**To Use keytool to Create a Server Certificate**

Run keytool to generate a new key pair in the default development keystore file, keystore.jks. This example uses the alias server-alias to generate a new public/private key pair and wrap the public key into a self-signed certificate inside keystore.jks. The key pair is generated by using an algorithm of type RSA, with a default password of changeit. For more information and other examples of creating and managing keystore files, read the keytool online help at <http://download.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html>.

**Note –**

RSA is public-key encryption technology developed by RSA Data Security, Inc.

From the directory in which you want to create the key pair, run keytool as shown in the following steps.

1. Generate the server certificate.

Type the keytool command all on one line:

|  |
| --- |
| ***java-home*/bin/keytool -genkey -alias server-alias -keyalg RSA -keypass changeit**  **-storepass changeit -keystore keystore.jks** |

When you press Enter, keytool prompts you to enter the server name, organizational unit, organization, locality, state, and country code.

You must type the server name in response to keytool’s first prompt, in which it asks for first and last names. For testing purposes, this can be localhost.

When you run the example applications, the host (server name) specified in the keystore must match the host identified in the javaee.server.name property specified in the file *tut-install*/examples/bp-project/build.properties.

1. Export the generated server certificate in keystore.jks into the file server.cer.

Type the keytool command all on one line:

|  |
| --- |
| ***java-home*/bin/keytool -export -alias server-alias -storepass changeit**  **-file server.cer -keystore keystore.jks** |

1. If you want to have the certificate signed by a CA, read the example at <http://download.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html>.
2. To add the server certificate to the truststore file, cacerts.jks, run keytool from the directory where you created the keystore and server certificate.

Use the following parameters:

|  |
| --- |
| ***java-home*/bin/keytool -import -v -trustcacerts -alias server-alias**  **-file server.cer -keystore cacerts.jks -keypass changeit -storepass changeit** |

Information on the certificate, such as that shown next, will appear:

|  |
| --- |
| Owner: CN=localhost, OU=Sun Micro, O=Docs, L=Santa Clara, ST=CA,  C=USIssuer: CN=localhost, OU=Sun Micro, O=Docs, L=Santa Clara, ST=CA,  C=USSerial number: 3e932169Valid from: Tue Apr 08Certificate  fingerprints:MD5: 52:9F:49:68:ED:78:6F:39:87:F3:98:B3:6A:6B:0F:90 SHA1:  EE:2E:2A:A6:9E:03:9A:3A:1C:17:4A:28:5E:97:20:78:3F:  Trust this certificate? [no]: |

1. Type **yes**, then press the **Enter** or **Return** key.

The following information appears:

|  |
| --- |
| Certificate was added to keystore[Saving cacerts.jks] |