connection 44

10. Rolling Back the VPFS Application Install. 45
Appendix A. References 46
VPFS Project Documents 46
Non-VPFS Project Documents 46

Appendix B. Acronyms and Abbreviations 47

List of Figures

Figure 6-1. WebLogic Server Administration Console Sign-In Page 15
Figure 6-2. JDBC Connection Pool Configuration page 16
Figure 6-3. Choose Database page 16
Figure 6-4. Define Connection Properties page 17
Figure 6-5. Create and Deploy page 18
Figure 6-6. JDBC Connection Pools configuration page 18
Figure 6-7. General page for the connection pool 19
Figure 6-8. Connection page 19
Figure 6-9. Advanced Options page 21
Figure 6-10. JDBC Data Sources page 22
Figure 6-11. Configure the data source page 23
Figure 6-12. Connect to connection pool page 23
Figure 6-13. Target the data source page 24
Figure 6-14. Configuration page table 24
Figure 7-1. WebLogic Server Administration Console Sign-In Page 29
Figure 7-2. JDBC Connection Pool Configuration page 30
Figure 7-3. Choose Database page 30
Figure 7-4. Define Connection Properties page 31
Figure 7-5. Create and Deploy page 32
Figure 7-6. JDBC Connection Pools configuration page 32
Figure 7-7. General page for the connection pool 33
Figure 7-8. Connection page 33
Figure 7-9. Advanced Options page 35
Figure 7-10. JDBC Data Sources page 36
Figure 7-11. Configure the data source page 36
Figure 7-12. Connect to connection pool page 37

Figure 7-13. Target the data source page 37
Figure 7-14. Configuration page table 38
Figure 9-1. Welcome to the Veterans Personal Finance System Page 43
Figure 9-2. VistAMigrate Pick Project Page 44

List of Tables

Table 1-1. The Installation Process 2
Table 4-1. Sample Grid for Recording M Connection Information for VistaLink 8

This page is left blank intentionally.

About This Document

Purpose

The purpose of the System Administration/Installation Guide is to provide step-by-step instructions for installing the various components for the Veterans Personal Finance System (VPFS).

Audience

This document is intended to be used by system administrators for installing and maintaining the Oracle database, VPFS and VistAMigrate.

Conventions Used in this Document

Format	Indicates…
bold	A control that you click, such as a button, icon, or link, or a field label. Example: Click **Deploy**.
`Courier` - normal	Text that appears on the screen or in a log file. Example: `Table created.`
`Courier` - **bold**	Text that you type. `SQL>` **`quit`**
`<Courier>` `italic and enclosed in angle brackets`	A variable that you replace with the specified text Enter password: *`<SYSDBA password>`*

1. Overview

1.1 Application Modernization Project

The Department of Veterans Hospital Administration (VHA) is in the process of re-hosting a number of legacy applications, moving from the MUMPS-based VistA environment to the VA‟s standard modern environment, which includes Oracle for the database environment and the Java 2 Platform, Enterprise Edition (J2EE) for the application software environment. The redesigned system, the Veterans Personal Financial System (VPFS), uses the Oracle 11g database and the BEA WebLogic 10.3.6 Application Server, built using the J2EE open source platform running on Red Hat Linux servers. The database is centralized for all facilities at the VA Enterprise Management Center, and the application is accessible via a secure, 508-compliant Web browser front end.

Table 1-1. The Installation Process

Task Sequence	Task Description	For information, see…
1	Verify that all pre-requisites and pre-conditions are met.	Chapter 2. Pre-Requisites/Conditions
2	Install all dependent interfaces.	Chapter 4. Dependency Installation Notes
3	Install the VPFS database using SQL*Plus scripts provided in the distribution package. • Use the batch script that prompts you through all steps, or • Use an individual script for each step.	Chapter 6. Database Setup
4	Install the VPFS application using the WebLogic Application Server Console.	Chapter 7. VPFS Application Installation
5	Install the VistAMigrate application using the WebLogic Application Server Console.	Chapter 8. VPFS VistAMigrate Installation
6	Perform all post-installation steps.	Chapter 9. Post-Installation Steps
7	Test the installation.	Chapter 10. Testing the VPFS Application Installation

2. Pre-Requisites/Conditions

This section provides a checklist for the pre-requisites and conditions that must be met before you can install and run the Veterans Personal Funds System (VPFS) application.

2.1 Target Installation Environment

The following environment must be in place:

☐ A BEA cluster of managed WebLogic servers (v 10.3.6 or higher) running on Red Hat Linux Advanced Server 4.0 or higher

☐ A load balancer (either hardware or software)

☐ A database server running Oracle Enterprise Database 10g

2.2 Software Tools

The following applications must be installed as part of the BEA WebLogic install:

☐ Java Development Kit (JDK) v1.7.0_80 or higher

☐ J2EE v1.7 or higher

☐ the Sun Java Virtual Machine (Sun JVM)

The following tools must be installed on the application server used by VistAMigrate. It is recommended that an Oracle administrator perform this install. This installation requires system database administrator privileges:

☐ Oracle SQL*Loader

☐ Oracle SQL*Plus

Note: These tools are included in the suite of administration tools that is installed when you select the Administrator installation option of the Oracle client. The Oracle client is provided in the Oracle Database Client Release for Linux or Windows download available on Oracle's web site.

---

2 Pre-Requisites/Conditions

2.3 Dependencies

The following dependencies and services must be installed and configured:

Note: For detailed installation instructions regarding Java and M dependencies, see the corresponding product documentation.

Services

☐ VistALink v1.6

☐ SDS v13.0 or higher (including SDS schema)

☐ Person Service Lookup v4.0.4.4 or higher

.

☐ Person Service Construct v2.0.0.8 or higher

☐ E-Signature v1.0.0.024 or higher

☐ SSOI v1.0.0

Kernel Authentication and Authorization for J2EE (KAAJEE)

☐ KAAJEE v1.0.1.003, KAAJEE SSPI v1.0 or higher

☐ VistA users are assigned to roles and granted access permissions in KAAJEE.
For information about user groups and permission, see the *VPFS Software Design Document (SDD) Appendix E – User Roles and Access*. **Vista Legacy**

☐ All M connection information must be available (such as IP address, port number, and access and verify codes) for each M system with which VPFS will interface.

☐ All required M patches (for RPCs, migration, etc.) must be applied to the VistA systems with which VPFS will communicate.

Note: With the latest release of VPFS version 1.2.0, SDS API jar has been upgraded to v18.0.

The interface interdependencies (on the Java side) are:

Interface Name	Dependent Interfaces
VistALink 1.6	None
Standard Data Service (SDS)	None
KAAJEE	VistALink 1.6, SDS Institution API
Patient Service Construct	VistALink 1.6
Patient Service Lookup	VistALink 1.6
E-Signature	VistALink 1.6

Note: VistALink is required by any interface that connects to VistA. KAAJEE uses VistALink and SDS internally. These two interfaces should be installed before the others. Once these two interfaces are installed, the other interfaces may be installed in any order.

This version of VPFS uses VistALink libraries that have reverse-IP check enabled.
Confirm all entries in the VistALink connectorConfig.xml file have the reverse IP information configured. Otherwise VistALink connector will not be able to connect out to those systems. This was done for security purposes and to pass Fortify compliance.

---

3. Upgrading a Prior Installation

If a newer version of any component of VPFS or any of the dependencies becomes available, please refer to the readme or release notes for that component.

VPFS Database upgrades will be distributed as patches for existing installations, as well as updated installation scripts for new installations. Each database patch is supplied with an associated readme file with specific installation instructions for that patch.

Upgrades to the VPFS or VistAMigrate applications can generally be performed by following the process outlined in sections 6.4 (Deploying the VPFS Application) or 7.5 (Deploying the VistAMigrate Application), respectively. Please refer to the readme file that is distributed with each application for any additional steps that may be required.

4. Dependency Installation Notes

Install interfaces in accordance with the installation instructions provided with the interface. Installation notes are provided in this section, as needed, only if specific installation information is needed for the VPFS installation.

Install the following interfaces in the same domain that you are using to deploy VPFS:

☐ VistALink

☐ SDS

☐ KAAJEE

☐ Patient Service Lookup

☐ Patient Service Construct

4.1 Interface Installation Notes

4.1.1 VistALink

VistALink is the transport mechanism for any interface that connects to the legacy system. Detailed installation instructions for both the Java and M components are included in the VistALink distribution.

- Before installing VistALink, make sure you have the M connection information (IP address, port number, access and verify code) for each M system with which VPFS will interface. These settings are used in the VistALink connectorConfig.xml configuration file to establish the VistALink connections.

Note: Table 4-1 provides a sample grid that you could use to record M connection information.

- **Install and configure VistALink before deploying VPFS.** If you deploy VPFS before you install VistALink you will receive an error when attempting to deploy VPFS.
- Ensure that M users are assigned to groups and roles so that they will have the proper permissions to access the VistALink menu options, otherwise VistALink RPC calls will fail. This is documented in detail in the VistALink installation instructions.
- VistALink has special installation steps to successfully deploy on WebLogic. See the VistALink documentation for installation instructions.

Table 4-1. Sample Grid for Recording M Connection Information for VistaLink

M Connection:		**M Connection:**	
IP Address:		IP Address:	

Port number:		Port number:	
Access code:		Access code:	
Verify code:		Verify code:	
M Connection:		**M Connection:**	
IP Address:		IP Address:	
Port number:		Port number:	
Access code:		Access code:	
Verify code:		Verify code:	

4.1.2 Standard Data Service (SDS)

The SDS API is a core service component that provides standard reference data that describes the VHA organizational structure.

- Create/modify the SDS schema. The SDS schema/user is referenced in the VPFS database scripts and may need to be changed to match the actual schema name.
- Pay close attention to the Standard Data Service Server Administration Installation Guide, which describes how the component is configured. VPFS will not function unless SDS is properly configured.

4.1.3 KAAJEE

- After installing the KAAJEE SSPI, be sure to update the KaajeeDatabase.properties to access the correct database.
- Prior to deploying the VPFS and VistAMigrate applications, make sure the kaajeeConfig.xml file is updated with appropriate STS service address and with the station numbers of the sites you need to access. This is covered in more detail during VPFS and VistAMigrate installation.

4.1.4 Person Service Lookup (PSL) / Person Service Construct (PSC)

- To use Person Service Lookup (PSL) and Person Service Construct (PSC) with VPFS special broker type configurations must be set for legacy VistA user accounts. See the Person Service installation guides for specific legacy account setup instructions.
- Prior to deploying the VPFS and VistAMigrate applications, make sure the PSL and PSC properties are updated as needed for your server configuration. This is covered in more detail in Section 6.4.1.

4.1.5 Electronic Signature

- The Java interface for Electronic Signature (e-signature) is incorporated in the VPFS application. No additional Java-side installation is required.

.

- To use Electronic Signature with VPFS a special broker type configuration must be set for legacy VistA user accounts. See the Electronic Signature installation guide for specific legacy account setup instructions.

5. Database Setup

Use the steps in this section to execute the SQL*Plus scripts and set up the VPFS database on the Oracle database server. The scripts used for the VPFS database refer to the default tablespace. If your site is using another tablespace for VPFS, edit the scripts to point to the correct tablespace prior to executing these scripts.

Note: You may prefer to group tasks differently than the sequence presented in this chapter based on user privileges or some other criteria. For example: A user with SYSDBA privileges may choose to log on as SYSDBA and create both schemas (VPFS and VistAMigrate) before creating the objects or granting privileges for those schemas. Be aware that there may be prerequisites, as noted.

Note: This procedure is written for new installations. Database patches may be available to upgrade an existing installation. Please refer to section 3 (Upgrading a Prior Installation) for upgrade installations.

Using an Installation Log

SQL*Plus allows you to create an installation log containing the output of the SQL script commands. After logging into the Oracle database using SQL*Plus, use the SPOOL command to generate a log file.

Note: Any time you restart SQL*Plus, you will need to set the spool output again. This setting is not maintained when you quit.

1. Create the log file. If you are using Oracle 10g and have an existing log file you want to append to, include "`append`" at the end of the command.

   ```
   SQL> spool install_vpfs.log
   ```

2. Execute the script(s) specified in the steps below.
3. Close the log file or quit SQL*Plus.

After installation is complete, view the installation log using your preferred text viewer. You can use the `more` command or an application such as gedit (Linux) or Notepad (Windows).

5.1 Preparing the Installation Scripts

The VPFS scripts need to be extracted to a machine that has SQL*Plus access to the Oracle database where the VPFS and VistAMigrate schemas will be created:

1. Create a temporary directory on your local drive.

   Note: A temporary directory can be on the SAN or on another drive as long as you are able to access the scripts and run the scripts for the database server.

2. Extract the contents from the database scripts distribution package to the temporary directory created above.

5. Database Setup

5.2 Creating VistAMigrate Schema

Requires DBA: To complete this step you must be logged on to the Oracle database as a user with DBA privileges.

Note: The script in this procedure creates a database user with username "vista_migrate" and password "vista_migrate" in the "USERS" tablespace. If you choose to change the username and password, or default or temporary tablespaces, be sure to edit the cr_vista_migrate_schema.sql and cr_vista_migrate_objects.sql scripts to match prior to executing them. Also make sure these changes are reflected in the Data Source section.

To create the VistAMigrate schema:

1. Change to the vistamigrate/scripts directory.
2. Start SQL*Plus and log on to the Oracle database as a user with DBA privileges.

   ```
   $ sqlplus

   Enter user-name: sys/password@databasename [as sysdba]
   ```

3. Run cr_vista_migrate_schema.sql to create the VistAMigrate schema:

   ```
   SQL> @cr_vista_migrate_schema.sql
   ```

4. Continue with the next section of this document or log out (`SQL> quit`).

5.3 Creating Objects for VistAMigrate User

Notes: The VistAMigrate schema should already have been created. If this is not the case, create it before continuing.

The script in this procedure creates database objects in the "vista_migrate" schema. If you chose a different VistAMigrate schema name, be sure to edit the cr_vista_migrate_objects.sql and pk_DM_UTILS.sql scripts before continuing.

To create objects for VistAMigrate User:

1. Start SQL*Plus (if not already running) and log on to the Oracle database as the VistAMigrate user.

   ```
   SQL> connect vista_migrate/vista_migrate@databasename
   ```

2. Run cr_vista_migrate_objects.sql to create all the relevant tables and objects for VistAMigrate.

   ```
   SQL> start cr_vista_migrate_objects.sql
   ```

3. Continue with the next section of this document or log out (`SQL> quit`).

5.4 Creating VPFS Schema

Requires DBA: To complete this step you must be logged on to the Oracle database as a user with DBA privileges.

Note: The script in this procedure creates a database user with username "vpfs" and password "vpfs" in the "USERS" tablespace. If you choose to change the username and password, or default or temporary tablespaces, be sure to edit the cr_vpfs_schema.sql and cr_all_objects.sql scripts to match prior to executing them. Also make sure these changes are reflected in the Data Source section.

To create the VPFS Schema:

1. Change to the vpfs/scripts directory.
2. Start SQL*Plus and log on to the Oracle database as a user with DBA privileges.

```
$ sqlplus

Enter user-name: sys/password@databasename [as sysdba]
```

3. Run cr_vpfs_schema.sql to create the VPFS schema.

```
SQL> @cr_vpfs_schema.sql
```

4. Continue with the next section of this document or log out (`SQL> quit`).

5.5 Granting Access to SDS Objects for VPFS

Notes: The SDS and VPFS schemas should already have been created. If this is not the case, create them before continuing. Contact the Health System Design & Development Standard Data Service for SDS database installation instructions.

The script in this procedure grants SELECT permissions on tables in the "sdsadm" schema to the "vpfs" user. If you chose different schema names for SDS or VPFS, be sure to edit the grant_sds_access.sql script to match prior to executing it.

To grant access to SDS objects:

1. Start SQL*Plus (if not already running) and log on to the Oracle database as a user with DBA privileges or the SDS user.

```
SQL> connect sys/password@databasename [as sysdba]
```

2. Run grant_sds_access.sql to grant read access to the SDS tables to the VPFS user.

```
SQL> start grant_sds_access.sql
```

3. Continue with the next section of this document or log out (`SQL> quit`).

5.6 Creating Objects for VPFS User

Notes: The SDS and VPFS schemas should already have been created. If this is not the case, create them before continuing.

The scripts in this procedure create database objects in the "vpfs" schema and reference the SDS and VistAMigrate schemas. The following scripts may need to be edited prior to executing them:

1. If you chose a different VPFS schema name, edit the create_objects/cr_all_objects.sql script.

2. If you chose a different SDS schema name (default is "sdsadm"), edit the create_objects/cr_all_objects.sql and reference_data/ins_vpfs_institution.sql scripts to match.

3. If you chose a different VistAMigrate schema name (default is "vista_migrate"), edit the stored_procedures/pk_convertPFOPData_body.sql script to match.

To create objects for VPFS User:

1. Start SQL*Plus (if not already running) and log on to the Oracle database as the VPFS user.

   ```
   SQL> connect vpfs/vpfs@databasename
   ```

2. From the scripts/create_objects folder, run cr_all_objects.sql to create all the relevant tables and objects for VPFS.

   ```
   SQL> start create_objects/cr_all_objects.sql
   ```

3. From the scripts/stored_procedures folder, run cr_all_spObjects.sql to create all the stored procedures required for VPFS.

   ```
   SQL> start stored_procedures/cr_all_spObjects.sql
   ```

4. From the scripts/triggers folder, run cr_all_triggers.sql, to create all the triggers required for VPFS.

   ```
   SQL> start triggers/cr_all_triggers.sql
   ```

5. From the scripts/reference_data folder, run ins_all_reference_data.sql script to insert the reference data to the appropriate tables.

   ```
   SQL> start reference_data/ins_all_reference_data.sql
   ```

6. To enable the sending of email notifications by VPFS, type the following commands at the SQL> prompt to set the email_enabled flag and the address for the SMTP server:

   ```
   SQL> UPDATE vsystem_parameter SET parameter_txt='true' WHERE
   parameter_cd='EMAIL_ENABLED';
   ```

   Replace the italicized text in brackets with the name of your SMTP server:

   ```
   SQL> UPDATE vsystem_parameter SET parameter_txt='[SMTP_SERVER]'
   WHERE parameter_cd='SMTP_SERVER';
   ```

   Example: if the SMTP server is smtp.va.gov, then the above SQL statement would read:

   ```
   SQL> UPDATE vsystem_parameter SET parameter_txt='smtp.va.gov'
   WHERE parameter_cd='SMTP_SERVER';
   ```

7. Commit the changes to the database.

```
SQL> commit;
```

8. Continue with the next section of this document or log out (`SQL> quit`).

5.7 Creating Jobs for the Stored Procedures

Note: The objects for the VPFS user should already have been created. If this is not the case, create them before continuing.

To create jobs for VPFS User:

1. Start SQL*Plus (if not already running) and log on to the Oracle database as the VPFS user.

   ```
   SQL> connect vpfs/vpfs@databasename
   ```

2. From the scripts/jobs folder, run cr_jobs.sql to create jobs for all stored procedures.

   ```
   SQL> start jobs/cr_jobs.sql
   ```

   (Optional) To verify that the jobs were created successfully, run the list_jobs.sql script.

   ```
   SQL> start jobs/list_jobs.sql
   ```

3. Continue with the next section of this document or log out (`SQL> quit`).

5.8 Creating VPFS Project in VistAMigrate

Notes: The objects for the VPFS and VistAMigrate users should already have been created. If this is not the case, create them before continuing.

The script in this procedure grants EXECUTE permissions on VistAMigrate stored procedures to the VPFS user. If you chose different schema names for VistAMigrate or VPFS, be sure to edit the ins_vpfs_data.sql script to match prior to executing it.

To create the VPFS project in VistAMigrate:

1. Change to the vistamigrate/scripts directory.
2. Start SQL*Plus and log on to the Oracle database as the VistAMigrate user.

   ```
   $ sqlplus

   Enter user-name: vista_migrate/vista_migrate@databasename
   ```

3. Run ins_vpfs_data.sql to create the VPFS project in VistAMigrate.

   ```
   SQL> start ins_vpfs_data.sql
   ```

4. Log out of SQL*Plus (`SQL> quit`).

6. VPFS Application Installation

Important: The VPFS application relies on nightly updates from PSC through VistALink. A VistALink application proxy user account (e.g. VPFS, APPLICATION PROXY) must be present for nightly updates to take place. This user account is created during the installation of Patch PRPF*3.0*16 in the legacy VistA database. If problems are encountered running the nightly update, contact IRM to verify that the application proxy user account was created successfully.

Perform the following steps from the WebLogic Server Administration Console:

1. Create a JDBC connection pool for VPFS.
2. Create a Data Source for the VPFS connection pool.
3. Deploy the VPFS application.
4. Create the VPFS logging directory.

6.1 Signing into the WebLogic Server Administration Console

To sign into the WebLogic Server Administration Console:

1. Open a web browser and go to **`http://`***`<host>`*`:`***`/console/`** The WebLogic Server Administration Console sign in page opens.

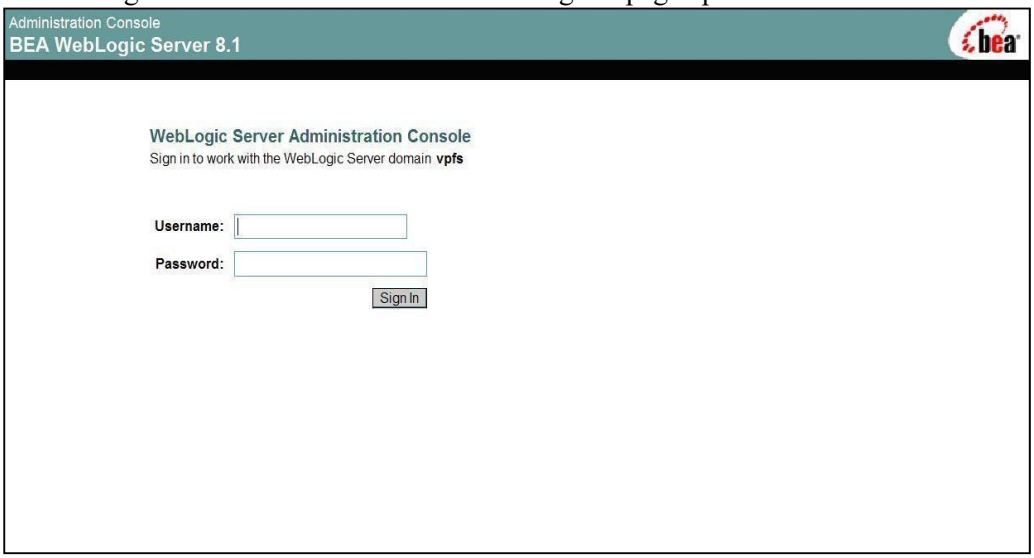


Figure 6-1. WebLogic Server Administration Console Sign In Page

2. Sign in as the administration user:
   a. **Username**: Enter your administration username.
   b. **Password**: Enter your administration password.

6.2 Creating a JDBC Connection Pool for VPFS

VPFS connects to the Oracle database through WebLogic using JDBC connection pools. JDBC connection pools manage resources by holding connections to the database open until they are needed by HealtheVet VistA applications.

Use the WebLogic Server Administration Console to create a JDBC connection pool for VPFS.

1. In the navigation tree on the left of the console, expand **Services** > **JDBC**.
2. Click the **Connection Pools** link.
   The JDBC Connection Pool Configuration page displays the list of connection pools that already exist.

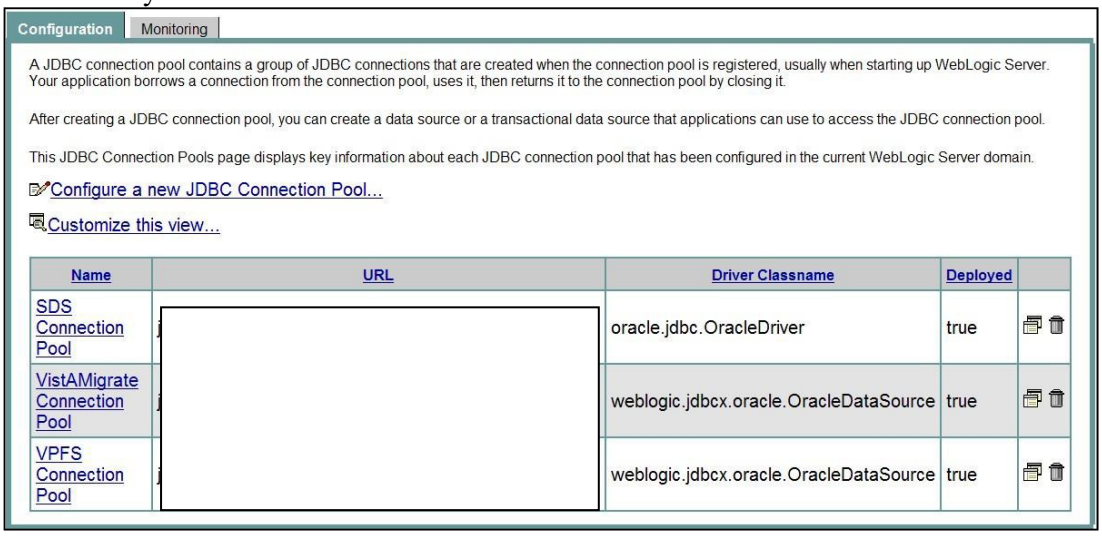


Figure 6-2. JDBC Connection Pool Configuration page

3. From the main panel, click the **Configure a new JDBC Connection Pool** link. The Choose Database page opens.

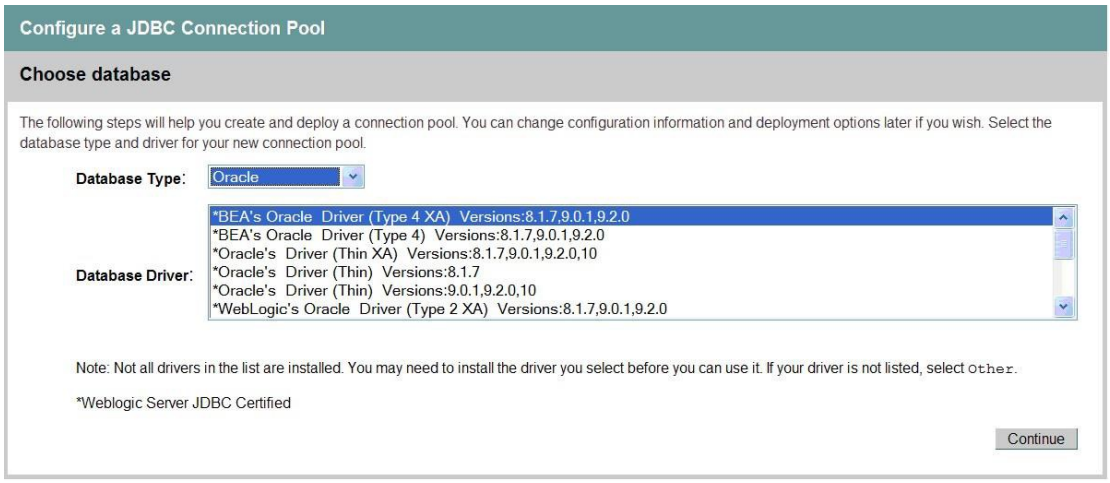


Figure 6-3. Choose Database page

4. Select the database type and driver:
   a. **Database Type**: Select Oracle.

b. **Database Driver**: Select *Oracle's Driver (Thin XA) Versions: 8.1.7,9.0.1,9.2.0,10

5. Click **Continue**.
   The Define Connection Properties page opens.

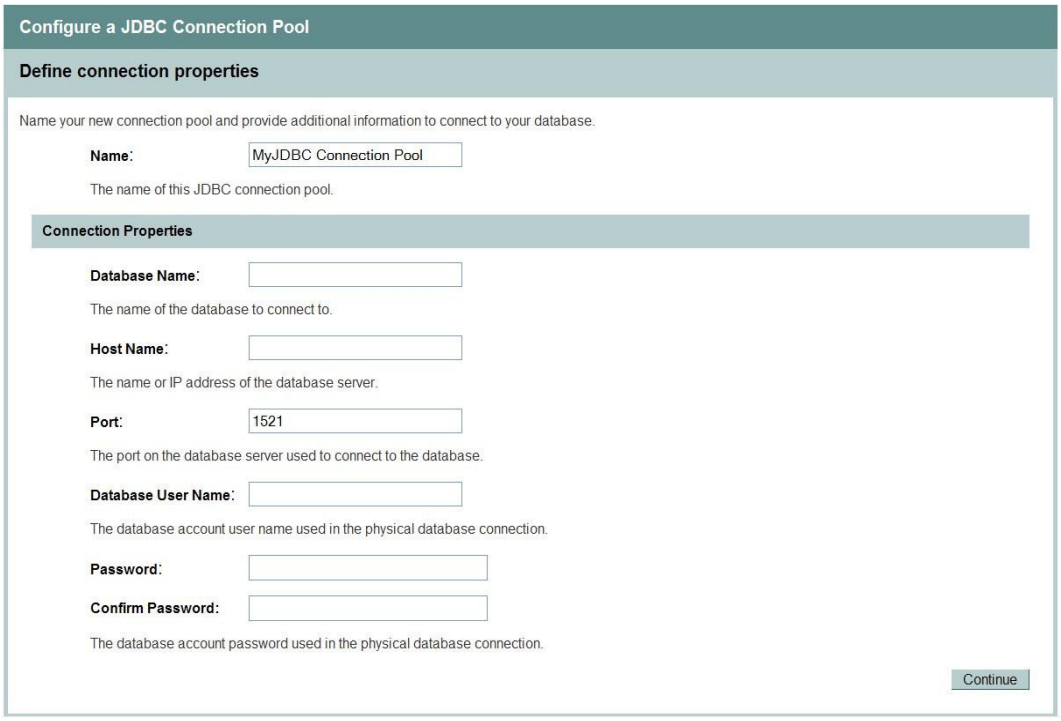


Figure 6-4. Define Connection Properties page

6. Enter Connection Properties:
   a. **Name**: Enter the name of the VPFS JDBC connection pool you are defining.
   b. **Database Name**: Enter the name of the database (or service name) to connect to.
   c. **Host Name**: Enter the name or IP address of the host machine.
   d. **Port**: Enter the port of the database server used to connect to the database. The Oracle default port is 1521.
   e. **Database User Name**: Enter your database username.
   f. **Password**: Enter your password.
   g. **Confirm Password**: Enter your password again to confirm.
7. Click Continue.
8. To test the database connection, click Test Driver Configuration.
   - If the configuration is correct, the Create and Deploy page opens. Continue with step 9.
   - If an error message appears at the top of the page, correct the error before you continue.

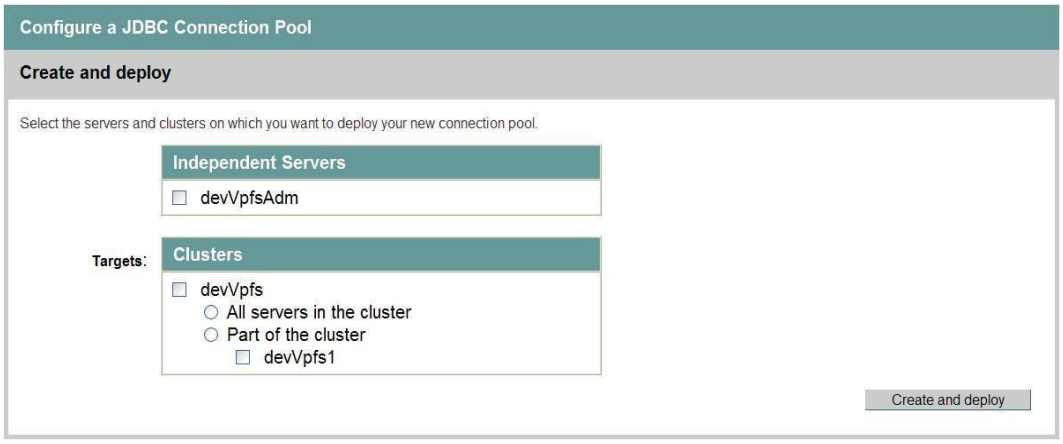


Figure 6-5. Create and Deploy page

9. Select the radio button for **All servers in the cluster** to target the connection pool to the cluster.
10. Click **Create and deploy**.
    Once created successfully, the JDBC Connection Pools configuration page opens. The new JDBC Connection Pool is added to the table on the JDBC Connection Pools configuration page.

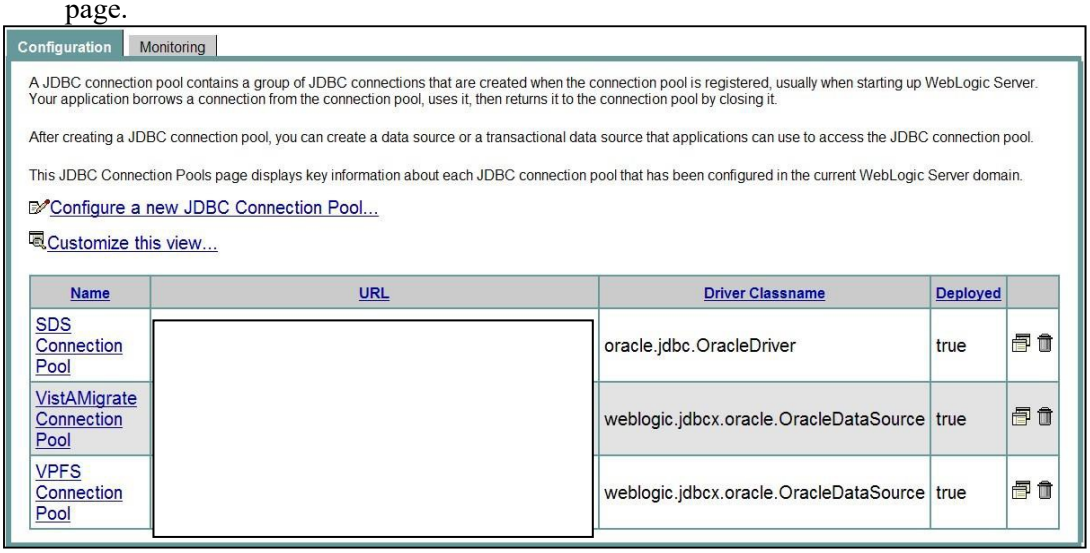


Figure 6-6. JDBC Connection Pools configuration page

11. Set the Advanced Options:
    a. Click the name of the JDBC Connection Pool that you created. The General page opens for the connection pool you requested.

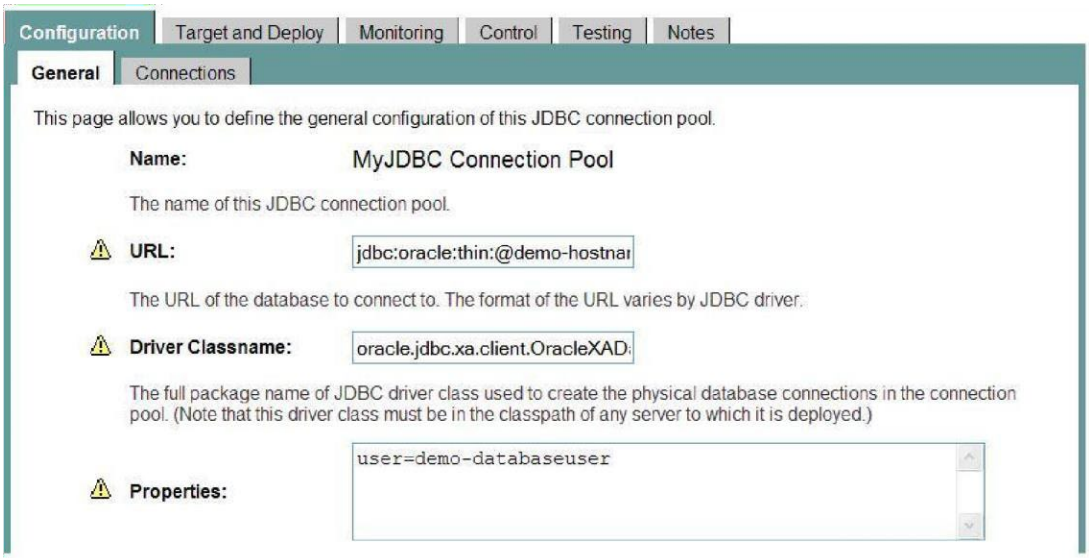


Figure 6-7. General page for the connection pool

b. Click the **Connections** tab.
The Connection page opens.

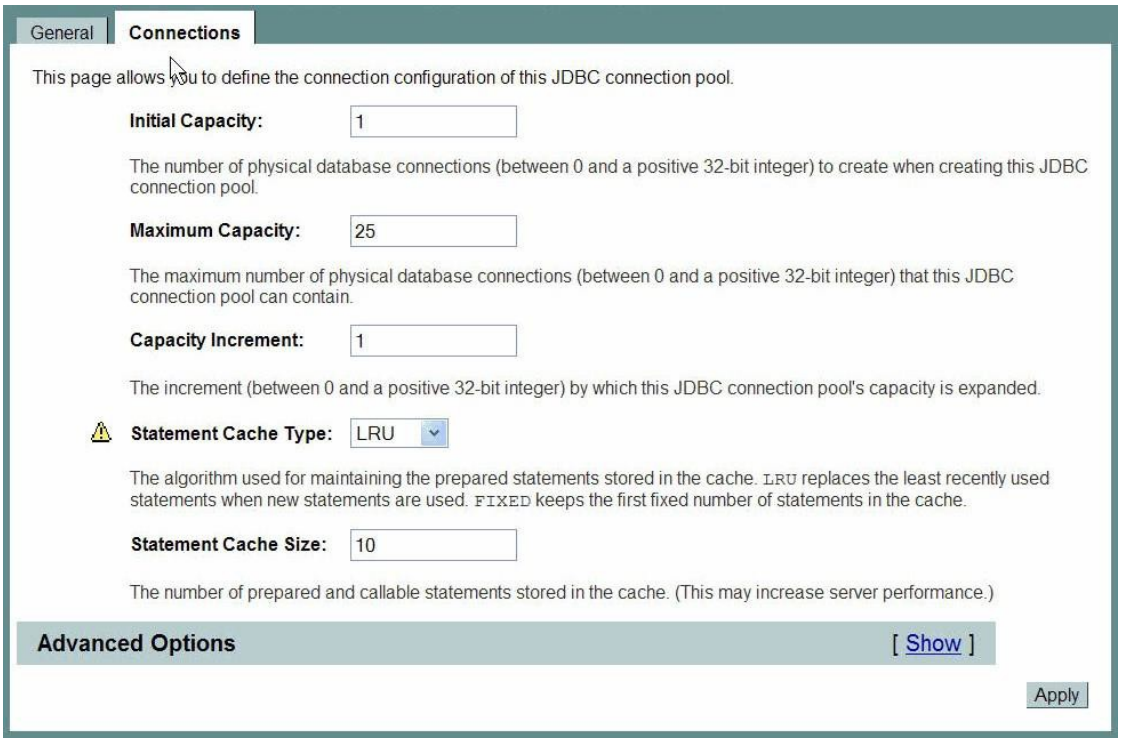
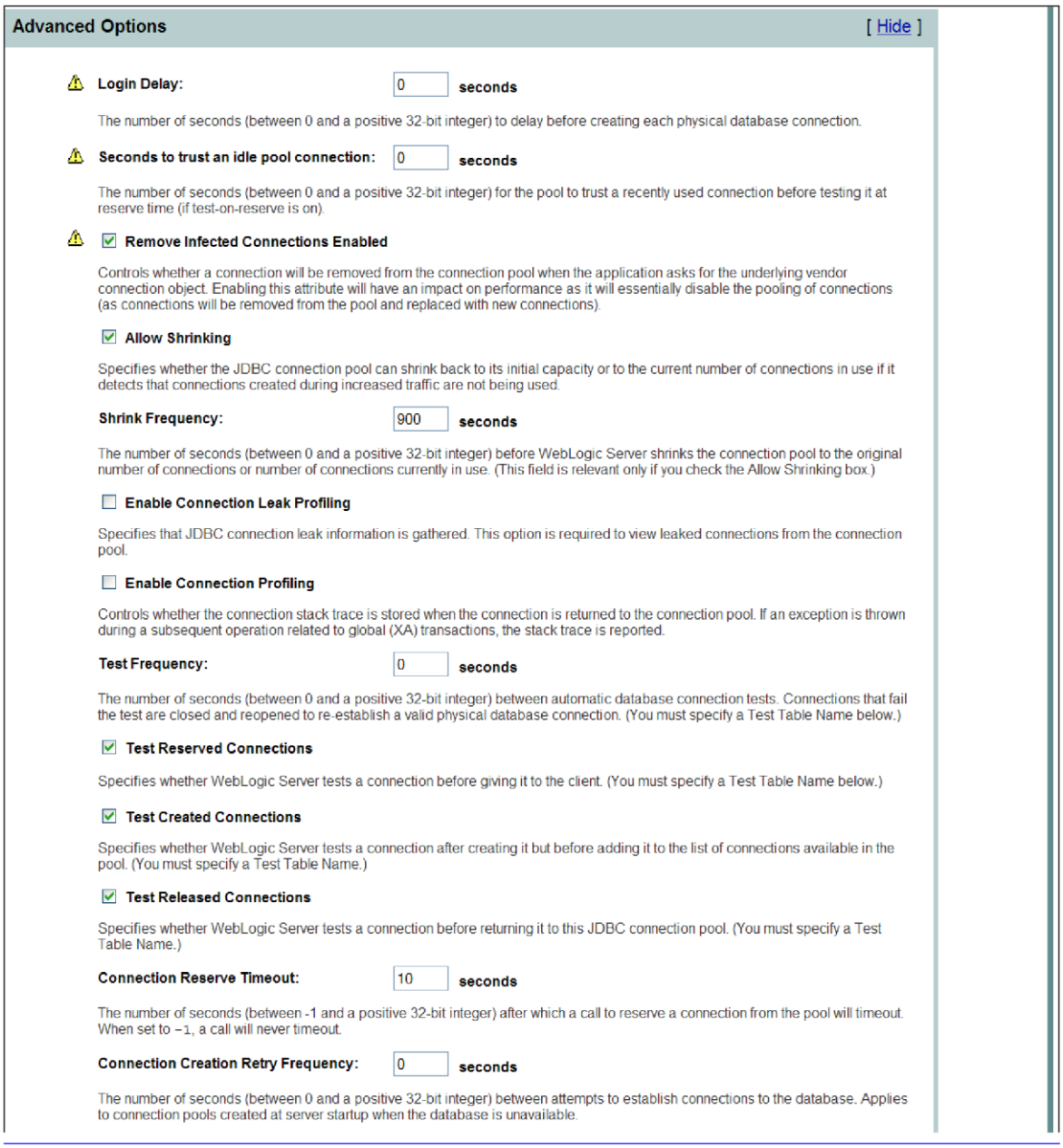


Figure 6-8. Connection page

c. Click the **Show** link to the right of **Advanced Options**.
The Advanced Options page opens.


Advanced Options
[Hide]
Login Delay:
0
seconds
The number of seconds (between 0 and a positive 32-bit integer) to delay before creating each physical database connection.
Seconds to trust an idle pool connection:
0
seconds
The number of seconds (between 0 and a positive 32-bit integer) for the pool to trust a recently used connection before testing it at reserve time (if test-on-reserve is on).
Remove Infected Connections Enabled
Controls whether a connection will be removed from the connection pool when the application asks for the underlying vendor connection object. Enabling this attribute will have an impact on performance as it will essentially disable the pooling of connections (as connections will be removed from the pool and replaced with new connections).
Allow Shrinking
Specifies whether the JDBC connection pool can shrink back to its initial capacity or to the current number of connections in use if it detects that connections created during increased traffic are not being used.
Shrink Frequency:
900
seconds
The number of seconds (between 0 and a positive 32-bit integer) before WebLogic Server shrinks the connection pool to the original number of connections or number of connections currently in use. (This field is relevant only if you check the Allow Shrinking box.)
Enable Connection Leak Profiling
Specifies that JDBC connection leak information is gathered. This option is required to view leaked connections from the connection pool.
Enable Connection Profiling
Controls whether the connection stack trace is stored when the connection is returned to the connection pool. If an exception is thrown during a subsequent operation related to global (XA) transactions, the stack trace is reported.
Test Frequency:
0
seconds
The number of seconds (between 0 and a positive 32-bit integer) between automatic database connection tests. Connections that fail the test are closed and reopened to re-establish a valid physical database connection. (You must specify a Test Table Name below.)
Test Reserved Connections
Specifies whether WebLogic Server tests a connection before giving it to the client. (You must specify a Test Table Name below.)
Test Created Connections
Specifies whether WebLogic Server tests a connection after creating it but before adding it to the list of connections available in the pool. (You must specify a Test Table Name.)
Test Released Connections
Specifies whether WebLogic Server tests a connection before returning it to this JDBC connection pool. (You must specify a Test Table Name.)
Connection Reserve Timeout:
10
seconds
The number of seconds (between -1 and a positive 32-bit integer) after which a call to reserve a connection from the pool will timeout. When set to -1, a call will never timeout.
Connection Creation Retry Frequency:
0
seconds
The number of seconds (between 0 and a positive 32-bit integer) between attempts to establish connections to the database. Applies to connection pools created at server startup when the database is unavailable.

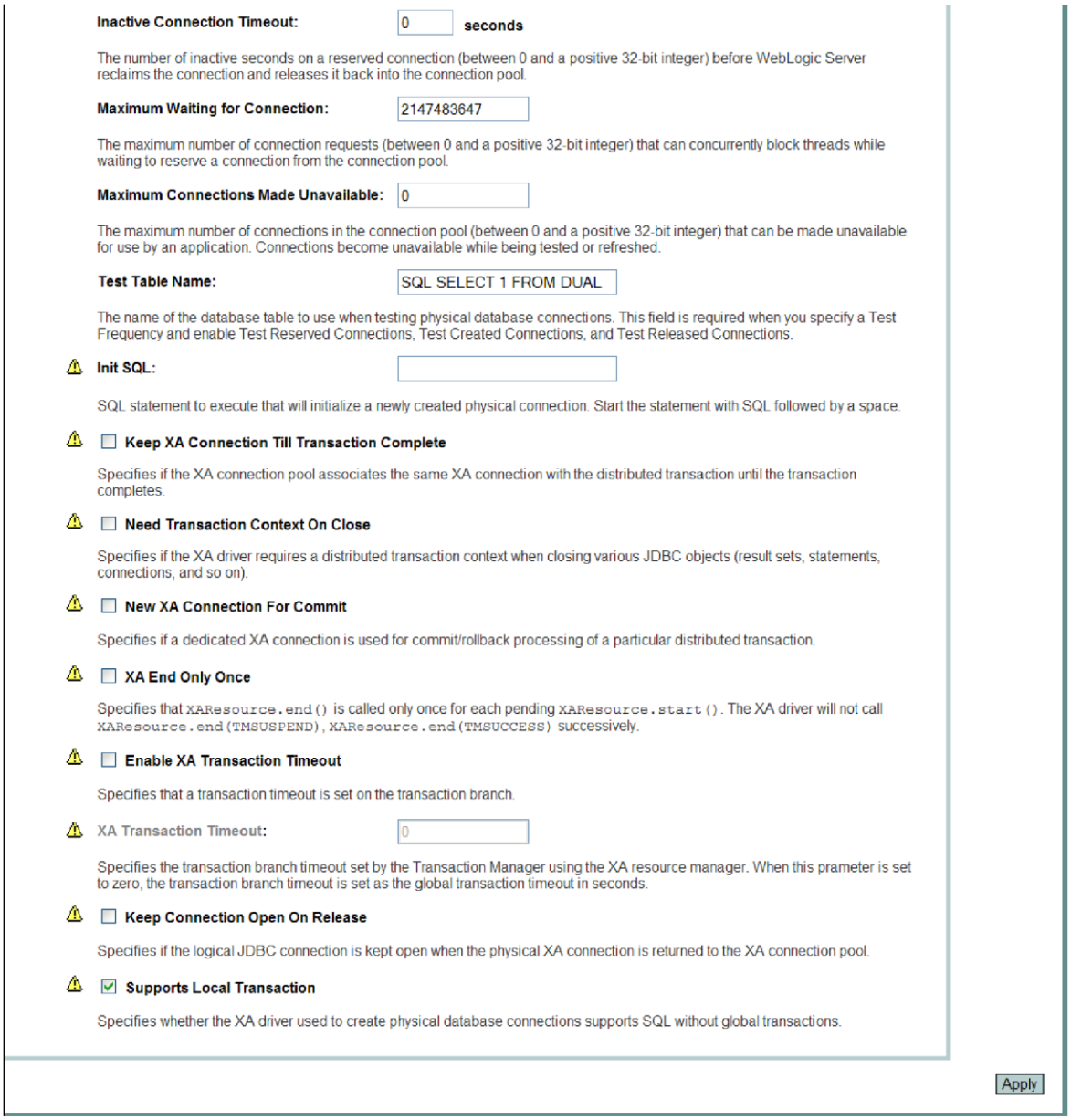


Figure 6-9. Advanced Options page

12. Check to select all of the following advanced features:
    a. **Test Reserved Connections**
    b. **Test Created Connections**
    c. **Test Released Connections**
    d. **Supports Local Transaction**.
13. Click **Apply**.
    The advanced options that you selected are applied.

6.3 Creating a Data Source for the Connection Pool

Use the WebLogic Server Administration Console to create a data source for the JDBC connection pool for VPFS.

1. In the navigation tree on the left of the console, expand **Services** > **JDBC**.
2. Click the **Data Source** link.
   The JDBC Data Sources page opens.

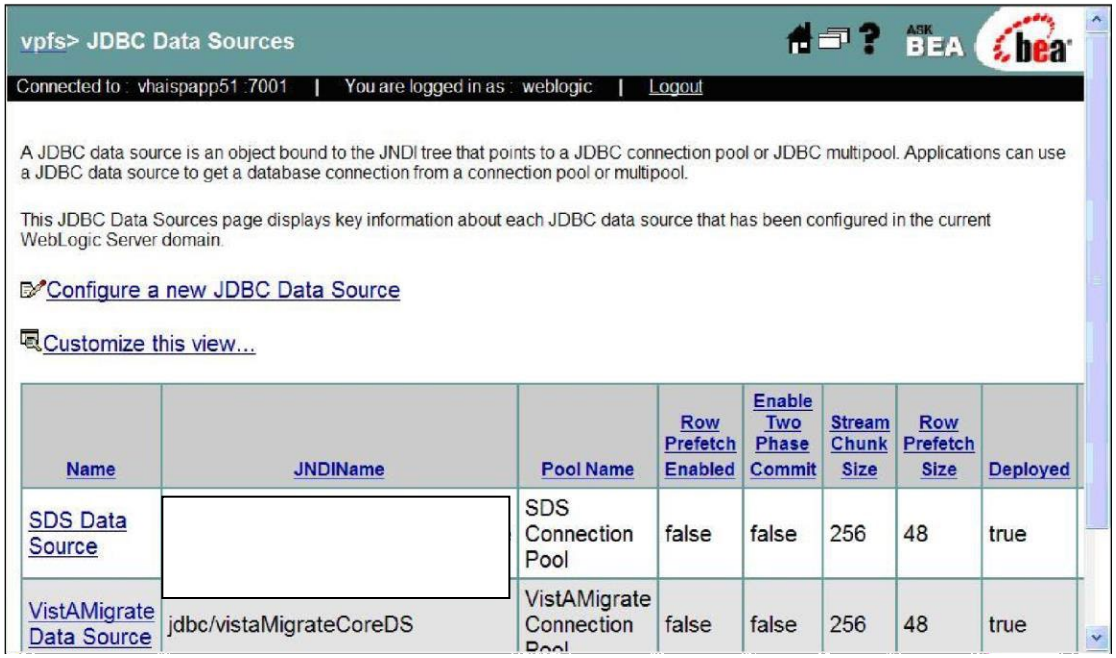


Figure 6-10. JDBC Data Sources page

3. Click the **Configure a new JDBC Data Source** link. The Configure the data source page opens.

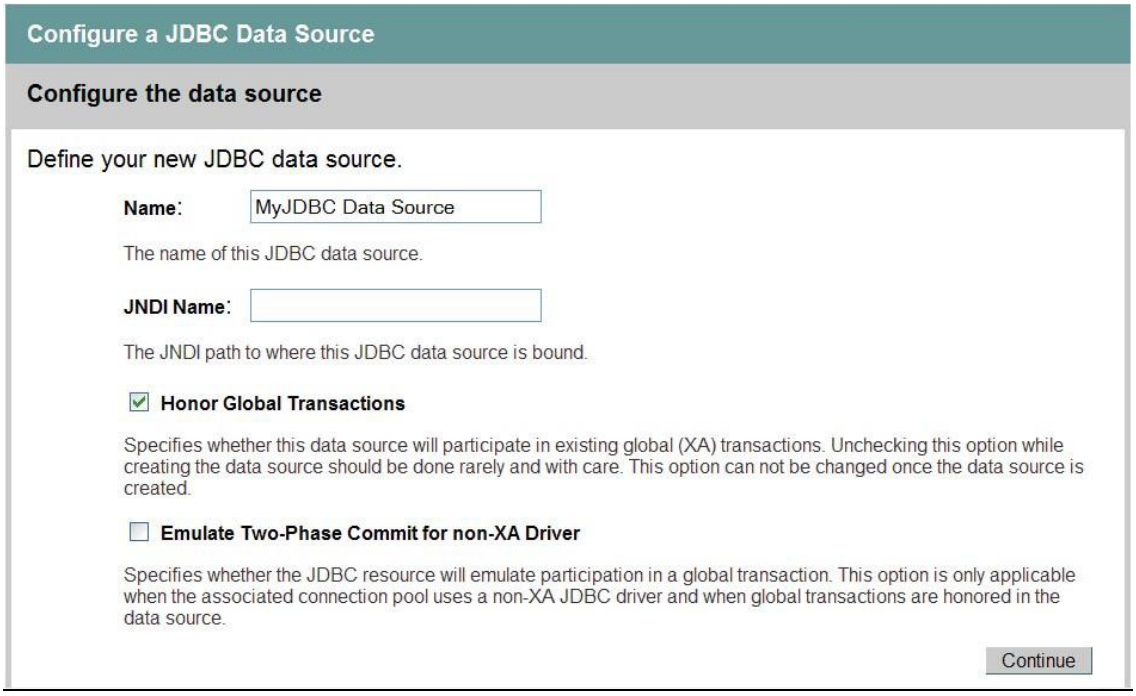


Figure 6-11. Configure the data source page

4. Define the new JDBC data source:
   a. **Name**: Enter the VPFS data source name.
   b. **JNDI Name**: Enter **jdbc/vpfsCoreDS**. (Required by VPFS code.)
5. Leave the default settings for the rest of the fields.
6. Click **Continue**.
7. Connect to connection pool:
   a. For the Pool Name, select the JDBC connection pool created for VPFS.
   b. Click **Continue**.
   The Connect to connection pool page opens.

Figure 6-12. Connect to connection pool page

8. Select the **Pool Name** for the VPFS pool you created.
9. Click **Continue**.
   The Target the data source page opens.

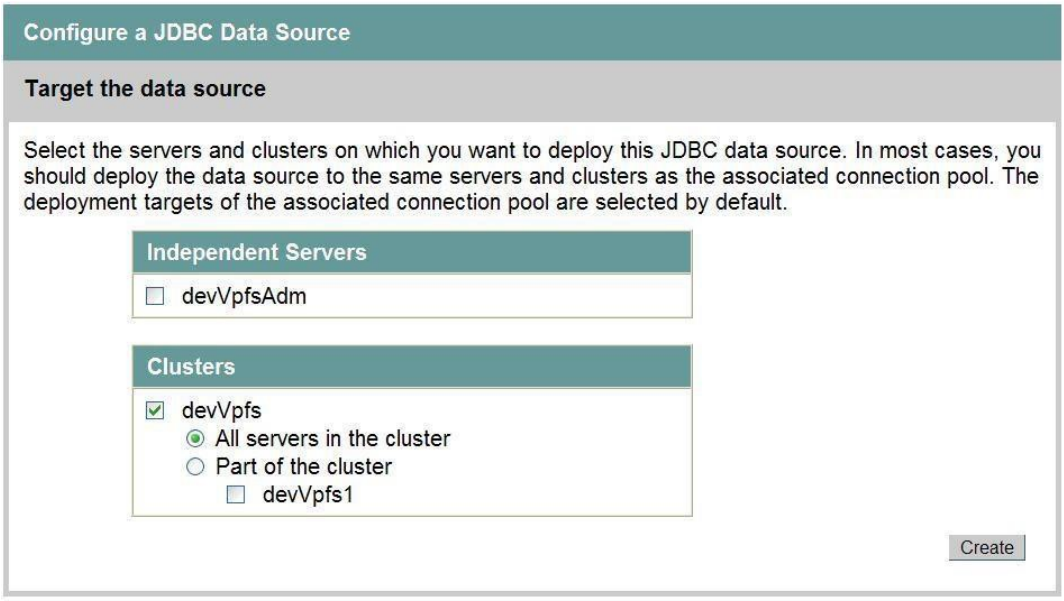


Figure 6-13. Target the data source page

10. Target the data source:
    a. Select the radio button next to **All servers in the cluster** to target the data source to the cluster.
    b. Click **Create**.
       The Configuration page opens. The new JDBC Data Source you created is listed in the table.

Name	JNDIName	Pool Name	Row Prefetch Enabled	Enable Two Phase Commit	Stream Chunk Size	Row Prefetch Size	Deployed	
MyJDBC Data Source	jdbc/vpfsCoreDS	MyJDBC Connection Pool	false	false	256	48	true	
SDS Data Source	gov.va.med.term.access.Database	SDS Connection Pool	false	false	256	48	true	
VistAMigrate Data Source	jdbc/vistaMigrateCoreDS	VistAMigrate Connection Pool	false	false	256	48	true	
VPFS Data Source	jdbc/vpfsCoreDS	VPFS Connection Pool	false	false	256	48	true	

Figure 6-14. Configuration page table

6.4 Deploying the VPFS Application

This section presents the steps necessary to deploy the application to the Application Server.

6.4.1 Before You Begin

Before you can display VPFS on the application server, you must:

- Update the PatientLookup.properties file to specify the location of the **Person Service Lookup** EJBs (pslEJB_4.0.x.x.ear) and authentication credentials.
- Update the PatSvcPkg.properties file to specify the location of the **Person Service Construct** EJBs (PatientServiceR2.ear) and authentication credentials.
- Update the kaajeeConfig.xml file to contain the station numbers for the VistA systems you need to access.
- Update the KAAJEE_CONTROLLER role in the weblogic.xml file with the WebLogic boot username.

To update the application configuration files:

The files that need to be edited are located in the VPFS .ear file. To edit them, you will need to extract the contents of the .ear file to a temporary directory. Be sure to preserve the directory structure when extracting the contents of the .ear file.

Note: An .ear file is basically a .zip file with a specific file and directory structure. You can use an application such as File Roller, WinZip, or the jar utility to work with .ear files as you would with a .zip file.

---

1. Create a temporary directory to extract the contents of the .ear file to. Copy the VPFS .ear file into this directory.

   ```
   # mkdir vpfstemp
   ```

2. Change directory to the temporary directory. Extract the VPFS .ear file to this directory.

   ```
   # cd vpfstemp
   ```

   ```
   # jar xf vpfs.ear
   ```

3. Update the PatientLookup.properties file in VPFS/WEB-INF/classes to specify the location of the Person Service Lookup EJBs (pslEJB_4.0.x.x.ear) and authentication credentials.

   - Set the `providerURL` property to the host and port of the server where the Person Service Lookup EJBs are deployed.
   - Set the `securityPrincipal` and `securityCredentials` properties to the username and password of the WebLogic console user or administrator.

   Refer to the Person Service Lookup documentation for detailed instructions as needed.

4. Update the PatSvcPkg.properties file in VPFS/WEB-INF/classes to specify the location of the Person Service Construct EJBs (PatientServiceR2.ear) and authentication credentials.

   - Set the `providerURL` property to the host and port of the server where the Person Service Lookup EJBs are deployed.
   - Set the `securityPrincipal` and `securityCredentials` properties to the username and password of the WebLogic console user or administrator.

   Refer to the Person Service Construct documentation for detailed instructions as needed.

5. To change the STS service address (e.g. from DEV to SQA), open the kaajeeConfig.xml file and edit the <sts-service-address> element with the appropriate value.

6. Update the kaajeeConfig.xml file in VPFS/WEB-INF/config to contain the appropriate `<station-number>` entries for the VistA systems you need to access. Ensure the appropriate VistaLink connector has been created for each station in this list. See VistaLink documentation for instructions on how to define a new connector.

   - Add or edit `<station-number>` entries as needed. There must be one entry per station number you need to access.

     ```
     <station-number>500</station-number>
     ```

   Refer to the KAAJEE Installation Guide for detailed instructions as needed.

7. Update the weblogic.xml file in VPFS/WEB-INF to set the KAAJEE_CONTROLLER role principal name to the username of a WebLogic Administrator user (or the WebLogic boot user). KAAJEE requires this to function correctly.

   - Look for a `<security-role-assignment>` with a `<role-name>` of KAAJEE_CONTROLLER. Change the `<principal-name>` to the name of a WebLogic Administrator user.

     ```
     <principal-name>weblogic</principal-name>
     ```

   - Look for a `<run-as-role-assignment>` with a `<role-name>` of KAAJEE_CONTROLLER. Change the `<run-as-principal-name>` to the name of a WebLogic Administrator user.

     ```
     <run-as-principal-name>weblogic</run-as-principal-name>
     ```

8. (Optional) Context-root „**vpfs'** in the META-INF/application.xml is the default context root URL for the VPFS application. This entry in the deployment descriptor can be updated appropriately based on the server URL requirements.

9. (Optional) If you are going to deploy VPFS as an .ear file, freshen the VPFS .ear file to update the changed files.

   ```
   # jar uf vpfs.ear *
   ```

6.4.2 Steps to Deploy

VPFS can be deployed as either an .ear file or an "exploded" deployment. To use the exploded deployment option, the contents of the temporary directory used above must be copied to the server. To use the .ear file deployment option, the vpfs.ear file may either be on the server or uploaded during deployment.

Complete these steps to deploy the application:

1. Using a web browser, connect to the WebLogic Server console.
2. Sign in as the administration user and supply the appropriate password.
3. From the left panel, expand the **Deployments** node and click the **Applications** link.
4. From the main panel, click the **Deploy a new Application…** link.
5. Select the archive for this application. Do **one** of the following:
   a. Navigate to the location of the vpfs.ear file or root directory of the exploded .ear file.

6. Click the upload your file(s) link.

   Click **Browse…** button and navigate to where the VPFS .ear file is located. Select the VPFS .ear file and click the **Open** button.

   Click **Upload**.

7. Select the VPFS application (.ear file or root directory) and click **Target Application**.
8. Select targets for this application:
   a. Click the radio button next to the **All servers in the cluster** to target the application to the cluster.
   b. Click Continue.
9. Review your choices and click **Deploy**.

   Deployment status: During the process of the deployment, the screen will refresh itself periodically until the application is deployed.

6.5 Logging

Logging for VPFS is configured in the log4j configuration file, log4j.xml. This file must be specified for each server using the JVM argument `-Dlog4j.configuration` (example: `Dlog4j.configuration=file:///u01/app/bea/user_projects/domains/domVPFS/log4j.xml`). There should only be a single log4j configuration file in use for any server. You can edit this file to change the default logging level or log file location if needed.

Log4j is configured by defining `<appender>`s and `<logger>`s. The default appenders and loggers for VPFS are defined in the log4j_VPFS.xml file supplied with the VPFS distribution package. These can be copied into the correct locations in an existing log4j configuration file. A sample log4j configuration file (log4j_sample.xml) is also provided that also includes sample appenders and loggers for our dependencies.

The default VPFS log file is called vpfs.log. To reduce the size of the log file and facilitate the job of the system admin, the log file is configured to automatically rollover every day. For example, at midnight of every day, the current vpfs.log will be rolled over to vpfs.log.yyyy-MM-dd (yyyy: 4-digit year, MM: 2digit month, and dd: 2-digit day) and a new vpfs.log will be created to log the new day's events.

The default configuration creates this file in the `log` directory. To create the VPFS logging directory:

1. Log in to one of the machines in the cluster VPFS was deployed to.
2. Create a directory named "log". As this is a common logging directory for the server, it may already exist in which case you do not need to try to create it again.
   - If VPFS is deployed on a managed server, create the directory under the node manager directory.

     # **cd <BEA_HOME>/weblogic81/common/nodemanager**

     # **mkdir log**

   - If VPFS is deployed on the domain admin server, create the directory in the root directory for the domain..

     # **cd <BEA_HOME>/user_projects/domains/vpfsDomain**

     # **mkdir log**

3. Repeat these steps as needed to create the log folder on each machine in the cluster.

7. VPFS VistAMigrate Installation

Note: Migrated VistA data will be stored in the staging environment on the hardware where VistAMigrate is deployed and that data contains sensitive patient information that must be kept secure. To provide maximum security, consider installing and using VistAMigrate on a computer that can be physically secured in a locked cabinet or secure room and powering off or disconnecting the computer from the network when not in use.

To install the VistAMigrate application, perform the following tasks from the WebLogic Server Administration Console:

1. Create a JDBC connection pool for VistMigrate.
2. Create a Data Source for the VistAMigrate Connection Pool.
3. Deploy the VistAMigrate application.
4. Create the VistAMigrate logging directory.

7.1 Logging in to the WebLogic Server Administration Console

To log in to the WebLogic Server Administration Console:

1. Open a web browser and go to **`http://`***`<host>`*`:`***`/console/`** The WebLogic Server Administration Console sign in page opens.

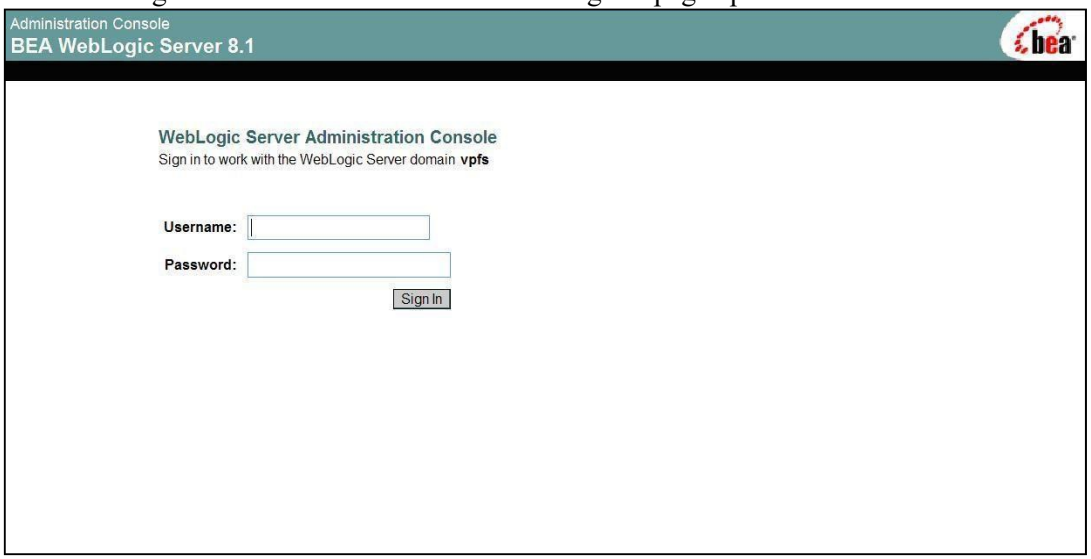


Figure 7-1. WebLogic Server Administration Console Sign In Page

2. Log in as the administration user:
   a. **Username**: Enter your administration username.
   b. **Password**: Enter your administration password.

7.2 Creating a JDBC Connection Pool for VistAMigrate

VPFS connects to the Oracle database through WebLogic using JDBC connection pools. JDBC connection pools manage resources by holding connections to the database open until they are needed by HealtheVet VistA applications.

Use the WebLogic Server Administration Console to create a JDBC connection pool for VistAMigrate.

1. In the navigation tree on the left of the console, expand **Services** > **JDBC**.
2. Click the **Connection Pools** link.
   The JDBC Connection Pool Configuration page displays the list of connection pools that already exist.

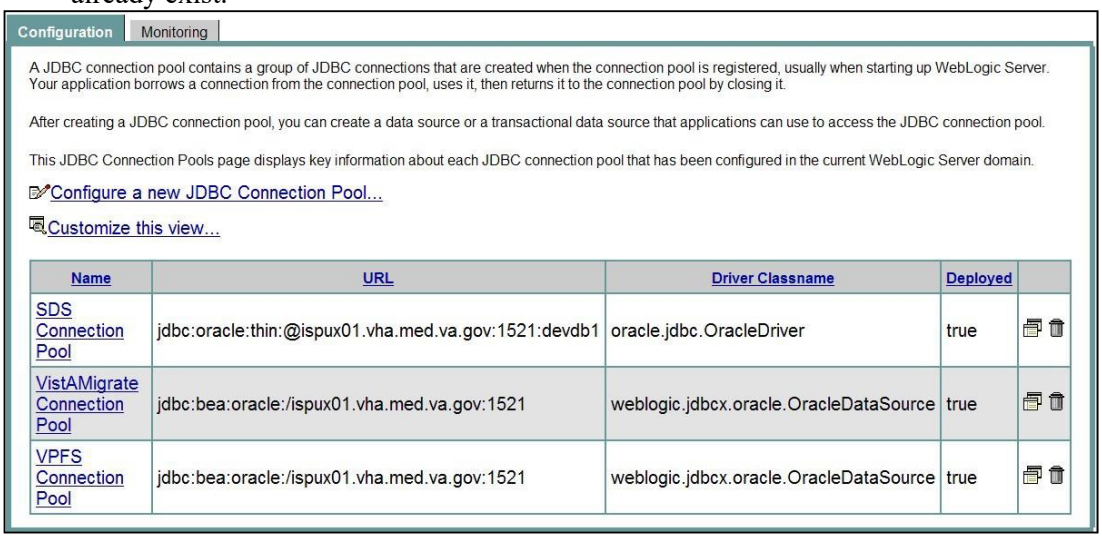


Figure 7-2. JDBC Connection Pool Configuration page

3. From the main panel, click the **Configure a new JDBC Connection Pool** link. The Choose Database page opens.

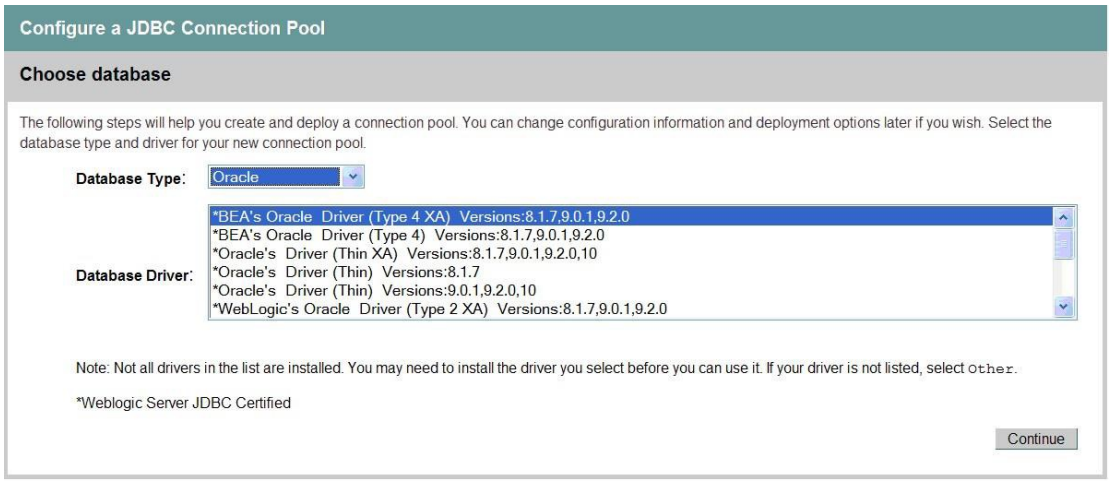


Figure 7-3. Choose Database page

4. Select the database type and driver:
   a. **Database Type**: Select Oracle.
   b. **Database Driver**: Select *Oracle's Driver (Thin XA) Versions: 8.1.7,9.0.1,9.2.0,10

5. Click **Continue**.
   The Define Connection Properties page opens.

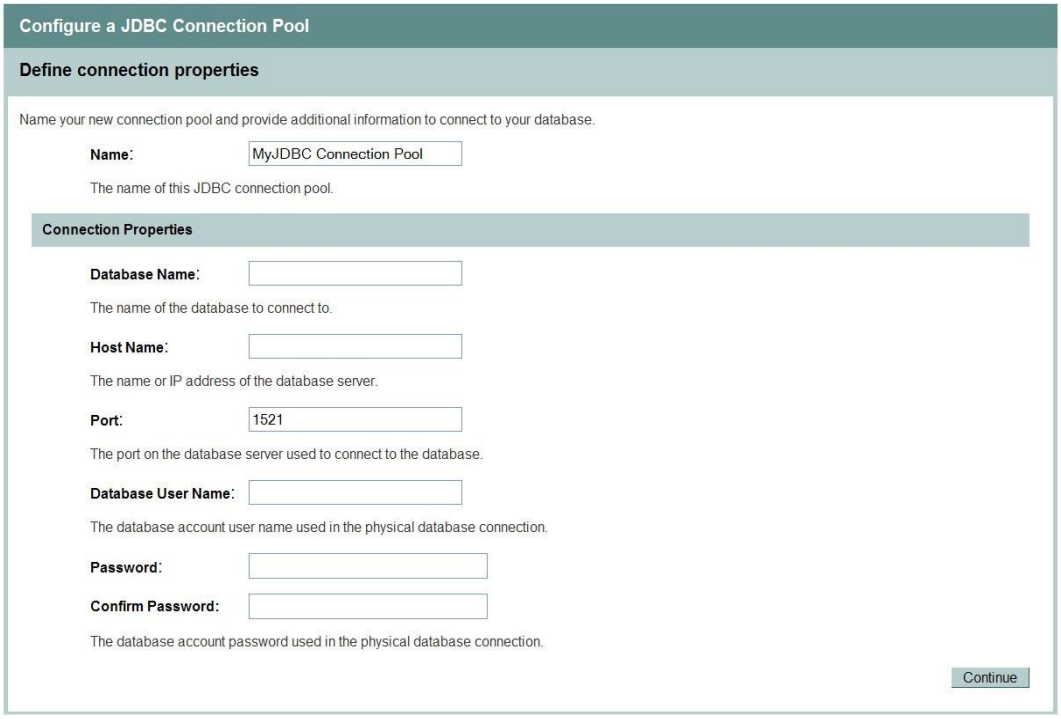


Figure 7-4. Define Connection Properties page

6. Enter Connection Properties:
   a. **Name**: Enter the name of the VistAMigrate JDBC connection pool you are defining.
   b. **Database Name**: Enter the name of the database (or service name) to connect to.
   c. **Host Name**: Enter the name or IP address of the host machine.
   d. **Port**: Enter the port of the database server used to connect to the database. The Oracle default port is 1521.
   e. **Database User Name**: Enter your database username.
   f. **Password**: Enter your password.
   g. **Confirm Password**: Enter your password again to confirm.
7. Click **Continue**.
8. To test the database connection, click **Test Driver Configuration**.
   - If the configuration is correct, the Create and Deploy page opens. Continue with step
   - 16.

   If an error message appears at the top of the page, correct the error before you continue.

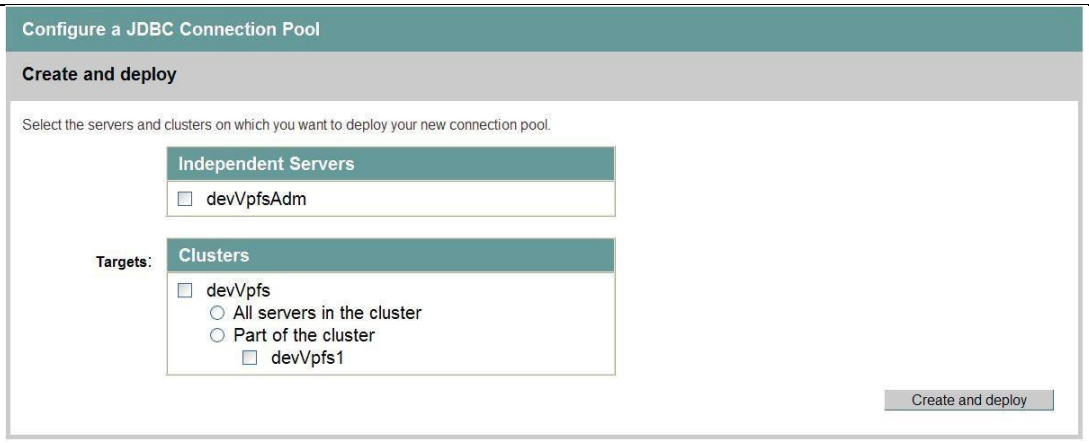


Figure 7-5. Create and Deploy page

9. Select the radio button for **All servers in the cluster** to target the connection pool to the cluster.
10. Click **Create and deploy**.
    Once created successfully, the JDBC Connection Pools configuration page appears. The new JDBC Connection Pool is added to the table on the JDBC Connection Pools configuration page.

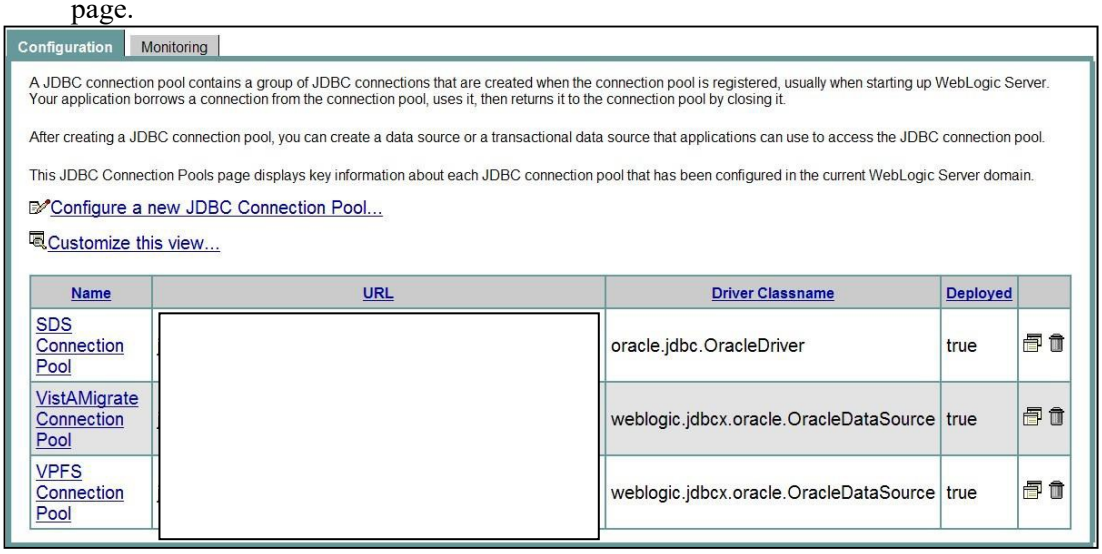


Figure 7-6. JDBC Connection Pools configuration page

11. Set the Advanced Options:
    a. Click the name of the JDBC Connection Pool that you created. The General page opens for the connection pool you requested.

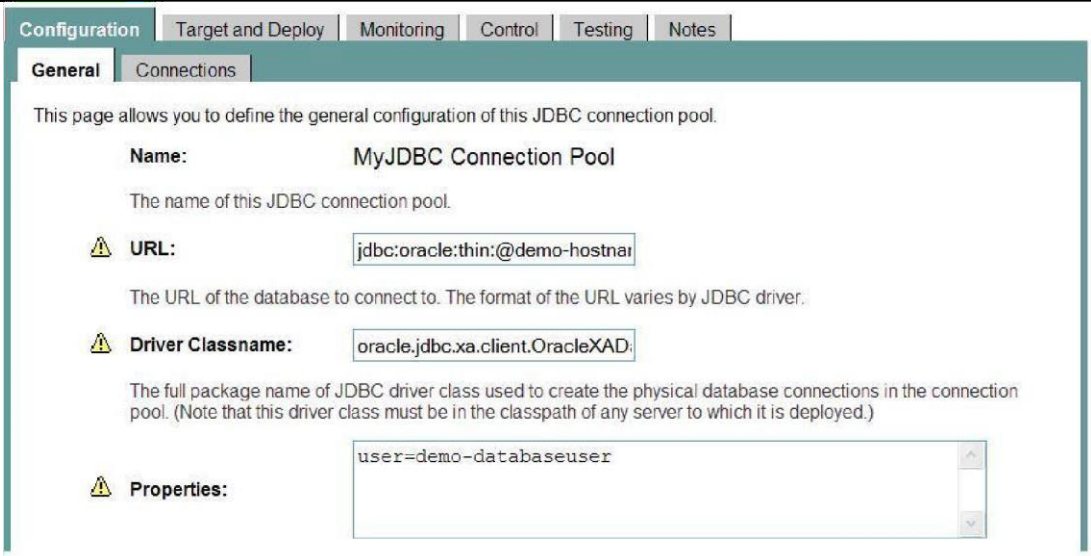


Figure 7-7. General page for the connection pool

b. Click the **Connections** tab.
The Connection page opens.

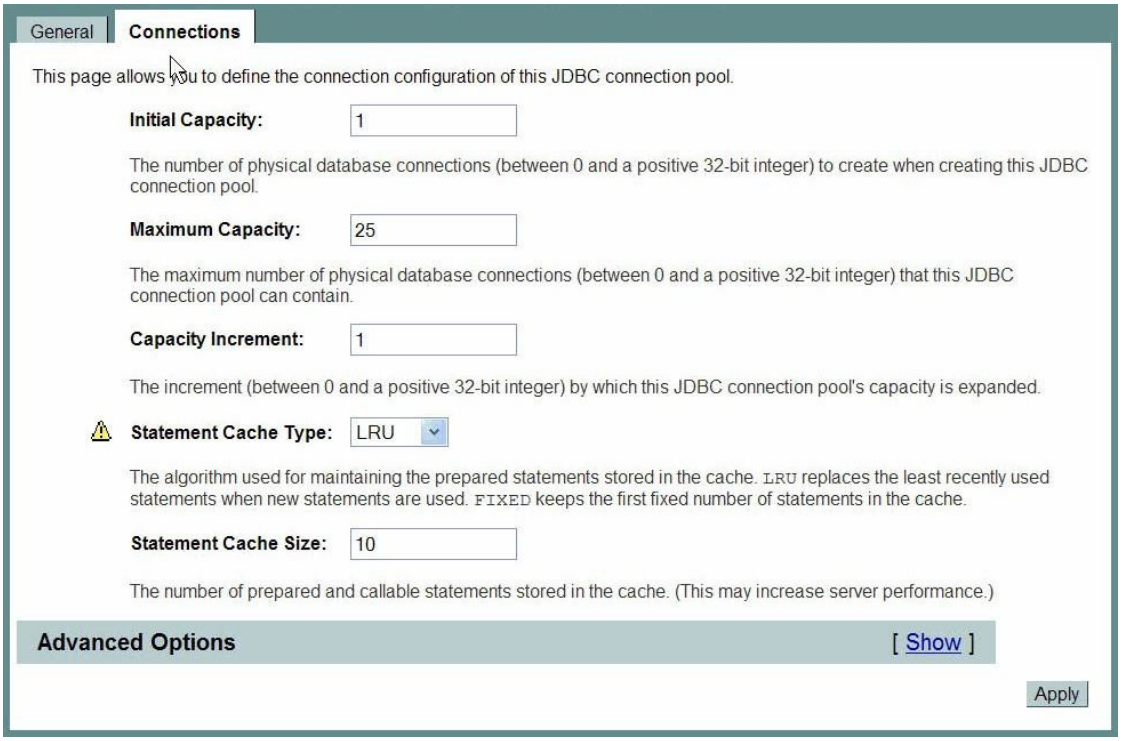
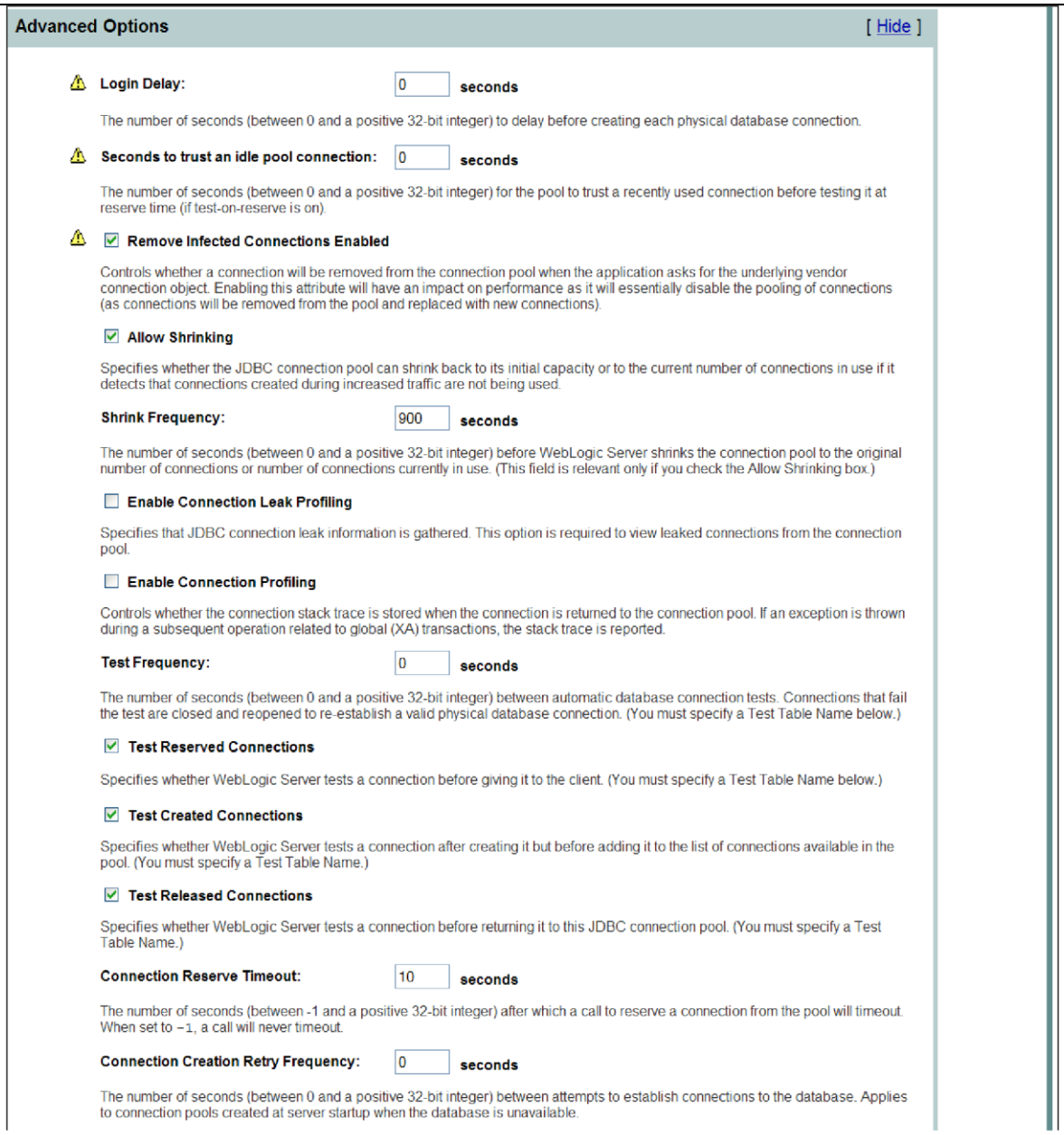


Figure 7-8. Connection page

c. Click the **Show** link to the right of **Advanced Options**.
The Advanced Options page opens.


Advanced Options
[Hide]
Login Delay: 0 seconds
The number of seconds (between 0 and a positive 32-bit integer) to delay before creating each physical database connection.
Seconds to trust an idle pool connection: 0 seconds
The number of seconds (between 0 and a positive 32-bit integer) for the pool to trust a recently used connection before testing it at reserve time (if test-on-reserve is on).
Remove Infected Connections Enabled
Controls whether a connection will be removed from the connection pool when the application asks for the underlying vendor connection object. Enabling this attribute will have an impact on performance as it will essentially disable the pooling of connections (as connections will be removed from the pool and replaced with new connections).
Allow Shrinking
Specifies whether the JDBC connection pool can shrink back to its initial capacity or to the current number of connections in use if it detects that connections created during increased traffic are not being used.
Shrink Frequency: 900 seconds
The number of seconds (between 0 and a positive 32-bit integer) before WebLogic Server shrinks the connection pool to the original number of connections or number of connections currently in use. (This field is relevant only if you check the Allow Shrinking box.)
Enable Connection Leak Profiling
Specifies that JDBC connection leak information is gathered. This option is required to view leaked connections from the connection pool.
Enable Connection Profiling
Controls whether the connection stack trace is stored when the connection is returned to the connection pool. If an exception is thrown during a subsequent operation related to global (XA) transactions, the stack trace is reported.
Test Frequency: 0 seconds
The number of seconds (between 0 and a positive 32-bit integer) between automatic database connection tests. Connections that fail the test are closed and reopened to re-establish a valid physical database connection. (You must specify a Test Table Name below.)
Test Reserved Connections
Specifies whether WebLogic Server tests a connection before giving it to the client. (You must specify a Test Table Name below.)
Test Created Connections
Specifies whether WebLogic Server tests a connection after creating it but before adding it to the list of connections available in the pool. (You must specify a Test Table Name.)
Test Released Connections
Specifies whether WebLogic Server tests a connection before returning it to this JDBC connection pool. (You must specify a Test Table Name.)
Connection Reserve Timeout: 10 seconds
The number of seconds (between -1 and a positive 32-bit integer) after which a call to reserve a connection from the pool will timeout. When set to -1, a call will never timeout.
Connection Creation Retry Frequency: 0 seconds
The number of seconds (between 0 and a positive 32-bit integer) between attempts to establish connections to the database. Applies to connection pools created at server startup when the database is unavailable.

7.

Figure 7-9. Advanced Options page

12. Check all of the following advanced features:
    a. **Test Reserved Connections**
    b. **Test Created Connections**
    c. **Test Released Connections**
    d. **Supports Local Transaction**.
13. Click **Apply**.
    The advanced options that you selected are applied.

7.3 Creating a Data Source for the Connection Pool

Use the WebLogic Server Administration Console to create a data source for the JDBC connection pool for VistAMigrate.

1. Click the **Data Source** link.
   The JDBC Data Sources page opens.

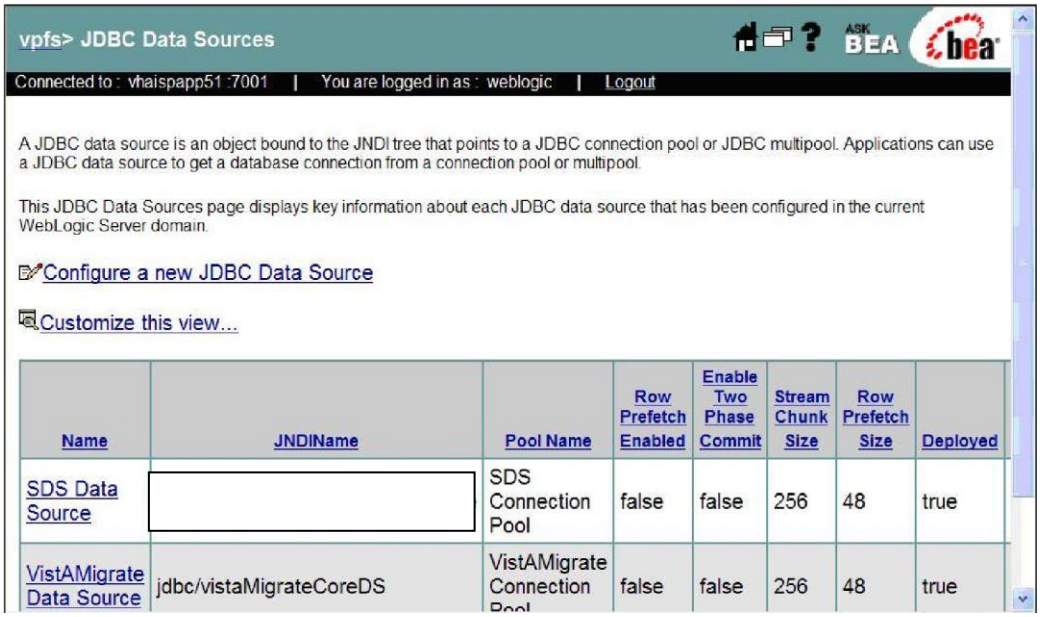


Figure 7-10. JDBC Data Sources page

2. Click the **Configure a new JDBC Data Source** link.
   The Configure the data source page opens.

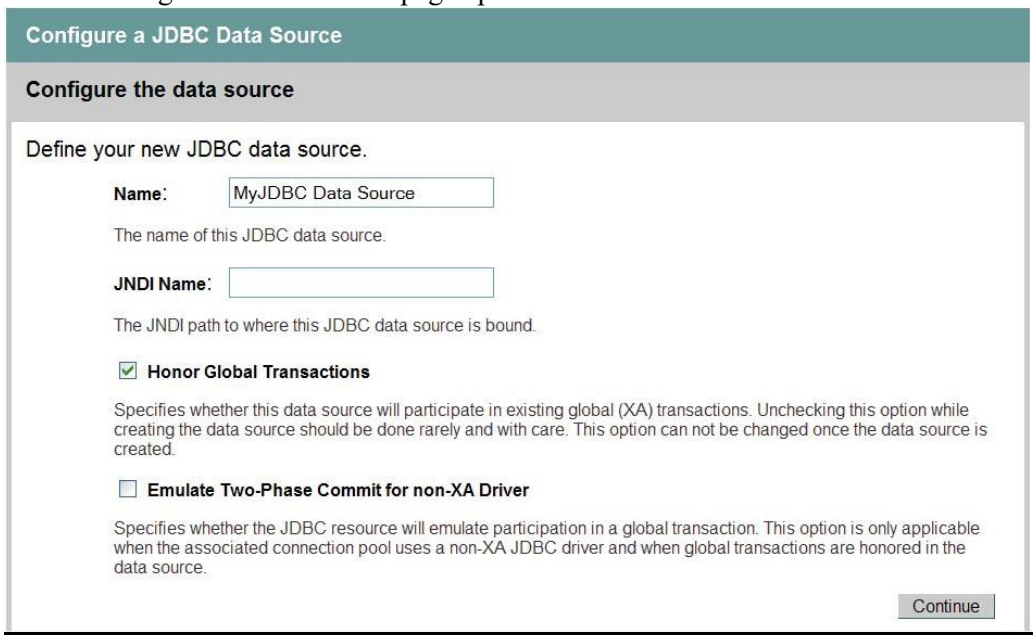


Figure 7-11. Configure the data source page

3. Define the new JDBC data source:
   a. **Name**: Enter the VPFS data source name.
   b. **JNDI Name**: Enter **jdbc/vistaMigrateCoreDS**. (Required by VPFS code.)
4. Leave the default settings for the rest of the fields.
5. Click **Continue**.
6. Connect to connection pool:
   a. For the Pool Name, select the JDBC connection pool created for VistAMigrate.
   b. Click **Continue**.
   The Connect to connection pool page opens.

Figure 7-12. Connect to connection pool page

7. Select the **Pool Name** for the VistAMigrate pool you created.
8. Click **Continue**.
The Target the data source page opens.

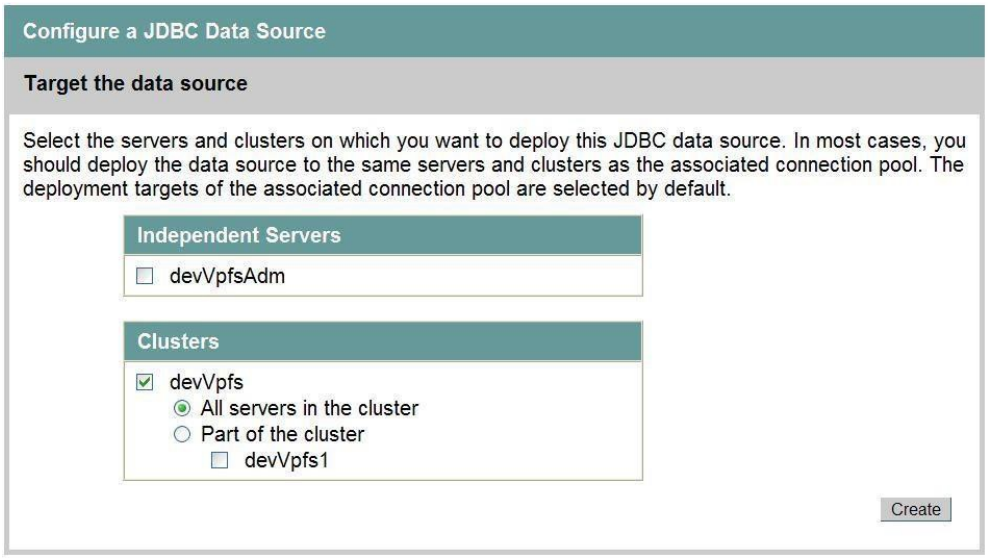


Figure 7-13. Target the data source page

9. Target the data source:
   a. Select the radio button next to the **All servers in the cluster** to target the data source to the cluster.
   b. Click **Create**.
      The Configuration page opens. The new JDBC Data Source you created is listed in the table.

Name	JNDIName	Pool Name	Row Prefetch Enabled	Enable Two Phase Commit	Stream Chunk Size	Row Prefetch Size	Deployed	
MyJDBC Data Source	jdbc/vpfsCoreDS	MyJDBC Connection Pool	false	false	256	48	true	
SDS Data Source		SDS Connection Pool	false	false	256	48	true	
VistAMigrate Data Source	jdbc/vistaMigrateCoreDS	VistAMigrate Connection Pool	false	false	256	48	true	
VPFS Data Source	jdbc/vpfsCoreDS	VPFS Connection Pool	false	false	256	48	true	

Figure 7-14. Configuration page table

7.4 Root and Code Folders Setup

This section presents the steps necessary to set up the root folder and code folder on the machine where VistAMigrate is deployed.

SQL*Loader and SQL*Plus must be installed for VistAMigrate to run. Install the Oracle client tools to install SQL*Loader and SQL*Plus.

1. Create a root folder where VistAMigrate will have full access permission (read, write, and execute). For example, /home/bea/**vistaMigrate**.

   ```
   # mkdir vistaMigrate
   ```

2. Create a VPFS project folder under the root folder.

   VistAMigrate requires a strict file hierarchy to operate. Each project (in this case, the migration for VPFS) requires a folder under the VistAMigrate folder created in the above step. For VPFS migration, create a VPFS folder under the root folder. For example, /home/bea/vistaMigrate/**VPFS**.

   ```
   # cd vistaMigrate
   ```

   ```
   # mkdir VPFS
   ```

3. Create a folder called CODE under the project folder. For example, /home/bea/vistaMigrate/VPFS/**CODE**.

   ```
   # cd VPFS
   ```

   ```
   # mkdir CODE
   ```

4. Copy the following files into the CODE folder from the distribution:

- setenv (Linux) or setenv.bat (Windows)
- runsqlldr.sh / runsqlldr.bat
- runsqlplus.sh / runsqlplus.bat
- vpfs_load_temp.ctl
- vpfs_analyze_data.sql
- vpfs_extract_errors.sql
- vpfs_verify_data.sql

5. Edit the setenv script to change the VM_ORACLE_HOME variable to point to the Oracle home where the Oracle client tools have been installed.

   Example: **VM_ORACLE_HOME='/home/oracle/product/10.1.0/OraClient_1'**

6. Start SQL*Plus (if not already running) and log on to the Oracle database as the VistAMigrate user.

   ```
   SQL> connect vista_migrate/vista_migrate@databasename
   ```

7. Update the username, password, and data source name to connect to the VPFS database. (Replace vpfs, vpfspassword, and vpfsdatabase as appropriate.)

   ```
   SQL> update dm_setup set db_user_nm='vpfs',
   db_user_pass_txt='vpfspassword', db_data_src_txt='vpfsdatabase'
   where project_nm='VPFS';

   SQL> commit;
   ```

7.5 Deploying the VistAMigrate Application

7.5.1 Before You Begin

Before you can display VPFS on the application server, you must:

- Update the kaajeeConfig.xml file to contain the station numbers for the VistA systems you need to access.
- Update the KAAJEE_CONTROLLER role in the weblogic.xml file with the WebLogic boot username.
- Update the VistaMigrateResources.properties file to set the location of the migration root folder.

The files that need to be edited are located in the VistAMigrate .ear file. To edit them, you will need to extract the contents of the .ear file to a temporary directory. Be sure to preserve the directory structure when extracting the contents of the .ear file.

1. Create a temporary directory to extract the contents of the .ear file to. Copy the VistAMigrate .ear file into this directory.

   ```
   # mkdir vmtemp
   ```

2. Change directory to the temporary directory. Extract the VistAMigrate .ear file to this directory.

   ```
   # cd vmtemp

   # jar xf vistaMigrate.ear
   ```

3. Update the kaajeeConfig.xml file in VistAMigrate/WEB-INF/config to contain the appropriate `<station-number>` entries for the VistA systems you need to access. Ensure the appropriate VistaLink connector has been created for each station in this list. See VistaLink documentation for instructions on how to define a new connector.

   - Add or edit `<station-number>` entries as needed. There must be one entry per station number you need to access.

     ```
     <station-number>500</station-number>
     ```

   Refer to the KAAJEE Installation Guide for detailed instructions as needed.

4. Update the weblogic.xml file in VistAMigrate/WEB-INF to set the KAAJEE_CONTROLLER role principal name to the username of a WebLogic Administrator user (or the WebLogic boot user). KAAJEE requires this to function correctly.

   - Look for a `<security-role-assignment>` with a `<role-name>` of KAAJEE_CONTROLLER. Change the `<principal-name>` to the name of a WebLogic Administrator user.

     ```
     <principal-name>weblogic</principal-name>
     ```

   - Look for a `<run-as-role-assignment>` with a `<role-name>` of KAAJEE_CONTROLLER. Change the `<run-as-principal-name>` to the name of a WebLogic Administrator user.

     ```
     <run-as-principal-name>weblogic</run-as-principal-name>
     ```

5. Update the VistaMigrateResources.properties file in VistAMigrate/WEB-INF/classes to specify the location of the migration root folder.

   - Set the `migrationRootPath` property to the path of the migration root folder created in section 7.4.

     ```
     migrationRootPath=/home/bea/vistaMigrate
     ```

Note: If you are running VistAMigrate on a Windows server, you need to use double backslashes in the migrationRootPath property. For example, if you have created the migration root directory at C:\vistaMigrate, this property would be:

```
migrationRootPath=C:\\vistaMigrate
```

---

6. (Optional) If you are going to deploy VistAMigrate as an .ear file, freshen the VistAMigrate .ear file to update the changed files.

   ```
   # jar uf vistaMigrate.ear *
   ```

7.5.2 Steps to Deploy

VistAMigrate can be deployed as either an .ear file or an "exploded" deployment. To use the exploded deployment option, the contents of the temporary directory used above must be copied

to the server. To use the .ear file deployment option, the vistaMigrate.ear file may either be on the server or uploaded during deployment.

The VistAMigrate application must be deployed to a standalone managed server instead of the cluster to access server specific resources.

Complete these steps to deploy the application:

1. Using a web browser, connect to the WebLogic Server console.
2. Sign in as the administration user and supply the appropriate password.
3. From the left panel, expand the **Deployments** node and click the **Applications** link.
4. From the main panel, click the **Deploy a new Application…** link.
5. Select the archive for this application. Do **one** of the following:
   a. Navigate to the location of the vistaMigrate.ear file or root directory of the exploded .ear file. (WebLogic will recognize the root directory of an exploded .ear file as an application.)
   b. Click the upload your file(s) link.

      Click **Browse…** button and navigate to where the VistAMigrate .ear file is located. Select the VistAMigrate .ear file and click the **Open** button.

      Click **Upload**.
6. Select the VistAMigrate application (.ear file or root directory) and click **Target Application**.
7. Select targets for this application:
   a. Click the radio button next to the server on which the application will be hosted. (Do *not* specify the cluster.)
   b. Click Continue.
8. Review your choices and click **Deploy**.

   Deployment status: During the process of the deployment, the screen will refresh itself periodically until the application is deployed.

7.6 Logging

Logging for VistAMigrate is configured in the log4j configuration file, log4j.xml. See section 6.5 for additional details.

Log4j is configured by defining `<appender>`s and `<logger>`s. The default appenders and loggers for VistAMigrate are defined in the log4j_VistAMigrate.xml file supplied with the VPFS distribution package. These can be copied into the correct locations in an existing log4j configuration file.

The default VistAMigrate log file is called vistaMigrate.log. See section 6.5 for additional details.

8. Post-Installation Steps

The interfaces that may require post-installation steps to integrate with VPFS are:

- VistALink
- KAAJEE

The post-installation steps for each interface are detailed in the following sections.

8.1 Post Interface Installation Steps

This section provides the steps that must be performed to integrate VPFS with the interfaces.

8.1.1 Connecting to VistA Sites

Ensure VistALink connectors have been created for the sites you need access to. VistALink requires an entry in the VistALink connectorConfig.xml for each VistA site to which you will connect to. Make sure to have the IP, port, access and verify codes for each site handy before you start adding new connections.

Be sure to include station numbers for any added VistA sites to the KAAJEE configuration file (kaajeeConfig.xml) also. This file is in VistAMigrate/WEB-INF/config for VistAMigrate and in VPFS/WEB-INF/config for VPFS.

9. Testing the VPFS Application Installation

9.1 Testing the Installation of the VPFS Application

1. Using a web browser, connect to the VPFS application through the proxy server or load balancer that points marshals' connections to the cluster.

   `http://[host]:[port]/vpfs`

2. After pressing *Sign In with VA PV Card*, you will be prompted to choose a certificate and for your PIV Pin. Upon pressing *Proceed*, you will be authenticated automatically.

3. Upon authentication with the system, you will see the VPFS welcome page, as shown in the following figure.

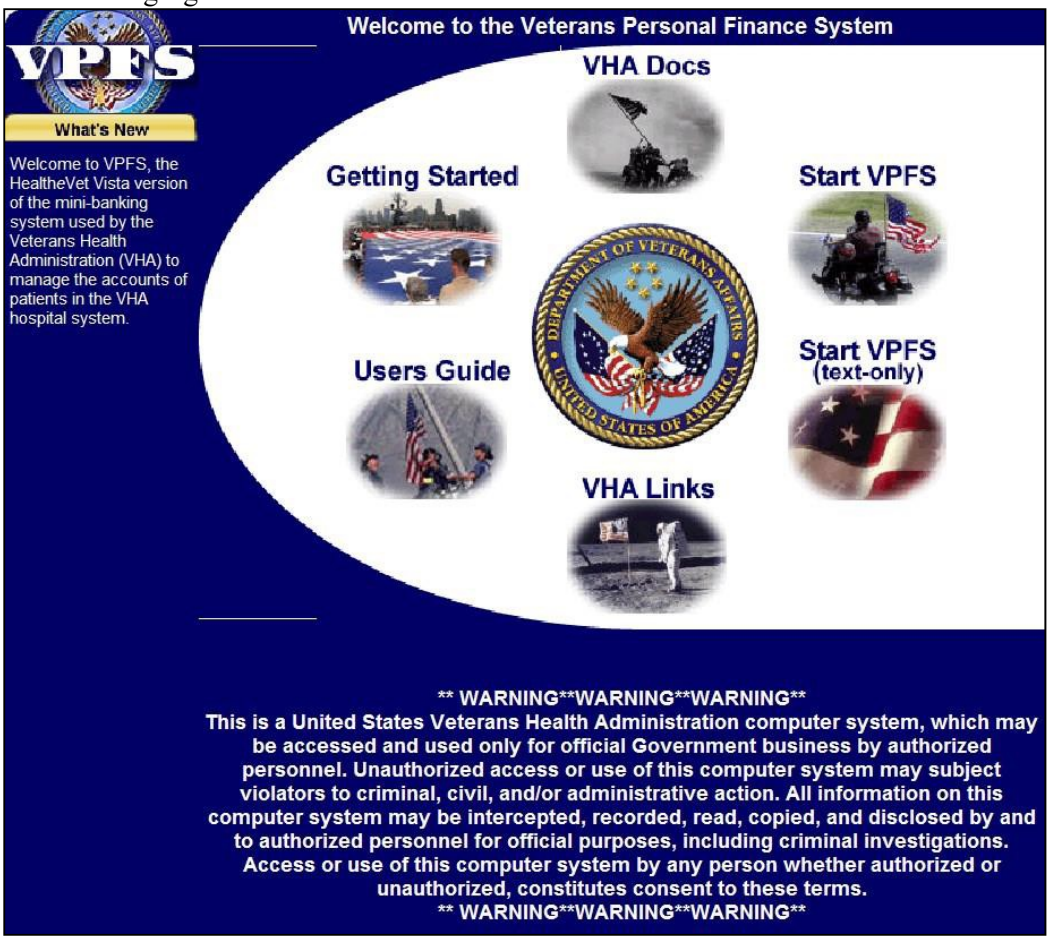


Figure 9-1. Welcome to the Veterans Personal Finance System Page

4. Click on **Start VPFS** to start the application.

9.2 Testing the Installation of the VistAMigrate Application

1. Using a web browser, connect to the VistAMigrate application through the proxy server or load balancer that points marshals' connections to the cluster.

2. KAAJEE will prompt you for a VistA access and verify code pair.
3. Upon authentication with the VistA system, you will see the VistAMigrate Pick Project page, as shown in the following figure.

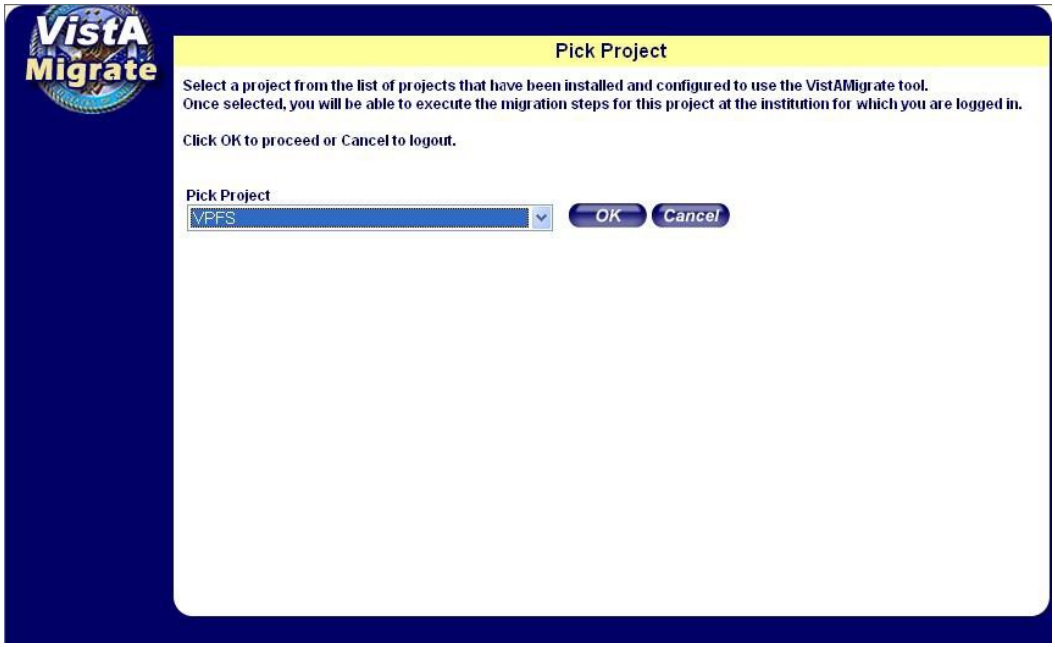


Figure 9-2. VistAMigrate Pick Project Page

9.3 Testing the SMTP Connection

This section will show how to test the SMTP connection. The SMTP connection is used to process the automated e-mails that are generated as part of the VPFS application.

1. Log on to any server in the cluster.
2. Open command prompt.
3. Telnet to the SMTP server via port 25.
   ```
   $ telnet smtp.[domain].gov 25
   ```
4. You will see the following:
   ```
   Trying 127.0.0.1...
   Connected to smtp.[domain].gov
   Escape character is '^]'
   ```
5. At the blank line type the following:
   ```
   EHLO localhost
   ```

.

6. To test sending e-mail type the following. A period (.) will end the message and send the message to the rcpt (recipient) list. To send multiple messages, enter each e-mail address one per line with rcpt to: command repeated.

```
MAIL FROM: abc@myserver.com

RCPT TO: xyz@myserver.com
```

```
DATA

Subject: Testing

<cr>   (This signifies that a blank line is separating the header
from the body of the message.)

Test message

.      (A period at the beginning of a line signifies the end
of the message.)
```

Note: You should get a successful indication and a message should be delivered to recipient user.

10. Rolling Back the VPFS Application Install

1. Using WebLogic Console Deployments section: Click the checkbox next to the current deployment of the VPFS and click the Stop button above or below.

2. Click the checkbox next to the previous deployment of VPFS and click the Start button above or below.

3.You may need to click the Update button above or below once this is done.

Appendix A. References

VPFS Project Documents

Final Detailed Overall System Architecture Document for the Veterans Personal Finance System (VPFS)

776-C21282 VPFS Software Design Document (SDD)

Non-VPFS Project Documents

KAAJEE Installation Guide
SDS API Installation Guide
VistALink Installation Guide
Person Service Lookup System Administrator / Developer Guide
Patient Service Installation Guide
Electronic Signature Installation Guide

Appendix B. Acronyms and Abbreviations

Term	Definition
API	Application Program Interface
C&A	Certification and Accreditation
CHISS	Common Health Information Security Services
CI	Configuration Item
COTS	Commercial Off The Shelf
CPU	Central Processing Unit
CRUD	Create, Read, Update, Delete
DAS	Delegated Administration Service
DBMS	Database Management System
DHTML	Dynamic HyperText Markup Language
DN	Distinguished Name
DOS	Disk Operating System
DSS	Decision Support System
EAR	Enterprise Archive
EJB	Enterprise Java Bean
ERD	Entity Relationship Diagram
ETL	Extract, Transform, Load
FK	Foreign Key
FTP	File Transfer Protocol
HIPAA	Health Insurance Portability and Accountability Act
HIS	Health Information System
HL7	Health Level 7
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure

ICN	Integration Control Number
IIG	Impact Innovations Group
IP	Internet Protocol
IFCAP	Integrated Patient Funds Distribution, Control Point Activity, Accounting and Procurement
IPF	Integrated Patient Funds
J2EE	Java ™ 2 Platform, Enterprise Edition
JAAS	Java Authentication and Authorization Service

Term	Definition
JCA	Java Connector Architecture
JDBC	Java Database Connectivity
JDK	Java Development Kit
JMS	Java Message Service
JNDI	Java Naming Directory Interface
JSP	Java Server Page
JSTL	Java Standard Tag Library
JTA	Java Transaction API
JVM	Java Virtual Machine
KAAJEE	Kernel Authentication and Authorization for J2EE
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LLC	Limited Liability Corporation
M	MUMPS
MAC	Media Access Controller
MUMPS	Massachusetts Universal Multi Programming System
MVC	Model View Controller
OCI	Oracle Call Interface
OCS	Office of Cyber Security

PFC	Patient Funds Clerk
PFCS	Patient Funds Clerk Supervisor
PATS	Patient Advocate Tracking System
PFOP	Personal Funds of Patients
PK	Primary Key
PKI	Public Key Infrastructure
PM	Project Manager
QA	Quality Assurance
QCI	Quality Control Inspector
QCP	Quality Control Plan
RBAC	Role Based Access Control
RDBMS	Relational Database Management System
RO	Regional Office
RPC	Remote Procedure Call
SCM	Software Configuration Management

Term	Definition
SCMP	Software Configuration Management Plan
SD&D	System Design & Development
SDD	System Design Document
SDK	Software Development Kit
SDL	Software Development Library
SDLC	Software Development Life Cycle
SDS	Standard Data Service
SEPG	Software Engineering Process Group
SI	System Implementation
SME	Subject Matter Expert
SOAP	Simple Object Access Protocol

SOW	Statement of Work
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SQL	Standard Query Language
SRS	Software Requirements Specification
SSL	Secure Sockets Layer
TBD	To Be Determined
TLD	Tag Descriptor Library
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VA	Veterans Affairs
VAX	VAX (Virtual Address eXtension) is an established line of mid-range server computers from the Digital Equipment Corporation (DEC).
VHA	Veterans Health Administration
VistA	Veterans Information Systems Technology Architecture
VMS	Virtual Machine System (operating system for VAX computers)
VPFS	Veterans Personal Finance System
WAN	Wide Area Network
WAR	Web Archive
XDE	Extensible Development Environment
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language