Configuring for Services Deployed Over HTTPS

To configureSOAtest and Virtualize to work withWebservices deployed using HTTPS (HTTP via the SSL), you need to identify and register the certificate being used for the HTTPS connection from the server:

1. CloseSOAtest and/or Virtualize if it is currently open.
2. Identify the location of the server certificate used for the HTTPS connection.
3. Ensure that this certificate's COMMON NAME parameter contains both the server's machine name and the subdomain (for example, `machine.company.com`).
4. Copy the certificate to the following location:

   Virtualize:

   `<virtualize_install_dir>/plugins/com.parasoft.xtest.libs.web_<virtualize_version_number>/root/lib`

   SOAtest:

   `<soatest_install_dir>/eclipse/plugins/com.parasoft.xtest.libs.web_<soatest_version_number>/root/lib.`

   This directory should contain a `cacerts` file in which the trusted certificates are stored.

5. Execute a command of the following format:
   `keytool -import -alias <certificate_alias> -file <certificate_file> -keystore cacerts`

   For example, if your certificate file is named `test.cert`, you would execute the following command from the`<soatest_install_dir>/plugins/com.parasoft.xtest.libs.web_<soatest_version_number>/root/lib` or `<virtualize_install_dir>/plugins/com.parasoft.xtest.libs.web_<virtualize_version_number>/root/lib` prompt:
   `keytool -import -alias serverTrustCert -file test.cert -keystore cacerts`
   This will import the certificate into the cacerts file with the alias "`serverTrustCert`".

   > **keytool path must be set**
   >
   > Before executing `keytool` commands, you must first set your path to include Java's `keytool` executable. You can use the version of the Java binaries that ship withSOAtest and Virtualize.To add the included Java binaries to your path, open a command line prompt and enter the following before referencing the keytool:
   >
   > `PATH =%PATH%; <Parasoft Test install dir>\<Parasoft Test version number>\plugins\com.parasoft.xtest.jdk.eclipse.core.<arch>_<java_version>\jdk\bin`
   >
   > Note that `<Parasoft Test install dir>` references the location where Parasoft Test is installed (e.g., "C:\Program Files\Parasoft\Test" on Windows), `<arch>` refers to the architecture (e.g., win32.x86, linux.x86, win32.x86_64, or linux.x86_64), and `<java_version>` references the Java version included with your Parasoft Test installation.

6. When prompted to enter a keystore password, enter `changeit`.
7. When asked whether you want to trust this certificate, enter `yes`. You will then see a message indicating that the certificate has been added to the keystore.
8. (Optional) Verify that the certificate has been added to the keystore by entering the following command, then checking the file that opens:
   `keytool -list -keystore cacerts`
9. LaunchSOAtest or Virtualize

   and try to access the service again.

If you experience issues working with services deployed over HTTS, verify the following:

1. Your server is running.
2. You used the full name of the machine when trying to communicate with this HTTPS.
3. The server certificate was created with the full name.
4. The name on the certificate is identical to the name the client tried to access it with.

If you cannot satisfy the above requirements (for example, if you don't have necessary permissions):

1. Choose **Parasoft> Preferences** to open the Preferences dialog.
2. Select **Parasoft> Seacurity** from thace left pane of the Preferences dialog, then select the **Trust all certificates** option in the right pane.
3. Click **OK** or **Apply** to apply this change.

SOAtest/Virtualize will then try to access any WSDL you specify, regardless of any problems with the certificate. However,SOAtest/Virtualizewill still try use the certificate while trying to send SOAP messages because it is required to do so.

> ⚠ **Note**
>
> You must add certificates to cacerts files on load test slave machines as well as on the master machine. Otherwise, SSL connections will not work when running a load test with slave machines.

If none of these procedures solve your problem, contact Parasoft in one of the ways described in Contacting Parasoft Technical Support.

Debugging SSL Issues

SOAtest and Virtualize run on a standard JVM. To show the SSL/TLS handshake details and help identify causes of SSL connection problems, enable JVM network and SSL debugging:

1. Open a command line console and navigate to the SOAtest installation directory.
2. Start the executable with the arguments:
   `-J-Dssl.debug=true -J-Djavax.net.debug=all -consolelog`

SOAtest/Virtualizewill start as usual, but whenever SSL connections are made, debugging output will be printed on the console. If you wish to save the trace output to a file (for example, `output.txt`), you may append the following to the end of the command :

```
> output.txt
```

For more information about managing keys and certificates using the Java keytool, see the Oracle Java documentation. refer to:

- **Windows:** http://docs.oracle.com/javase/8/docs/technotes/tools/windows/keytool.html
- **Linux, Mac:** https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html

JMS SSL

See JMS.