# EDT Client Setup Manual

This manual describes the required configuration steps in order to enable MCEDT client in your local development environment. The intended audience includes developers and system administrators.

## Prerequisites

1. You will need a tool for working with key stores. Keytool that comes with Java will work, but it’s recommended to get a more powerful tool. Try out KeyStore Explorer:

<http://keystore-explorer.sourceforge.net/>

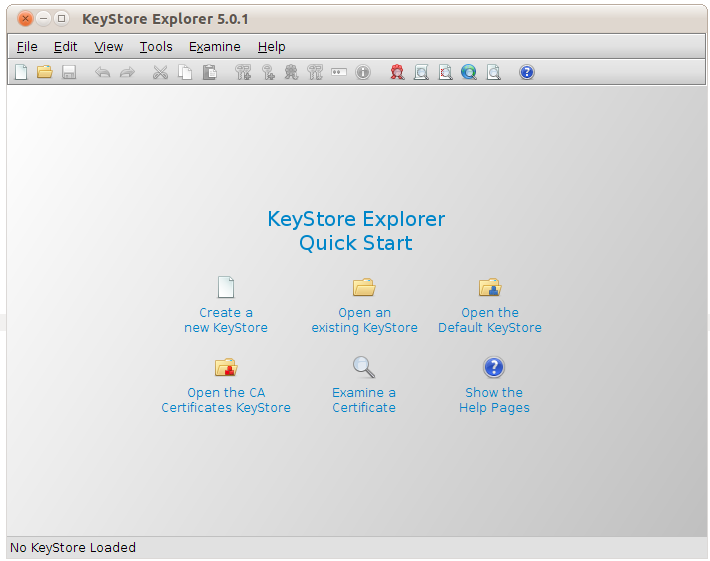
After installing this tool, you will be prompted to install jurisdiction policy. For that, navigate to Oracle’s web site, download the policy zip, and use installer to point to that zip, and click upgrade button.

1. You will require up-to-date MCEDT login credentials. You should be provided a set of credentials by the MCEDT conformance testing team.

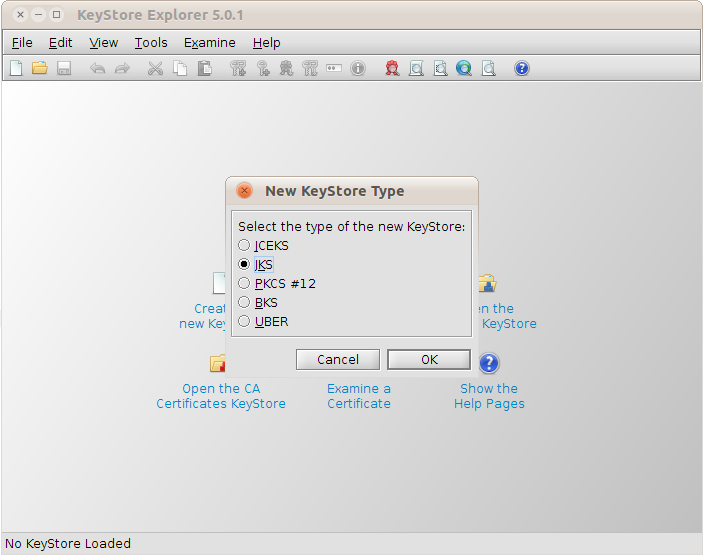
Once you have two items above, you should be ready to start working with the MCEDT service. In order to make a successful call you will need to carry out two setup steps: generate a keystore containing a private-public key pair and import MCEDT public certificate there. Once it’s done, the project must be configured to know where to locate the required files.

## Creating key store

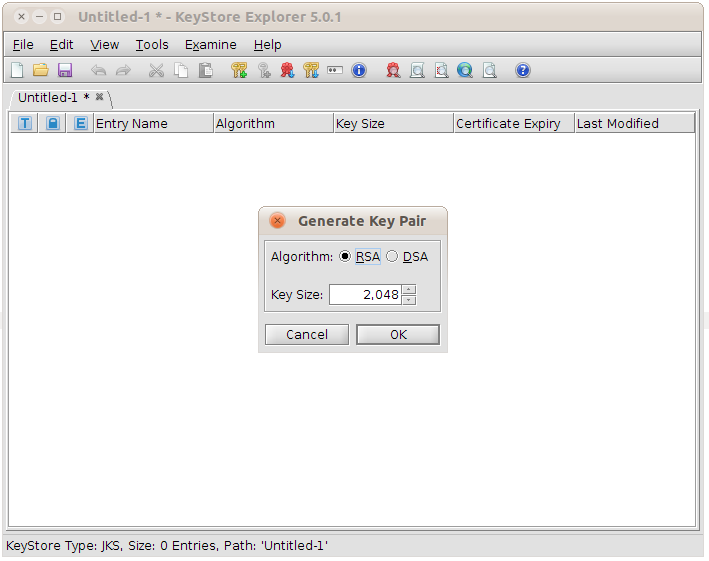
1. Launch key store explorer



1. Create new keystore by clicking File > New > JKS and then Ok



1. Create a new key pair by clicking Tools > Generate Key Pair, then pick RSA algorithm and 2kb key size (2048)



1. You will be prompted to enter key pair certificate information. You can leave the default settings for

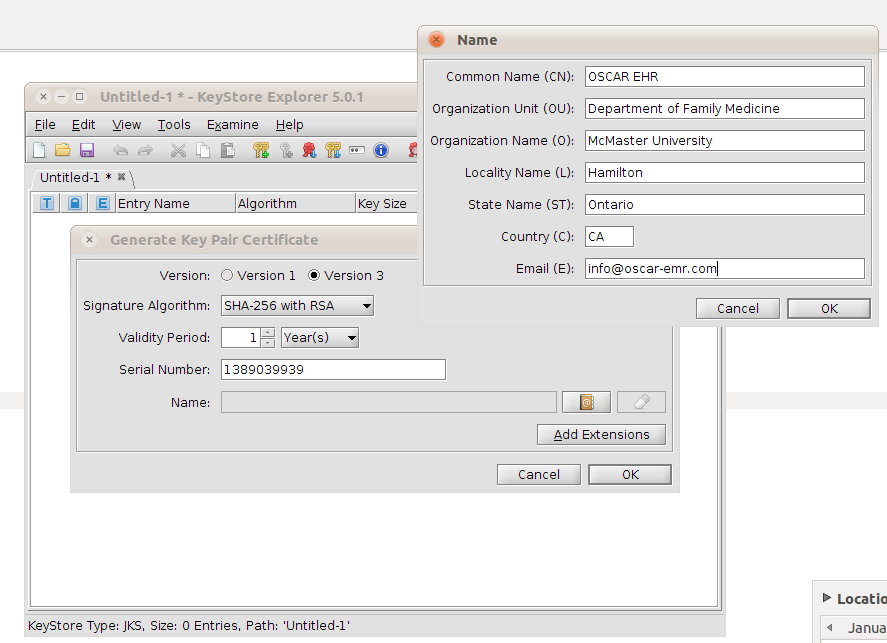
Version: Version 3

Signature Algorithm: SHA-256 with RSA

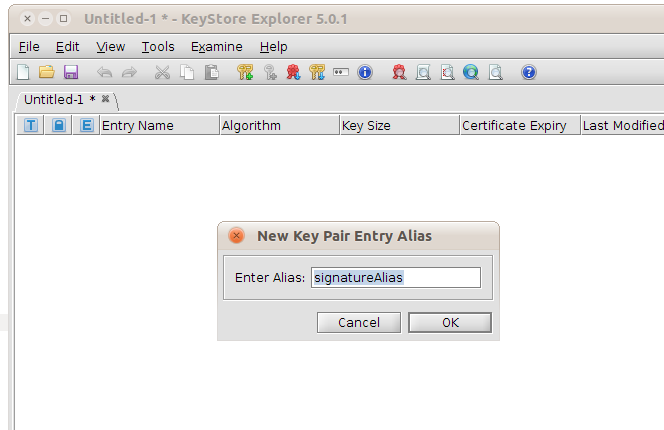
Validity: 1 year

Serial No: Leave what’s there by default

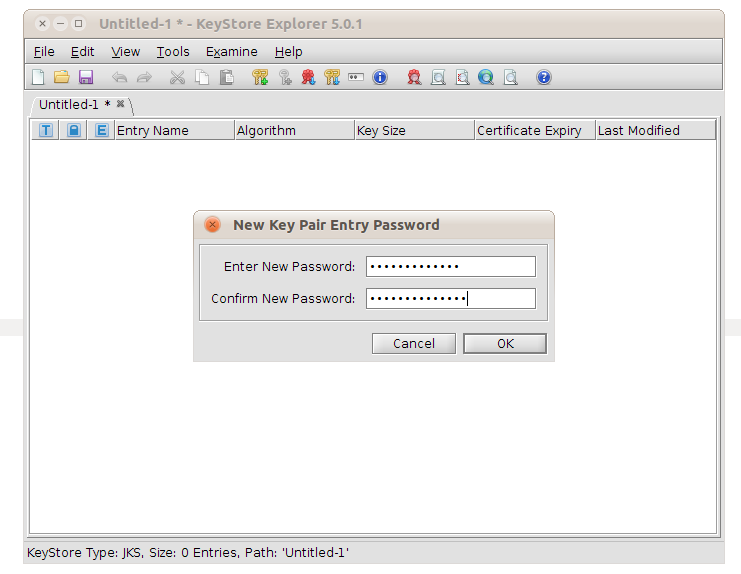
1. Click the address book button and enter the name information. You can enter your information. This key will be used for signing the outgoing requests. Optionally, if you have an existing certificate, feel free to enter it into the keystore



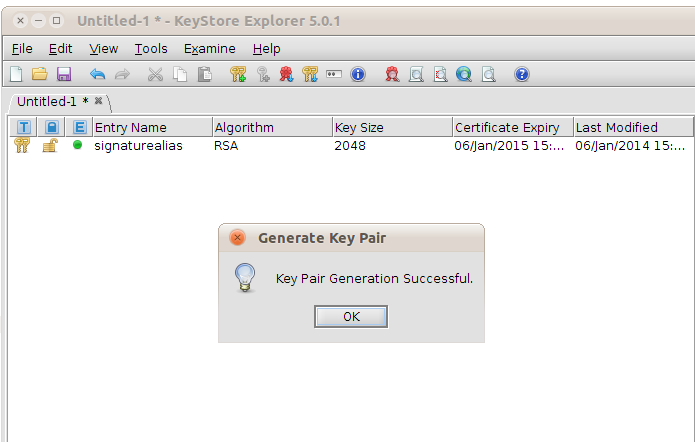
1. Now click OK button twice. It will display “New Key Pair Entry Alias” screen. Specify a meaningful alias and click OK. I will use “signatureAlias” here:



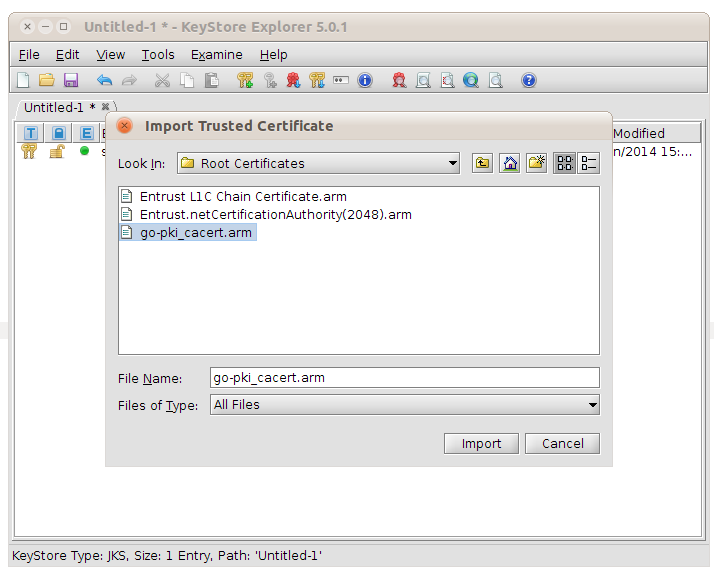
1. Now enter the password. I will use “aliasPassword”



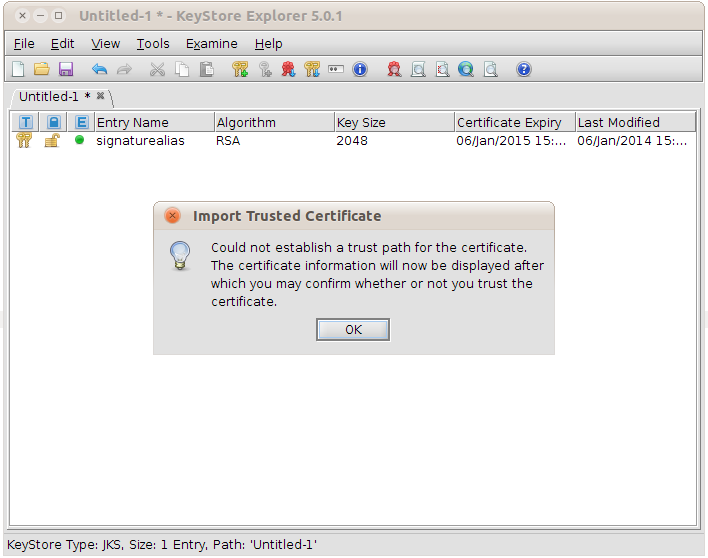
1. This completes the key pair generation:



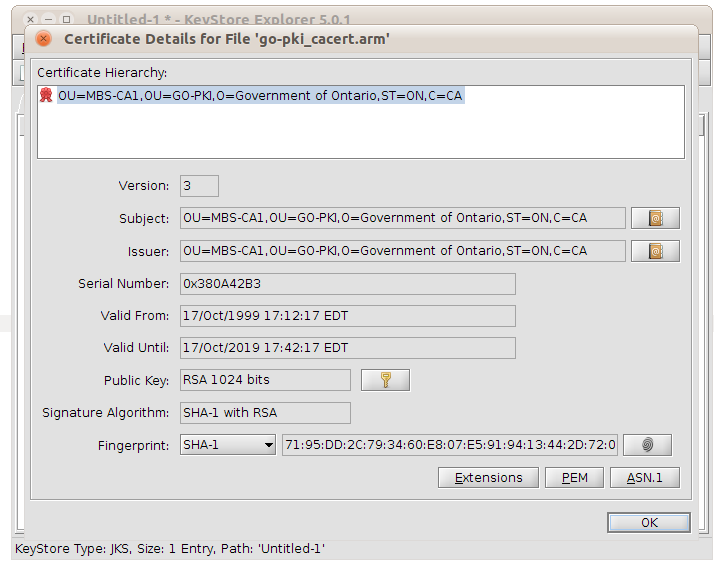
1. Now import the MCEDT certificate by accessing Tools > Import Trusted Certificate. Then locate go-pki\_cacert.arm file within MCEDT source files (edt-stubs/src/main/resources/from\_ohip\_web\_site) and click Import



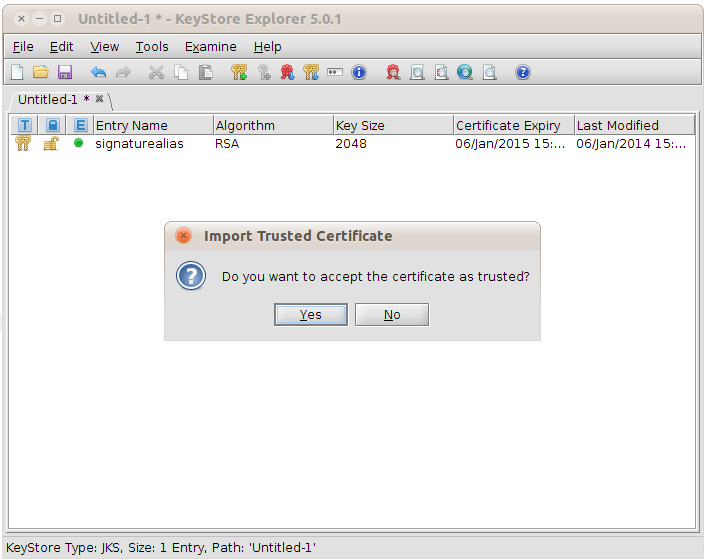
1. KeyStore Explorer will display an info screen. Click “OK”



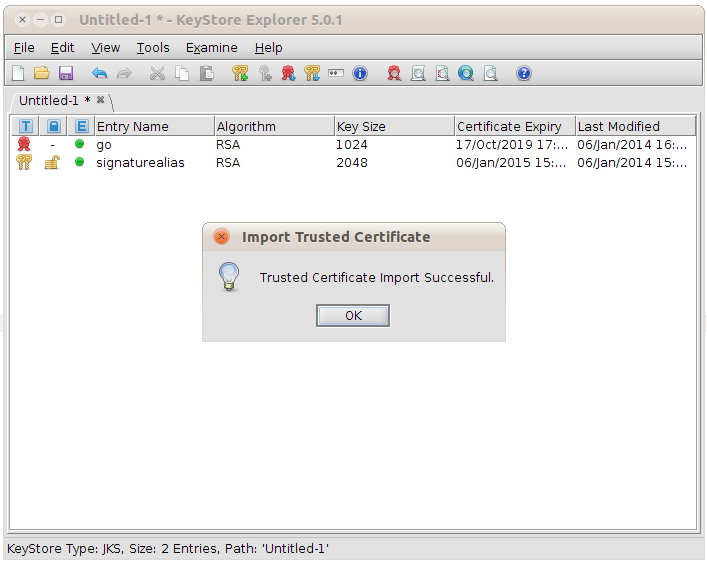
1. And OK again to confirm the certificate details



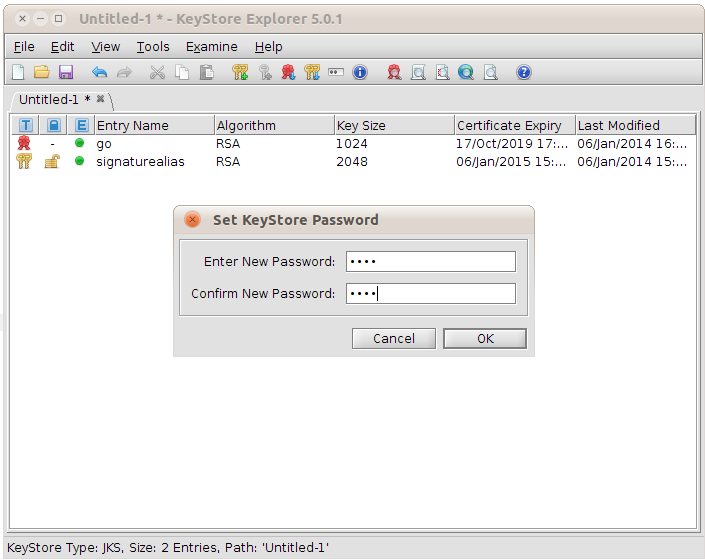
1. Click “Yes” to accept the certificate as trusted



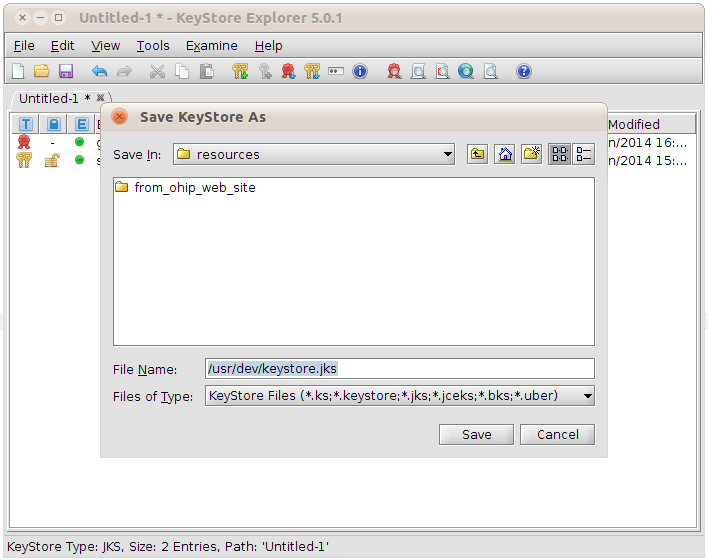
1. Enter an alias (any alias would do):



1. This completes the import step
2. Now save the key store by clicking File > Save
3. It will prompt you for the keystore password. Enter “pass” there



1. Click OK and choose a keystore path



1. This completes the keystore setup process

## Project configuration

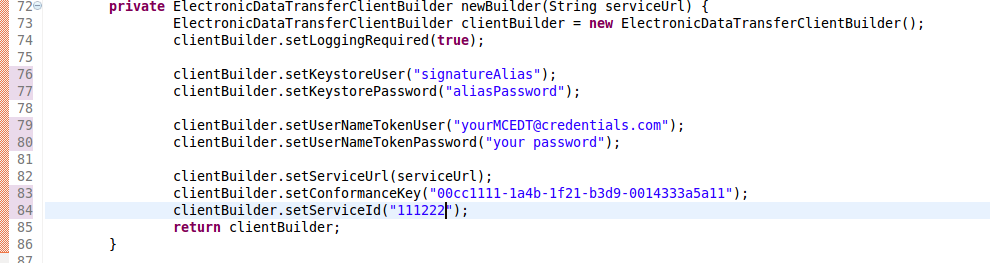
After the key store has been setup, it’s necessary to configure the client. Please open clientKeystore.properties and update the properties below:

org.apache.ws.security.crypto.merlin.keystore.password=pass

org.apache.ws.security.crypto.merlin.keystore.alias=signatureAlias

org.apache.ws.security.crypto.merlin.keystore.file=/usr/dev/keystore.jks

Now open the ElectronicDataTransferClientTest and navigate to the newBuilder method. Modify the source code that specifies the keystore user and password, MCEDT credentials (token info, conformance key and service ID) appropriately:



And run the unit test. It should complete successfully, producing valid download output:

