



Energy Portfolio Management

Server Installation Guide

Release 5.7.0.0

EMDDB-0226-2003-14

March 2020

© Copyright 2020 ABB

All Rights Reserved
Confidential and Proprietary

Legal Disclaimer

The product described in this documentation may be connected to, and/or communicate information and data via, a network interface, which should be connected to a secure network. It is your sole responsibility to ensure a secure connection to the network and to establish and maintain appropriate measures (such as but not limited to the installation of firewalls, application of authentication measures, encryption of data, installation of antivirus programs, etc.) to protect the product, the network, your systems, and the interface against any kind of security breach, unauthorised access, interference, intrusion, leakage, damage, or corruption or theft of data. We are not liable for damages or losses related to any such security breach, unauthorised access, interference, intrusion, leakage, damage, or corruption or theft of data.

Contents

Chapter 1: Installation	4
First Steps.....	4
EPM Ops Server Installation Program.....	5
Java Development Kit (JDK)	5
Important Oracle JDK License Update	5
Database Installation	6
Database Components and Features	6
Processor	6
Memory (SGA/PGA).....	6
INIT.ORA Parameters	7
Tablespaces	7
Undo/Rollback Tablespace	7
Redo Logs	7
Database Maintenance	8
Backup/Recovery	8
Configuration of Oracle-accessible directories	8
Application Schema Installation.....	9
Creating the User	9
Tablespaces and Quotas	9
Required Privileges	9
Applying Core EPM Ops Schema Objects.....	10
Applying Market Operations Schema Objects	10
Application Schema – First Use	11
Updating Market Interfaces	12
Install J2EE Server (Apache Tomcat)	13
Installation	13
Configure Tomcat for SSL	17
Deploy and Configure Applications	21
Chapter 2: Tomcat Login Configuration	23
Standard Active Directory Configuration	23
Alternate Active Directory Configuration	25
LDAP Configuration	26
Chapter 3: Client Installation	27

Chapter 1: Installation

This document describes the steps required to install the Retail Operations/CSB/DRMS/Market Operations Server (from here on referred to as EPM Ops Server).

First Steps

Before you begin installing the EPM Ops Server, you should take the following steps:

1. Ensure hardware for the EPM Ops Server meets our recommended requirement guidelines. Obtain access to the hardware for the EPM Ops Server with Administrator privileges.
2. Obtain the names of Oracle tablespaces that will be used to store application data in the Oracle RDBMS server. By default, the application uses tablespaces named NERO_DATA and NERO_INDEX. A DBA can either create these tablespaces in the Oracle RDBMS server or you can choose to use different tablespaces (specified during installation of EPM Ops Server media).
3. Obtain the information required to authenticate against the network.
 - ♦ A new network security role/group named **VENTYX_ADMIN** must be created. This role is used to identify the set of users that are authorized to manage the Datasources for the application.
 - ♦ A new network user named **ventyxadmin** must be created that belongs to the **VENTYX_ADMIN** role/group.
 - ♦ Obtain the Active Directory or LDAP connection settings required to authenticate against the client network (see [Tomcat Login Configuration](#))
4. Run the [EPM Ops Server Installation Program](#). If the installation media is a single executable then it is a self-installing executable and you simply must run it. If the installation media is an optical disc or disc image, run the **SETUP.EXE** program found thereon. For optical discs, this program should automatically run when the media is inserted into a CD/DVD drive (unless "Auto-Run" functionality is disabled). For 64-bit Windows systems, the server installation program must be run As Administrator.

Note: It is recommended to have a malware prevention solution during operation of the product.

EPM Ops Server Installation Program

The EPM Ops Server Installation Program must be run before attempting the configuration described in this document.

The EPM Ops Server Installation Program will extract the necessary files to a location on the server that you specify. This location will be referred to in this document as the **Installation Directory**.

Once the EPM Ops Server Installation Program has been run, you will need to install the Database, install the Application Schema, install Tomcat and deploy the EPM Ops Server applications.

Java Development Kit (JDK)

The EPM Ops Server requires JDK version 11.

Important Oracle JDK License Update

The Oracle JDK License has changed for releases starting April 16, 2019.

The new [Oracle Technology Network License Agreement for Oracle Java SE](#) is substantially different from prior Oracle JDK licenses. The new license permits certain uses, such as personal use and development use, at no cost – but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. An FAQ is available [here](#).

Commercial license and support are available with a low-cost [Java SE Subscription](#).

Oracle also provides the latest **OpenJDK** release under the open source [GPL License](#) at [jdk.java.net](#).

Example JDK Options:

- Oracle JDK using Java SE Subscription:
<https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html>
- OpenJDK:
<https://jdk.java.net/java-se-ri/11>
- Azul Zulu Community JDK:
<https://www.azul.com/downloads/zulu-community/?&version=java-11-lts&os=&os=windows&architecture=x86-64-bit&package=jdk>

Database Installation

The purpose of this document is to specify only several important aspects of the database setup. This document is not a substitute for hardware and system-software documentation and installation manuals. Database configuration should follow Oracle-provided guidelines and recommendations, e.g. SAME, OFA, etc.

Database Components and Features

Database should include all features automatically selected by the default installation of the Oracle Standard Edition, including but not limited to JVM, XDB, etc. If installation of some feature is deemed undesirable, prior arrangements should be made to make sure that partial configuration is supported or work-around is available.

Oracle Database Enterprise Edition provides numerous options and prior analysis of necessary features is highly recommended.

Processor

It is recommended to use a fast processor with minimum of 2 cores (or 2 CPUs) at 3GHz or greater than 3 GHz. Additional processors for the server would increase performance.

Memory (SGA/PGA)

It is recommended 8GB of memory on the server.

It is recommended to use Automatic Shared Memory Management in 11.2.02.

Example for the Server that can provide about 8 GB of available memory for the Oracle instance:

- SGA Size: 6000 Mb PGA Size: 2000 Mb

If more memory can be allocated to the Oracle instance, SGA should be scaled up first. Increasing PGA beyond 0.5 GB only makes sense if instance is allowed to grow above 6-8 Gb of RAM and there is a justifiable need for the larger PGA (e.g. very large sorts).

It is extremely important to make sure that memory allocated to Oracle will not be paged by the OS. At least, following steps need to be taken:

1. Suppress excessive memory consumption by the OS file caching (e.g. on Windows Server platforms make sure that Maximize data throughput for file sharing option is **not** set).
2. Use platform-specific init.ora parameters to pin SGA in memory (e.g. LOCK_SGA or PRE_PAGE_SGA as appropriate). Similar flexibility is not always available for the PGA memory, but steps need to be taken to guarantee free RAM availability, or paging can be disabled altogether on large-RAM servers.

Note: Based on our experience, it is much safer to change memory locking parameters after the instance is built and running.

It is advisable to have ample unused memory on the server vs. allocating every last byte to the Oracle instance and risking paging of SGA or PGA.

INIT.ORA Parameters

Substantial augmentation of init.ora parameters is no longer necessary since introduction of 11.2.02 most of the automatic settings are reasonably efficient and tuning/tweaking can be done once instance is up and running. Following list contains few examples of parameters that may require tuning:

Parameter	Comments
CURSOR_SHARING	SIMILAR or FORCE if becomes necessary. There are some uncommon situations when this setting may be changed from EXACT, but that will become apparent based on hit ratios.
DB_WRITER_PROCESSES	May be increased, but I/O waits will become apparent
LOG_BUFFER	May be increased, but waits will be obvious

Once instance is operational, other parameters should be incrementally tuned based on the Oracle guidelines and built-in diagnostics.

Tablespaces

Name	Extent Management	Block Size
NERO_DATA	Automatic with maxsize set	8K
NERO_INDEX	Automatic with maxsize set	8K
NERO_LARGE_DATA	Uniform ~8M or more	8K or more
NERO_LARGE_INDEX	Uniform ~4M or more	8K or more

It is recommended to use the same block sizes for all tablespaces, 8K default is reasonable, but larger size may be used, if database is large and resources are plentiful (fast I/O, lots of RAM).

All tablespaces should be locally managed with automatic segment space management. Initially, all tables and indices should be placed into NERO_DATA or NERO_INDEX tablespace respectively. Every table/index that is expected to grow in size should be moved to the _LARGE_ tablespace. TEMP tablespace should be large enough to accommodate data-intensive and long running processes usually, 4GB or more.

It is highly advisable to follow the suggested naming convention for tablespaces. Also, it is advisable to monitor space usage in NERO_DATA and NERO_INDEX tablespaces and keeping the cumulative size of all objects under 500Mb per non-LARGE tablespace. Once all growing segments are identified and moved, typical Retail Operations schema will consume about 300Mb per non-LARGE tablespace and only LARGE ones will grow substantially. Space quotas may be placed on non-LARGE tablespaces to prevent unexpected growth or automatic datafile growth may be disabled.

Additional tablespaces may be created for very large and volatile segments (e.g. large partitioned tables), but they should be kept to a minimum 2-5 largest tables/indexes, if absolutely necessary.

Undo/Rollback Tablespace

Rollback segments should not be used anymore. The best approach is to setup a large (4 GB or more) UNDO tablespace and reduce/increase undo retention, if default is unacceptable (uncommon).

Redo Logs

Redo log members should be about 1Gb. Size can be scaled up or down based on the database monitoring, but 1Gb is a reasonable starting point for large instances.

Database Maintenance

Optimizer statistics must be gathered regularly, in order to maintain high level of performance. It is recommended to use automatically scheduled statistics gathering process in the Oracle 11g.

Some indexes and tables may have to be re-built/coalesced as necessary, according to the Oracle guidelines and recommendations (uncommon).

Backup/Recovery

Appropriate steps should be taken in order to implement a reliable backup/recovery strategy. Retail Operations is fully compatible with all Oracle-provided and third-party tools.

Configuration of Oracle-accessible directories

The account (OS user) under which Oracle database instance runs should have full access to at least one shared directory and all its sub-directories. In addition, application end-users and/or automated interfaces must have access to the same directory structure.

Typically, such configuration involves configuring a folder on a file server and granting the Oracle account the necessary level of access. Similarly, users can be granted various levels of access. Such configuration is possible on all platforms and network infrastructures, but actual steps are very much platform specific. For example: it may be necessary to run Oracle instance under a valid domain account on Windows platforms, or Oracle user on UNIX may require access via Samba to a shared folder on a Windows file server.

One of the local folders on the database server itself may also be used as a shared location for processing files, but such configuration is only suitable for test or very restricted small-volume production systems. Granting any level of access to the database server to end-users may be prohibited by IT security policies.

Such configuration is essential for efficient and reliable file-based interfaces, since all processes and business logic reside within the database. Configuration of accessible shared folders may not be necessary if implementation does not include any automated file-based interfaces.

Application Schema Installation

This page describes how to create a new application schema. It covers details needed by a DBA for creating the new Oracle schema as well as instructions for building all application tables and objects and compiling the application into the Oracle schema.

Creating the User

The application schema is an Oracle user that owns all the application tables, objects, and PL/SQL code. This user requires several privileges for the application build scripts to succeed. There are also several privileges required for the application to execute properly:

Tablespaces and Quotas

The application schema must have permission to create tables and objects in the NERO_DATA, NERO_INDEX, NERO_LARGE_DATA, and NERO_LARGE_INDEX tablespaces. The user should be allowed 300mb or more in the NERO_DATA and NERO_INDEX tablespaces. Since the LARGE tablespaces will grow over time, an unlimited quota may be appropriate.

See [Database Installation](#) for more details on tablespace requirements.

It is strongly advised that the application schema's default tablespace be set to **NERO_DATA**.

Required Privileges

The following privileges must be granted from a DBA to the application schema:

Privilege	Explicit Grant?	Optional?
CREATE SESSION	N	N
DEBUG CONNECT SESSION	N	Y
CREATE TABLE	Y	N
CREATE TRIGGER	Y	N
CREATE VIEW	N	N
CREATE SEQUENCE	N	N
CREATE TYPE	N	N
CREATE SYNONYM	N	N
CREATE PROCEDURE	Y	N
CREATE JOB	Y	N
SELECT ON SYS.V_\$LOCK	Y	N
SELECT ON SYS.V_\$PX_PROCESS	Y	N
SELECT ON SYS.V_\$SESSION	Y	N
EXECUTE ON SYS.DBMS_LOCK	Y	Y

The second column indicates whether the grant can be made via an Oracle role or whether it must be explicitly granted directly to the application schema user. Explicit grants are required for functionality performed by stored procedures (because roles are not in effect when stored procedures are executed in Oracle).

The third column indicates optional privileges. For instance, access to the DBMS_LOCK privilege is optional. If this privilege is **not** granted, then the application must be configured to skip synchronization logic that requires the privilege. The DEBUG privilege is optional and would generally only be needed in a Development or Test environment for troubleshooting purposes.

Applying Core EPM Ops Schema Objects

Once the user is created and has been granted necessary privileges, the schema objects (tables, triggers, PL/SQL procedures and packages, etc.) must be created. These are easily constructed using the supplied scripts. The server setup program will install the scripts into a folder that, by default, is named:

```
C:\Program Files\Ventyx\[Product]Server\vm.n.r\OracleScripts
```

where {Product} is the name of the Product (RetailOperations, CSB, DRMS or MarketOperations),

where **m.n** indicates the installed version of the application and **r** is the Release. If there is no Release number, it implies that it is 0(zero).

Using SQL *Plus or another Oracle client program, connect to the application schema, and then execute the following scripts:

```
[OracleScripts Folder]\System\crebas.sql
```

This script will build core tables, indexes, and constraints.

```
[OracleScripts Folder]\System\build.sql
```

This script compiles all PL/SQL objects and loads initial set of data (default configuration, default entities, etc.)

Applying Market Operations Schema Objects

If the product is Market Operations, then we need to apply Market Operations objects to the schema. Before running this step, recompile any invalid objects. Close the previous SQL *Plus or Oracle client program session and open a new session to connect to the application schema. Execute the following scripts:

```
[OracleScripts Folder]\MarketManager\Common\crebas.sql
```

This script will build Market Operations tables, indexes, and constraints.

```
[OracleScripts Folder]\MarketManager\Common\build.sql
```

This will take care of compiling PL/SQL objects for Market Operations.

Before running the next step, recompile any invalid objects. Close the previous SQL *Plus or Oracle client program session and open a new session to connect to the application schema. For SEM Market configuration, run the following script:

```
[OracleScripts Folder]\MarketManager\SEM\SetupMarket.sql
```

This will run configuration for SEM market. For TDIE Market configuration, run the following script:

```
[OracleScripts Folder]\MarketManager\TDIE\SetupMarket.sql
```

This will run configuration for TDIE market.

Application Schema – First Use

Before you can connect to the new schema, you must configure it in the web application. This is done via the Datasources Setup screens.

The first time you connect to the new application schema, you must login as **ventyxadmin**. This will be the only valid application user. Once connected, you can add additional users via the Admin screens. After adding new users and assigning them to roles, those users can then login to the application.

Note: The application schema is configured with four system users: MailMonitor, ProcQueuesMonitor, Reactor, and System. These users are for various system processes and cannot be used to login. It is recommended not to modify these system users.

Before using the application, there are some other initial administration tasks that should be considered:

- Numerous reports in the user interface use a special database table named **SYSTEM_DATE_TIME**. This table, by default, will not be populated.
 - ♦ Use the **Admin Data Exchange** option named "Populate System-Date-Time Table" to populate this table. You will need to choose all Time Zones that users are expected to use when querying reports. You will also need a sufficiently long time horizon to minimize the frequency of re-populating this table. You should set the Begin Date to the earliest day for which reports will be run in the application. You can set the End Date to a date in the future. Once users need to start reporting on days after your chosen End Date, you will need to re-populate this table and choose a later End Date.
- By default, all scheduled jobs are disabled. Use the **Background Job Status** view to enable jobs as needed:
 - ♦ **LOB_STAGING_CLEANUP_JOB**
The job named LOB_STAGING_CLEANUP_JOB should be enabled. This is a general maintenance job that insures certain temporary staging tables get cleaned up on a regular basis. With this job disabled, the application may retain old and unneeded temporary staging information used by certain functions.
 - ♦ **PROCESS_EVENT_CLEANUP_JOB**
The job named PROCESS_EVENT_CLEANUP_JOB should be enabled if the Process Log is expected to grow very large. This job compacts the log, removing less important messages/data associated with older processes. Configuration of the retention rules is done via the System Dictionary.
 - ♦ **MAIL_MONITOR_JOB**
The job named MAIL_MONITOR_JOB **must** be enabled if the application will be used to send e-mails – including the use of e-mail alerts.
 - ♦ **APPLY_AUTO_DATA_LOCKS_JOB**
The job named APPLY_AUTO_DATA_LOCKS_JOB **must** be enabled if Automatic Data Lock Groups will be used. With this job disabled, automatic data locking will not occur.
 - ♦ **REACTOR_JOB**
The job named REACTOR_JOB **must** be enabled if the Reactor will be used. Certain interfaces, like synchronization between ABB Retail Operations and Schedule Management, require the Reactor.
 - ♦ **PROCESS_QUEUES_MONITOR_JOB**
The job named PROCESS_QUEUES_MONITOR_JOB should be enabled if Job Threads will be used. With this job disabled, significant runtime errors in a database job could cause a Job Thread to stall. The status of Job Threads can be monitored in the user interface from the Process Queue Monitor.

- By default, execute privileges on the **DBMS_LOCK** package is required. If you choose not to grant this database permission then you must apply certain configuration options. Otherwise, runtime errors may occur as a result of not having access to DBMS_LOCK.
 - ♦ Job Threads **cannot** be used without privileges to DBMS_LOCK. The logic that allows synchronization/serialization of job threads uses Oracle locks.
 - ♦ Serializing the execution of Calculation Processes also relies on Oracle locks. This option must be disabled.

Updating Market Interfaces

If the product is Market Operations, then update Market Interfaces by taking the following steps:

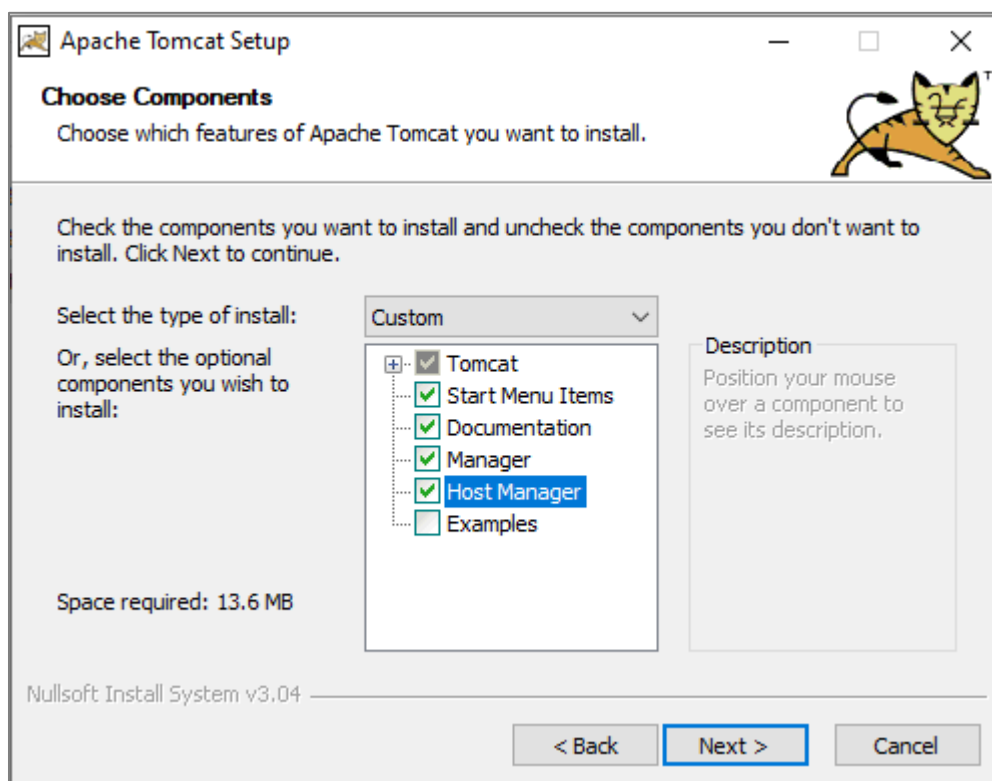
- Login to the application as an administrator and click the *Admin* link from the home page tree. Navigate to the tab named *ROML Publish and Subscribe* and click the button labeled **Import from File**.
- Import one or more ROML files found in ***MarketManager\SEM*** and ***MarketManager\TDIE*** sub-folder of the OracleScripts installed from the server setup.

Install J2EE Server (Apache Tomcat)

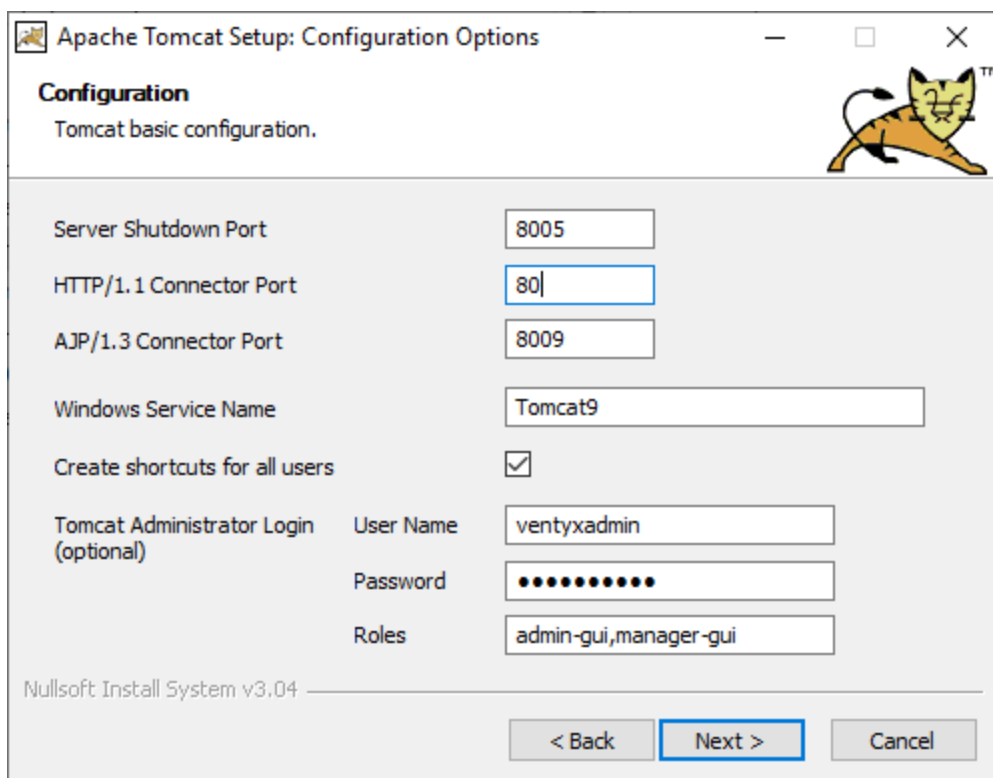
The EPM Ops Server installation requires Apache Tomcat 9.

Installation

1. Download **JDK 11** and complete the installation process using the defaults. If the JDK distribution does not contain an installer (e.g., OpenJDK), unzip the distribution to where you want it to reside.
2. Unzip the **CustomTomcat.zip** file included in the **Installation Directory**. This will contain the files required for customizing the Apache Tomcat installation. You can unzip this to any directory (from here referred to as **Custom Tomcat Directory**).
3. Download an Apache Tomcat 9 installation appropriate to your OS from <https://tomcat.apache.org/download-90.cgi>. For 64-bit Windows machines, the 32-bit/64-bit Windows Service Installer is a preferred choice.
4. Once downloaded, run the installer executable.
5. Customize the Apache Tomcat setup by referring to the following screenshots:
 - a. Select the components shown below:



b. Configure Tomcat using the following settings:



Apache Tomcat Setup: Configuration Options

Configuration
Tomcat basic configuration.

Server Shutdown Port: 8005

HTTP/1.1 Connector Port: 80

AJP/1.3 Connector Port: 8009

Windows Service Name: Tomcat9

Create shortcuts for all users: ☒

Tomcat Administrator Login (optional)

User Name: ventyadmin

Password: 13579

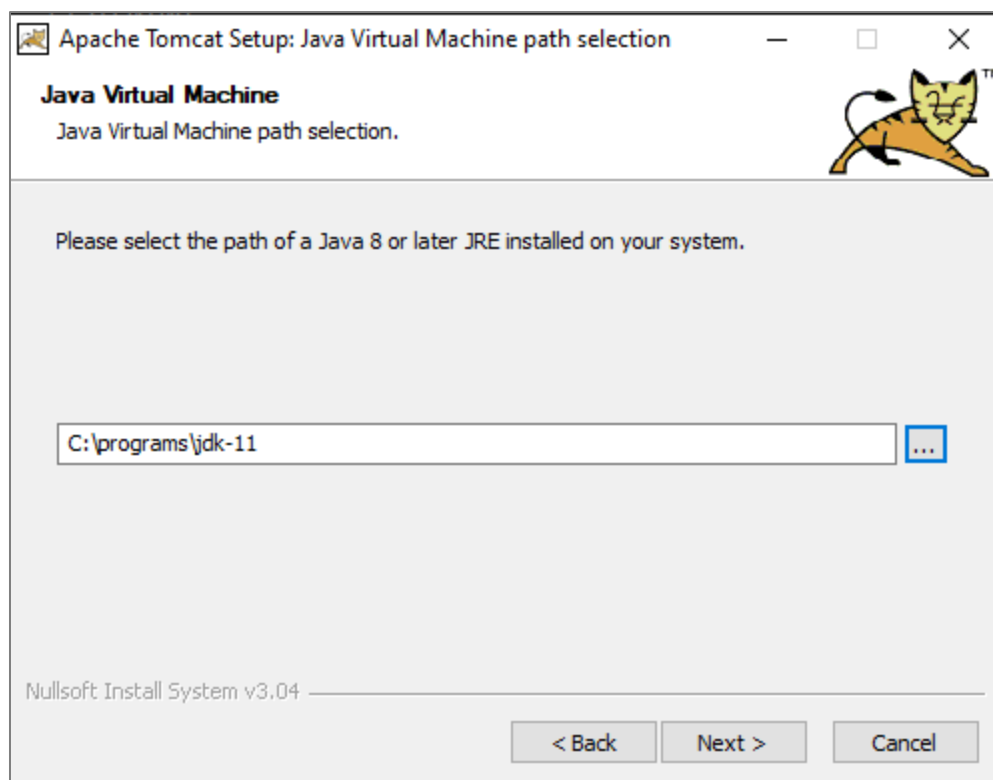
Roles: admin-gui,manager-gui

Nullsoft Install System v3.04

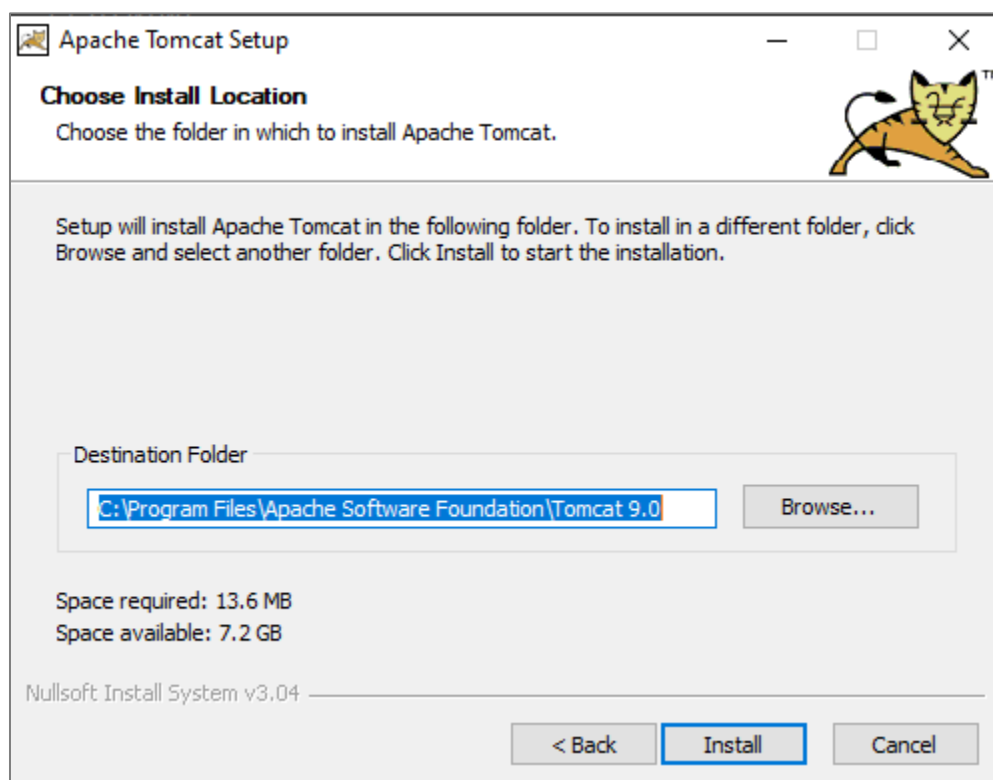
< Back Next > Cancel

- Set the **HTTP/1.1 Connector Port**. ABB recommends Port **80** as shown in the screenshot. Please note that this may conflict with another service on the physical machine. You must make sure that the port is available for use. For example, many Windows Server machines already have IIS setup and running on Port 80. If you are not using IIS, then you can disable it.
- Check the "Create shortcuts for all users". This is required for starting the Apache Tomcat Monitor application later.
- Specify the Tomcat Administrator Login User Name and Password ventyadmin/Admin13579.

- c. Select the JDK path that was used in step 1 during the JDK installation.



- d. Select the installation directory (from here referred to as the **Tomcat Installation Directory**). The default value will work.



6. When the installation is complete uncheck the "Run Apache Tomcat" and "Show Readme" check boxes and click **Finish**.
7. Ensure that the Tomcat server is stopped. This can be checked by going to: Windows Control Panel > Administrative Tools > Services. You should see the Apache Tomcat service stopped. We will now begin copying files from the **CustomTomcat Directory** to the **Tomcat Installation Directory**.
8. Copy the 3 files from the **CustomTomcat Directory \conf files** folder to **Tomcat Installation Directory\conf** directory (Copy and Replace existing files).
9. Delete the folder **Tomcat Installation Directory\work\Catalina** if it exists.
10. Copy jar files from the "CustomTomcat Directory \lib" folder to "Tomcat Installation Directory\lib".
11. Copy the text in **CustomTomcat Directory \JVM configuration.txt** file to the clipboard.

Note: There are several places in this text file that reference the **Tomcat Installation Directory** as "C:\Program Files\Apache Software Foundation\Tomcat 9.0". These will need to change if Tomcat was installed in any other location during step 5.d.

12. Turn on the Apache Tomcat Monitor application. This can be done by going to Start menu > Programs > Apache Tomcat 9.0 ... > Monitor Tomcat.

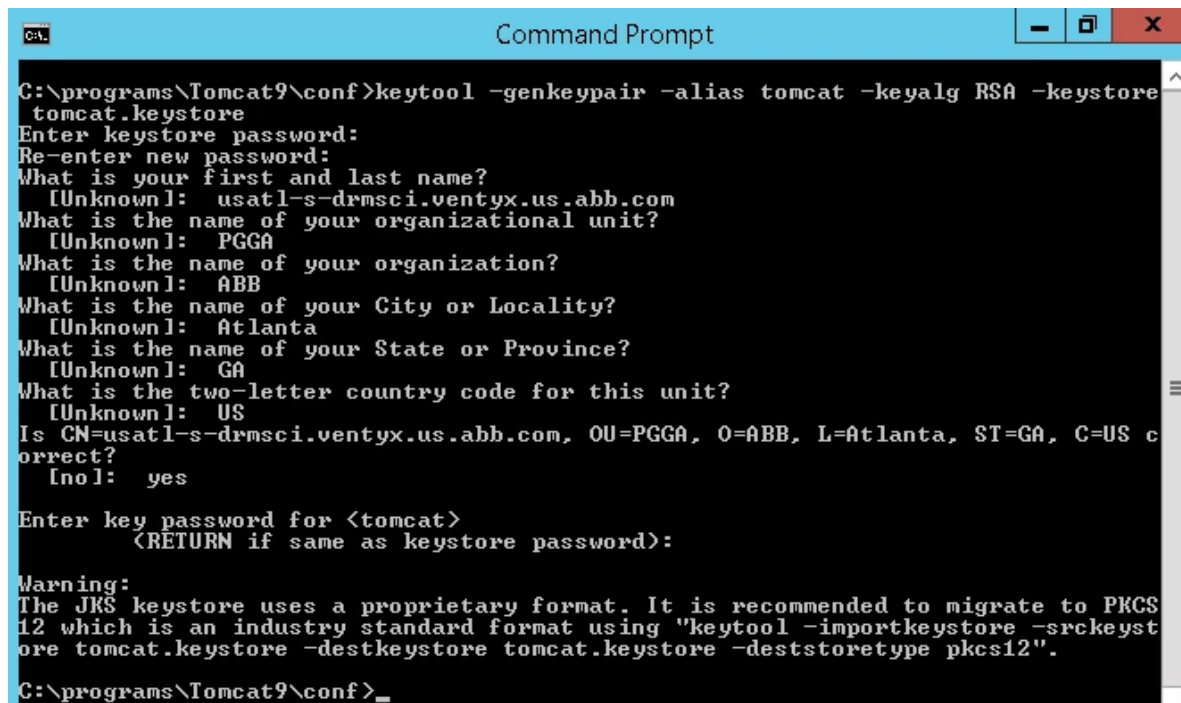
Note: You will not have this shortcut in your Start menu unless you checked the "Create shortcuts for all users" option during step 5b.

13. Right-click on **Tomcat Monitor > Configure...** and select the **Java** tab. Replace the existing **Java Options** with the contents from the clipboard.
Or navigate to Start menu > Programs > Apache Tomcat 9.0 ... > Configure Tomcat and perform the step above
14. If the Tomcat server sits inside a proxy server, add the following lines to the Java Options section:
`-Dhttp.proxyHost=<proxy host name or address>`
`-Dhttp.proxyPort=<proxy port>`
15. Set Initial memory pool and Maximum memory pool to 256MB and 512MB respectively. Click **OK**.
16. Right-click on the **Tomcat Monitor** again and start service.
17. Verify that Tomcat is running by navigating to <http://<hostname>>.

Configure Tomcat for SSL

1. This step generates a keystore with a self-signed certificate for testing purposes only.

See <http://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html> for configuring a production system with a certificate signed by a valid certificate authority. (This step assumes %JAVA_HOME%\bin is on the PATH environment variable.) Open a command prompt and cd to the Tomcat conf directory. Type the command `keytool -genkeypair -alias tomcat -keyalg RSA -keystore tomcat.keystore` and press **Enter**. Respond to the prompts; when prompted for first and last name, enter the computer host name. Using the host name for first and last name will prevent a warning when accessing the site.



```

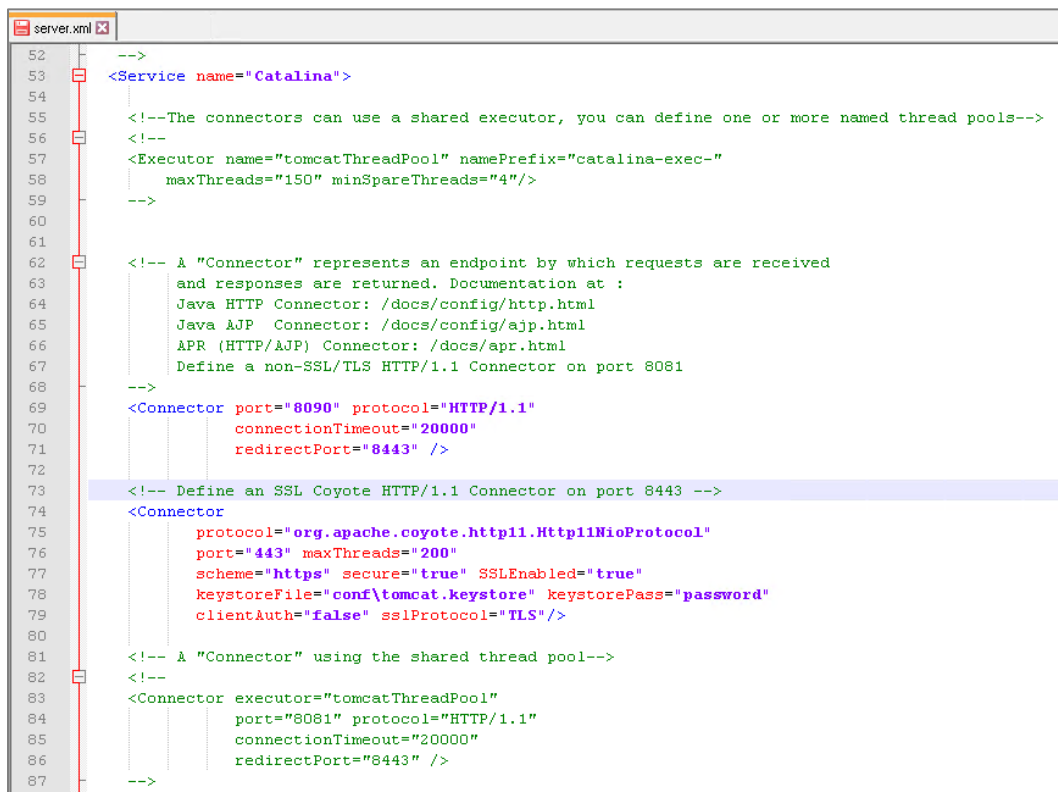
C:\>
C:\programs\Tomcat9\conf>keytool -genkeypair -alias tomcat -keyalg RSA -keystore
tomcat.keystore
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  usatl-s-drmsci.ventyx.us.abb.com
What is the name of your organizational unit?
  [Unknown]:  PGGA
What is the name of your organization?
  [Unknown]:  ABB
What is the name of your City or Locality?
  [Unknown]:  Atlanta
What is the name of your State or Province?
  [Unknown]:  GA
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=usatl-s-drmsci.ventyx.us.abb.com, OU=PGGA, O=ABB, L=Atlanta, ST=GA, C=US c
orrect?
  [No]:  yes
Enter key password for <tomcat>
  <RETURN if same as keystore password>:

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS
12 which is an industry standard format using "keytool -importkeystore -srckeyst
ore tomcat.keystore -destkeystore tomcat.keystore -deststoretype pkcs12".

C:\programs\Tomcat9\conf>_
  
```

2. Edit the Tomcat conf/server.xml file. Find the <Connector> entry for the standard HTTP port. Add the following <Connector> entry to define the SSL Connector. Be sure to provide the same keystorePass used above. Save the file. See <http://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html> for additional configuration options.

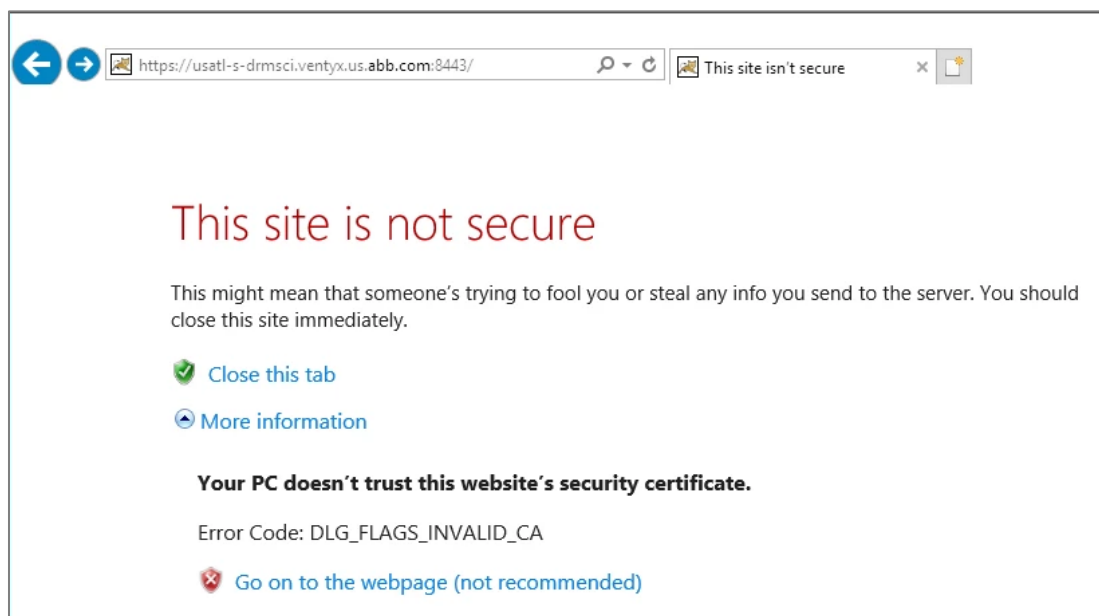
```
<Connector
protocol="org.apache.coyote.http11.Http11NioProtocol"
port="443" maxThreads="200"
scheme="https" secure="true" SSLEnabled="true"
keystoreFile="conf\tomcat.keystore" keystorePass="password"
clientAuth="false" sslProtocol="TLS"/>
```



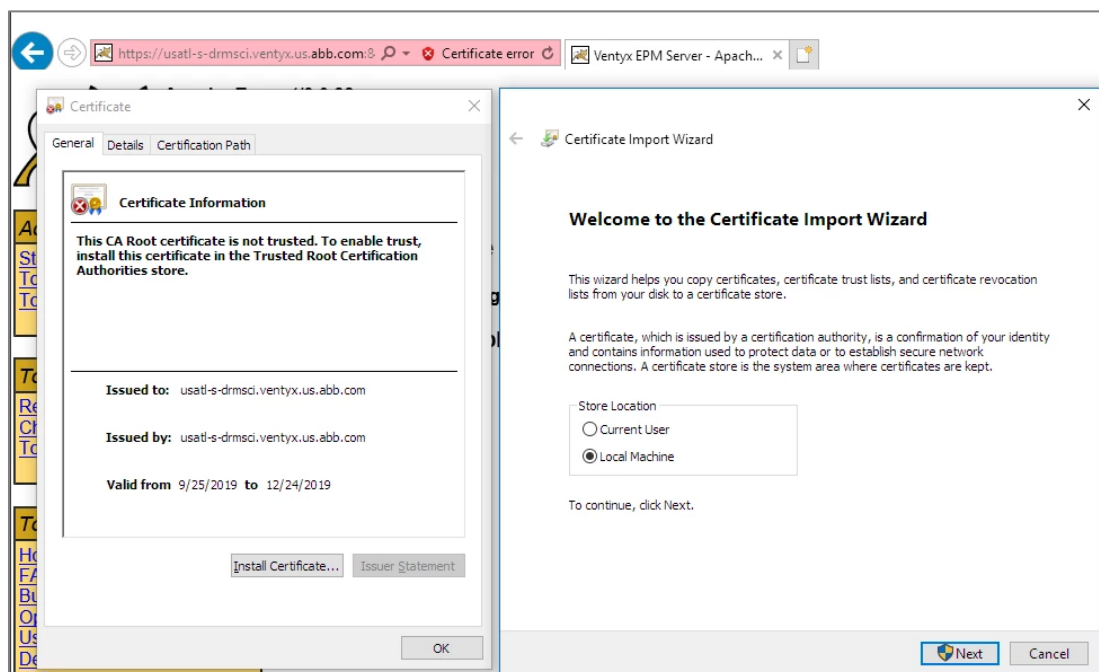
```
server.xml
52  <!--
53  <Service name="Catalina">
54
55  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
56  <!--
57  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
58  maxThreads="150" minSpareThreads="4"/>
59  -->
60
61
62  <!-- A "Connector" represents an endpoint by which requests are received
63  and responses are returned. Documentation at :
64  Java HTTP Connector: /docs/config/http.html
65  Java AJP Connector: /docs/config/ajp.html
66  APR (HTTP/AJP) Connector: /docs/apr.html
67  Define a non-SSL/TLS HTTP/1.1 Connector on port 8081
68  -->
69  <Connector port="8090" protocol="HTTP/1.1"
70  connectionTimeout="20000"
71  redirectPort="8443" />
72
73  <!-- Define an SSL Coyote HTTP/1.1 Connector on port 8443 -->
74  <Connector
75  protocol="org.apache.coyote.http11.Http11NioProtocol"
76  port="443" maxThreads="200"
77  scheme="https" secure="true" SSLEnabled="true"
78  keystoreFile="conf\tomcat.keystore" keystorePass="password"
79  clientAuth="false" sslProtocol="TLS"/>
80
81  <!-- A "Connector" using the shared thread pool-->
82  <!--
83  <Connector executor="tomcatThreadPool"
84  port="8081" protocol="HTTP/1.1"
85  connectionTimeout="20000"
86  redirectPort="8443" />
87  -->
```

3. Remove the HTTP/1.1 protocol Connector to prevent insecure connections. Note that leaving this connector will allow login communication over an insecure connection which will expose user names and passwords in clear text.
4. Right-click on the **Tomcat Monitor** again and start service.

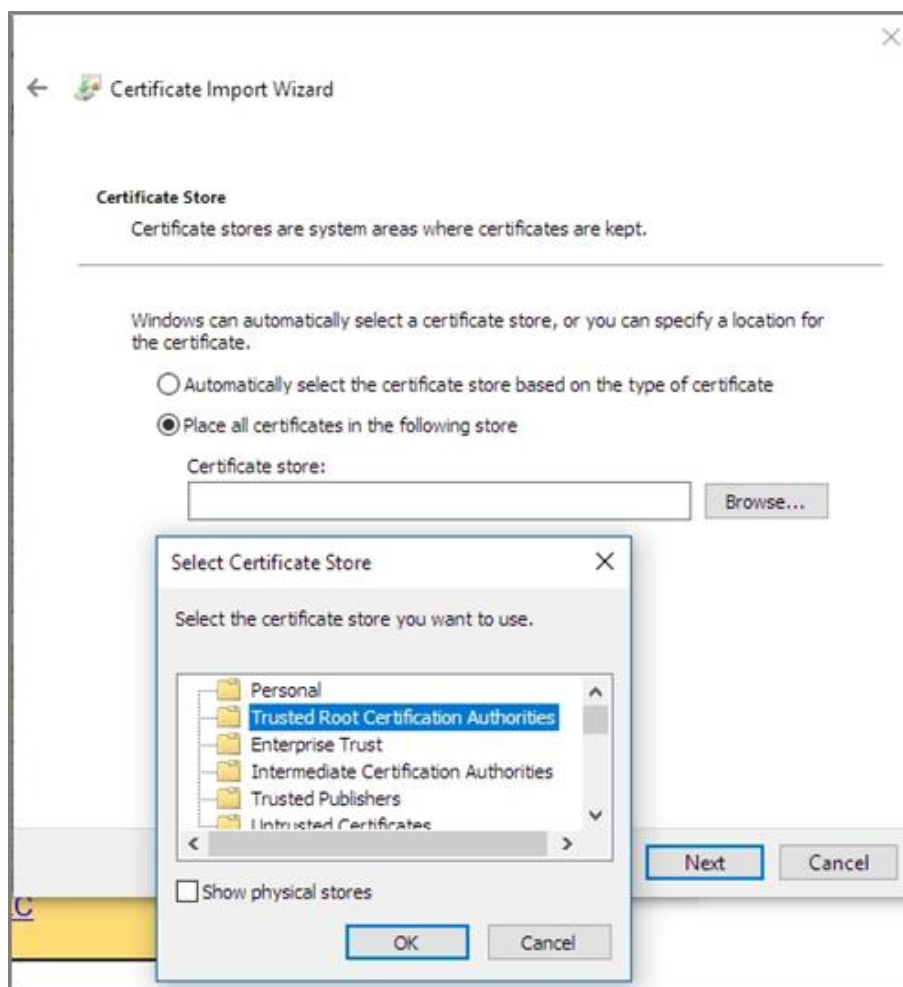
5. Verify that Tomcat is running by navigating to <https://<hostname>>.
 - a. If you are using a self-signed certificate, you will get a warning that the site is not secure because of that. You can expand the More information link and Go on to the webpage.



- b. (optional) To prevent seeing this warning in the future, click the **Certificate error** in the URL bar. Click the **View Certificates** link. Click the **Install Certificate...** button. Choose the **Local Machine** radio button and click **Next**.



- c. Place the certificate in the **Trusted Root Certification Authorities** store:



Deploy and Configure Applications

1. Stop the Tomcat service.
2. Copy the war files you want to deploy from the Web folder of the Server Installation directory to the webapps folder of the Tomcat installation directory.
 - a. The main application is named APP#.#.#.#.war where APP is equal to ro, mm, csb, or drms, depending on whether the file is Retail Operations, Market Operations, CSB, or DRMS respectively, and #.#.#.# is the version number. An example of a Retail Operations WAR file for version 5.7.0.0 is ro5.7.0.0.war.
 - b. The mex war file should be copied if you are using or plan to use the market exchanges.
 - c. The help war file should be copied for access to the application help.
 - d. The web services (WS) war file should be copied to use delivered or custom web services.
3. Start the Tomcat service. Tomcat will deploy the war files and create directories in the webapps folder for each of the war files copied.
4. Open a web browser and navigate to the Tomcat Manager page at <https://localhost/manager/html>. Login using ventyxadmin/Admin13579.
5. Verify the war files copied are listed under Applications and the Running column contains true for each of them. If one is not listed or is not running, check the Tomcat log files in the logs directory for errors.
6. Configure authentication for the main application.
 - a. Stop the Tomcat server.
 - b. Open the file **Tomcat Installation Directory\webapps\APP#.#.#.#\WEB-INF\web.xml**.
 - c. Scroll down to the "<!-- Environment Entries -->" section and edit the two circled fields in the image to match the desired authentication type. You may choose from the following values for this. If you choose other than the TEST_PAM, please be sure to see [Chapter 2](#) to make sure the other PAMs are configured correctly.
 - i. TEST_PAM
 - ii. ACT_DIR_PAM
 - iii. LDAP_PAM

```

81 - <env-entry>
82   <description>Whether or not the application is in 'Direct-Oracle-Login' mode.</description>
83   <env-entry-name>directOracleLogin</env-entry-name>
84   <env-entry-type>java.lang.Boolean</env-entry-type>
85   <env-entry-value>>false</env-entry-value>
86 </env-entry>
87 - <env-entry>
88   <description>The JAAS Authentication type for the Main application Login. Values include: TE
89   <env-entry-name>authType</env-entry-name>
90   <env-entry-type>java.lang.String</env-entry-type>
91   <env-entry-value>TEST_PAM</env-entry-value>
92 </env-entry>
93 - <env-entry>
94   <description>The JAAS Authentication type for the Datasource Management Login. Values includ
95   <env-entry-name>datasourceAuthType</env-entry-name>
96   <env-entry-type>java.lang.String</env-entry-type>
97   <env-entry-value>TEST_PAM</env-entry-value>
98 </env-entry>
99 - <env-entry>
100  <description>The role used to secure access to the Datsource Manager tool.</description>
101  <env-entry-name>adminRole</env-entry-name>
102  <env-entry-type>java.lang.String</env-entry-type>
103  <env-entry-value>VENTYX_ADMIN</env-entry-value>
104 </env-entry>

```

- d. **Save** the file.

7. If the web services war file was deployed, configure it as follows.
 - a. Stop the Tomcat service.
 - b. Go to Tomcat Installation Directory\webapps\roWS5.7.0.0\META-INF, and edit the context.xml file with a plain-text editor like Notepad.
 - i. Change user to schema name
 - ii. Change password to schema password
 - iii. Change url to appropriate Datasource information
8. Start the Tomcat service.
9. Configure data sources for the main application
 - a. Open the application Login page in a web browser. For example, <https://localhost/ro5.7.0.0/>
 - b. Click the **Manage Datasources** link.
 - c. Login to the **Manage Datasources** tool with a valid Admin user account. If you configured this website to use Active Directory or LDAP, then the Admin user account should be the **ventyxadmin** user that you created during the First Steps. If you configured this website to use the Test PAM, then use the **ventyxadmin** user account with the password of **password**.
 - d. Click the **Insert** button.
 - e. Enter the appropriate Datasource information.
 - f. Click the **Save** button.
 - g. Click the **Logout** button.
 - h. Verify your new Datasource by returning to the application Login page, verify the new datasource appears in the dropdown list, and verify you can log in with a valid user and password. If there are any errors, detailed information can be found in the Tomcat stdout log under Tomcat Installation Directory\logs.
10. If the web services war file was deployed, test connectivity to the web services.
 - a. Navigate to <https://localhost/roWS5.7.0.0/test>
 - b. Login with a valid user and password.
 - c. You should see a list of the web services.

Chapter 2: Tomcat Login Configuration

The EPM Ops Server authenticates users using the Java Authorization and Authentication Service (JAAS). JAAS is a set of APIs that enable services to authenticate and enforce access controls upon users. It implements a Java technology version of the standard Pluggable Authentication Module (PAM) framework and supports user-based authorization.

More information about JAAS can be found at: <http://www.oracle.com/technetwork/java/index.html>

The EPM Ops Server currently supports the following JAAS PAM configurations:

- Active Directory
- LDAP
- Default Ventyx Authentication Provider (when no client provider is present)
- Test PAM (when no authentication is required)

By default, all EPM Ops Web Applications default to the test PAM. This module successfully authenticates any set of credentials provided that the password is **password**. With this provider, users belong to no security groups *except* the **ventyxadmin** user, which belongs to a security group named **VENTYX_ADMIN**. With this configuration, the **ventyxadmin** user can be used to access the Datasource Setup screens of an EPM Ops Web Application that is configured with the default authentication settings.

To setup JAAS Authentication for LDAP or Active Directory:

1. Stop the Tomcat service if it is running.
2. Using any text editor, edit the network settings in the **Tomcat Installation Folder\login.conf** file. This file is where all JAAS modules are defined.

Standard Active Directory Configuration

For Active Directory Authentication edit the **ACTIVE_DIR_PAM** entry. This entry contains fifteen (15) fields that can be modified in order to establish a connection with Active Directory. However, most of the fields are populated with reasonable defaults and typically, only a few fields need to be edited.

Edit the following four (4) fields:

- **connectionURL** – This is the full URL to the Active Directory. You will need the full Protocol, Host Name, and Port. The port is typically 389. However, this depends entirely on which port is being used by the client's Active Directory.

Example:

```
connectionURL="ldap://123.45.67.89:389"
```

- **connectionDN** – This is the template for the DN used by the application to connect to the LDAP. Typically this value should be **{0}@domain**. For example, if a network user logs using the name **pmanning@ventyx.com** then the connectionDN property would be **{0}@ventyx.com**.

Example:

```
connectionDn="{0}@abc.com"
```

- **userBase** – This is the base set of LDAP attributes required to lookup a user in the Active Directory. Once the PAM module **binds** to the Active Directory using the **connectionDN**, the **userBase** property is used to search the Active Directory to retrieve the users full DN. This property should be the fully qualified path to the Organization Unit (OU) that contains the network users.

Example:

userBase="OU=Accounting Users,OU=ABC Division,DC=abc,DC=com"

Note: If the network users belong to the **Users** container within Active Directory, not in a specific Organizational Unit (OU), then the userBase will use the **CN** attribute (ie. "CN=Users,DC=abc,DC=com").

- **roleBase** – This is the base set of LDAP attributes required to lookup a role in the Active Directory. This property should be the fully qualified path to the Organization Unit (OU) that contains the security Groups. Once the PAM module finds the user in the Active Directory, the "roleBase" property is used to search for all of the user's roles.

Example:

roleBase="OU=Security Groups, OU=ABC Division,DC=abc,DC=com"

Note: The application will specifically look for the **VENTYX_ADMIN** role that was previously set up in the LDAP (see [First Steps](#)).

An example of complete PAM entry:

```
ACT_DIR_PAM{
  com.newenergyassoc.auth.module.NELdapLoginModule required
  debugMode=true
  description="Active Directory Login Module"
  connectionURL="ldap://123.45.67.89:389"
  initialContextFactory=com.sun.jndi.ldap.LdapCtxFactory
  connectionProtocol=""
  authentication=simple
  userBase="OU=Accounting Users,OU=ABC Division,DC=abc,DC=com"
  userSearchMatching="userPrincipalName={0}"
  userSearchSubtree=false
  roleBase="OU=Security Groups, OU=ABC Division,DC=abc,DC=com"
  roleName=cn
  roleSearchMatching="(member={0})"
  roleSearchSubtree=false
  userRoleName=""
  connectionDn="{0}@abc.com"
  ;
};
```


Alternate Active Directory Configuration

Note: This is not the preferred way of configuring Active Directory Authentication (see [Standard Active Directory Configuration](#)). Only use this method when the Active Directory will not allow application users to bind() directly using the **connectionDN** property. Instead of using the **connectionDN** property, the alternate method allows the administrator to provide the credentials of a user that can bind() to the Active Directory.

Edit the following fields:

- **connectionUsername** – The Distinguished Name of a known user in the Active Directory that has **bind()** privilege to the Active Directory. This user must have privileges to **bind()** and search the Active Directory. The Distinguished Name is made up of several parts:

cn – common name
ou – organizational unit(s)
dc – domain content

Note: For the EPM Ops Server application, the **connectionUsername** should be the Distinguished Name of the **ventyxadmin** user account that was previously setup (see [First Steps](#)).

- **connectionPassword** – The password of the user that has **bind()** privilege to the Active Directory.

Example:

```
ACT_DIR_PAM{
  com.newenergyassoc.auth.module.NELdapLoginModule required
  debugMode=true
  description="Active Directory Login Module"
  connectionURL="ldap://123.45.67.89:389"
  initialContextFactory=com.sun.jndi.ldap.LdapCtxFactory
  connectionProtocol=""
  authentication=simple
  userBase="OU=Accounting Users,OU=ABC Division,DC=abc,DC=com"
  userSearchMatching="userPrincipalName={0}"
  userSearchSubtree=false
  roleBase="OU=Security Groups, OU=ABC Division,DC=abc,DC=com"
  roleName=cn
  roleSearchMatching="(member={0})"
  roleSearchSubtree=false
  userRoleName=""
  connectionUsername="ventyxadmin"
  connectionPassword="password123"
  ;
};
```

LDAP Configuration

For LDAP Authentication edit the LDAP_PAM entry.

Edit the following fields:

- **connectionURL** – This is the full URL to the LDAP. You will need the full Protocol, Host Name, and Port. The port is typically 389. However, this depends entirely on which port is being used by the client's LDAP.

Example:

```
connectionURL="ldap://123.45.67.89:389"
```

- **connectionDN** – This is the template for the DN used by the application to connect to the LDAP.

Example:

```
connectionDn="uid={0},OU=Accounting Users,OU=ABC Division,DC=abc,DC=com."
```

- **userBase** – This is the base set of LDAP attributes required to lookup a user. Once the PAM module binds to the LDAP using the connectionDN, the userBase property is used to search the LDAP to retrieve the users full DN. This property should be the fully qualified path to the Organization Unit (OU) that contains the network users.

Example:

```
userBase="OU=Accounting Users,OU=ABC Division,DC=abc,DC=com"
```

- **roleBase** – This is the base set of LDAP attributes required to lookup a role. This property should be the fully qualified path to the Organization Unit (OU) that contains the roles. Once the PAM module finds the user in the LDAP, the **roleBase** property is used to search for all of the user's roles.

Example:

```
roleBase="OU=Security Groups, OU=ABC Division,DC=abc,DC=com"
```

Note: The application will specifically look for the **VENTYX_ADMIN** role that was previously set up in the LDAP (see [First Steps](#)).

Example:

```
LDAP_PAM{
  com.newenergyassoc.auth.module.NELdapLoginModule required
  debugMode=true
  description="LDAP Login Module"
  connectionURL="ldap://123.45.67.89:389"
  initialContextFactory=com.sun.jndi.ldap.LdapCtxFactory
  connectionProtocol=""
  authentication=simple
  userBase="OU=Accounting Users,OU=ABC Division,DC=abc,DC=com"
  userSearchMatching="uid={0}"
  userSearchSubtree=false
  roleBase="OU=Security Groups, OU=ABC Division,DC=abc,DC=com"
  roleName=cn
  roleSearchMatching="(memberOf={0})"
  roleSearchSubtree=false
  userRoleName=""
  connectionDN="uid={0},OU=Accounting Users,OU=ABC
Division,DC=abc,DC=com"
  ;
};
```

Chapter 3: Client Installation

Each client machine which will access the application from the server will need some software installed.

1. JDK version 11 must be installed. See the [Java Development Kit](#) section in Chapter 1 for options. The same option does not need to be chosen for both the server and client.
2. Set the JAVA_HOME system environment variable to the path of the JDK installed in step 1.
 - a. Click Start > Settings.
 - b. In the Find a setting textbox, type **env** and choose the item **Edit the system environment variables**.
 - c. Click the **Environment Variables...** button on the Advanced tab of the System Properties screen.
 - d. In the bottom **System variables** grid, click **New...**
 - e. Enter **JAVA_HOME** for the Variable name.
 - f. Click **Browse Directory...** and choose the directory where the JDK version 11 was installed in step 1.
 - g. Click **OK**.
 - h. Click **OK**.
3. Run the EPM Ops Client installer.