

Using keytool to generate an Identity Keystore

What is a Keystore?

A keystore is a repository of security certificates, either authorization certificates or public key certificates, which are used mainly in SSL encryption.

The Java Development Kit (JDK) maintains a default CA keystore stored in

`<JAVA_HOME>/jre/lib/security/cacerts.`

- The well-known password is “changeit”.
- Best practice:
 - Copy the default keystore to a new location.
 - Reset the password.
 - Configure WebLogic to use the new location.

keytool Utility

`keytool` is a standard Java SE SDK utility for managing:

- The generation of private keys and the corresponding digital certificates
- Keystores (databases) of private keys and the associated certificates

The `keytool` utility can display certificate and keystore contents.

You can specify an algorithm that is different from Digital Signature Algorithm (DSA) when generating digital keys by using `keytool`.

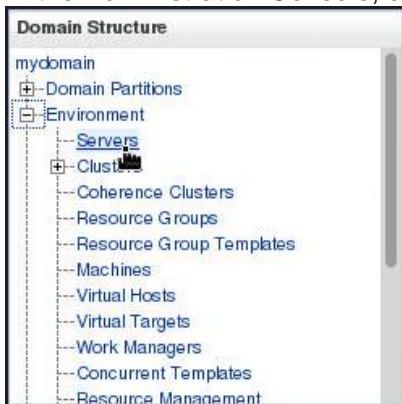
Perform the following steps to create a new key pair using the Java `keytool` utility and configure `server1` to use your custom keystore:

1. Execute the following command to run `keytool` to create a keystore and a key pair within the keystore (all in one line). You can use the [genkey.sh](#) script for convenience.

Note: You can execute this command from any location/directory. The certificate will get saved in the directory/location where you executed the command. In this tutorial, the command is executed under `/scratch/scripts` directory. Hence, the file `wls_identity.jks` got generated and saved in the same directory.

```
$ keytool -genkey -v -alias wlskey -keyalg RSA -keysize 2048 -sigalg MD5withRSA -  
dname "CN=wls-sysadm" -keypass wlskeypass -validity 365 -keystore  
wls_identity.jks -storepass wlsstorepass
```

2. In the Administration Console, click **Environment > Servers** under **Domain Structure**.



- Click **server1** in the **Servers** table on the **Summary of Servers** page.

Summary of Servers

Configuration Control

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

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

[Customize this table](#)

Servers (Filtered - More Columns Exist)

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

New Clone Delete Showing 1 to 3 of 3 Previous | Next

	Name	Type	Cluster	Machine	State	Health	Listen Port
<input type="checkbox"/>	AdminServer(admin)	Configured			RUNNING	✓ OK	7001
<input type="checkbox"/>	server1	Configured	cluster1	machine1	RUNNING	✓ OK	7004

- On the **Settings for server1** page, select the **Keystores** tab.

Settings for server1

Configuration Protocols Logging Debug Monitoring Control Deployments Services Security

Notes

General Cluster Services **Keystores** SSL Federation Services Deployment Migration Tuning

Overload Concurrency Health Monitoring Server Start Web Services Coherence

- In the Change Center, click **Lock & Edit**.

Change Center

View changes and restarts

Click the **Lock & Edit** button to modify, add or delete items in this domain.

Lock & Edit

Release Configuration

- On the **Keystores** page, specify the following properties and click **Save**.

Description	Choices or Values
Keystores	Custom Identity and Java Standard Trust
Custom Identity Keystore	/scratch/scripts/wls_identity.jks
Custom Identity Keystore Type	JKS
Custom Identity Keystore PassPhrase	wlsstorepass
Java Standard Trust Keystore PassPhrase	changeit

Settings for server1

Configuration

Protocols

Logging

Debug

Monitoring

Control

Deployments

Services

Security

Notes

General

Cluster

Services

Keystores

SSL

Federation Services

Deployment

Migration

Tuning

Overload

Concurrency

Health Monitoring

Server Start

Web Services

Coherence

Save

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This page lets you view and define various keystore configurations. These settings help you to manage the security of message transmissions.

Keystores:

Custom Identity and Java Standard Trust

Which configuration rules should be used for finding the server's identity and trust keystores? [More Info...](#)

Change

Identity

Custom Identity Keystore:

:/ratch/scripts/wls_identity.jks

The source of the identity keystore. For a JKS keystore, the source is the path and file name. For an Oracle Key Store Service (KSS) keystore, the source is the KSS URI. [More Info...](#)

Custom Identity Keystore Type:

JKS

The type of the keystore. Generally, this is JKS. If using the Oracle Key Store Service, this would be KSS. [More Info...](#)

Custom Identity Keystore Passphrase:

••••••••••

The encrypted custom identity keystore's passphrase. If empty or null, then the keystore will be opened without a passphrase. [More Info...](#)

Confirm Custom Identity Keystore Passphrase:

••••••••••

Trust

Java Standard Trust Keystore:

/u01/app/jdk/jre/lib/security/cacerts

The location of the java standard trust keystore. [More Info...](#)

Java Standard Trust Keystore Type:

jks

The type of the java standard trust keystore. Generally, this is JKS. [More Info...](#)

Java Standard Trust Keystore Passphrase:

••••••••

The password for the Java Standard Trust keystore. This password is defined when the keystore is created. [More Info...](#)

Confirm Java Standard Trust Keystore Passphrase:

••••••••

Save

7.

Configuring SSL for a managed server

Perform the following steps to configure `server1` to enable and support SSL using your custom identity keystore:

1. On the **Settings for server1** page, select the **SSL** tab.



2. On the **SSL** page, specify the following properties and click **Save**.

Description	Choices or Values
Identity and Trust Locations	Keystores
Private Key Alias	wlskey
Private Key Passphrase	wlskeypass

- 3.

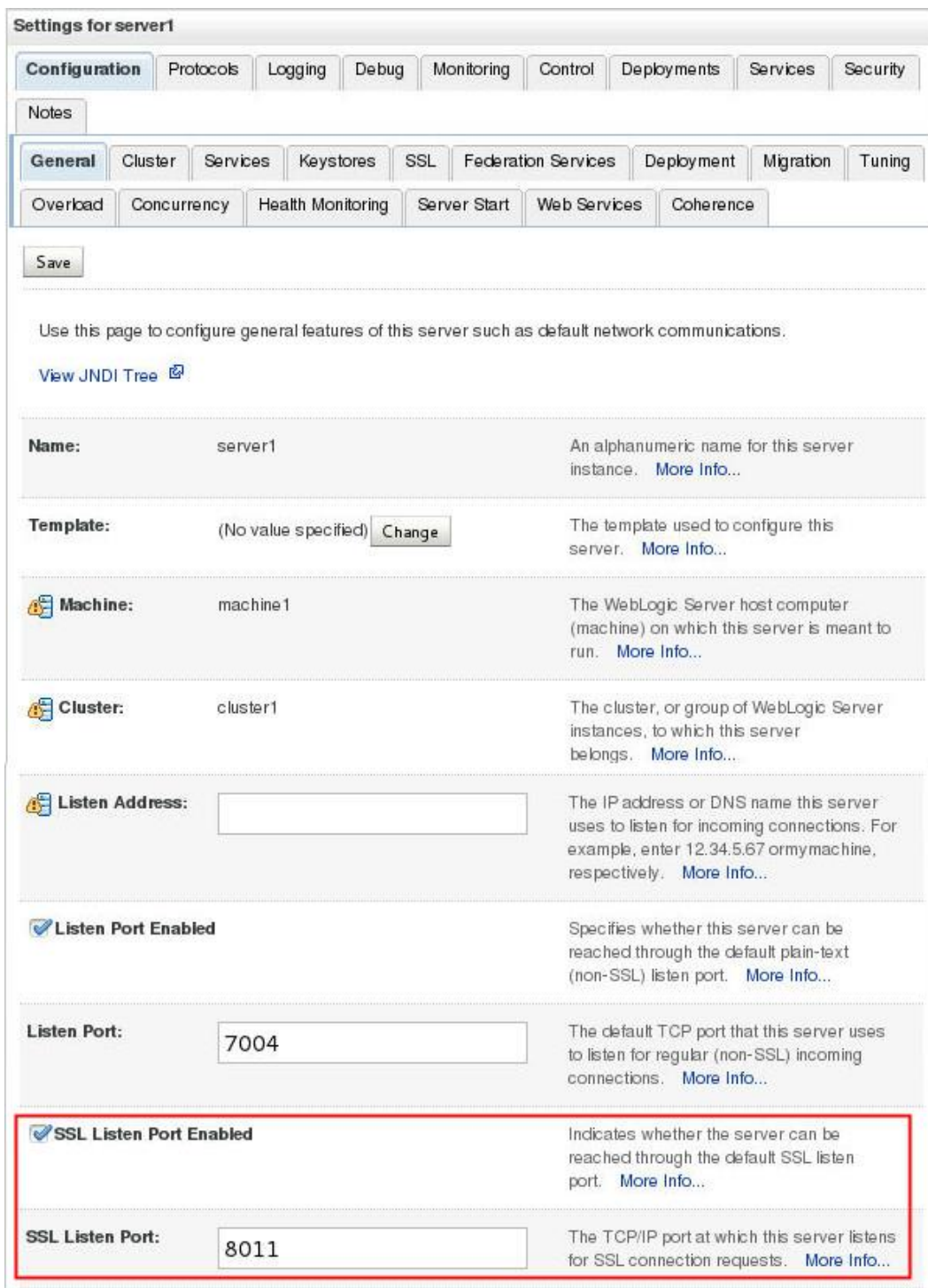
- 4.

5. On the **Settings for server1** page, select the **General** tab.



The screenshot shows the 'Settings for server1' page. At the top, there is a row of tabs: Configuration, Protocols, Logging, Debug, Monitoring, Control, Deployments, Services, and Security. Below this is a 'Notes' section. Underneath, there is another row of tabs: General, Cluster, Services, Keystores, SSL, Federation Services, Deployment, Migration, and Tuning. The 'General' tab is currently selected and highlighted. Below this second row, there is a third row of tabs: Overload, Concurrency, Health Monitoring, Server Start, Web Services, and Coherence.

6. Select the check box next to **SSL Listen Port Enabled** and set the SSL Listen Port as **8011**. Then click **Save**.



The screenshot shows the 'Settings for server1' page with the 'General' tab selected. Below the tabs, there is a 'Save' button. A message states: 'Use this page to configure general features of this server such as default network communications.' Below this is a link 'View JNDI Tree' with an external link icon. The configuration details are as follows:

Name:	server1	An alphanumeric name for this server instance. More Info...
Template:	(No value specified) Change	The template used to configure this server. More Info...
Machine:	machine1	The WebLogic Server host computer (machine) on which this server is meant to run. More Info...
Cluster:	cluster1	The cluster, or group of WebLogic Server instances, to which this server belongs. More Info...
Listen Address:	<input type="text"/>	The IP address or DNS name this server uses to listen for incoming connections. For example, enter 12.34.5.67 or mymachine, respectively. More Info...
<input checked="" type="checkbox"/> Listen Port Enabled		Specifies whether this server can be reached through the default plain-text (non-SSL) listen port. More Info...
Listen Port:	<input type="text" value="7004"/>	The default TCP port that this server uses to listen for regular (non-SSL) incoming connections. More Info...
<input checked="" type="checkbox"/> SSL Listen Port Enabled		Indicates whether the server can be reached through the default SSL listen port. More Info...
SSL Listen Port:	<input type="text" value="8011"/>	The TCP/IP port at which this server listens for SSL connection requests. More Info...