How to Connect to Server using SSL and Client Certificate

This article applies to: **Deskzilla 1.x-2.x, JIRA Client 1.x-2.x**, and connecting to Bugzilla and JIRA servers via https://... connections.
For clarity, this article is written about JIRA Client and JIRA, but it applies to Deskzilla / Bugzilla as well.

Problem

You need to use a client SSL certificate to connect to the server. In most cases, client certificate is not used. But in some high-security configurations it may be required.

When a connection is attempted to a server that requires a client certificate, it may result in the following error from JIRA Client:

```
Connection problem: Software caused connection abort: recv failed
```

JIRA Client does not have a direct way to specify a certificate. This article explains how to set up JIRA Client for using client certificate with standard Java tools.

The server certificate, which is used by the server to authenticate the connection, may be self-signed. In that case you will also need to apply the solution for self-signed server certificate.

Solution

The solution is to create a secure file for storing your client certificate and specify it in command-line properties for JIRA Client.

1. Obtain PKCS12 (.p12) file with the client certificate

You may already have it, but if you don't, it's easy to export the certificate from your browser. (We are assuming that you can access JIRA with your browser, hence the browser does have the certificate.)

When specifying password for the exported certificate, enter **at least 6 characters**, and also **without any space or special characters**. Otherwise you may not be able to proceed later.

To export client certificate from **Firefox**, open Tools | Options menu, Advanced tab, and click on View Certificates. Select certificate that matches your server and click Backup button. Enter a file name for .p12 file, and then backup password. Remember backup password.

To export client certificate from **Internet Explorer**, open Tools | Internet Options dialog, Content tab, then click on Certificates button. Find your certificate and click Export. Follow the wizard. Include private key in export. Select PKCS 12 format (although the extension will be PFX, you can rename it to P12). Don't include extra options. Enter password and remember it.

As a result of this step, you should have .p12 file, let's call it my-certificate.p12 and the password for it.

2. Create Java keystore using keytool

Use "keytool" program to transform the .p12 file. Keytool may be found in <JIRA Client Install Dir>\jre\bin (C:\Program Files\JIRA Client\jre\bin), or in any Java installation on your computer.

Run the following command:

```
keytool -importkeystore -srckeystore c:\path\to\p12\my-certificate.p12 -srcstoretype pkcs12  -destkeystore c:
\temporary\path\jiraclient.jks -deststoretype jks
```

Substitute path to the exported .p12 file instead of c:\path\to\p12\my-certificate.p12, and any temporary directory instead of c:\temporary\path.

Keytool will ask for password two times - for the source and destination keystores. You should enter the password you created on the previous step - every time. It will be **the same password** for both key stores.

If passwords are different, it may result in the same errors on the client side and the following error in the server logs:

```
SSL Library Error: 336105671 error:140890C7:SSL routines:SSL3_GET_CLIENT_CERTIFICATE:
peer did not return a certificate No CAs known to server for verification?
```

As a result of this step, you should have file jiraclient.jks in a temporary directory.

3. Place jiraclient.jks into installation directory and adjust shortcut for launching JIRA Client

Copy jiraclient.jks from the previous step to C:\Program Files\JIRA Client, or whereever you have JIRA Client installed.

To run JIRA Client with the client certificate, you will need to pass three additional parameters via command-line. To avoid typing them each time, you will need to modify the shortcut or script that you use to start JIRA Client.

On Windows

Find "JIRA Client" start menu item (or other shortcut that you use to launch JIRA Client), *Right-Click* on it and select *Properties*. The *Shortcut* tab will appear, with the *Target* field containing something like *"C:\Program Files\JIRA Client\bin\jiraclient.exe"*.

Click on the Target field and modify it so it says

```
"C:\Program Files\JIRA Client\bin\jiraclient.exe" -J-Dforce.http.jre.executor=true -J-Djavax.net.ssl.
keyStore=jiraclient.jks -J-Djavax.net.ssl.keyStorePassword=<your password created at step 1>
```

Use copy&paste from this article to avoid typos. Substitute your password in place of <your password created at step 1>.

Note the quotes in this example. Don't put additional parameters inside the quotes around the .exe file path.

On Linux

Modify "jiraclient.sh" script. Find line that says *JAVA_OPTIONS="-Xmx600m -Duse.metal=true"*. Modify it so it says

```
JAVA_OPTIONS="-Xmx600m -Duse.metal=true -Dforce.http.jre.executor=true -Djavax.net.ssl.keyStore=/path/to
/jiraclient.jks -Djavax.net.ssl.keyStorePassword=<your password created at step 1>"
```

Note the quotes are around all the line. Use full path to specify the location of jiraclient.jks.

On Mac

Right-click on JIRA Client application and select *Show Package Contents*. Open *Contents* folder. Double-click on the *Info.plist* file. Plist editor should start. Open *Java* section, then *Properties* subsection. Use "+" button to add the following properties:

Name	Value
force.http.jre.executor	true
javax.net.ssl.keyStore	/path/to/jiraclient.jks
javax.net.ssl.keyStorePassword	<your password created at step 1>

Use full path to specify the location of jiraclient.jks.

For self-signed server certificates

If the server uses a self-signed certificate (or a certificate signed by an unknown CA), you will need to explicitly import server's certificate into the Java's trust keystore. (See instructions.) By default, the trust keystore is called *cacerts* and it resides in C:\Program Files\JIRA Client\jre\lib\security\cacerts. With the same method you used for setting the three properties described above, it's possible to specify a different location for cacerts: you need to set *javax.net.ssl.trustStore* property to *</path/to/your/cacerts>*, and, if the password is not default (*changeit*), set *javax.net.ssl.trustStorePassword* property.

This is it!

Start JIRA Client. Try to connect. If it doesn't work, double-check that

- jiraclient.jks file exists and has at least 500 bytes;
- It is correctly pointed to using command-line properties;
- you really launch the same shortcut / script that you have edited.

If all looks correct but it doesn't work, please contact support.