



## **Publisher ID:**

**What, why, how much and how to install it.**

### **Q. What is a Publisher ID?**

**A.** The Publisher ID is a certificate that proves the existence of the developer and the company the developer works for and is used to sign the applications that are submitted to Java Verified for R&D, signing and testing. In other words, the Publisher ID is your passport to using Java Verified services.

The prerequisite for using Java Verified services is to have a Publisher ID.

The key things to note about the Publisher ID are:

- It costs 200USD/year and one PublisherID per company is needed.
- It is valid for one year. When it expires it can no longer be used and must be renewed.
- Signing with a Publisher ID carries no additional cost.
- An application signed with a Publisher ID cannot be installed to a live device (unless installed to a MIDP 1 device where the signature has no effect).

### **Q. How do I obtain or renew a Publisher ID?**

**A.** To obtain or renew a Publisher ID, you will need to pay the required fee and pass a simple vetting process, which includes the submission of documentation proving the existence of the company and your right to submit applications on its behalf.

To obtain or renew a Publisher ID go here:

[http://www.trustcenter.de/en/products/tc\\_publisher\\_id\\_for\\_java\\_verified.htm](http://www.trustcenter.de/en/products/tc_publisher_id_for_java_verified.htm)

For more details about what you are spending your money on, go to:

[http://www.trustcenter.de/en/products/tc\\_publisher\\_id\\_for\\_symbian.htm](http://www.trustcenter.de/en/products/tc_publisher_id_for_symbian.htm)

*Note: Java Verified uses the Publisher ID provided by TC TrustCenter. So if you have already obtained a TrustCenter Publisher ID (for example from Symbian Signed) you can use this ID.*

### **Q. How do I use the Publisher ID?**

**A.** After the successful purchase of the Publisher ID it will appear in the certificate store of your Web browser. In order to use the Publisher ID to sign applications, you will firstly need to port it to the Wireless Tool Kit or NetBeans.

To do this, please follow the instructions below, which are based on the following set up:

- Internet Explorer version 6
- Java Platform Standard Edition 6, keytool.exe application

## Option 1: Porting the Publisher ID to a .PFX file

### *In Microsoft Internet Explorer*

1. Go to the last item in the “Tools” menu, which is called “Internet Options”. In “Internet Options” click on the “Content” tab and then select the “Certificates” button, which will allow you to view all certificates (see image 1 below).
2. Select the certificate issued by “TC Trustcenter” and press “Export” to start porting of the certificate.
3. You will be given a number of options when exporting the certificate from the Internet Explorer, so please follow this guidance:
  - Include the private key with the certificate
  - Use Personal Information Exchange format (.PFX)
  - Include all certificates in the certification path
  - No need to select strong protection
  - No need to delete the private key if the export is successful
  - Save the .PFX file to the same directory where the keytool.exe application is located. This should make the next steps easier.

If you have followed the instructions above, the certificate is now in a .PFX file and located in the same directory as the keytool.exe application.

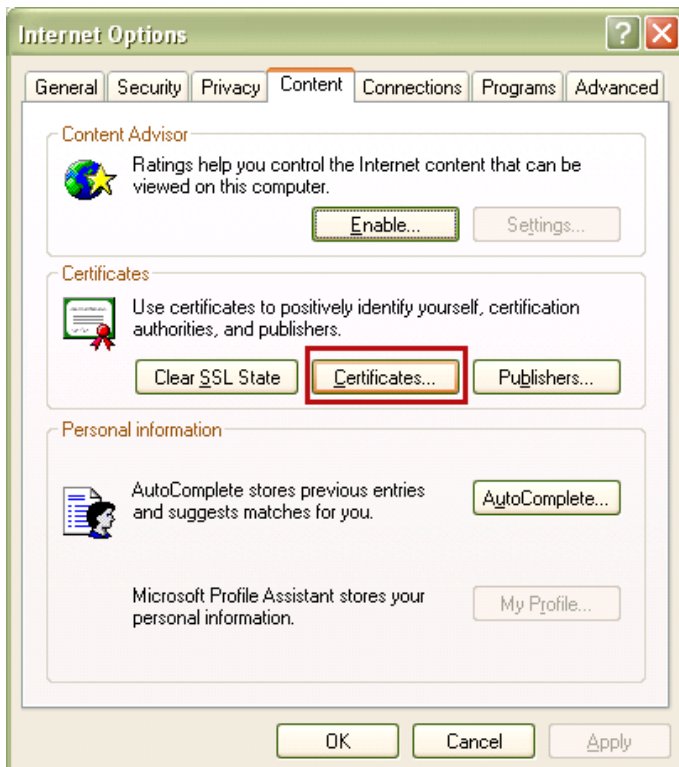
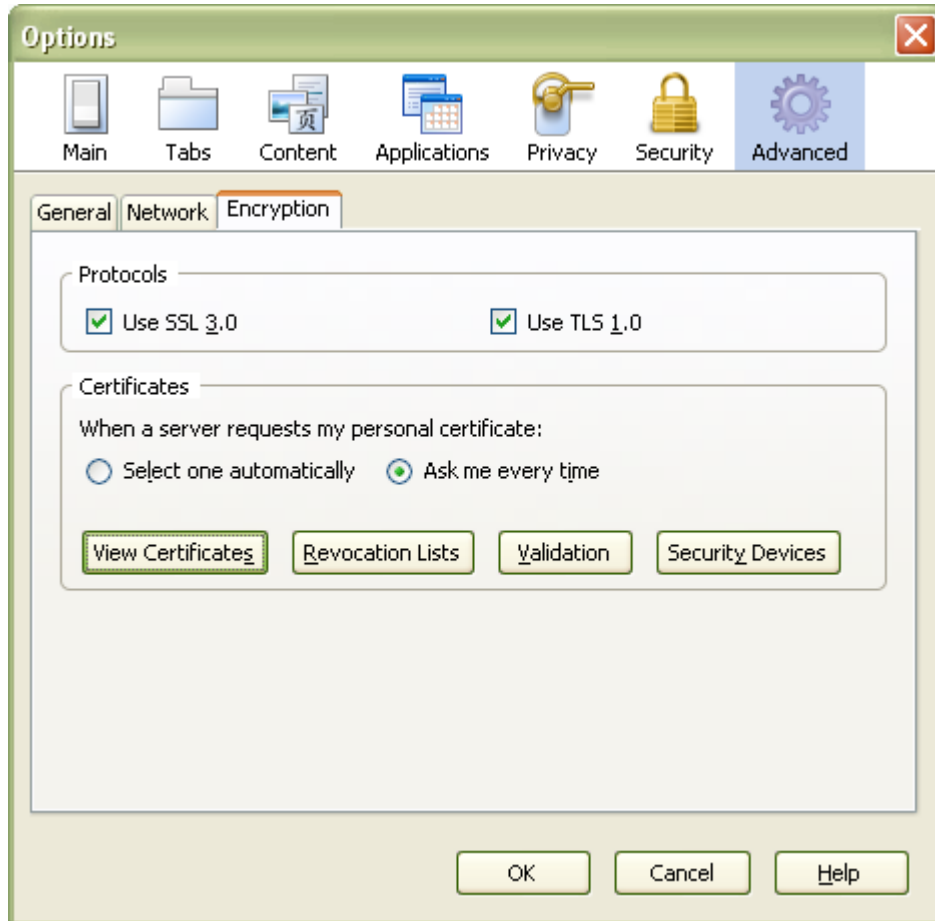


Image 1: The Certificates button

### *In Firefox*

1. Go to the last item in the “Tools” menu, which is called “Options”. In “Options” click on the “Advanced” selection and then select the “Encryption” tab, which will show the

2. "View Certificates" option (see image 2 below). That opens up a new window which allows you to view all certificates.
3. Select the "Your Certificates" tab and from there select the certificate issued by "TC TrustCenter" and press "Backup" to start saving the certificate with a selected name to the desired location.



*Image 2: The View certificates button*

### **Option 2: Copying the Publisher ID to a keystore**

To copy the Publisher ID from the .PFX file to a specific keystore file, you will need to use the key tool application and follow these steps:

1. Open the command prompt and run the keytool.exe application. You will need to define a password and remember it, so please write it down somewhere safe.
2. Use the following command to copy the Publisher ID to a designated keystore file:

```
Keytool -importkeystore -srckeystore "The_Name_Of_The_PFX_file.PFX" -destkeystore Name_For_The_Key_store.jks -srcstoretype PKCS12 -deststoretype JKS
```

*Note: As a shortcut, in some versions of Windows it may be possible to copy / paste the text above directly into a command window.*

Now the Publisher ID should be in a designated keystore file.

You may notice that the Publisher ID has an odd alias. If this is the case, you can change this by following these steps:

1. Open the command prompt and locate the keytool.exe and the keystore-file.
2. Use the following command to define the current alias (*note: you will need the keystore password that you noted down earlier*):

```
keytool -list -keystore Name_Of_The_Key_store.jks
```

3. In the case where the keystore has only one item in it, the outcome from the keytool command will display the current alias in the first item of the output using brackets to distinguish the alias name.
4. The alias can then be changed with the following command:

```
keytool -changealias -alias "Enter_The_Alias" -destalias "New_Alias" -keystore Name_Of_The_Key_store.jks
```

### **Option 3: Using the keystore file with Wireless Tool Kit**

1. Start Wireless Tool Kit and run the utility called "Sign MIDlet Suite"
2. Under the "File" menu of "Sign MIDlet Suite" select "Load keystore" and then "from file". This can be used to bring the Publisher ID to the use of the "Sign MIDlet Suite". (*Note: The keystore password is required when loading the keystore.*)
3. You will be given an option to select the security domain. This is not important (as it is not possible to sign applications with Publisher ID and install them to a device as the devices do not have the corresponding root certificate), so just keep things simple and select "minimum".

Now the Publisher ID is available for signing the applications.

### **Option 4: Using the keystore file with NetBeans**

1. Launch NetBeans. In the "properties" window, select the "Signing" category.
2. Use the keystore manager to add a new keystore from a file (common file formats to import are ".jks, .ks, .keystore, .p12 and .pkcs12")
3. After the keystore is imported it must be unlocked. When this is done check the "Sign Distribution" box, keystore and alias in the properties -> signing menu.

By selecting "Sign Distribution", the application will be signed at build time using the Publisher ID.

Any questions? Contact us by completing the 'talk to us' form at [www.javaverified.com](http://www.javaverified.com)