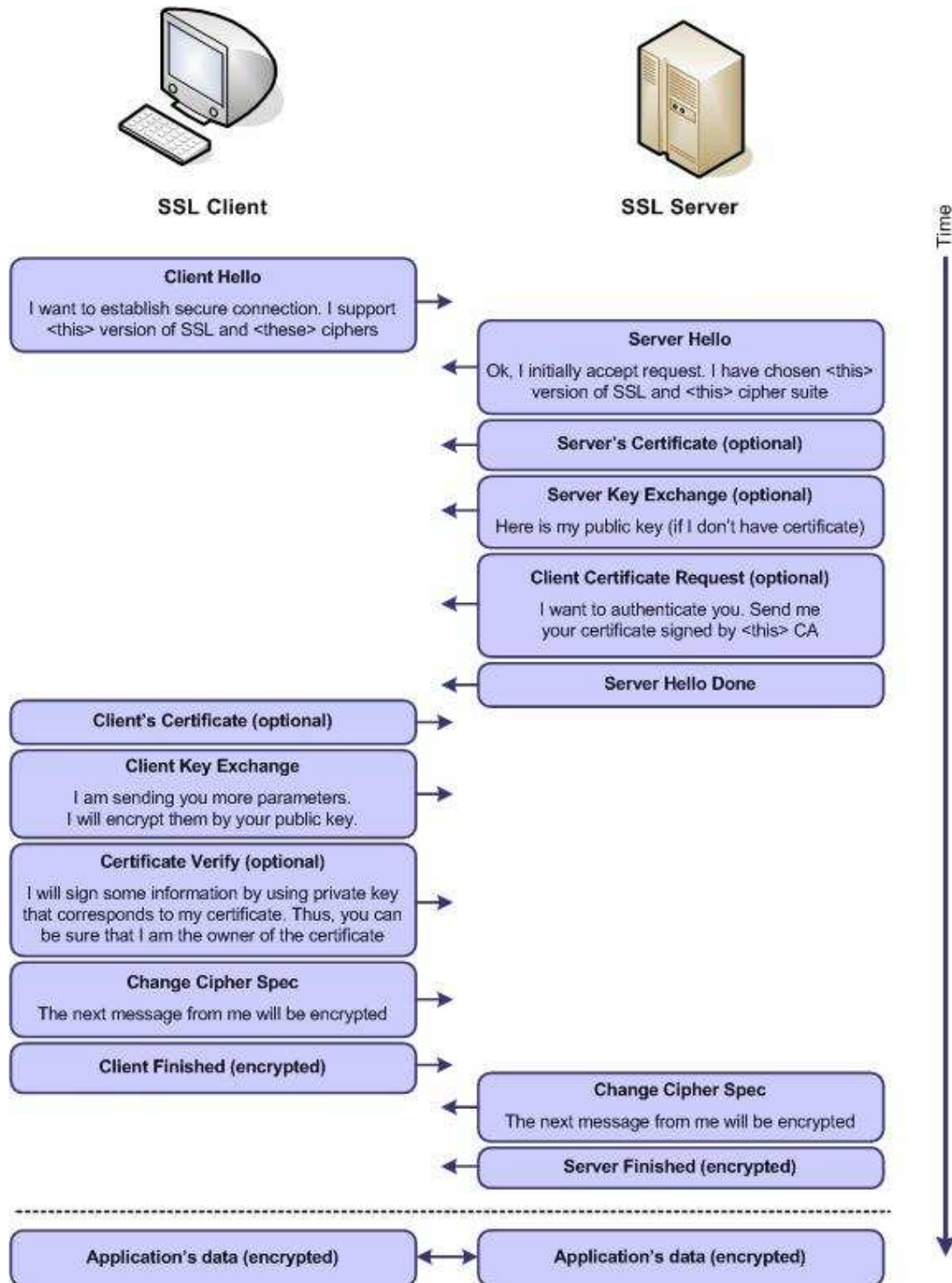


Two Way SSL

In Two Way SSL (mutual authentication) the client verifies the identity of the server, and then the server verifies the credentials of the client. The figure below gives an overview of the Two Way SSL process.



Implementation

Example below shows how to configure Two Way SSL for client connecting to Weblogic/Glassfish Server. Both servers provide default keystore (database of private keys and certificate) which are complete in themselves for SSL implementation in testing environment. In production environment you should implement your own certificate signed by your own CA.

More information on configuring SSL on Weblogic at:

http://download-llnw.oracle.com/docs/cd/E11035_01/wls100/secmanage/ssl.html

Java provides **keytool**, a key and certificate management utility. It enables users to administer their own public/private key pairs and associated certificates for use in self-authentication.

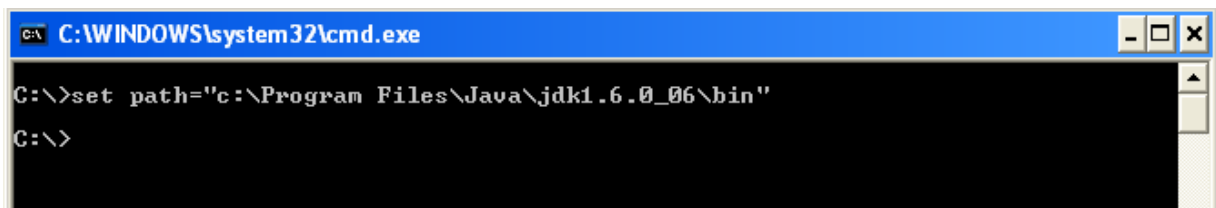
keytool stores the keys and certificates in a so-called *keystore*.

More information on keytool visit:

<http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html>

Following are the steps to implement Two Way SSL:

1. Set the path to use keytool: set the path to your jdk



```
C:\WINDOWS\system32\cmd.exe
C:\>set path="c:\Program Files\Java\jdk1.6.0_06\bin"
C:\>
```

2. Configure the weblogic to implement Two Way SSL

Start Weblogic -> Login to console -> Click on Environment -> Servers -> SSL ->Advanced

Make sure in Two Way Client Cert behavior option **Client Certs Requested and Enforced** is selected

Health of Running Servers

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

as a client to another application server). [More Info...](#)

Custom Hostname Verifier: The name of the class that implements the `weblogic.security.SSL.HostnameVerifier` interface. [More Info...](#)

Export Key Lifespan: Indicates the number of times WebLogic Server can use an exportable between a domestic server and an exportable client before generating key. The more secure you want WebLogic Server to be, the fewer time key should be used before generating a new key. [More Info...](#)

Use Server Certs Sets whether the client should use the server certificates/key as the identity when initiating a connection over https. [More Info...](#)

Two Way Client Cert Behavior: **Client Certs Requested and Enforced** The form of SSL that should be used. [More Info...](#)

Cert Authenticator: The name of the Java class that implements the `weblogic.security.ad.CertAuthenticator` class, which is deprecated in the release of WebLogic Server. This field is for Compatibility security only, only used when the Realm Adapter Authentication provider is configured. [More Info...](#)

SSLRejection Logging Enabled Indicates whether warning messages are logged in the server log when connections are rejected. [More Info...](#)

Allow Unencrypted Null Cipher Test if the `AllowUnEncryptedNullCipher` is enabled. [More Info...](#)

Inbound Certificate Validation: Indicates the client certificate validation rules for inbound SSL. [More Info...](#)

Outbound Certificate Validation: Indicates the server certificate validation rules for outbound SSL. [More Info...](#)

For Glassfish

Make sure that client authentication is enabled

The screenshot displays the Glassfish Administration Console interface. On the left is a navigation tree with the following structure:

- Common Tasks
- Registration
- Application Server
- Applications
 - Web Applications
- Resources
 - JDBC
- Configuration
 - Web Container
 - HTTP Service
 - HTTP Listeners
 - admin-listener
 - http-listener-2**
 - http-listener-1
 - Virtual Servers
 - Monitoring
 - Security
 - Update Tool

The main content area shows the configuration for 'http-listener-2' under 'Configuration > HTTP Service > HTTP Listeners > http-listener-2'. There are two tabs: 'Listener Settings' and 'SSL'. The 'SSL' tab is active, showing the following settings:

- Client Authentication:** Enabled
Requires the client to authenticate itself to the server
- Certificate nickname:**
Takes a single value, identifies the server's keypair and certificate
- SSL3:** Enabled
- SSL2:** Enabled
- TLS:** Enabled

Below the SSL settings is the 'Ciphersuites' section:

- Ciphersuites**
- If no cipher suite is added, it means ALL cipher suite will be chosen.

At the bottom, there are two lists for cipher suites:

- Available Common Ciphersuites:**
 - SSL_RSA_WITH_RC4_128_MD5
 - SSL_RSA_WITH_RC4_128_SHA
 - TLS_RSA_WITH_AES_128_CBC_SHA
 - SSL_RSA_WITH_3DES_EDE_CBC_SHA
- Selected Common Ciphersuites:** (Empty list)

Navigation buttons between the lists include: 'Add >', 'Add All >>', '< Remove', and '<< Remove All'.

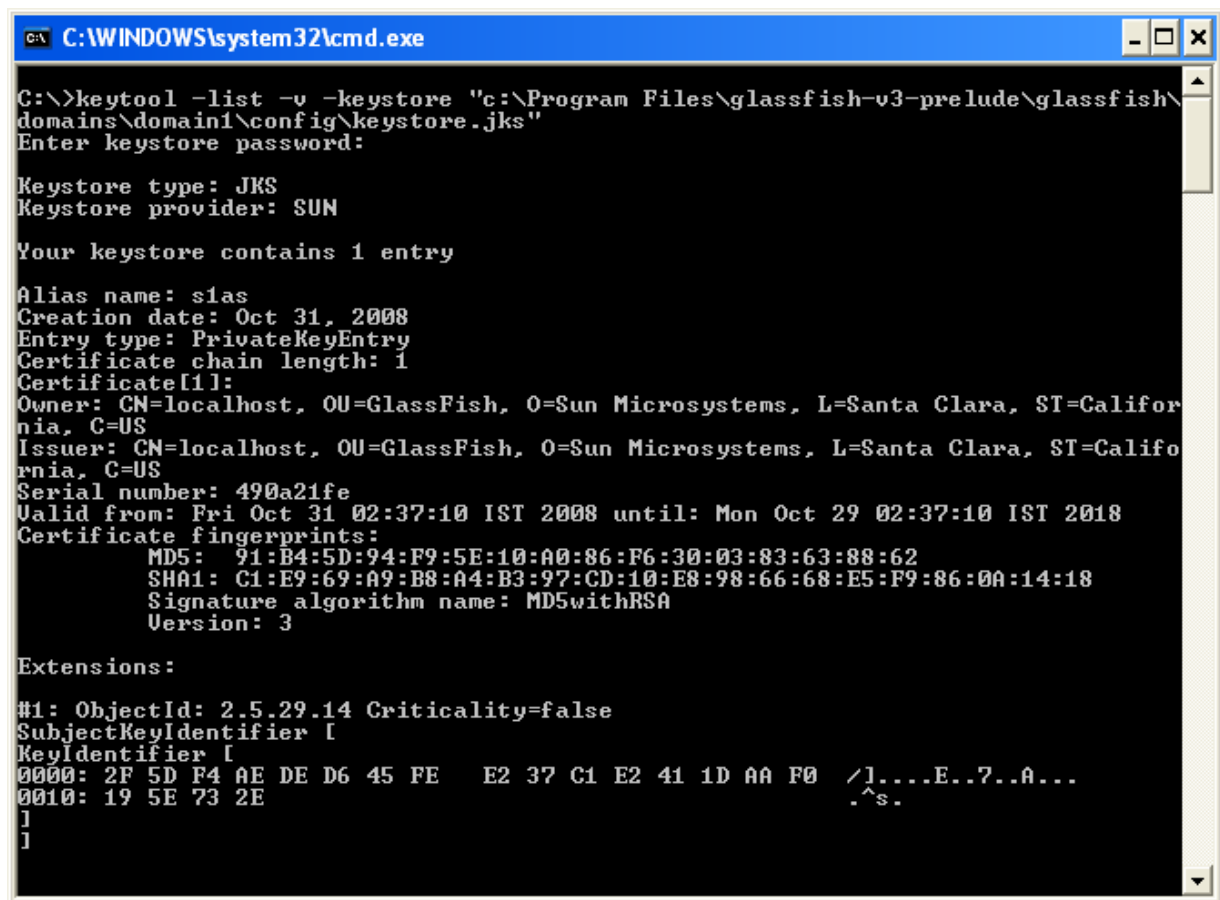
3. To view the information about certificate(s) in default keystore

a) Glassfish Keystore

```
C:\>keytool -list -v -keystore "c:\Program Files\glassfish-v3-  
prelude\glassfish\domains\domain1\config\keystore.jks
```

Keystore password is masterpassword of domain that is defined by user during domain creation.

(For netbeans glassfish the password is "changeit")

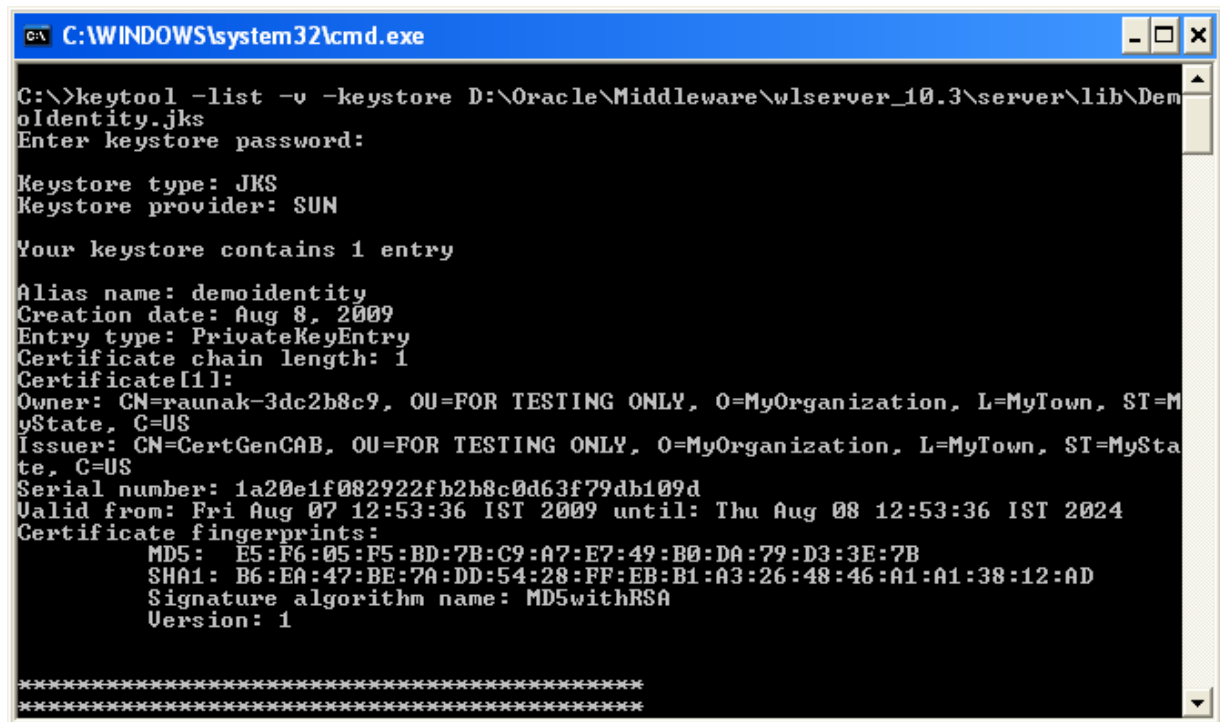


```
C:\WINDOWS\system32\cmd.exe  
C:\>keytool -list -v -keystore "c:\Program Files\glassfish-v3-  
prelude\glassfish\domains\domain1\config\keystore.jks"  
Enter keystore password:  
Keystore type: JKS  
Keystore provider: SUN  
Your keystore contains 1 entry  
Alias name: s1as  
Creation date: Oct 31, 2008  
Entry type: PrivateKeyEntry  
Certificate chain length: 1  
Certificate[1]:  
Owner: CN=localhost, OU=GlassFish, O=Sun Microsystems, L=Santa Clara, ST=Califor  
nia, C=US  
Issuer: CN=localhost, OU=GlassFish, O=Sun Microsystems, L=Santa Clara, ST=Califo  
rnia, C=US  
Serial number: 490a21fe  
Valid from: Fri Oct 31 02:37:10 IST 2008 until: Mon Oct 29 02:37:10 IST 2018  
Certificate fingerprints:  
MD5: 91:B4:5D:94:F9:5E:10:A0:86:F6:30:03:83:63:88:62  
SHA1: C1:E9:69:A9:B8:A4:B3:97:CD:10:E8:98:66:68:E5:F9:86:0A:14:18  
Signature algorithm name: MD5withRSA  
Version: 3  
Extensions:  
#1: ObjectId: 2.5.29.14 Criticality=false  
SubjectKeyIdentifier [  
KeyIdentifier [  
0000: 2F 5D F4 AE DE D6 45 FE E2 37 C1 E2 41 1D AA F0 /]....E..7..A...  
0010: 19 5E 73 2E .^s.  
]  
]
```

b) Weblogic Keystore

```
C:\>keytool -list -v -keystore  
D:\Oracle\Middleware\wlserver_10.3\server\lib\DemoIdentity.jks
```

Default Password for DemoIdentity.jks is DemoIdentityKeyStorePassPhrase

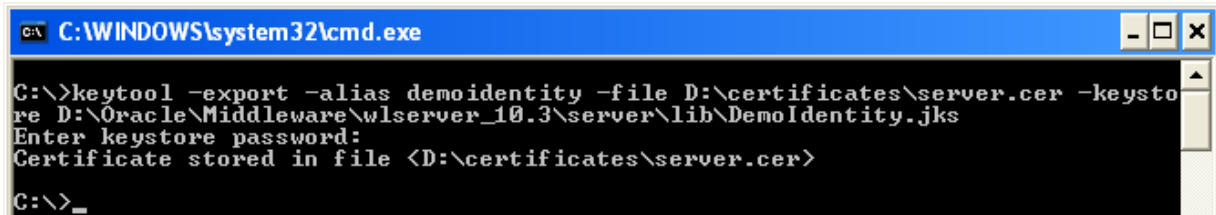


```
C:\WINDOWS\system32\cmd.exe  
C:\>keytool -list -v -keystore D:\Oracle\Middleware\wlserver_10.3\server\lib\Dem  
oIdentity.jks  
Enter keystore password:  
  
Keystore type: JKS  
Keystore provider: SUN  
  
Your keystore contains 1 entry  
  
Alias name: demoidentity  
Creation date: Aug 8, 2009  
Entry type: PrivateKeyEntry  
Certificate chain length: 1  
Certificate[1]:  
Owner: CN=raunak-3dc2b8c9, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=My  
yState, C=US  
Issuer: CN=CertGenCAB, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MySta  
te, C=US  
Serial number: 1a20e1f082922fb2b8c0d63f79db109d  
Valid from: Fri Aug 07 12:53:36 IST 2009 until: Thu Aug 08 12:53:36 IST 2024  
Certificate fingerprints:  
    MD5:   E5:F6:05:F5:BD:7B:C9:A7:E7:49:B0:DA:79:D3:3E:7B  
    SHA1:  B6:EA:47:BE:7A:DD:54:28:FF:EB:B1:A3:26:48:46:A1:A1:38:12:AD  
Signature algorithm name: MD5withRSA  
Version: 1  
  
*****  
*****
```

4. Export the certificate in keystore to a file. This certificate file will be imported to client keystore.

a) Weblogic Certificate

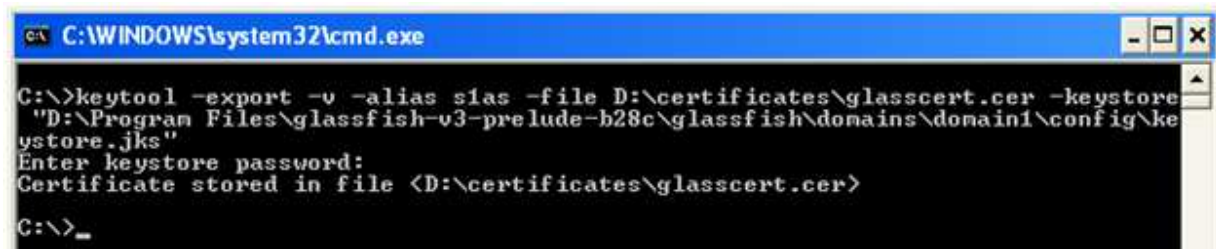
```
C:\>keytool -export -alias demoidentity -file D:\certificates\server.cer -keystore D:\Oracle\Middleware\wlserver_10.3\server\lib\DemoIdentity.jks
```



```
C:\WINDOWS\system32\cmd.exe
C:\>keytool -export -alias demoidentity -file D:\certificates\server.cer -keystore D:\Oracle\Middleware\wlserver_10.3\server\lib\DemoIdentity.jks
Enter keystore password:
Certificate stored in file <D:\certificates\server.cer>
C:\>_
```

b) Glassfish Certificate

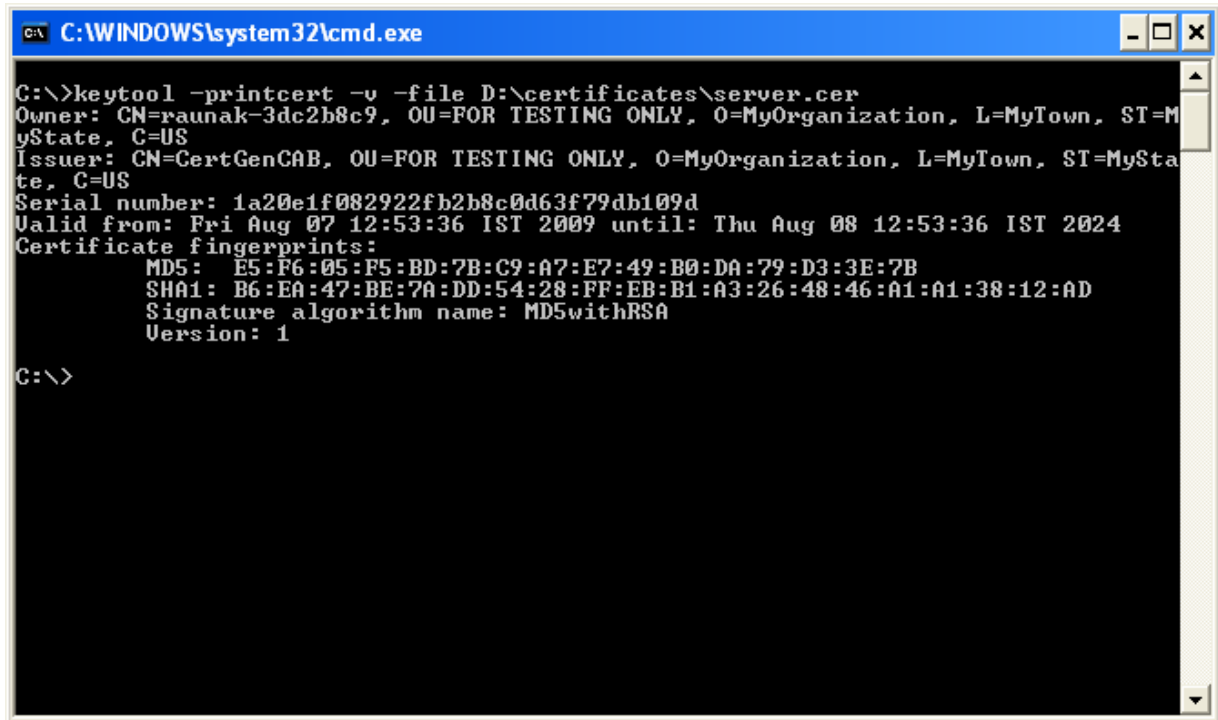
```
C:\> keytool -export -v -alias s1as -file D:\certificates\glasscert.cer -keystore "D:\Program Files\glassfish-v3-prelude-b28c\glassfish\domains\domain1\config\keystore.jks"
```



```
C:\WINDOWS\system32\cmd.exe
C:\>keytool -export -v -alias s1as -file D:\certificates\glasscert.cer -keystore "D:\Program Files\glassfish-v3-prelude-b28c\glassfish\domains\domain1\config\keystore.jks"
Enter keystore password:
Certificate stored in file <D:\certificates\glasscert.cer>
C:\>_
```

5. To print the information about the certificate created

```
C:\>keytool -printcert -v -file D:\certificates\server.cer
```



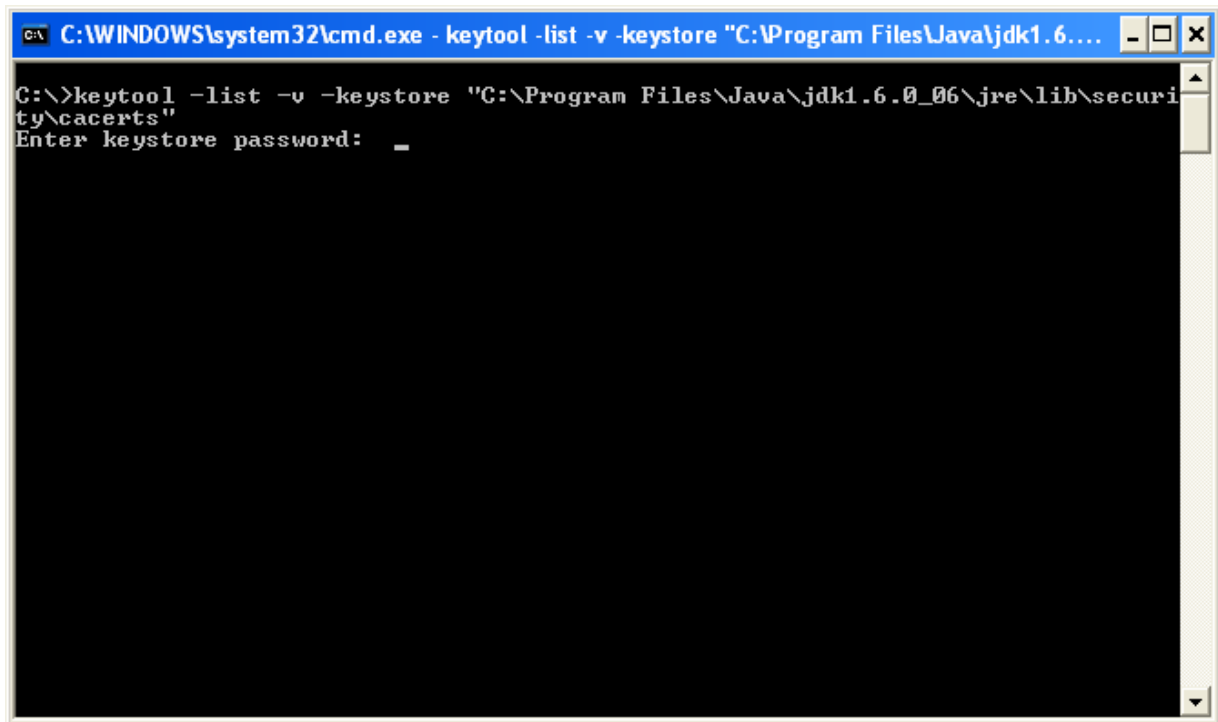
```
C:\WINDOWS\system32\cmd.exe
C:\>keytool -printcert -v -file D:\certificates\server.cer
Owner: CN=raunak-3dc2b8c9, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MyState, C=US
Issuer: CN=CertGenCAB, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MyState, C=US
Serial number: 1a20e1f082922fb2b8c0d63f79db109d
Valid from: Fri Aug 07 12:53:36 IST 2009 until: Thu Aug 08 12:53:36 IST 2024
Certificate fingerprints:
    MD5:  E5:F6:05:F5:BD:7B:C9:A7:E7:49:B0:DA:79:D3:3E:7B
    SHA1: B6:EA:47:BE:7A:DD:54:28:FF:EB:B1:A3:26:48:46:A1:A1:38:12:AD
Signature algorithm name: MD5withRSA
Version: 1

C:\>
```


6. To view the information about certificates in the client keystore

(Java provides its own truststore which is placed in "C:\Program Files\Java\jdk1.6.0_06\jre\lib\security" directory with name cacerts)

```
C:\>keytool -list -v -keystore "C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts"
```



```
C:\WINDOWS\system32\cmd.exe - keytool -list -v -keystore "C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts"
C:\>keytool -list -v -keystore "C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts"
Enter keystore password: _
```

```
C:\WINDOWS\system32\cmd.exe
SubjectKeyIdentifier [
KeyIdentifier [
0000: BE A8 A0 74 72 50 6B 44 B7 C9 23 D8 FB A8 FF B3 ...trPkD..#.....
0010: 57 6B 68 6C Wkhl
]
]

#3: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
SSL CA
S/MIME CA
Object Signing CA]

#4: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: BE A8 A0 74 72 50 6B 44 B7 C9 23 D8 FB A8 FF B3 ...trPkD..#.....
0010: 57 6B 68 6C Wkhl
]
]

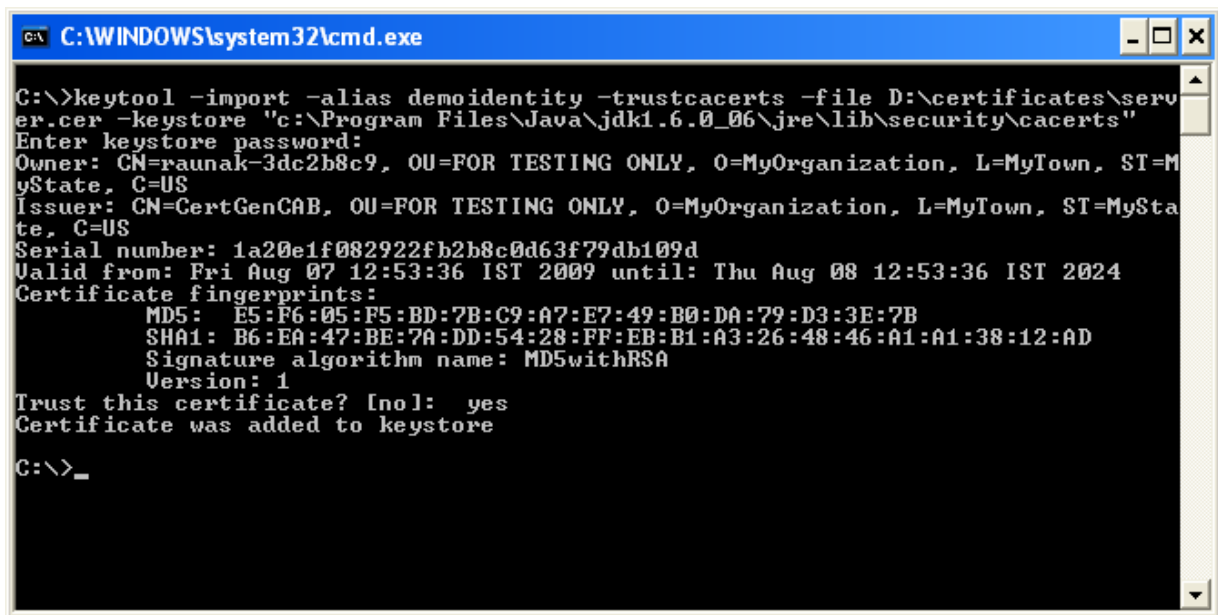
*****
*****

C:\>
```

7. Import the server certificate into the client cacert

a) Weblogic Certificate

```
C:\>keytool -import -alias demoidentity -trustcacerts -file D:\certificates\server.cer -keystore "c:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts"
```



```
C:\WINDOWS\system32\cmd.exe
C:\>keytool -import -alias demoidentity -trustcacerts -file D:\certificates\server.cer -keystore "c:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts"
Enter keystore password:
Owner: CN=raunak-3dc2b8c9, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MyState, C=US
Issuer: CN=CertGenCAB, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MyState, C=US
Serial number: 1a20e1f082922fb2b8c0d63f79db109d
Valid from: Fri Aug 07 12:53:36 IST 2009 until: Thu Aug 08 12:53:36 IST 2024
Certificate fingerprints:
    MD5:  E5:F6:05:F5:BD:7B:C9:A7:E7:49:B0:DA:79:D3:3E:7B
    SHA1: B6:EA:47:BE:7A:DD:54:28:FF:EB:B1:A3:26:48:46:A1:A1:38:12:AD
    Signature algorithm name: MD5withRSA
    Version: 1
Trust this certificate? [no]: yes
Certificate was added to keystore
C:\>_
```

b) Glassfish Certificate

```
C:\>keytool -import -v -trustcacerts -alias s1as -keystore "C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts" -file D:\certificates\glasscert.cer
```

```
C:\WINDOWS\system32\cmd.exe
C:\>keytool -import -v -trustcacerts -alias sias -keystore "C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts" -file D:\certificates\glasscert.cer
Enter keystore password:
Owner: CN=localhost, OU=GlassFish, O=Sun Microsystems, L=Santa Clara, ST=California, C=US
Issuer: CN=localhost, OU=GlassFish, O=Sun Microsystems, L=Santa Clara, ST=California, C=US
Serial number: 48ffe311
Valid from: Thu Oct 23 08:06:01 IST 2008 until: Sun Oct 21 08:06:01 IST 2018
Certificate fingerprints:
    MD5: 6C:19:37:05:6B:F6:96:2D:D1:22:3F:B3:D8:9B:60:B3
    SHA1: 00:89:E8:22:9B:5C:0C:CA:0F:11:7C:A3:30:FC:A1:F5:8B:03:3C:D9
    Signature algorithm name: MD5withRSA
    Version: 3

Extensions:
#1: ObjectID: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 30 14 E0 BD 00 EF 9E 71    DB E8 3E B0 81 B6 30 D3    0.....q..>...0.
0010: 8A A9 6A BB                    ..j.
]
]

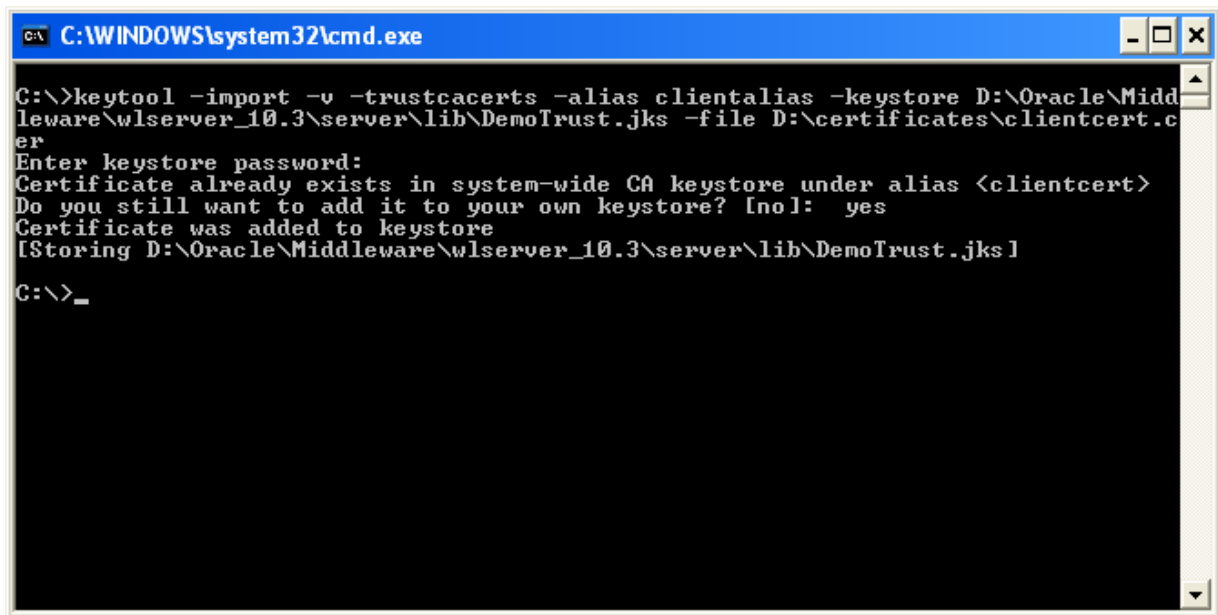
Trust this certificate? [no]: yes
Certificate was added to keystore
[Storing C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts]
```

8. Import the client certificate into the server cacert

a) Importing to Weblogic truststore

(Note: Default password for DemoTrust.jks is DemoTrustKeyStorePassPhrase)

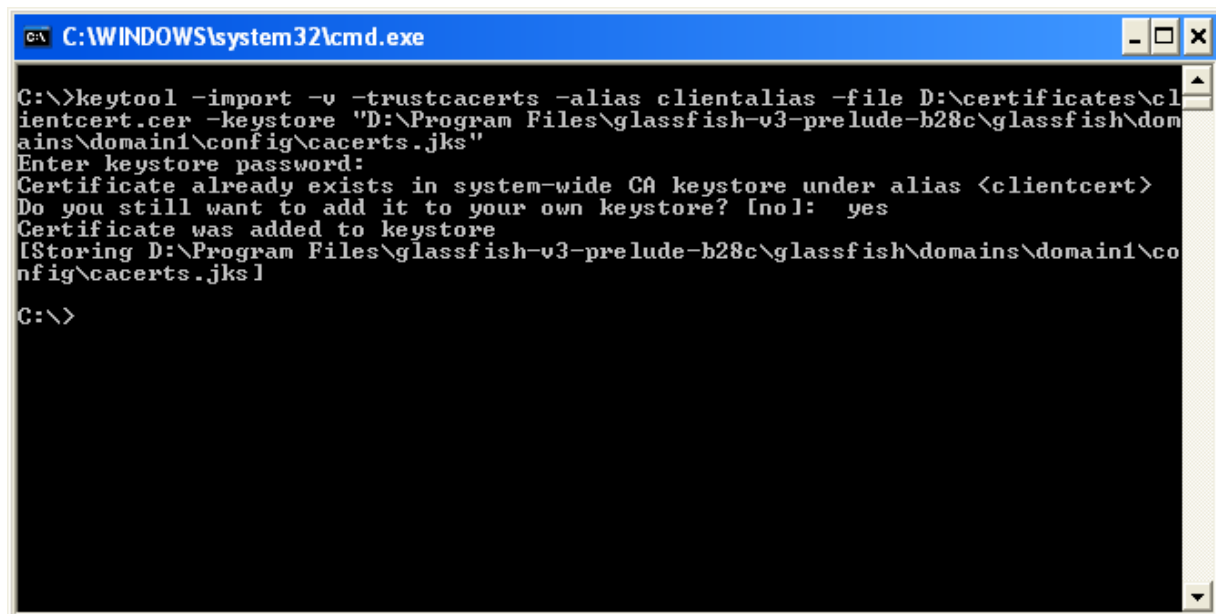
```
C:\>keytool -import -v -trustcacerts -alias clientalias -keystore  
D:\Oracle\Middleware\wlserver_10.3\server\lib\DemoTrust.jks -file D:\certificates\clientcert.cer
```



```
C:\WINDOWS\system32\cmd.exe  
C:\>keytool -import -v -trustcacerts -alias clientalias -keystore D:\Oracle\Middleware\wlserver_10.3\server\lib\DemoTrust.jks -file D:\certificates\clientcert.cer  
Enter keystore password:  
Certificate already exists in system-wide CA keystore under alias <clientcert>  
Do you still want to add it to your own keystore? [no]: yes  
Certificate was added to keystore  
[Storing D:\Oracle\Middleware\wlserver_10.3\server\lib\DemoTrust.jks]  
C:\>_
```

b) Importing to Glassfish truststore

```
keytool -import -v -trustcacerts -alias clientalias -file D:\certificates\clientcert.cer -keystore "D:\Program Files\glassfish-v3-prelude-b28c\glassfish\domains\domain1\config\cacerts.jks"
```



```
C:\WINDOWS\system32\cmd.exe
C:\>keytool -import -v -trustcacerts -alias clientalias -file D:\certificates\clientcert.cer -keystore "D:\Program Files\glassfish-v3-prelude-b28c\glassfish\domains\domain1\config\cacerts.jks"
Enter keystore password:
Certificate already exists in system-wide CA keystore under alias <clientcert>
Do you still want to add it to your own keystore? [no]: yes
Certificate was added to keystore
[Storing D:\Program Files\glassfish-v3-prelude-b28c\glassfish\domains\domain1\config\cacerts.jks]
C:\>
```

Additional Information

Note: If you are using self-signed certificate include following property in JAVA_OPTIONS of setDomainEnv of weblogic else weblogic will show Basic CA constraint error and restart the server.

-Dweblogic.security.SSL.enforceConstraints=off

```
set JAVA_OPTIONS=%JAVA_OPTIONS% %JAVA_PROPERTIES% -
Dweblogic.security.SSL.enforceConstraints=off
```

Error

If you get the following error while running the client for the web service

```
javax.xml.ws.WebServiceException: Failed to access the WSDL at: https://localhos
t:7002/BasicOperations/BasicOperationService?wsdl. It failed with:
    Received fatal alert: handshake_failure.
    at com.sun.xml.internal.ws.wSDL.parser.RuntimeWSDLParser.tryWithMex(Unkn
own Source)
    at com.sun.xml.internal.ws.wSDL.parser.RuntimeWSDLParser.parse(Unknown S
ource)
    at com.sun.xml.internal.ws.client.WSServiceDelegate.parseWSDL(Unknown So
urce)
    at com.sun.xml.internal.ws.client.WSServiceDelegate.<init>(Unknown Sourc
e)
    at com.sun.xml.internal.ws.client.WSServiceDelegate.<init>(Unknown Sourc
e)
    at com.sun.xml.internal.ws.spi.ProviderImpl.createServiceDelegate(Unknow
n Source)
    at javax.xml.ws.Service.<init>(Unknown Source)
    at basicoperationservice.wsdl.BasicOperationService.<init>(BasicOperatio
nService.java:46)
    at client.ClientMain.subtraction(ClientMain.java:51)
    at client.ClientMain.main(ClientMain.java:91)
Caused by: javax.net.ssl.SSLHandshakeException: Received fatal alert: handshake_
failure
    at com.sun.net.ssl.internal.ssl.Alerts.getSSLException(Unknown Source)
    at com.sun.net.ssl.internal.ssl.Alerts.getSSLException(Unknown Source)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.recvAlert(Unknown Source)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.readRecord(Unknown Source)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.performInitialHandshake(Un
known Source)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.startHandshake(Unknown Sou
rce)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.startHandshake(Unknown Sou
rce)
    at sun.net.www.protocol.https.HttpsClient.afterConnect(Unknown Source)
    at sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.connect
(Unknown Source)
    at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown So
urce)
    at sun.net.www.protocol.https.HttpsURLConnectionImpl.getInputStream(Unkn
own Source)
    at java.net.URL.openStream(Unknown Source)
    at com.sun.xml.internal.ws.wSDL.parser.RuntimeWSDLParser.createReader(Un
known Source)
    at com.sun.xml.internal.ws.wSDL.parser.RuntimeWSDLParser.resolveWSDL(Unk
nown Source)
    ... 9 more
```

Solution

Make sure that certificates are imported correctly on both client and server side. Error signifies that either server hello or client hello was incomplete.

To check for detailed debug information for SSL include the following property during invocation of client

```
-Djavax.net.debug=ssl or -Djavax.net.debug=handshake
```

```
java -Djavax.net.debug=ssl <client class>
```

or

You can include it in your code also

```
System.setProperty("javax.net.debug","ssl");
```

Error

```
javax.xml.ws.WebServiceException: Failed to access the WSDL at: https://localhost:7002/BasicOperations/BasicOperationService?wsdl. It failed with:
    sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
        at com.sun.xml.internal.ws.wSDL.parser.RuntimeWSDLParser.tryWithMex(Unknown Source)
        at com.sun.xml.internal.ws.wSDL.parser.RuntimeWSDLParser.parse(Unknown Source)
        at com.sun.xml.internal.ws.client.WSServiceDelegate.parseWSDL(Unknown Source)
        at com.sun.xml.internal.ws.client.WSServiceDelegate.<init>(Unknown Source)
        at com.sun.xml.internal.ws.client.WSServiceDelegate.<init>(Unknown Source)
        at com.sun.xml.internal.ws.spi.ProviderImpl.createServiceDelegate(Unknown Source)
        at javax.xml.ws.Service.<init>(Unknown Source)
        at basicoperationservice.wSDL.BasicOperationService.<init>(BasicOperationService.java:46)
        at client.ClientMain.subtraction(ClientMain.java:51)
        at client.ClientMain.main(ClientMain.java:91)
Caused by: javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
    at com.sun.net.ssl.internal.ssl.Alerts.getSSLException(Unknown Source)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.fatal(Unknown Source)
    at com.sun.net.ssl.internal.ssl.Handshaker.fatalSE(Unknown Source)
    at com.sun.net.ssl.internal.ssl.Handshaker.fatalSE(Unknown Source)
    at com.sun.net.ssl.internal.ssl.ClientHandshaker.serverCertificate(Unknown Source)
    at com.sun.net.ssl.internal.ssl.ClientHandshaker.processMessage(Unknown Source)
    at com.sun.net.ssl.internal.ssl.Handshaker.processLoop(Unknown Source)
    at com.sun.net.ssl.internal.ssl.Handshaker.process_record(Unknown Source)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.readRecord(Unknown Source)
```



```

    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.performInitialHandshake(Un
known Source)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.startHandshake(Unknown Sou
rce)
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.startHandshake(Unknown Sou
rce)
    at sun.net.www.protocol.https.HttpsClient.afterConnect(Unknown Source)
    at sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.connect
(Unknown Source)
    at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown So
urce)
    at sun.net.www.protocol.https.HttpURLConnectionImpl.getInputStream(Unkn
own Source)
    at java.net.URL.openStream(Unknown Source)
    at com.sun.xml.internal.ws.wSDL.parser.RuntimeWSDLParser.createReader(Un
known Source)
    at com.sun.xml.internal.ws.wSDL.parser.RuntimeWSDLParser.resolveWSDL(Unk
nown Source)
    ... 9 more
Caused by: sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find vali
d certification path to requested target
    at sun.security.validator.PKIXValidator.doBuild(Unknown Source)
    at sun.security.validator.PKIXValidator.engineValidate(Unknown Source)
    at sun.security.validator.Validator.validate(Unknown Source)
    at com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.validate(Unknown So
urce)
    at com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.checkServerTrusted(
Unknown Source)
    at com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.checkServerTrusted(
Unknown Source)
    ... 24 more
Caused by: sun.security.provider.certpath.SunCertPathBuilderException: unable to
find valid certification path to requested target
    at sun.security.provider.certpath.SunCertPathBuilder.engineBuild(Unknown
Source)
    at java.security.cert.CertPathBuilder.build(Unknown Source)
    ... 30 more

```

Solution

Error signifies that the client was not able to find a valid certificate keystore path. Include the following properties during client invocation

```

-Djavax.net.ssl.keyStore="C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts" -
Djavax.net.ssl.keyStorePassword=changeit -Djavax.net.ssl.trustStore="C:\Program
Files\Java\jdk1.6.0_06\jre\lib\security\cacerts" -
Djavax.net.ssl.trustStorePassword=changeit

```

For example:

```

java -Djavax.net.ssl.keyStore="C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts"
-Djavax.net.ssl.keyStorePassword=changeit -Djavax.net.ssl.trustStore="C:\Program
Files\Java\jdk1.6.0_06\jre\lib\security\cacerts" -
Djavax.net.ssl.trustStorePassword=changeit <client class>

```

Error:

When Weblogic is acting as client (i.e. when service deployed on weblogic is accessing the service deployed on another server) in Two Way SSL, you may get the following error "**No suitable identity certificate chain has been found.**"

Solution

Go to SSL tab of your server where application is deployed and enable **Use Server Certs**

The screenshot displays the WebLogic Administration Console interface for configuring SSL settings. On the left, a navigation tree shows the path to the SSL configuration page. Below the tree are two panels: 'How do I...' with a list of tasks including 'Configure two-way SSL', and 'System Status' showing the health of running servers as 'OK (1)'. The main configuration area is divided into sections: 'Identity' (Private Key Location, Private Key Alias, Private Key Passphrase, Certificate Location), 'Trust' (Trusted Certificate Authorities), and 'Advanced' (Hostname Verification, Custom Hostname Verifier, Export Key Lifespan, Use Server Certs, Two Way Client Cert Behavior). The 'Use Server Certs' checkbox is checked, and 'Two Way Client Cert Behavior' is set to 'Client Certs Requested and Enforced'.

Error

If you get the following error while running your client:

```
javax.xml.ws.WebServiceException: Failed to access the WSDL at: https://localhost:7002/BasicOperations/BasicOperationService?wsdl. It failed with:  
    java.security.cert.CertificateException: No subject alternative names present.  
    at com.sun.xml.internal.ws.wsdl.parser.RuntimeWSDLParser.tryWithMex(RuntimeWSDLParser.java:136)  
    at com.sun.xml.internal.ws.wsdl.parser.RuntimeWSDLParser.parse(RuntimeWSDLParser.java:122)  
    at com.sun.xml.internal.ws.client.WSServiceDelegate.parseWSDL(WSServiceDelegate.java:226)  
    at com.sun.xml.internal.ws.client.WSServiceDelegate.<init>(WSServiceDelegate.java:189)  
    at com.sun.xml.internal.ws.client.WSServiceDelegate.<init>(WSServiceDelegate.java:159)  
    at com.sun.xml.internal.ws.spi.ProviderImpl.createServiceDelegate(ProviderImpl.java:81)
```

Solution:

Include the following code in your code

```
static {  
    //WORKAROUND. TO BE REMOVED.  
  
    javax.net.ssl.HttpURLConnection.setDefaultHostnameVerifier(new  
    javax.net.ssl.HostnameVerifier() {  
  
        public boolean verify(String hostname,  
            javax.net.ssl.SSLSession sslSession) {  
            return true;  
        }  
    });  
}
```

Or

In weblogic

Start Weblogic -> Login to console -> Click on Environment -> Servers -> SSL ->Advanced

Migratable Targets
 Machines
 Work Managers
 Startup & Shutdown Classes
 Deployments
 Services
 Security Realms
 Interoperability
 Diagnostics

How do I...
 • Configure identity and trust
 • Set up SSL
 • Verify host name verification is enabled
 • Configure a custom host name verifier
 • Configure two-way SSL

System Status
Health of Running Servers
 Failed (0)
 Critical (0)
 Overloaded (0)
 Warning (0)
 OK (1)

Identity and Trust Locations: Keystores
 Indicates where SSL should find the server's identity (certificate key) as well as the server's trust (trusted CAs). [More Info...](#)

Identity

Private Key Location: from Demo Identity Keystore
 The keystore attribute that defines the location of the private key. [More Info...](#)

Private Key Alias: DemoIdentity
 The keystore attribute that defines the string alias used to store the server's private key. [More Info...](#)

Private Key Passphrase: [REDACTED]
 The keystore attribute that defines the passphrase used to retrieve the server's private key. [More Info...](#)

Certificate Location: from Demo Identity Keystore
 The keystore attribute that defines the location of the trusted certificate. [More Info...](#)

Trust

Trusted Certificate Authorities: from Demo Trust Keystore and Java Standard Trust Keystore
 The keystore attribute that defines the location of the certificate authorities. [More Info...](#)

Advanced

Hostname Verification: None
 Specifies whether to ignore the installed implementation of the `weblogic.security.SSL.HostnameVerifier` interface (when this server acts as a client to another application server). [More Info...](#)

Custom Hostname Verifier: [REDACTED]
 The name of the class that implements the `weblogic.security.SSL.HostnameVerifier` interface. [More Info...](#)

Export Key Lifespan: 500
 Indicates the number of times WebLogic Server can use an exportable client before generating a new key. The more secure you want WebLogic Server to be, the

Client Run

```
C:\Documents and Settings\Raunak\Desktop\client>java -Djavax.net.ssl.keyStore="C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts" -Djavax.net.ssl.keyStorePassword=changeit -Djavax.net.ssl.trustStore="C:\Program Files\Java\jdk1.6.0_06\jre\lib\security\cacerts" -Djavax.net.ssl.trustStorePassword=changeit client.ClientMain
Enter your choice:
  1. Addition
  2. Subtraction
2
Enter the first number for the operation..
32
Enter the second number for the operation..
28
Client subtract Port is JAX-WS RI 2.1.6 in JDK 6: Stub for https://localhost:7002/BasicOperations/BasicOperationService
Result = 4
Result of subtraction is 4
```