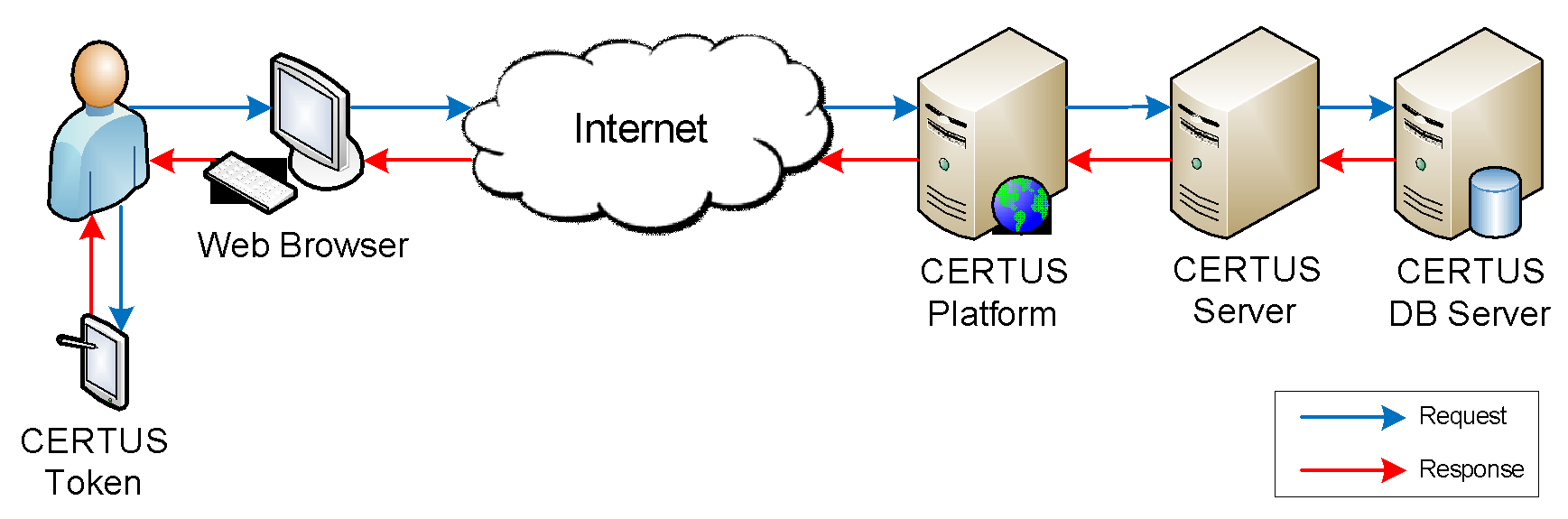
Installation Instructions

CERTUS Voting System



This document consists three parts:

1. Setting up Operating Environment.

2. Setting up Development Environment.

3. Deploying Instructions.

# Part 1 – Setting up Operating Environment:

Instruction on setting up servers in the operating environment is described in this section. Following sections describe steps to be followed in setting up the database server, application server, and the web server.

## Certus Database

Any operating system can be used on this machine, as long as it supports MySQL server. For the purposes of this project, we hosted the CERTUS database on the same machine as CERTUS Server, which runs on Ubuntu Server 12.4:

### Installed Services:

* OpenSSH
* MySQL Server:
  + Version: 5.5.35-0ubuntu0.12.04.2
  + Certus Database Name: \*\*\*\*\*\*
  + Current Accounts:
    - Username: \*\*\*\*\*\*/ password: \*\*\*\*\*\*/ root account.
    - Username: \*\*\*\*\*\*/ password: \*\*\*\*\*\*/ has access on repo6908 database only.

## Certus Application Server

Any operating system can be used on this machine, as long as it supports OpenJDK. For the purposes of this project, we used Ubuntu Server 12.4:

* Domain Name: cosec.seas.gwu.edu
* Public IP: 128.164.159.149
* MAC address: \*\*\*\*\*\*
* Account: \*\*\*\*\*\*/\*\*\*\*\*\*
* The needed services on this machine:
  + OpenSSH
  + OpenJDK Runtime Environment (at the time of this writing we are using IcedTea 2.4.4 64-Bit build 24.45-b08).
* MySQL server.

## Certus Platform (Web Server)

Certus Platform requires operating environment which can support running an instance of Apache Tomcat web server. For the purpose of quick development, we hosted Certus Platform and Certus Server within the same operating environment - Ubuntu Server 12.4:

In order to run the Platform, we have to setup the following prerequisites of the project:

* OpenSSH
* OpenJDK Runtime Environment (at the time or this writing we are using IcedTea 2.4.4 64-Bit build 24.45-b08).
* Setup Apache Tomcat 7.0
* Setup Apache Tomcat to accept https connections and make sure that all unencrypted traffic is forced through https.
* Setup Apache Tomcat user login credentials in <tomcatBasePath>/tomcat-users.xml in order to get access to host-manager GUI. This user must have roles of admin-gui and manager-gui. User can be added using this line: <user username="username" password="\*\*\*\*\*\*" roles="admin-gui,manager-gui"/>
* Using host-manager GUI, we can deploy WAR files and run the platform.

# 

# Part 2 - Setting up Development Environment

This section describes steps involved in setting up the development environment in Eclipse IDE.

## Create Projects in Eclipse

Required Software

Java JDK version: v1.7.0.x

Apache Tomcat: v7.0

MySQL Driver: mysql-connector-java-5.1.23-bin.jar

Email Library: javax.mail.jar

IDE: Eclipse Kepler

There are four projects:

1. CERTUS-Server: contains the Java source code for the application server implementation.
2. CERTUS-Token: contains the Java source code for the hardware token simulator implementation.
3. CERTUS-web : contains the Java source code for the web server implementation.
4. CERTUS-Common: contains the Java source code for the shared classes between application server and web server.

Projects may be created from the source code in local hard drive or from a remote code repository.

### Creating Project from Local Hard Drive

Copy the source code into your local hard drive

Then from Eclipse IDE, select File -> Import.

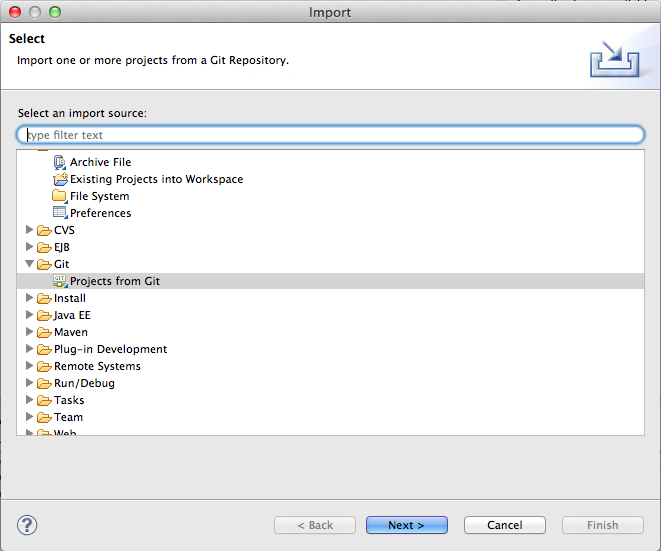
Then select *Existing Project into Workspace* under *General* and then browse the path of source code directory. Follow the same steps for all four projects.

### Creating Project from Remote Code Repository

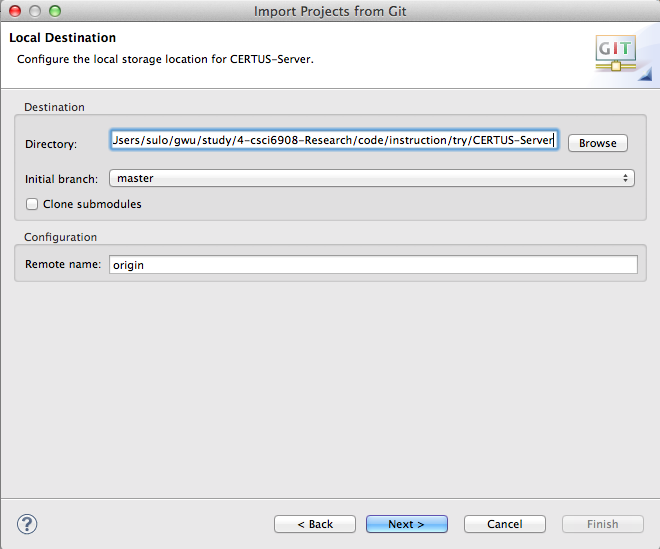
Get the URL for the code repository from the project supervisor.

Select File -> Import

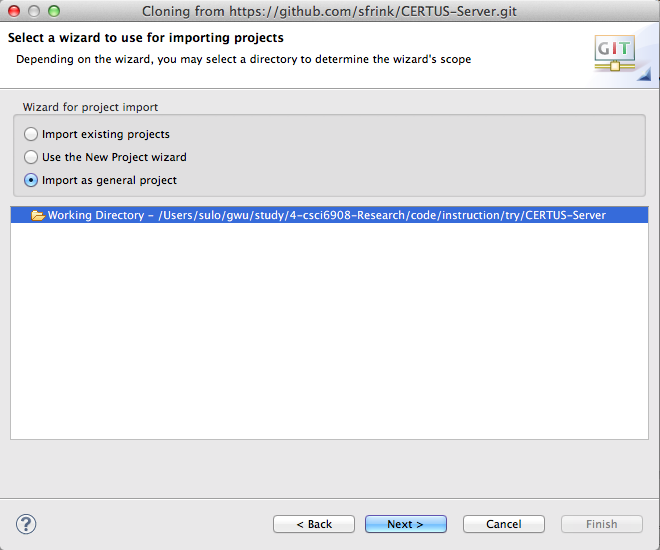
Select the Family of Repository (Instruction provided here is based on GitHub), and then click Next.



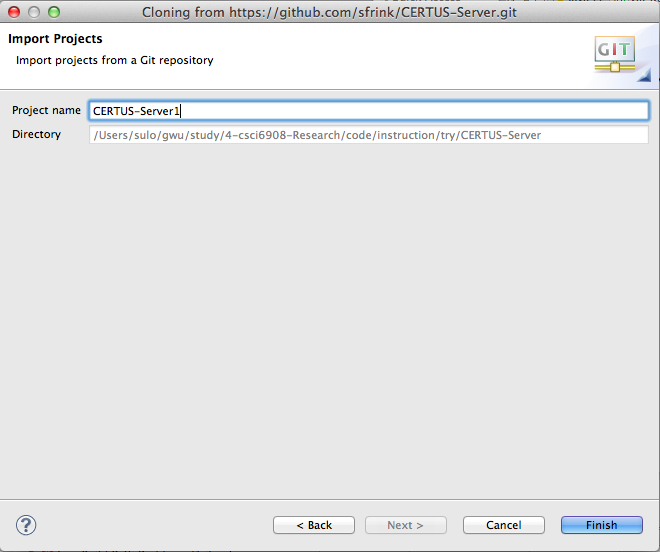
Mention the path to the directory where source should be placed in the local computer, and then click Next.



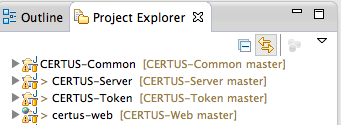
Select the option *Import as general project* and click on Next.



Give a name for the project and then click Finish.



Once all the projects opened, it will appear similar to following:



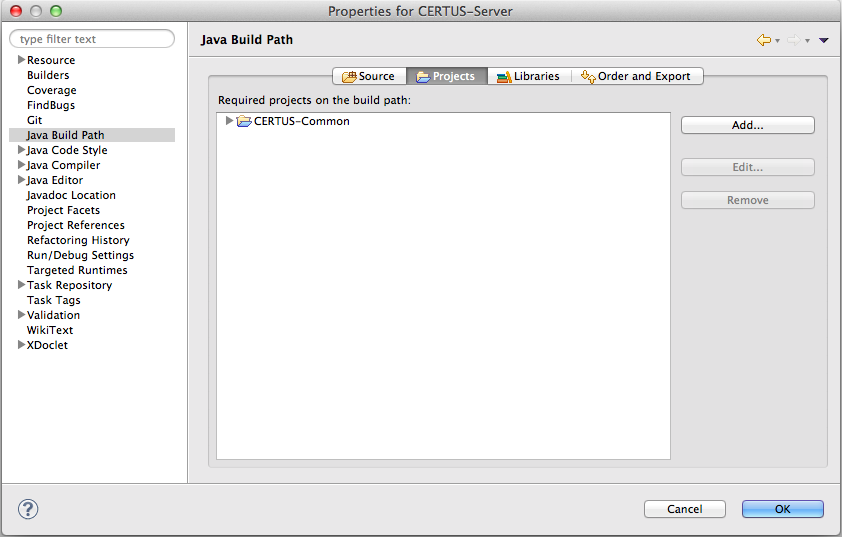
## 

## Setup Project Properties

### 

### Setup Server Project Properties

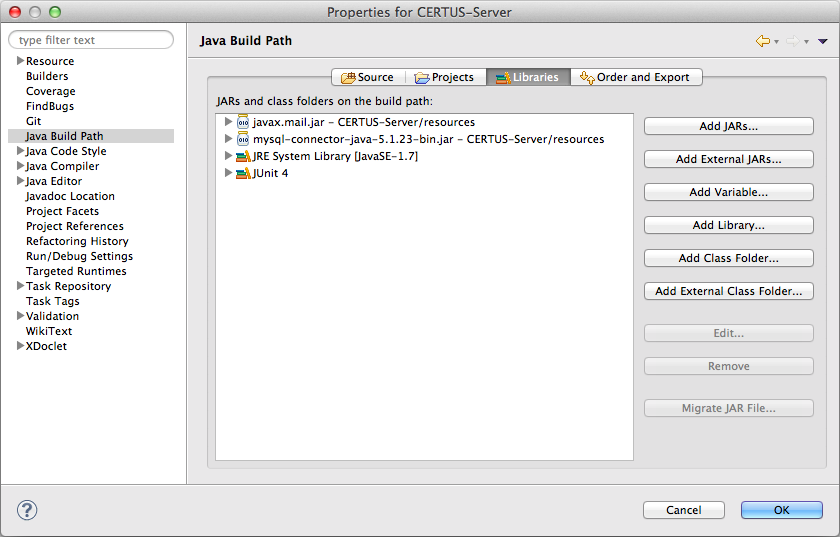
The CERTUS-Server project properties are set to refer to the CERTUS-Common project. Under the *Java Build Path*, the *Project* tab lists required projects on the build path.



Following list of third party libraries are required to compile and run the CERTUS-Server project.

1. myql-connector-java-5.1.23.jar
2. javax.mail.jar

These .jar files are included to the path <path\_to\_the\_server\_source\_code>/CERTUS-Server/resources. The *Libraries* tab of the *Properties* dialog box lists required third party libraries.

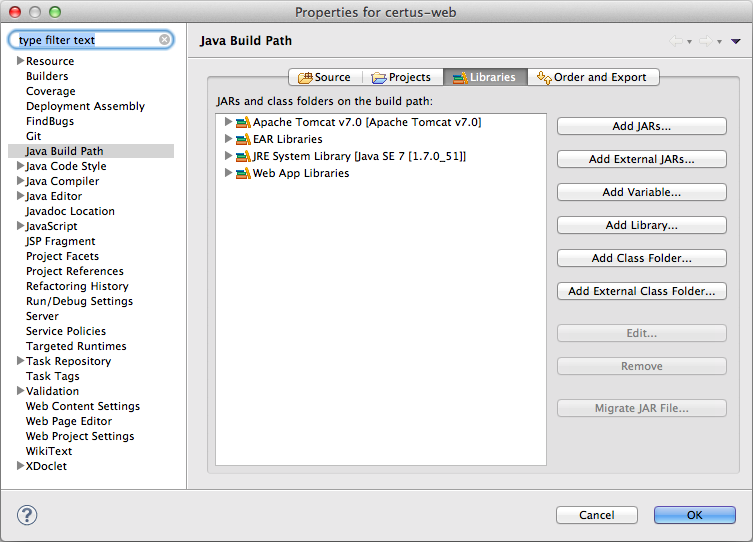


### 

### Setup Web Server Project properties

Similarly, the CERTUS-Web project requires referring to the CERTUS-Common project and requires pointing to the third party libraries.

Further the CERTUS-Web project should be correctly configured with relevant libraries for Apache Tomcat and JRE.



#### 

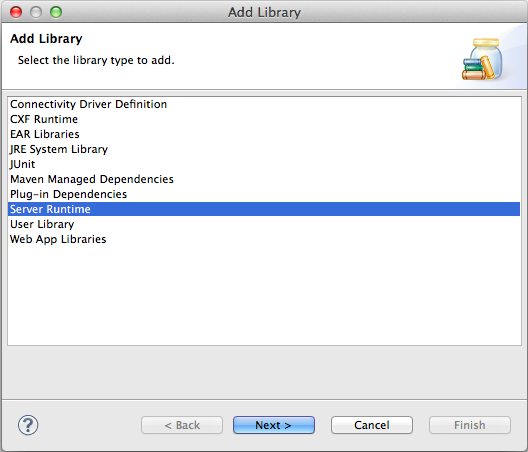
#### Tomcat

Tomcat should be configured to run in the development environment.

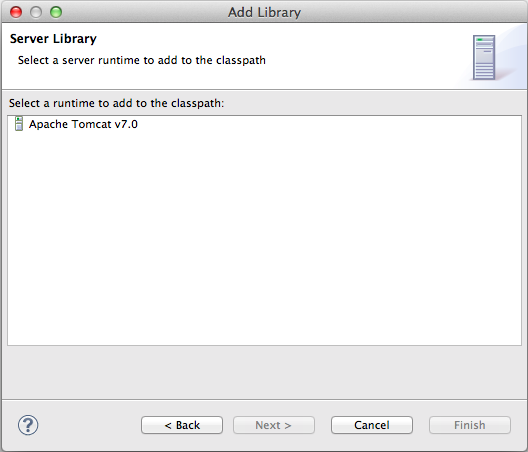
Select the project properties and then select the Java Build Path.

Select the Libraries tab.

Select the Add Library… button, Add Library dialog box appears.



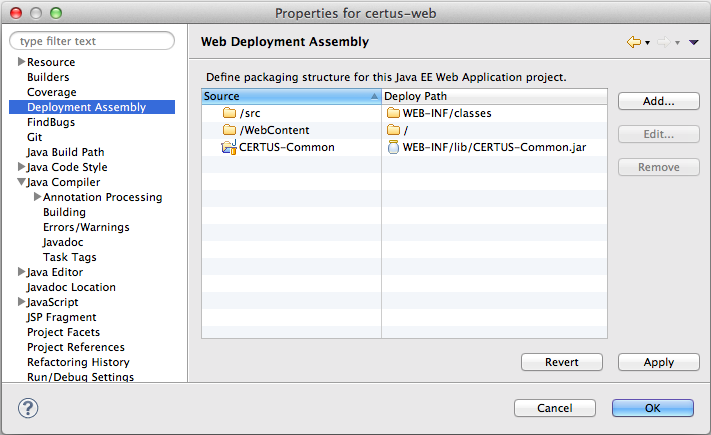
Select Server Runtime and click on Next.



Select the Apache Tomcat v7.x and click on Finish.

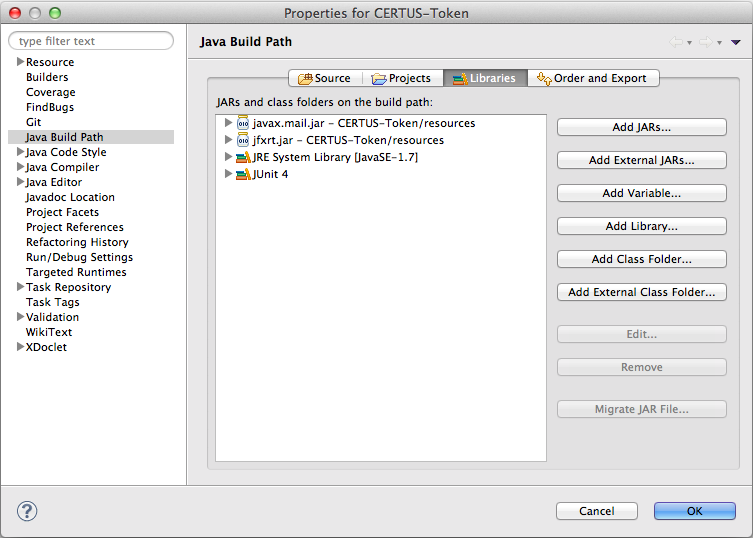
#### Project Deployment Assembly

Project Deployment Assembly should include the CERTUS-Common project in order to deploy the CERTUS-Server with corresponding common classes.



### Setup Token Project Properties

Java FX library is required to compile and run the Token (hardware simulator). The required jar file, jfxrt.jar is located in the directory: <path\_to\_the\_token\_source\_code>/CERTUS-Token/resources.



## 

## How to run the Server from Eclipse

Follow these steps to run both the application server and the web server on the development environment which is Eclipse.

### Step 1 : Run the Application Server

Start the Database server

Update the configuration.properties file with correct parameters.

Select the server file: <path\_to\_server\_source\_code>/CERTUS-Server/src/rmi/CertusServer.java

Select Run -> Run As -> Java Application

### Step 2 : Run the Web Server

Ensure the application server is running on specified IP/port as in the configuration.properties file.

Select the project from the Project Explorer

Select Run -> Run As -> Run on Server menu options

### Step 3 : Access the Web Server

When the web server started successfully, following url can be used to access the web site over the loca machine.

url :<http://localhost:8080/certus-web/login>

### Step4 : Run the Token

Token is required to sign the encrypted vote.

To access the Token, select the Main.java file at

<path\_to\_token\_source\_code>/CERTUS-Token/src/application/Main.java

Select Run -> Run As -> Java Application.

# Part 3 - Deploying Instructions

## 

## How to deploy Certus Server:

Step 1: Install Eclipse Kelper.

Step 2: Install JUnit.

Step 3: Import Certus-Server project into Eclipse.

Step 4: Make sure to check the following dependencies:

- Deployment assembly: Certus-Common Project.

- Library: mysql-connector-java-5.1.23 - used to establish connections to MySQL database.

- Library: javamail-1.4.7 - used for sending emails.

Step 5: Double check parameters in config.properties file located at the project root. Distribution of Certus Server comes with all required files for RMI which are located in <projectRoot>/resources/rmi-files/

Step 6: Now we can run directly from Eclipse by running rmi.CertusServer as Java Application.

Alternatively, we can export Certus-Server into JAR file and deploy as follows:

Step 7: Export project as runnable JAR file with main class: rmi.CertusServer

Step 8: run JAR file as $java -jar <filename>.jar

## How to deploy Certus Platform:

Step 1: Install Eclipse Kepler.

Step 2: Setup Apache Tomcat plugin for Eclipse and add at least one instance of Apache Tomcat Server.

Step 3: Import Certus-Web project into Eclipse.

Step 4: Make sure to check the following dependencies:

- Library: Apache Tomcat

- Deployment assembly: Certus-Common Project

Step 5: Double check parameters in config.properties file located at the project root. Certus-Platform distribution comes prepackaged with all files required for RMI connections and are located in <projectRoot>/rmi-files/

Step 6: Run Certus Server.

Step 7: Now we can run the project directly from Eclipse by manually running any of the servlets.

Also, we can export this project as .WAR file and deploy it to remote Apache Tomcat using standard host-manager GUI console (commonly found at <hostname>/host-manager).

## How to deploy token

Step 1: Install Eclipse Kelper.

Step 2: Install Java FX SDK ([Click Here](http://www.oracle.com/technetwork/java/javafx/install-javafx-sdk-1-2-139156.html))

Step 3: Install JUnit.

Step 4: Install e(fx)clipse for Eclipse ([Click Here](http://www.eclipse.org/efxclipse/install.html))

Step 5: Import the Certus-Token project into Eclipse, and run application.Main as Java Application

We can also export this project into JAR file and run it on any computers that support Java.

Since Certus-Token uses JavaFX libraries, it is important to export this project using command line tool provided by JavaFX called javafxpackager. This tool can commonly be found in any standard JDK in bin directory. Command usage looks as follows:

<JdkBasePath>/bin/javafxpackager -createjar -appclass application.Main -srcdir <TokenBasePath>/bin -outdir . -outfile token -v