**Standards & Terminology Services (STS)**

**VETS Deployment Services Production Release**

*Set Up Guide*



### Version 2.0

### December 2010

**Department of Veterans Affairs**

**Office of Information and Technology (OI&T) Office of Enterprise Development (OED)**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| July 2010 | 0.1 | V9 text uploaded for V10 start. | REDACTED |
| Sept. 2010 | 0.2 | Initial changes from Schwann. | REDACTED |
| Oct. 2010 | 0.3 | Rewrite text for V10. Edit, format. | REDACTED |
| Nov. 2010 | 0.4 | Update screen shots, step text. | REDACTED |
| Dec. 2010 | 0.5 | Screen shots, validation. | REDACTED |
| Dec. 2010 | 0.6 | Sent for Review | REDACTED |
| Dec. 2010 | 1.0 | Final for PDF | REDACTED |

**Table of Contents**

[Introduction 3](#_bookmark0)

[PreInstallation 3](#_bookmark1)

[VETS V10 Build and Data Conversion Instructions 3](#_bookmark2)

[Create VETS V10 Schemas Instructions 4](#_bookmark3)

[Copy VETS V9 Data into the VETS V10 Schemas Instructions 5](#_bookmark4)

[STS VETS Deployment Process 6](#_bookmark5)

[Prerequisites 6](#_bookmark6)

[Create a Basic WebLogic Domain 8](#_bookmark7)

[Create WebLogic Managed Servers 8](#_bookmark8)

[Create JDBC Data Sources 13](#_bookmark9)

[Load files into WebLogic Domain Directories 15](#_bookmark10)

[Create Server Start/Stop and Application Deployment Scripts 16](#_bookmark11)

[Creating STS Application Users 17](#_bookmark12)

[Configure and Start Node Manager 21](#_bookmark13)

[Log Configuration 22](#_bookmark14)

[Configure WebLogic for Production Mode 25](#_bookmark15)

[Start Managed WebLogic Servers and Deploy Applications 26](#_bookmark16)

[Glossary 28](#_bookmark17)

[STS Terminology Glossary 28](#_bookmark18)

[Appendix A – WebLogic Server Installation Configuration 34](#_bookmark19)

[Appendix B – WebLogic Domain Installation Configuration 34](#_bookmark20)

[Appendix C – Start/Stop and Deploy Scripts 35](#_bookmark21)

[Appendix D – STS V10 Detailed Configuration 35](#_bookmark22)

# Introduction

This manual describes the STS VETS Version 10 Set Up procedures. The procedures include:

* Build and Data Conversion
* STS Deployment
* Copying New Configuration and Help Files
* Starting and Stopping WebLogic Servers
* Creating STS Application User

# PreInstallation

STS VETS V10 uses an Oracle 11gR1 RAC database server. WebLogic 10.3.2 and Java 6 Update 20 must be installed.

# VETS V10 Build and Data Conversion Instructions

This section describes the steps you need to perform to create the VETS V10 database schemas and copy the VETS V9 data into the VETS V10 schema. The VETS V9 and V10 schemas can co- exist in the same database while the VETS V10 environment is validated and the VETS V9 environment can be archived.

These tables list the V9 schema names and their corresponding V10 schema names in the environments that the V10 data creates from a V9 schema.

|  |  |
| --- | --- |
| **PRODUCTION (HDRP06)** | |
| **V9 SCHEMA NAME** | **V10 SCHEMA NAME** |
| DS\_V09 | DS\_V10 |
| NTRT\_V09 | NTRT\_V09 |
| VHAT\_V09 | No longer used |
| VTS\_V09 | VTS\_V09 |
| VUID\_V09 | Moved into VTS\_V09 schema |

|  |  |
| --- | --- |
| **SQA (ETSD07 on vhaislbll10)** | |
| **V9 SCHEMA NAME** | **V10 SCHEMA NAME** |
| DS\_V09 | DS\_V10 |

|  |  |
| --- | --- |
| NTRT\_V09 | NTRT\_V09 |
| VHAT\_V09 | No longer used |
| VTS\_V09 | VTS\_V09 |
| VUID\_V09 | Moved into VTS\_V09 schema |

|  |  |
| --- | --- |
| **Integration (STSI01 on vhaislbll26)** | |
| **V8 SCHEMA NAME** | **V10 SCHEMA NAME** |
| DS\_V09 | DS\_V10 |
| NTRT\_V09 | NTRT\_V09 |
| VHAT\_V09 | No longer used |
| VTS\_V09 | VTS\_V09 |
| VUID\_V09 | Moved into VTS\_V09 schema |

## Create VETS V10 Schemas Instructions

Follow the steps below to create the V10 schemas:

1. These steps are executed from a PC or UNIX server that has Oracle client installed and tnsnames.ora connection information to the database for the new VETS V10 schema.
2. Execute the Oracle SQL script. build\_v10\_releasenumber\_full\_databasename.sql
   * The releasenumber is the latest database build number for V10.
   * The databasename is the name of the database that the build will be created on.
   * The build\_v10\_releasenumber\_full\_databasename.sql file calls other SQL files to build the environment.
   * All parameters needed to create the objects are specified in the

build\_v10\_releasenumber\_full\_databasename.sql file.

1. Review the script file before the script is executed to ensure that all of the information is correct.
   * All SQL files used in this step are stored in Perforce.
   * The objects that are created by this script can be viewed via this HTML data model contained in the zip file VETS\_001009.zip.

Follow the step below to create the VETS V10 data model on a PC:

1. Unload the zip files into a directory on your computer.

<FILE://\\vhaislmul1\Projects\ETS (STS)\VETS\Version 10\Toad Data Modeler\Reports\HTML\VETS\_10.zip>

1. Open the file DS\_NTRT\_VTS\_V10.html in a browser window to view the data model.

## Copy VETS V9 Data into the VETS V10 Schemas Instructions

The conversion process below was designed so that the V9 and the V10 data reside in the same database. Follow the steps below to copy V9 data in to the V10 schemas:

1. Copy the following files into the data\_pump\_dir directory on the database server:

changegroup\_seq\_create.sql checksumrequest\_seq\_create.sql codesystem\_seq\_create.sql onceptstate\_seq\_create.sql concept\_seq\_create.sql copy\_v9\_to\_v10\_database.sh copy\_v9\_v10\_ds.sql

copy\_v9\_v10\_ds\_stsi01.sql (only for database stsi01 & stsi02) copy\_v9\_v10\_ntrt.sql

copy\_v9\_v10\_vts.sql copy\_v9\_v10\_vuid.sql deploymentconcept\_seq\_create.sql deploymenthistory\_seq\_create.sql deployment\_seq\_create.sql drop\_v10\_objs.sql grant\_select\_any\_table.sql property\_seq\_create.sql regionchecksum\_seq\_create.sql relationship\_seq\_create.sql request\_delete\_rows.sql revoke\_select\_any\_table.sql sitedatarequest\_seq\_create.sql sitedata\_seq\_create.sql state\_seq\_create.sql type\_seq\_create.sql v10\_gather\_stats\_001002.sql

version\_seq\_create.sql vhat\_v10\_cleanup.sql request\_subset\_update.sql

1. The following files need to be edited to update the variables with the correct values:

copy\_v9\_to\_v10\_database.sh

1. Execute the shell script copy\_v9\_to\_v10\_database.sh on the database server to copy the data using this command:

copy\_v9\_to\_v10\_database.sh 1>copy\_v9\_to\_v10\_database.log 2>&1

This script can be executed against an empty V10 schema, or when data is in the V10 schema and data needs to be reloaded. The main tasks that this script performs are:

* Copies data directly from V9 tables into V10 tables using Insert/Select statements for tables that do not contain LONG RAW data.
* The data in the ds\_v10 and vts\_v10 schemas are truncated.
* Export/Import V9 NTRT table into a V10 NTRT table because table contains LONG RAW data
* Sets all sequences to correct values
* Sets all object permissions
* Moves indexes to correct tablespace
* Gathers schema statistics for all V10 schema All files used in this step are stored in Perforce.

# STS VETS Deployment Process

The STS VETS Deployment Process consists of installing a VETS V10 domain and configuring the scripts.

## Prerequisites

All software packages are installed into the base directory /u01/app. A different directory can be used as the base directory. If you use a different directory you need to replace any references to the base directories in this document to the directory you are using.

All installation commands should run under the WebLogic UNIX system ID. Before installation, run the following command on the target server to make sure it is running Linux on x86\_64 hardware. See the following sample output:

*uname -a*

Linux vhaislbll25.vha.med.va.gov 2.6.18-194.3.1.el5 #1 SMP Sun May 2 04:17:42 EDT 2010 x86\_64 x86\_64 x86\_64 GNU/Linux

* Verify /etc/hosts file

Run the following command to make sure there is a separate line in /etc/hosts that resolves both short hostname and fully qualified domain name to the host IP address.

*cat /etc/hosts | grep `hostname`*

REDACTED REDACTED

* Install Java 6 Update 20

Download the Java installer file jdk-6u20-linux-x64.bin from the Java web site*.* Copy the file into the application base directory and make it executable.

Run the following:

./jdk-6u20-linux-x64.bin

Press the **Enter** button to progress through the License Agreement pages.

At the prompt Do you agree to the above license terms? [yes or no], select Yes. Press the **Enter** button to continue.

Java is installed into the /u01/app/jdk1.6.0\_20 directory. The original installer file can be deleted at this point.

Run the following command to create a symbolic link to the actual java location.

ln -s jdk1.6.0\_20 jdk1.6

* Install WebLogic version 10.3.2

Download the WebLogic installer file wls1032\_generic.jar from the Oracle web site. Copy the file into the application base directory.

In the same base directory, create a file named wls1032.xml with the content shown in [Appendix A.](#_bookmark19)

Run the following command:

/u01/app/jdk1.6/bin/java -jar wls1032\_generic.jar \

-mode=silent -silent\_xml=wls1032.xml

The command takes a few minute to run.

When the command is finished WebLogic is installed into /u01/app/wls1032 and the WebLogic home directory is /u01/app/wls1032/wls.

The WebLogic installer file can be deleted to free up disk space.

The following directories are defined for reference throughout this document:

AppDir=/u01/app JavaHome=/u01/app/jdk1.6 BeaHome=/u01/app/wls1032 WlsHome=/u01/app/wls1032/wls

## Create a Basic Web Logic Domain

As WebLogic user ID, set up CLASSPATH by following command: CLASSPATH=/u01/app/wls1032/utils/config/10.3/config-launch.jar CLASSPATH=$CLASSPATH:/u01/app/wls1032/wls/server/lib/weblogic.jar

CLASSPATH=$CLASSPATH:/u01/app/wls1032/modules/features/weblogic.server.modul es\_10.3.2.0.jar

export CLASSPATH

Create a Python command file named basedomain.py ([Appendix B](#_bookmark20)) in the application base directory with the following commands.

**Note**: Substitute with appropriate base directory values wherever needed.

Create a WebLogic domain by running the following command. If it is successful, a new domain directory should be found in /u01/app/domains, e.g. v10.Prod.

/u01/app/jdk1.6/bin/java weblogic.WLST basedomain.py

Start the new WebLogic domain with following commands:

cd /u01/app/domains/v10.Prod

nohup /u01/app/domains/v10.Prod/bin/startWebLogic.sh & tail -f nohup.out

The command will take a few minutes to run.

**Note**: Your Username and Password are created when you run the command.

Log in to the WebLogic Administration Console using the Administration Username and Password created in the basedomain.py script.

## Create Web Logic Managed Servers

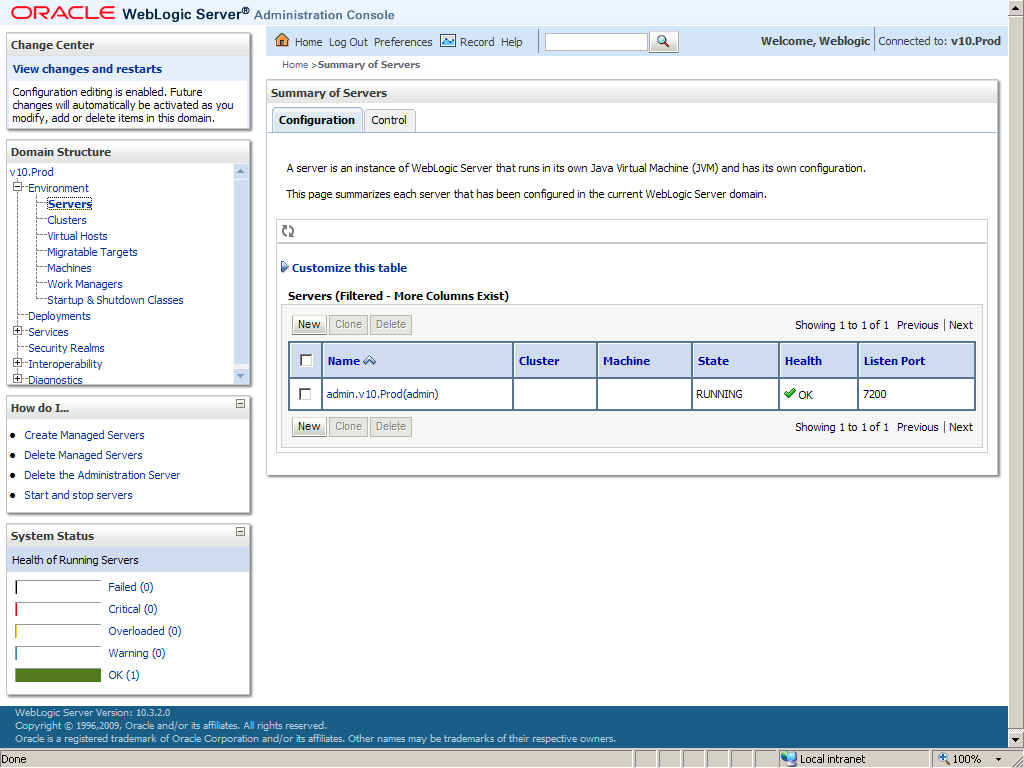
To create the WebLogic managed servers:

1. Log in to the WebLogic Administration Console. Enter **Username** WebLogic and **Password**

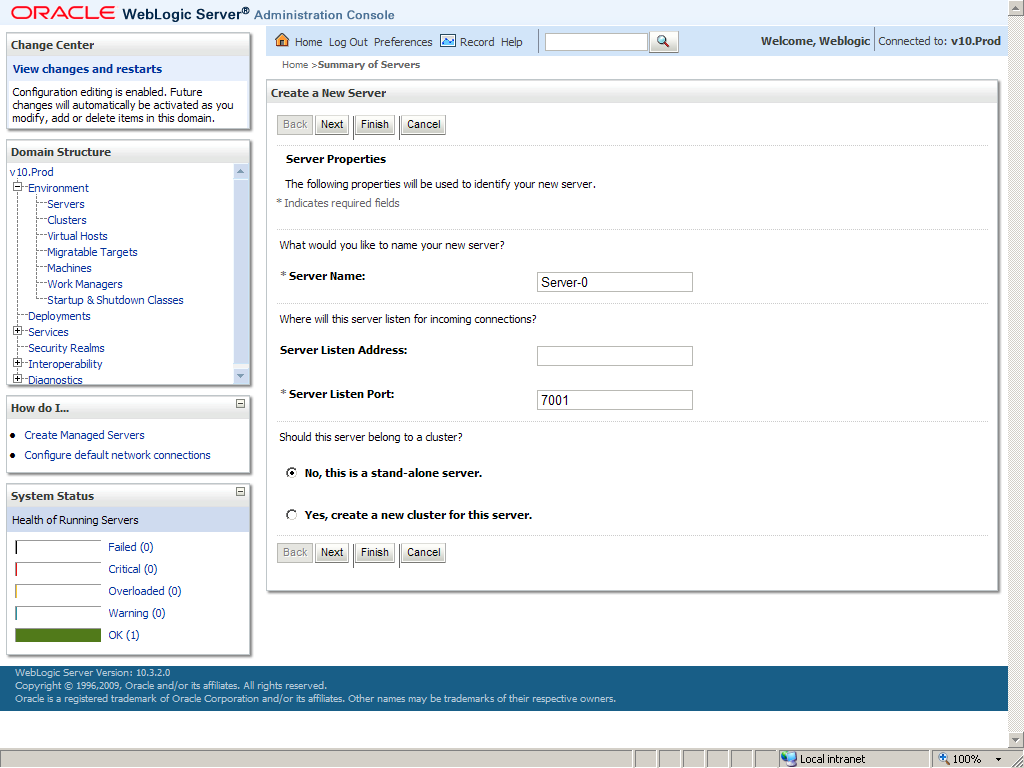
admin123.

Following is a list of the managed servers you want to create:

|  |  |
| --- | --- |
| Name | Listen Port |
| deployment | 7201 |
| ntrt | 7202 |
| vuid | 7203 |
| browser | 7204 |
| ted | 7208 |

1. In the **Domain Structure** box, expand **Environment**.
2. Click and open **Servers**.
3. In the **Summary of Servers** box, click the **Configuration** tab.
4. Click the **New** button at the bottom of the **Configuration** box.
5. In the **Create a New Server** box, complete the required fields (\* Indicates required fields) with the information for the first managed server you want to create.

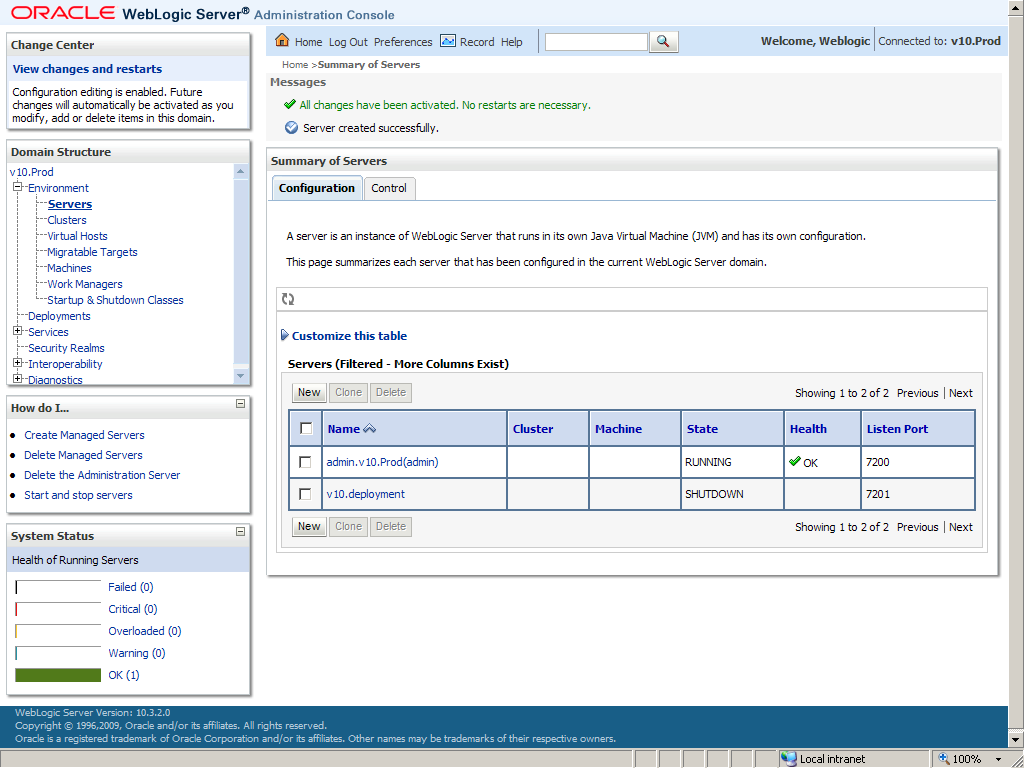
This example uses the first server listed under step 1, above.

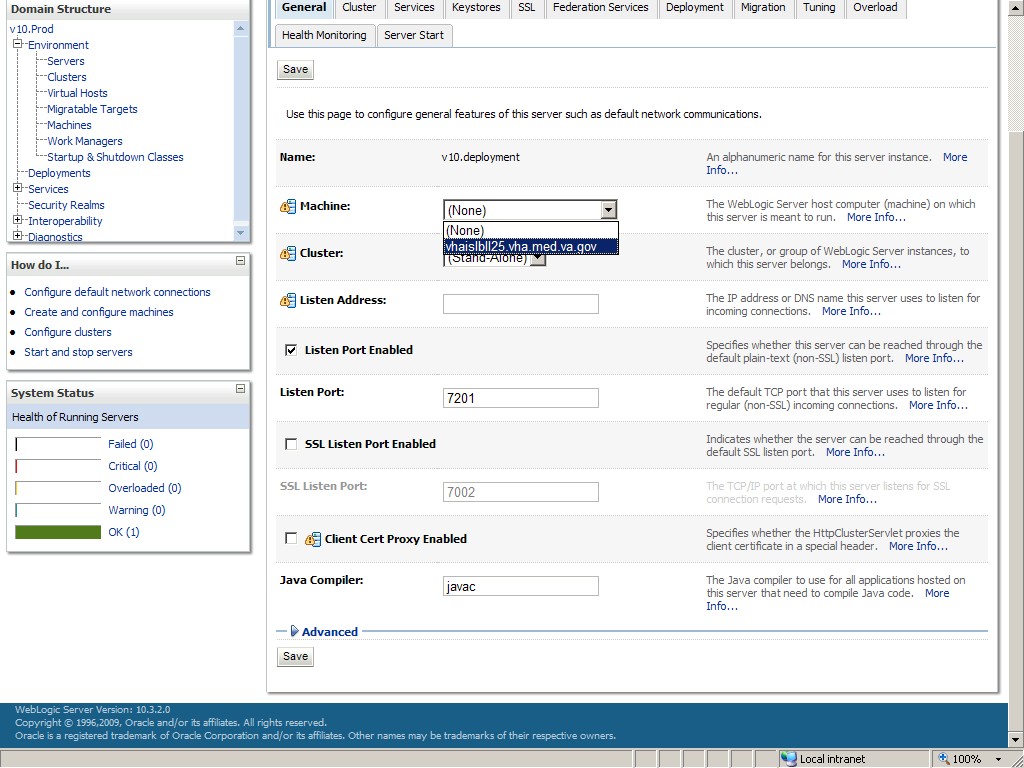


1. Complete the **Server Listen Address** field.

The Server Listen Address is the IP address of the WebLogic host server.

1. Leave the No, this is a stand-alone server. option selected.
2. Click the **Finish** button.



1. In the **Servers** box, click the name (link) of the newly created server.
2. In the **Machine** pull down menu, choose the machine name where this server should reside.
3. Click the **Save** button.
4. Click the **Server Start** tab.
5. Complete the following fields with the correct information.

**Java Home:** /u01/app/jdk1.6

#### Java Vendor: Sun

**BEA Home:** /u01/app/wls1032

**Root Directory:** /u01/app/domains/v10.prod

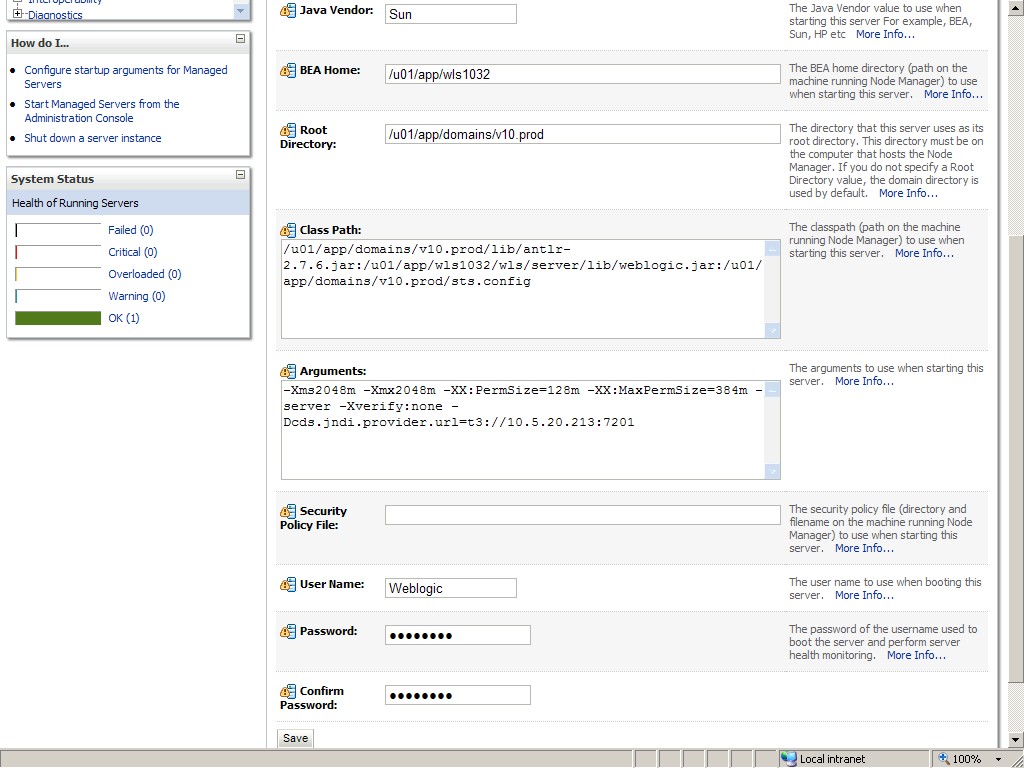
#### Classpath:

/u01/app/domains/v10.prod/lib/antlr- 2.7.6.jar:/u01/app/wls1032/wls/server/lib/weblogic.jar:/u01/app/domains/v10.prod/sts. config

#### Arguments:

-Xms2048m -Xmx2048m -XX:PermSize=128m -XX:MaxPermSize=384m -server - Xverify:none -Dcds.jndi.provider.url=t3://10.5.20.180:7201

**User Name:** WebLogic **Passwords:** admin123 **Confirmed Passwords:** admin123



1. Click the **Save** button.
2. Repeat the above steps to create the other managed servers.

## Create JDBC Data Sources

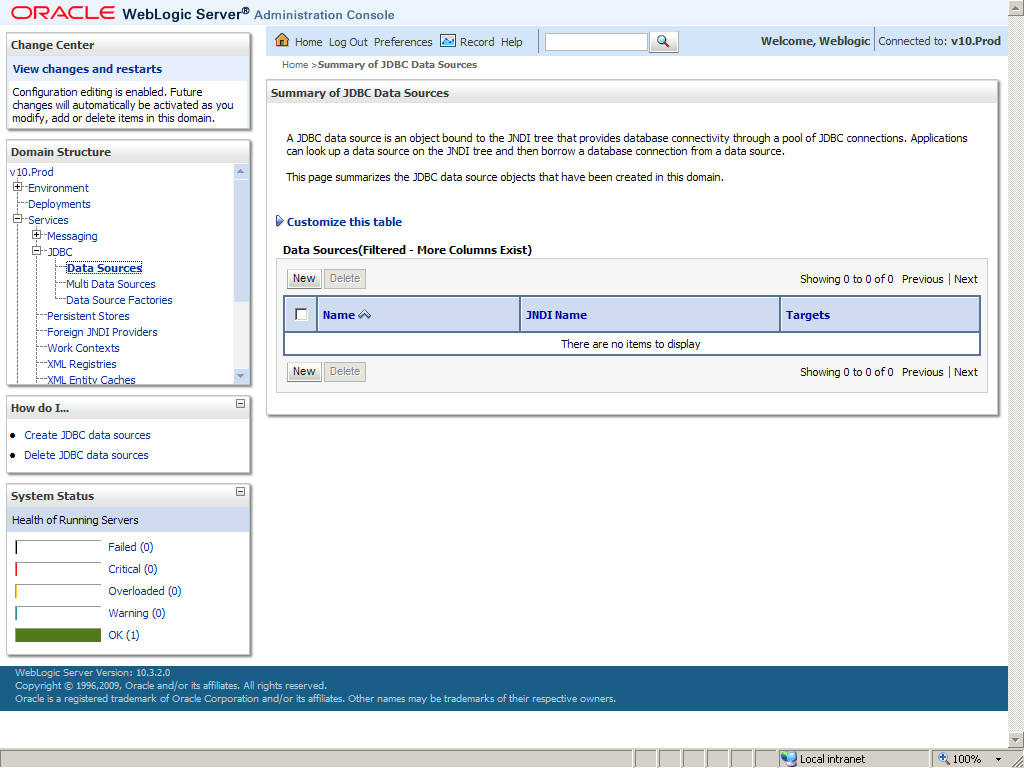
Follow the steps below to create JDBC data sources:

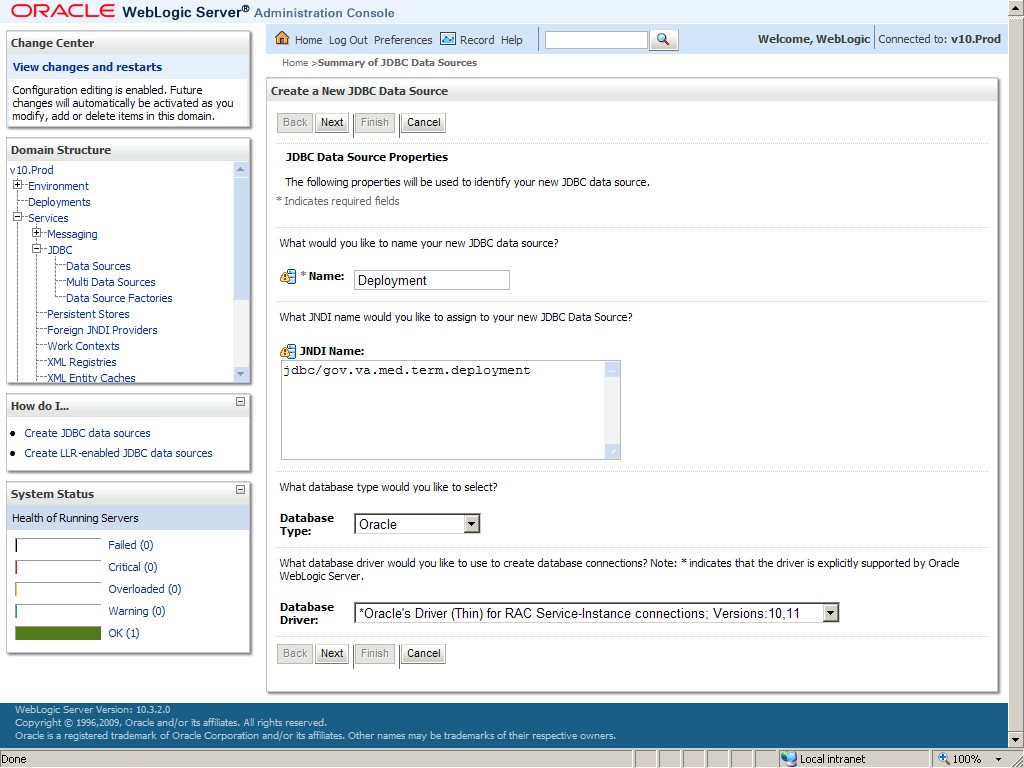
1. Log in to the WebLogic Administration Console. Enter **Username** WebLogic and **Password**

admin123.

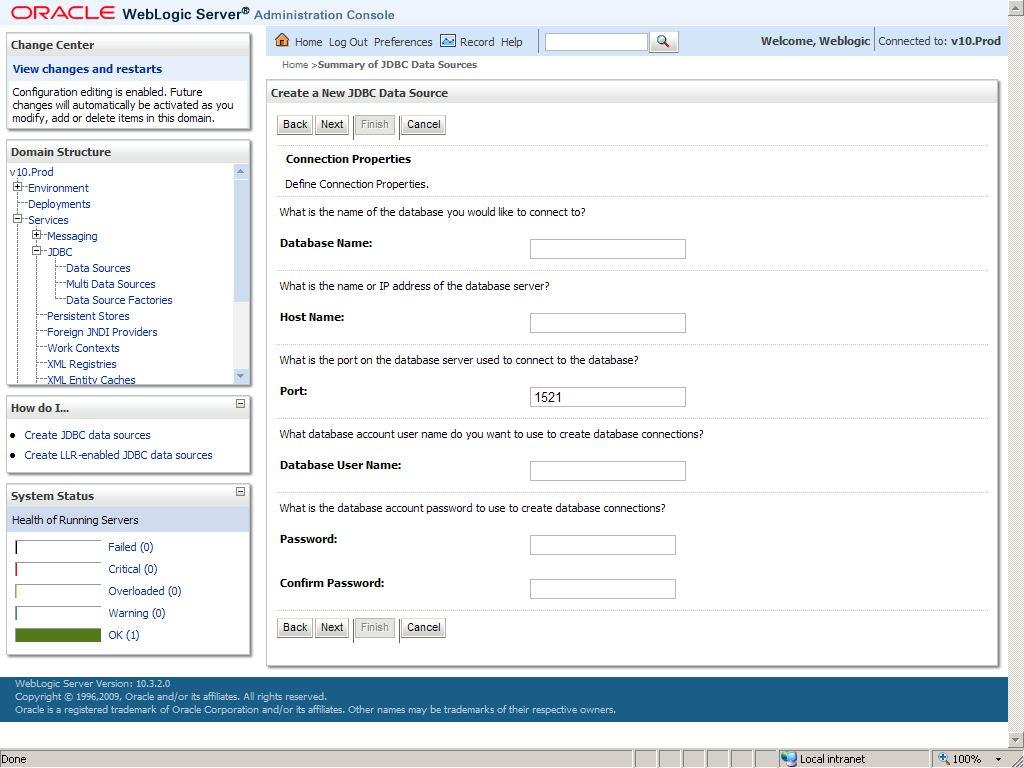
Following is a list of the data sources you want to create:

|  |  |  |
| --- | --- | --- |
| Name | JNDI Name | Targets |
| Deployment | jdbc/gov.va.med.term.deployment | deployment |
| NTRT | jdbc/gov.va.med.term.ntrt | ntrt |
| VETS | jdbc/gov.va.med.term.services | browser, ntrt,  ted, vuid |

1. In the **Domain Structure** box, expand **Services**.
2. Expand **JDBC**.
3. Click and open **Data Sources**.
4. In the **Summary of JDBC Data Sources** box, click the **New** button.
5. Complete the **Name** and **JNDI Name** fields.

This example uses the first data source under step 1, above.

1. Click the **Next** button.
2. Click the **Next** button on the following page.
3. Complete the **Connections Properties** page.



1. Click the **Next** button.
2. In the **Select Targets** box, select the **Server** where this data source will be deployed.
3. Click the **Finish** button.
4. Repeat these steps to create the other data sources.

## Load file s into Web Logic Domain Directories

* 1. Run the following commands to create several new directories under the domain directory.

cd /u01/app/domains/v10.prod

mkdir apps checksum sts.config sts.log

* 1. Copy the files listed in each of the following directories, from the build .WAR files and source repository.

**Note**: For any update to application.properties or terminologyconfig.xml, sts.deployment

should be restarted.

apps/ntrt.war apps/sts.browser.war apps/sts.deployment.war apps/sts.ted.war

apps/vuid.war lib/antlr-2.7.6.jar lib/log4j-1.2.14.jar lib/loglayout.jar

sts.config/application.properties sts.config/BrowserConfig.xml sts.config/browserhelp.html sts.config/browserIntro.html sts.config/deploymenthelp.html sts.config/log4j.properties sts.config/ntrtdeployhistory.html sts.config/tedhelp.html sts.config/TerminologyConfig.xml

* 1. Edit the application.properties file.

Ensure all variables are appropriate for the target environment, particularly the following variables.

application.server.name=PROD V10 listenerPort = 49990 msh.sendingFacility.namespaceId=660VM1

gov.va.med.term.access.maint.messaging.hl7.factory.BusinessWareMessageDispatcher/ url=http://vhaislviev1:8080/fwclient2/Framework2ServletHTTPtoChannel

## Create Server Start/Stop and Application Deployment Scripts

1. For each managed server create start, stop, and bounce scripts in

/u01/app/domains/v10.prod/bin directory.

Examples of the scripts are shown in [Appendix C.](#_bookmark21) For a different server, make sure all variables are changed accordingly.

1. Create a master deploy script in the same domain bin directory. See [Appendix C.](#_bookmark21)
2. Run chmod u+x … on all above scripts to make sure that they are executable.
3. Run the following commands to store user credentials referenced from the above scripts.

cd /u01/app/domains/v10.prod/bin mkdir .security

export CLASSPATH=$CLASSPATH:/u01/app/wls1032/wls/server/lib/weblogic.jar echo y | /u01/app/jdk1.6/bin/java -Duser.home=.security \

weblogic.Admin STOREUSERCONFIG -username WebLogic -password admin123 chmod 700 .security

chmod 400 .security/\*

## Creating STS Application Users

To create STS Application Users:

1. Log in to the WebLogic Administration Console. Enter **Username** WebLogic and **Password**

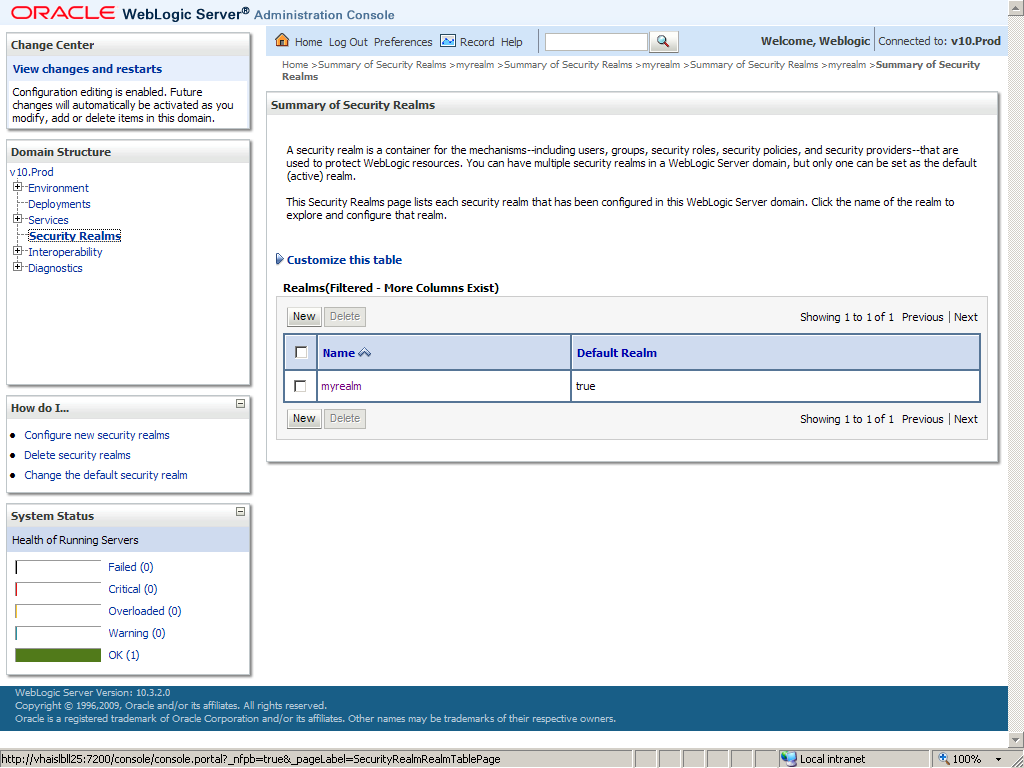
admin123.

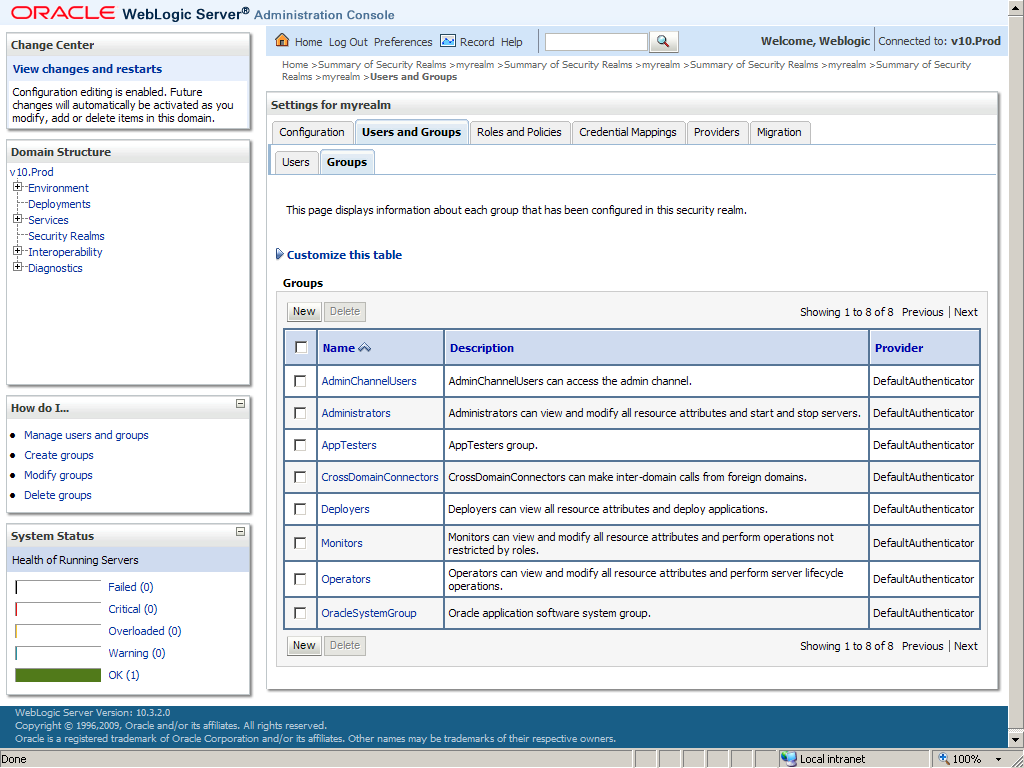
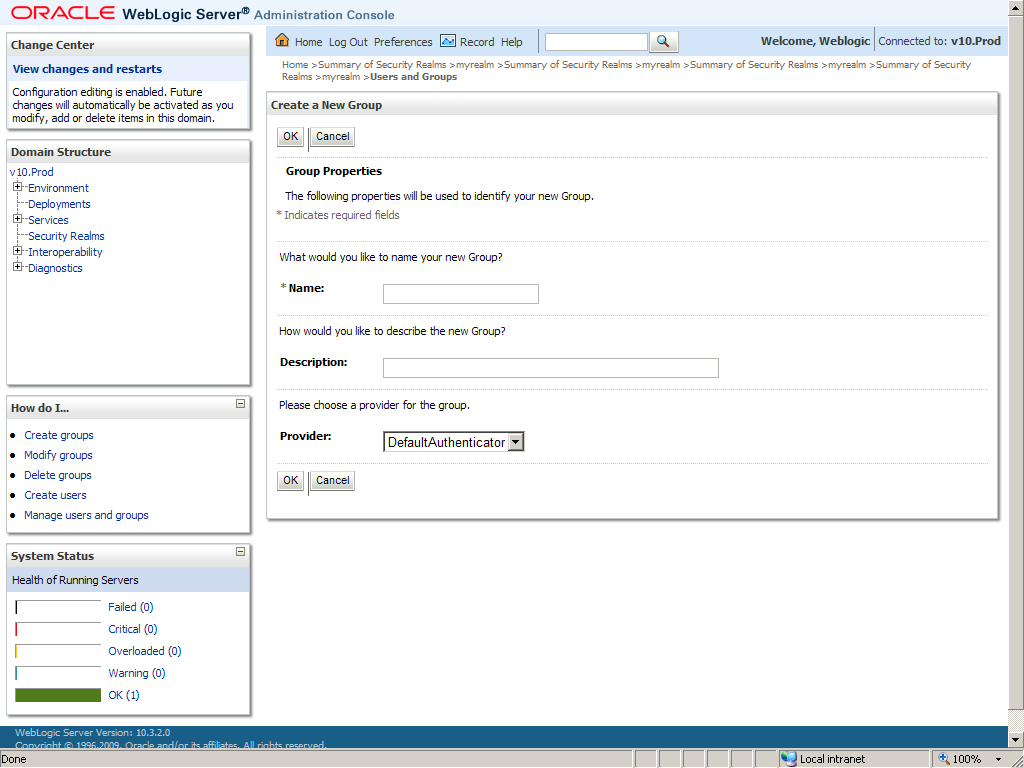
Following is the group and user you want to create.

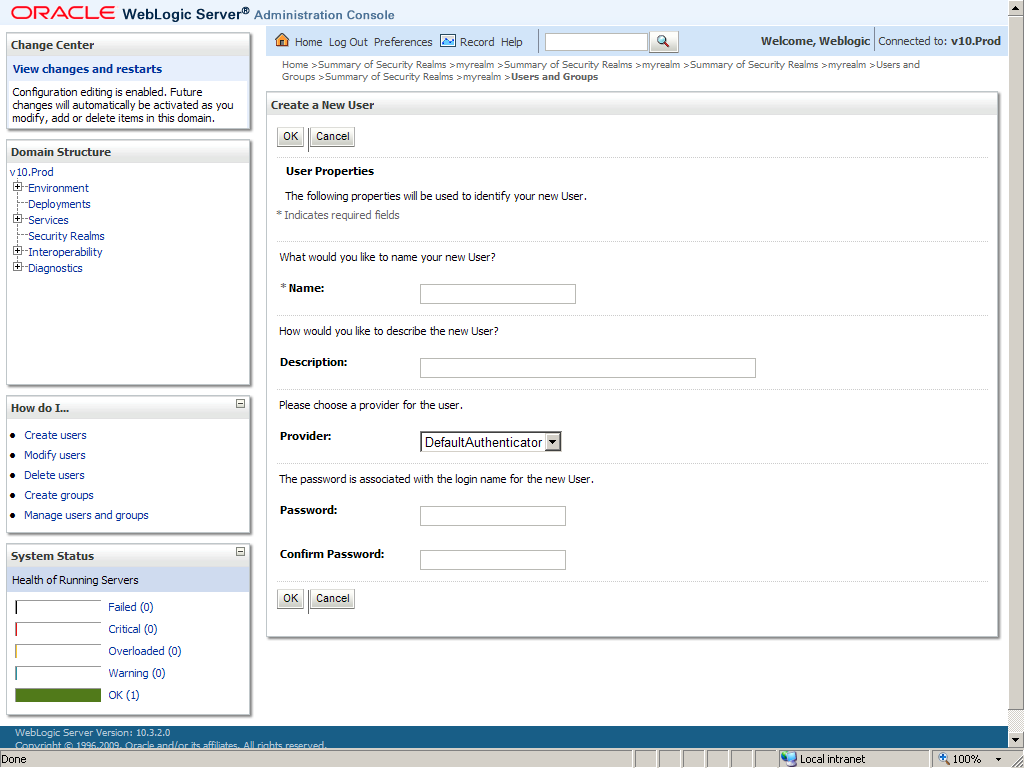
**Group:** DSAdmin

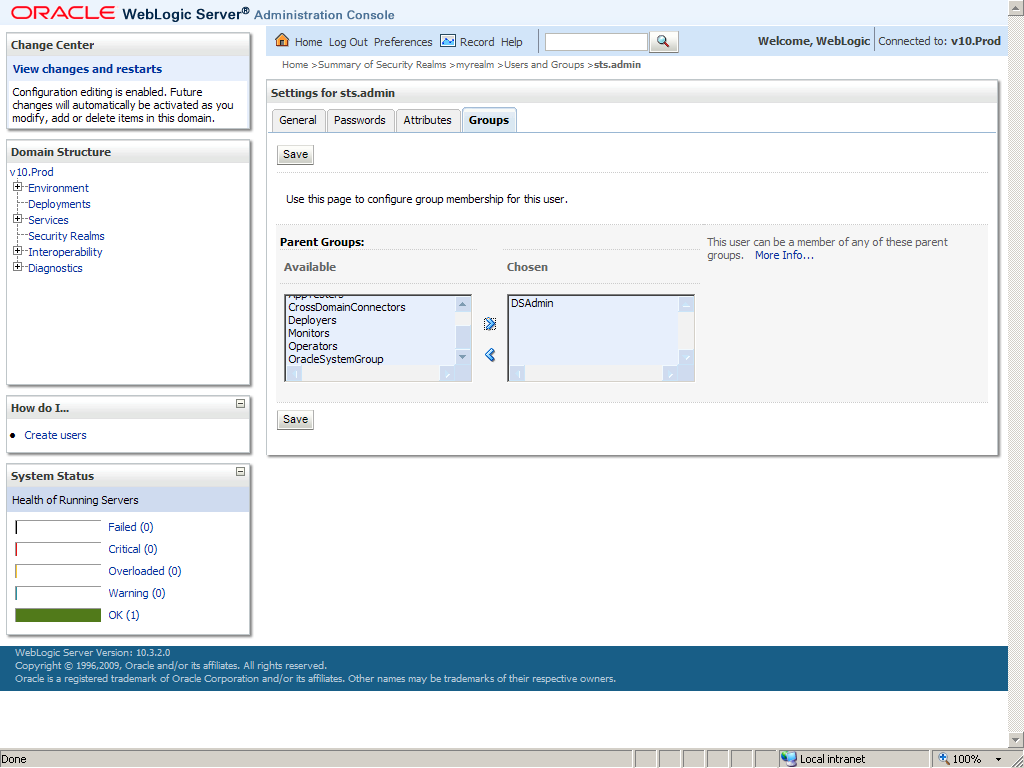
**User:** sts.admin

1. In the **Domain Structure** box, click **Security Realms**.
2. In the **Summary of Security Realms** box, click **myrealm**.



1. Click the **Users and Groups** tab.
2. Click the **Groups** tab.
3. Click the **New** button.
4. Complete the **Name** and **Description** fields.
5. Click the **OK** button.
6. On the **Users and Groups** page, click the **Users** tab.
7. Click the **New** button.
8. Complete the **Name**, **Description**, and **Password** fields.



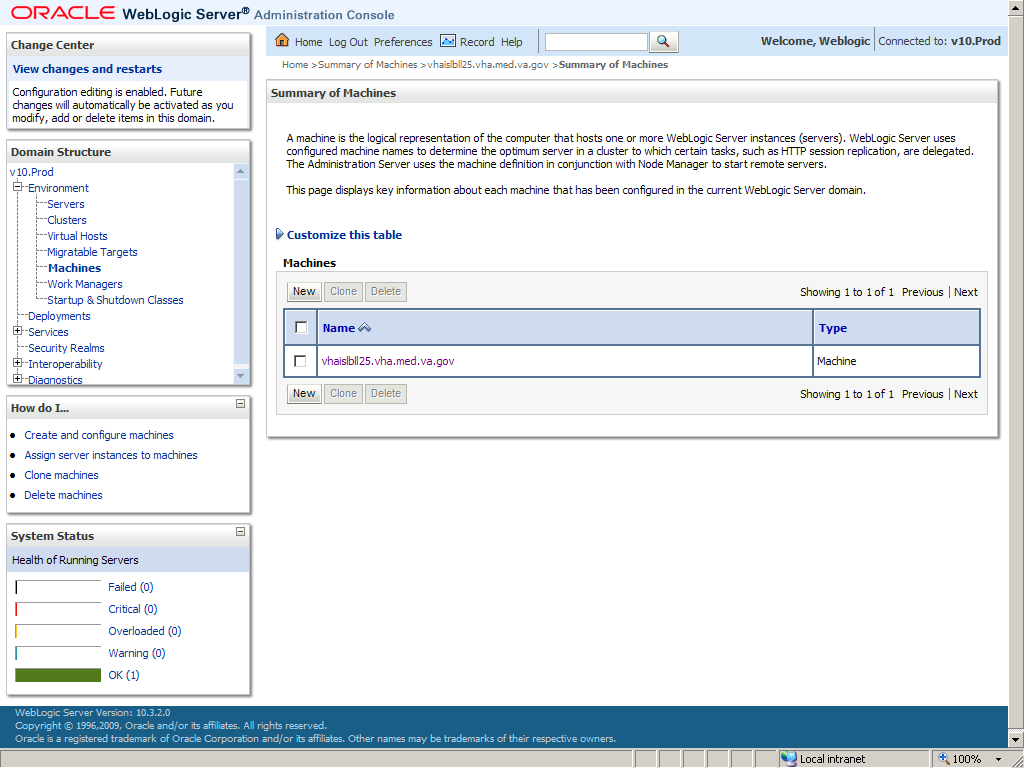
1. Click the **OK** button.
2. Click on the name of the newly created user.
3. Click the **Groups** tab.
4. Select the Group you created in the previous steps.
5. Click the **>** button to move that Group into the **Chosen** field.
6. Click the **Save** button.
7. Repeat these steps to create other users.

## Con figure and Start Node Manager

To configure the node manager:

1. Log in to the WebLogic Administration Console. Enter **Username** WebLogic and **Password**

admin123.

1. In the **Domain Structure** box, expand **Environment**.
2. Click the **Machines** link.
3. In the **Summary of Machines** box, click the machine name.
4. Click the **Node Manager** tab.
5. Complete the fields.



1. Click the **Save** button.

If the default listen port is not available on the server, change it to an available number.

1. Open a terminal console and run the following commands to start the node manager. cd /u01/app/wls1032/wls/server/bin

nohup /u01/app/wls1032/wls/server/bin/startNodeManager.sh &

tail -f nohup.out

1. If the default port 5556 is not used, edit the following file to update ListenPort with the same number configured in the Administration Console.

/u01/app/wls1032/wls/common/nodemanager/nodemanager.properties

1. Kill and restart the node manager to pick up any change.

## Log Con figuration

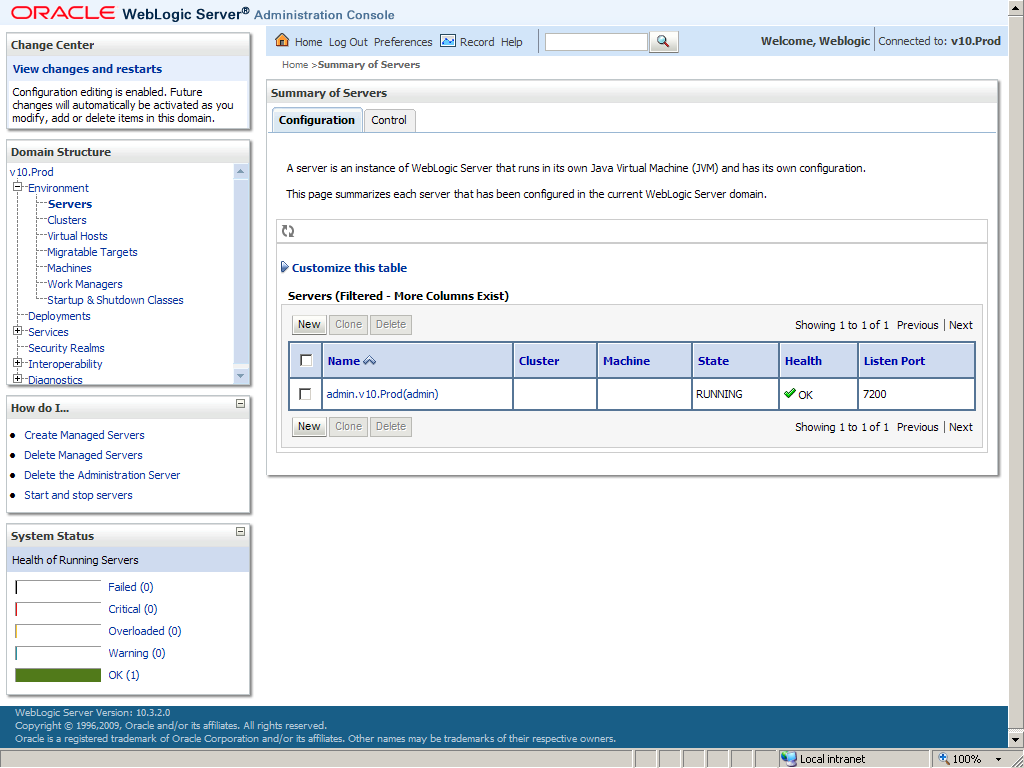
To configure the WebLogic server logs:

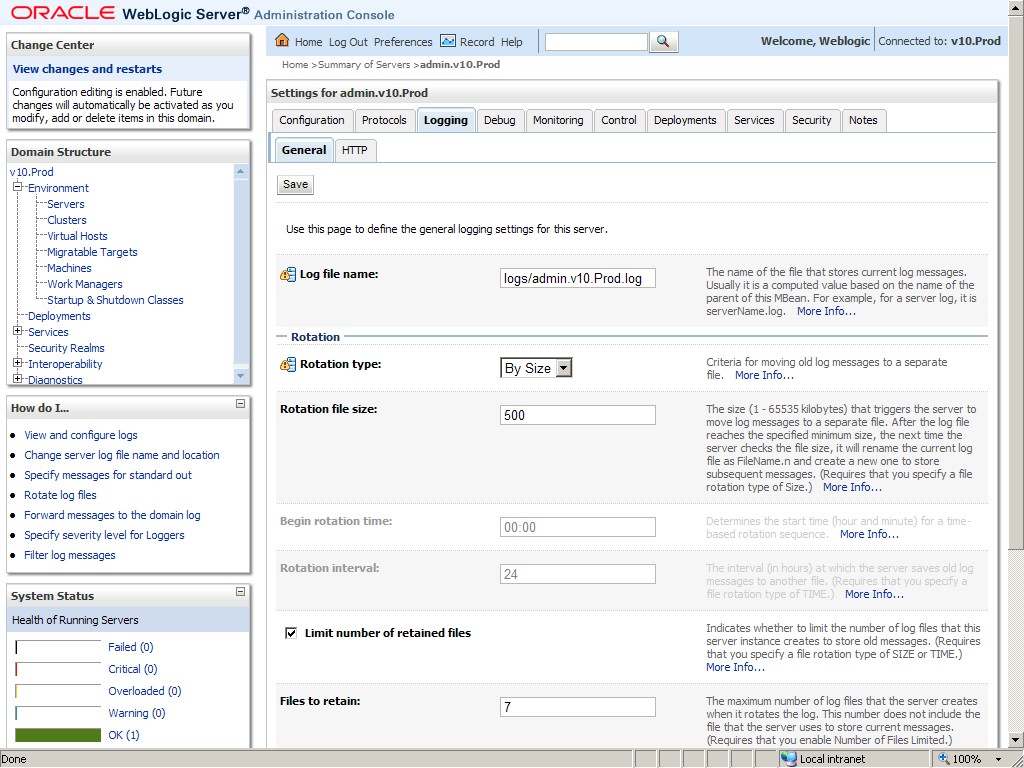
1. Run the following commands in a terminal console to create a directory to hold all WebLogic server logs.

cd /u01/app/domains mkdir logs

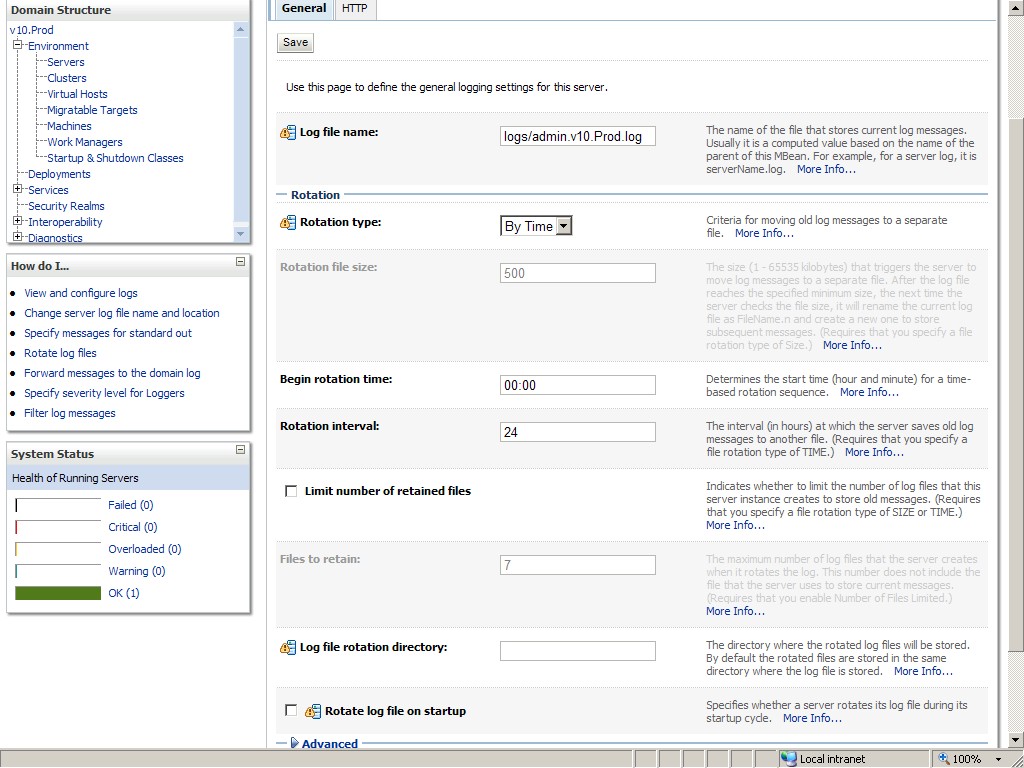
1. Log in to the WebLogic Administration Console. Enter **Username** WebLogic and **Password**

admin123.

1. In the **Domain Structure** box, expand **Environments**.
2. Click **Servers**.
3. Click the admin server name.
4. Click the **Logging** tab.



1. Click the **General** tab.
2. Change the **Log file name** to use the newly created log directory.
3. In the **Rotation type** pull down menu, select **By Time**.
4. Uncheck **Limit number of retained files**.
5. Uncheck **Rotate log file on startup**.
6. Click the **Save** button.



1. Click the **HTTP** tab.
2. Change the **Log file name** to use the newly created log directory.
3. In the **Rotation type** pull down menu, select **By Time**.
4. Uncheck **Limit number of retained files**.
5. Uncheck **Rotate log file on startup**.
6. Click the **Save** button.
7. Repeat the above steps for all managed servers.

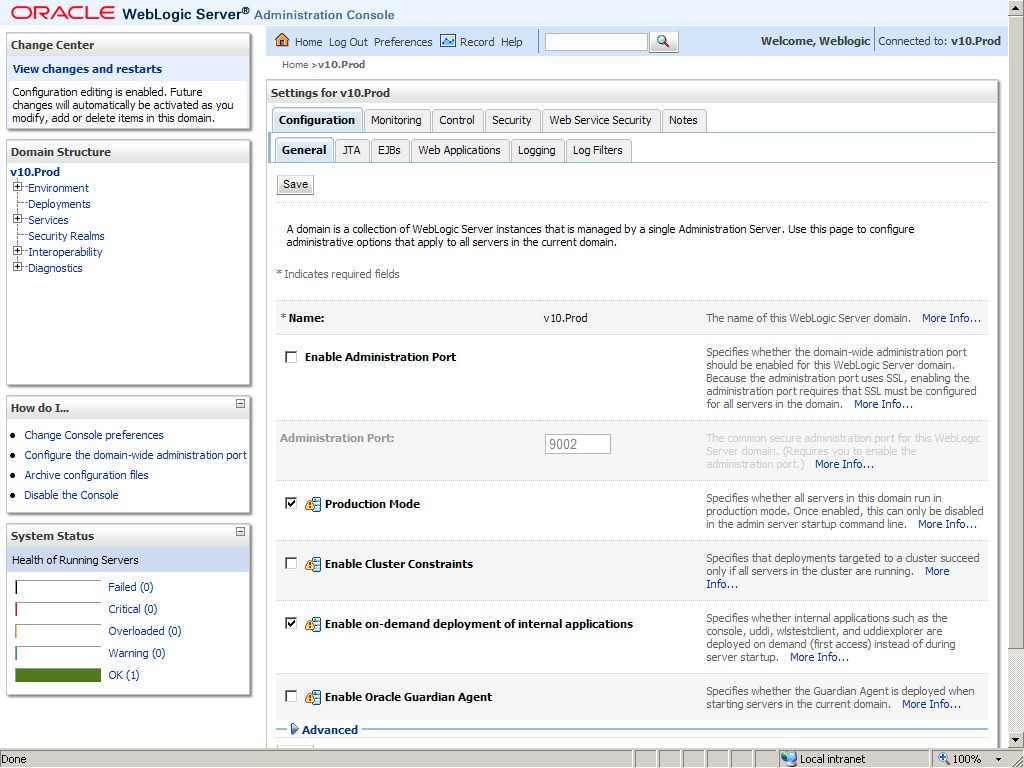
## Configure WebLogic for Production Mode

To configure WebLogic for Production mode:

1. Log in to the WebLogic Administration Console. Enter **Username** WebLogic and **Password**

admin123.

1. In the **Domain Structures** box, click the domain name.
2. Click the **General** tab.
3. Select the box for **Production Mode**.



1. Click the **Save** button.

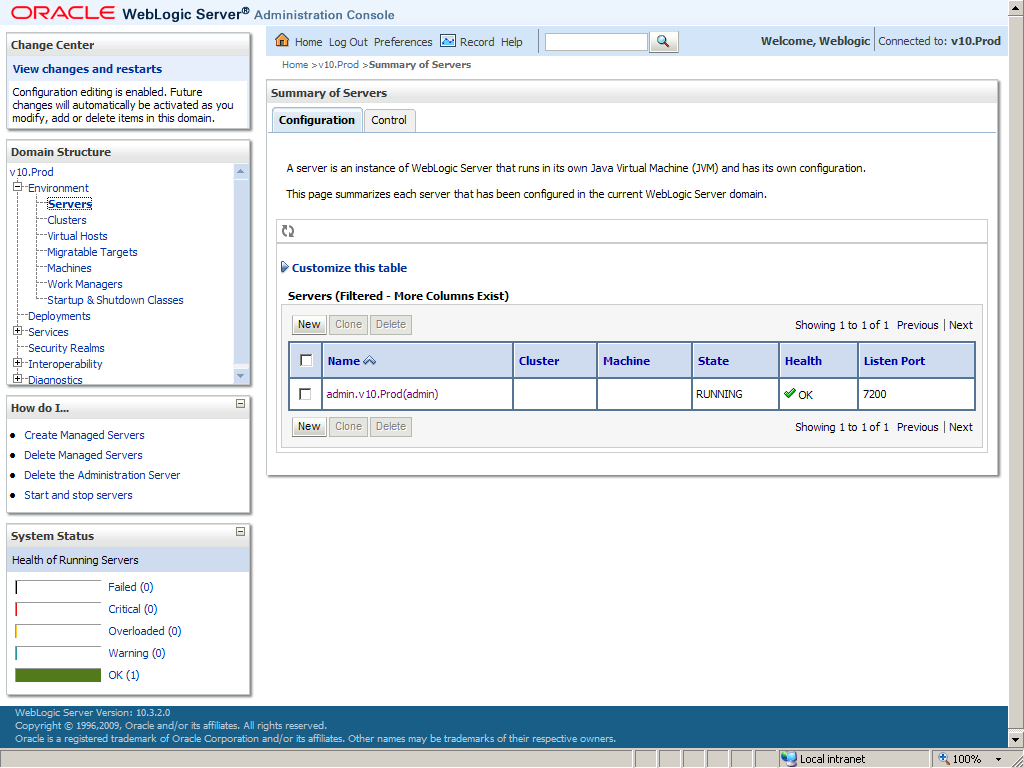
## Start Managed WebLogic Servers and Deploy Applications

To start the managed servers and deploy the applications:

1. Log in to the WebLogic Administration Console. Enter **Username** WebLogic and **Password**

admin123.

1. In the **Domain Structure** box, expand **Environment**.
2. Click **Servers**.



1. In the **Summary of Servers** box, click the **Control** tab.
2. Check the box by the server name.
3. Click the **Start** button.
4. Run the following commands in a terminal console.

cd /u01/app/domains/v10.prod/bin

./all\_deploy.sh

# Glossary

## STS Terminology Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Application Program Interface (API)** | An API is:   1. The interface or set of functions, between the application software and the application platform. 2. The means by which an application designer enters and retrieves information. |
| **archetype** | An archetype is:   1. A syntactically and semantically structured aggregation of vocabulary or other data that is the basic unit of clinical information. See also: template 2. A formal, reusable model of a concept for a given domain. |
| **attribute** | A named characteristic of a concept that can be assigned a value. See also: property (preferred). |
| **authoring** | The process of creating and editing terminology content. See also: development environment. |
| **candidate version** | Terminology Deployment Server (TDS) content that has passed internal testing and is sent to Software Quality Assurance (SQA) for quality assurance testing. |
| **change set** | A generic term for any terminology content that is deployed by TDS; specifically an Initial Deployment, a Candidate Version, or a Finalized Version. |
| **characteristic** | An attribute or behavior of something. See also: property. |
| **child** | The subtype in a parent-child relationship. The child (subtype) is narrower and more specific while the parent (supertype) is broader and more general. The child inherits the characteristics of the parent. |
| **classification** | Groupings of concepts for a given purpose where entries are found in one category. |
| **code set** | Any set of codes used for encoding data elements, such as tables of terms, medical concepts, medical diagnosis codes, or medical procedure codes. |
| **component** | An identifiable item in the main body of SNOMED CT or in an authorized extension. Components include: concepts, descriptions, relationships, subsets, histories, and extensions. |
| **Computerized Patient Record System (CPRS)** | The CPRS is the people, data, rules and procedures, processing and storage devices, and communication and support facilities that provide the capture, storage, processing, communication, security, and presentation of computer-based patient record information. |
| **concept** | An abstract unit of thought. |
| **concept equivalence** | Concept equivalence occurs when two concepts have the same meaning. |
| **concept to concept linking** | Concept to concept linking is when one concept is explicitly associated with another concept. Types of concept to concept linking are the creation of Map Sets, Translation Services, and Pre and Post Coordinated terms. |
| **context** | A context can be:   1. The environment in which it is appropriate to display a specific designation for a concept. 2. A specified part or field of a patient record, application, protocol, query, or communication in SNOMED CT. |

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **data cleanup** | Activities that are taken to correct, normalize, and eliminate terms from a reference file before it is matched to a new standard. See also: standardization. |
| **data model** | A schema that describes the way data is represented. |
| **data standardization** | The process of defining, creating, deploying, and maintaining a common terminology resource. |
| **datatype** | A data storage format that can contain a specific type or range of values. |
| **deploy** | Deploy means:   1. Within general software development, to send electronically as a unit. 2. Within STS, to publish terminology content from the development to production environments. |
| **deployment** | A deployment is:   1. The process of publishing terminology content from the development environment to the production environment. 2. Groups of concepts that are ready to be tested and potentially added to the terminology. |
| **description** | The text that represents a concept in human readable form. See also: designation (preferred). |
| **designation** | A representation of a concept. See also: description, display form, expression, surface form, term |
| **development environment** | All the software and hardware components needed to create or edit a terminology. See also: authoring. |
| **display form** | A representation of a concept. See also: designation (preferred), description, expression, surface form, term. |
| **domain** | A domain is:   1. A specialized discipline of medicine. 2. A set of terms belonging to a specialized discipline of medicine. 3. A set of terms associated within a VistA application. |
| **entity relationship model** | A graphical representation of work or information flow. Consists of entities (things), attributes (data), and relationships (connections between entities). Often used to model basic work or information flow. See also: information model, terminology model. |
| **Enterprise Terminology Services (ETS)** | The term ETS is no longer used. This team is now referred to as Standardization and Terminology Services (STS). |
| **expression** | Human readable representation of a concept or the name of a concept. See also: designation (preferred), description, surface form. |
| **finalized version** | TDS content that has passed SQA testing and is sent to production sites for field use. |
| **Health Data Repository (HDR)** | The HDR is a repository of clinical information normally residing on one or more independent platforms for use by clinicians and other personnel in support of patient- centric care. |

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Health Level Seven (HL7)** | HL7 is:   1. One of the American National Standards Institute (ANSI) accredited Standards Developing Organizations (SDO) operating in the healthcare arena. 2. An interoperability specification for transactions produced and received by computer systems. |
| **homophone** | One of two or more words pronounced alike but different in meaning, derivation, or spelling. |
| **homonym** | One of two or more words spelled and pronounced alike but different in meaning. |
| **International Classification of Diseases – 9th edition** (**ICD-9)** | ICD-9 classifies morbidity and mortality information for statistical purposes and for indexing of hospital records by disease and operations for data storage and retrieval. |
| **International Classification of Diseases – 9th edition – Clinical Modification (ICD- 9-CM)** | ICD-9-CM is a clinical modification of the World Health Organization’s ICD-9. It purpose is to classify morbidity data for indexing medical records, medical care review, and ambulatory and other medical care programs as well as for basic health statistics. |
| **initial deployment** | TDS content that has passed initial review and is sent to testing sites for internal evaluation. |
| **Internal Entry Number (IEN)** | A number used to identify an entry within a file. Every record has a unique internal entry number. In a VistA file, an IEN is a numerical identifier. |
| **information model** | A structured specification, expressed graphically and/or narratively, of the information requirements of a domain. An information model describes the required classes of information and the properties of those classes, optionally including attributes, relationships, and other essential information. See also: entity relationship model, terminology model |
| **lexicon** | A lexicon is:   1. The vocabulary of a language. See: terminology. 2. Commonly used to refer to VistA’s Lexicon Utility. |
| **Logical Observation Identifiers, Names, And Codes (LOINC)** | The LOINC database provides a set of universal names and ID codes for identifying laboratory and clinical observations. LOINC codes are used to facilitate the exchange and pooling of clinical laboratory results, such as blood hemoglobin or serum potassium, for clinical care, outcomes management, and research. |
| **map entry** | The link between concepts from a source code system to one or more concepts from a target code system. Map entries may be from two standard code systems or from within the same code system. A map entry is an instance of the data in a map set. |
| **map entry order** | The numeric order of the target code(s) for a source code. |
| **map set** | A collection of map entries with associated metadata. |
| **metadata** | Attributes that describe the format and content of information to enable sharing of information between users and applications. |
| **modifier** | A word or phrase associated with a concept that changes its meaning. |
| **nomenclature** | A system of names and groupings, which is structured according to pre-established naming rules. See also: classification, taxonomy |
| **non-domain** | Content that is not part of a clinical domain. |

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **non-VistA** | Content that is not deployed to VistA. |
| **normalization** | The process of identifying lexical variations of concepts that may include identification of synonyms. |
| **ontology** | Ontology is:   1. An explicit formal specification of how to represent the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. See also: terminology 2. All terms in a domain including the relationships among them. |
| **parent** | The supertype in a parent-child relationship. The child (subtype) is narrower and more specific while the parent (supertype) is broader and more general. The child inherits the characteristics of the parent. |
| **partial deployment** | Deploying one or more subsets within a Version instead of deploying the entire Version. |
| **post-coordination** | The representation of a complex concept as a combination of two or more concepts. See also: pre-coordination. |
| **pre-coordination** | The representation of a complex concept as a single concept. See also: post-coordination. |
| **preferred term** | The preferred human readable representation of a concept or the preferred name of a concept. Often used as the default display form of a concept. Synonyms: preferred designation, preferred expression. |
| **production environment** | The software and hardware that is used by end users, as opposed to developers and testers, to access terminology services in the VHA enterprise. |
| **property** | A named characteristic of a concept that can be assigned a value. |
| **qualifier** | A word or phrase associated with a concept that does not change its meaning. |
| **reference file** | Non-patient VistA data file that contains reference or Terminology information not Patient Data. |
| **reference terminology** | Reference terminology is:   1. A comprehensive, consistent, and logically organized set of concepts that is designed to completely embody the expressive detail of a given domain, supported by a set of relationships that defines the elements within the domain and shows how their meanings relate to each other. 2. A controlled medical vocabulary intended for use as a reference to enable storage, retrieval, and analysis of clinical data. |
| **relationship** | An association between concepts. See also: semantics, semantic relationship. |
| **Standards Development Organization (SDO)** | Any entity whose primary activities are developing and maintaining standards that address the interests of a wide base of users outside the standards development organization |
| **semantics** | The meanings assigned to terminology content. See also: semantic relationship. |
| **semantic relationship** | An association between two concepts that has a specific meaning. |
| **service oriented architecture (SOA)** | The VistA architecture is an SOA whereby applications that provide functionality for use by other applications are created as a service that conforms to a set of VHA standardized design patterns. |

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Systemized Nomenclature of Medicine (SNOMED) Clinical Terms (CT)** | SNOMED CT is a dynamic, scientifically validated clinical reference terminology that makes health care knowledge more usable and accessible. |
| **standard code system (SCS)** | An organized collection of terms or concepts established by an authoritative source such as an SDO. |
| **standardization** | The process of defining, creating, deploying, and maintaining a common terminology resource. |
| **Standards and Terminology Services (STS)** | STS includes project teams that were previously known as Data Standardization (DS) and ETS as well as the VETS and Enterprise Reference Terminology (ERT) subproject teams. |
| **subset** | A collection of concepts or designations that share a specified purpose or set of characteristics. |
| **subtype** | The child in a parent-child relationship. The subtype (child) is narrower and more specific while the supertype (parent) is broader and more general. The subtype contains all the characteristics of the supertype. |
| **supertype** | The parent in a parent-child relationship. The supertype (parent) is broader and more general while the subtype (child) is narrower and more specific. All the characteristics of the supertype are included in the subtype. |
| **surface form** | The term that 3M uses for a human readable representation of a concept, or the name of a concept. See also: designation (preferred). |
| **synonym** | A term or an expression that is an acceptable alternative to the preferred designation. |
| **taxonomy** | A hierarchical classification of concepts. |
| **template** | A template is:   1. A structured aggregation of one or more archetypes, with optional order, to represent clinical data. An HL7 template is a data structure, based on the HL7 RIM that expresses the data content that is needed in a specific clinical or administrative context. Templates are drawn from the RIM and make use of HL7 vocabulary domains. Templates are also described as constraints on HL7 artifacts. 2. A locally produced constraint specification that specifies which archetypes go together in an application dialog or message specification. |
| **term** | A human readable representation of a concept or name of a concept. See also: designation (preferred). |
| **terminology** | Set of concepts, designations, and relationships for a specialized subject area. The terms that are characterized by special reference within a discipline are called the terms of the discipline and, collectively, they form the terminology. Terms that function in general reference over a variety of languages are simply words and their totality is a vocabulary. |
| **terminology deployment services** | Central distribution point for all terminology services. Updates are uploaded to the terminology deployment server, which in turn distributes them to targeted VistA sites. |
| **terminology model** | A terminology model provides a consistent structure and specifies the formal representation of a concept. The STS terminology model comprises of components such as concepts, designations, properties, and relationships. Other components of the STS terminology model include Subsets and Concept to Concept linking. |
| **terminology server** | The software application and hardware that provide access to terminology content through a published set of API. |

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **test environment** | The software and hardware that is used by developers and testers as opposed to end users to test terminology services in the VHA enterprise. |
| **translation** | After two terminologies have been mapped, a translation between the two is possible. |
| **Unified Medical Language System (UMLS)**  **Metathesaurus** | The UMLS Metathesaurus is a very large, multi-purpose, and multi-lingual vocabulary database that contains information about biomedical and health related concepts, their various names, and the relationships among them. It reflects and preserves the meanings, concept names, and relationships from its source vocabularies. It also supplies information that computer programs can use to create standard data, interpret user inquiries, interact with users to refine their questions, and convert the users' terms into the vocabulary used in relevant information sources. |
| **value** | A quantitative or qualitative state that is assigned to a property. |
| **value domain** | All allowable values for a terminology, datatype, or value set. May be an infinite set of values. |
| **value set** | A finite set of allowable values. Typically, a value set has a small number of values. If it has a large number of values, it may be a terminology. |
| **version** | A version is:   1. Formal changes in a terminology. May be used to find and track inactivated codes, determine the current code set, or track the history of a concept. 2. Also applies to formal revisions in computer code or programs. 3. An STS deployment that has passed internal testing. Can refer to a Candidate Version or a Finalized Version. |
| **Veterans Health Administration (VHA) Enterprise Terminology Services (VETS)** | VETS focuses on requirements for the deployment of and runtime access to terminology content in ERT for all VHA clinical applications. |
| **VHA Terminology (VHAT)** | VHAT is the terminology that is created and maintained by STS. |
| **Veterans Health Information Systems and Technology Architecture (VistA)** | VistA is a term used to describe the VA’s health care information system. It encompasses in-house developed applications developed by VA staff, office automation applications, locally developed applications, and commercial-off-the-shelf applications. |
| **vocabulary** | A list of words or phrases with their meanings. See also: terminology. |
| **Web Services Description Language (WSDL)** | WSDL is an XML-based language that provides a model for describing Web services. The meaning of the acronym has changed from version 1.2 where the D meant Definition. |

# Appendix A – WebLogic Server Installation Configuration

File wls1032.xml content:

<?xml version="1.0" encoding="UTF-8"?>

<bea-installer>

<input-fields>

<data-value name="BEAHOME" value="/u01/app/wls1032"/>

<data-value name="WLS\_INSTALL\_DIR" value="/u01/app/wls1032/wls"/>

<data-value name="COMPONENT\_PATHS" value="WebLogic Server/Core Application Server|WebLogic Server/Administration Console|WebLogic Server/Configuration Wizard and Upgrade Framework|WebLogic Server/WebLogic Web Server Plugins|WebLogic Server/UDDI and Xquery Support"/>

<data-value name="LOCAL\_JVMS" value="/u01/app/jdk1.6"/>

</input-fields>

</bea-installer>

# Appendix B – WebLogic Domain Installation Configuration

File basedomain.py content: readTemplate('**/u01/app/wls1032/wls**/common/templates/domains/wls.jar') create ('**vhaislbll25.vha.med.va.gov**','Machine')

set('AdminServerName', ‘admin.v10.Prod’) cd('/Server/AdminServer')

set('Name', ‘admin.v10.Prod’)

cd('/Server/'+’admin.v10.Prod’) set('ListenAddress', 'REDACTED**0**')

set('ListenPort', REDACTED)

cd('/Security/base\_domain/User/weblogic') set('Name','**weblogic**') cd('/Security/base\_domain/User/'+'**weblogic**') cmo.setPassword('**admin123**')

writeDomain('**/u01/app/domains/v10.Prod**') closeTemplate()

# Appendix C – Start/Stop and Deploy Scripts

Script start\_sts.deployment.sh content

echo Starting weblogic server v10.deployment ...

/u01/app/domains/v10.prod/bin/nodemanager/wlscontrol.sh \

-d v10.prod \

-r /u01/app/domains/v10.prod \

-s v10.deployment START

Script stop\_sts.deployment.sh content

/u01/app/domains/v10.prod/bin/nodemanager/wlscontrol.sh \

-d v10.prod \

-r /u01/app/domains/v10.prod \

-s v10.deployment KILL

Script bounce\_sts.deployment.sh content

/u01/app/domains/v10.prod/bin/stop\_sts.deployment.sh sleep 30

/u01/app/domains/v10.prod/bin/start\_sts.deployment.sh Script deploy\_sts.deployment.sh content

cd /u01/app/domains/v10.prod/bin

. /u01/app/wls1032/wls/server/bin/setWLSEnv.sh

/u01/app/jdk1.6/bin/java -Duser.home=.security \

weblogic.Deployer -adminurl t3://10.5.20.180:7200 -stop -name sts.deployment

/u01/app/jdk1.6/bin/java -Duser.home=.security \

weblogic.Deployer -adminurl t3://10.5.20.180:7200 -undeploy -name sts.deployment

./bounce\_sts.deployment.sh

/u01/app/jdk1.6/bin/java -Duser.home=.security \

weblogic.Deployer -adminurl t3://10.5.20.180:7200 -deploy -name sts.deployment \

-source /u01/app/domains/v10.prod/apps/sts.deployment.war \

-targets v10.deployment -nostage Script all\_deploy.sh content

/u01/app/domains/v10.prod/bin/deploy\_sts.deployment.sh

/u01/app/domains/v10.prod/bin/deploy\_sts.browser.sh

/u01/app/domains/v10.prod/bin/deploy\_ntrt.sh

/u01/app/domains/v10.prod/bin/deploy\_sts.ted.sh

/u01/app/domains/v10.prod/bin/deploy\_vuid.sh

# Appendix D – STS V10 Detailed Configuration

[Host] Name=vhaislbll25

Domain=.vha.med.va.gov FQDN=%(Name)s%(Domain)s

IP=10.5.20.180

AppDir=/u01/app JavaHome=%(AppDir)s/jdk1.6 BeaHome=%(AppDir)s/wls1032 WlsHome=%(BeaHome)s/wls WlsInstaller=wls1032\_generic.jar

[Domain] Tag=v10

BaseDir=@@Host\_AppDir@@/domains Name=%(Tag)s.prod Home=%(BaseDir)s/%(Name)s PortGroup=72

Serial=0 ProductionModeEnabled=true

[NodeManager] ListenPort=5556

[AdminServer] Serial=00

Name=admin.@@Domain\_Name@@ ListenAddress=@@Host\_IP@@ ListenPort=@@Domain\_PortGroup@@00 SSL\_ListenPort=@@Domain\_PortGroup@@99 AdministrationPort=@@Domain\_PortGroup@@88 Url=t3://@@Host\_IP@@:%(ListenPort)s SSL\_Url=t3s://@@Host\_IP@@:%(AdministrationPort)s

Start\_Arg=-server -Xms360m -Xmx360m -XX:PermSize=128m -XX:MaxPermSize=256m \

-Xverify:none -Dcds.jndi.provider.url=%(Url)s

[User] Admin\_Name=weblogic Admin\_Pass=admin123 App\_Group=DSAdmin App\_Users=sts.admin App\_User\_Pswd\_Char=

[ManagedServer] SerialList=01,02,03,04,05

Start\_ArgMem=-Xms640m -Xmx640m -XX:PermSize=128m -XX:MaxPermSize=384m -server - Xverify:none

Start\_JavaHome=@@Host\_JavaHome@@ Start\_JavaVendor=Sun Start\_BeaHome=@@Host\_BeaHome@@ Start\_RootDirectory=@@Domain\_Home@@ Start\_Username=@@User\_Admin\_Name@@ Start\_Password=@@User\_Admin\_Pass@@ Start\_ClassPath\_A=%(Start\_RootDirectory)s/lib/antlr-2.7.6.jar

Start\_ClassPath\_B=@@Host\_WlsHome@@/server/lib/weblogic.jar Start\_ClassPath\_C=%(Start\_RootDirectory)s/sts.config [[need create dir]] Start\_ClassPath=%(Start\_ClassPath\_A)s:%(Start\_ClassPath\_B)s:%(Start\_ClassPath\_C)s

Name01=deployment ListenAddress01=@@Host\_IP@@ ListenPort01=@@Domain\_PortGroup@@01 SSL\_ListenPort01=@@Domain\_PortGroup@@11 AdministrationPort01=@@Domain\_PortGroup@@21 Machine01=@@Host\_FQDN@@ Url01=t3://@@Host\_IP@@:%(ListenPort01)s

Start\_ArgMem01=-Xms2048m -Xmx2048m -XX:PermSize=128m -XX:MaxPermSize=384m -server -Xverify:none Start\_Arg01=%(Start\_ArgMem01)s -Dcds.jndi.provider.url=%(Url01)s

Name02=browser ListenAddress02=@@Host\_IP@@ ListenPort02=@@Domain\_PortGroup@@04 SSL\_ListenPort02=@@Domain\_PortGroup@@14

AdministrationPort02=@@Domain\_PortGroup@@24 Machine02=@@Host\_FQDN@@ Url02=t3://@@Host\_IP@@:%(ListenPort02)s Start\_Arg02=%(Start\_ArgMem)s -Dcds.jndi.provider.url=%(Url02)s

Name03=ntrt ListenAddress03=@@Host\_IP@@ ListenPort03=@@Domain\_PortGroup@@02

SSL\_ListenPort03=@@Domain\_PortGroup@@12 AdministrationPort03=@@Domain\_PortGroup@@22 Machine03=@@Host\_FQDN@@ Url03=t3://@@Host\_IP@@:%(ListenPort03)s Start\_Arg03=%(Start\_ArgMem)s -Dcds.jndi.provider.url=%(Url03)s

Name04=ted ListenAddress04=@@Host\_IP@@ ListenPort04=@@Domain\_PortGroup@@08

SSL\_ListenPort04=@@Domain\_PortGroup@@18 AdministrationPort04=@@Domain\_PortGroup@@28 Machine04=@@Host\_FQDN@@ Url04=t3://@@Host\_IP@@:%(ListenPort04)s Start\_Arg04=%(Start\_ArgMem)s -Dcds.jndi.provider.url=%(Url04)s

Name05=vuid ListenAddress05=@@Host\_IP@@ ListenPort05=@@Domain\_PortGroup@@03

SSL\_ListenPort05=@@Domain\_PortGroup@@13 AdministrationPort05=@@Domain\_PortGroup@@23 Machine05=@@Host\_FQDN@@ Url05=t3://@@Host\_IP@@:%(ListenPort05)s Start\_Arg05=%(Start\_ArgMem)s -Dcds.jndi.provider.url=%(Url05)s

[JDBC]

SerialList=01,02,03 Url=jdbc:oracle:thin:@vhaislbll25.vha.med.va.gov:1521:stsd01 DriverName=oracle.jdbc.OracleDriver KeepXaConnTillTxComplete=

XaEndOnlyOnce=true XaSetTransactionTimeout=true

XaRetryDurationSeconds=300 XaTransactionTimeout=120 InitialCapacity=1 CapacityIncrement=1 MaxCapacity=15 TestConnectionsOnReserve=true

TestTableName=SQL SELECT 1 FROM DUAL GlobalTransactionsProtocol=TwoPhaseCommit

Name01=Deployment User01=ds\_@@Domain\_Tag@@ Password=.........

Password\_Key01=c2s106 JNDINames01=jdbc/gov.va.med.term.deployment Target01=@@ManagedServer\_Name01@@

Name02=VETS

User02=vts\_@@Domain\_Tag@@ Password=.........

Password\_Key02=c2s92 JNDINames02=jdbc/gov.va.med.term.services

Target02=@@ManagedServer\_Name01@@,@@ManagedServer\_Name02@@,\ @@ManagedServer\_Name03@@,@@ManagedServer\_Name04@@,\ @@ManagedServer\_Name05@@

Name03=NTRT

User03=ntrt\_@@Domain\_Tag@@ Password=.........

Password\_Key03=c2s145 JNDINames03=jdbc/gov.va.med.term.ntrt Target03=@@ManagedServer\_Name03@@ GlobalTransactionsProtocol03=OnePhaseCommit

[Log] BaseDir=@@Domain\_BaseDir@@/logs RotationType=byTime NumberOfFilesLimited=false RotateLogOnStartup=false