ReadMe – MasterPassJava

This document outlines the steps needed to configure and run the MasterPass test application on a Windows 7 development computer

# Prerequisites The following items must be installed prior to running the MasterPass test application

# Eclipse Indigo JEE or greater (latest is preferred)

1. Java 1.7 Enterprise Edition (or above)
2. Tomcat v7.0

# Using MasterPass Test App

The MasterPass Web project can be used to test every step in the MasterPass API workflow. Use the following steps to configure Eclipse and your computer HOST file to create a local web application.

## Configuring the HOST File

In order for the MasterPass test application to receive a callback from the MasterPass Wallet, you must add an entry your HOST file to tell DNS how the call should be routed on your local computer.

1. Open the following file in Notepad: C:\Windows\System32\drivers\etc\hosts

*Note: due to Windows 7 security restrictions, it may be necessary to create a copy of the hosts file and place it in C:\temp. After you have completed your edit, paste the hosts file back into the* ***etc*** *directory and overwrite the existing file.*

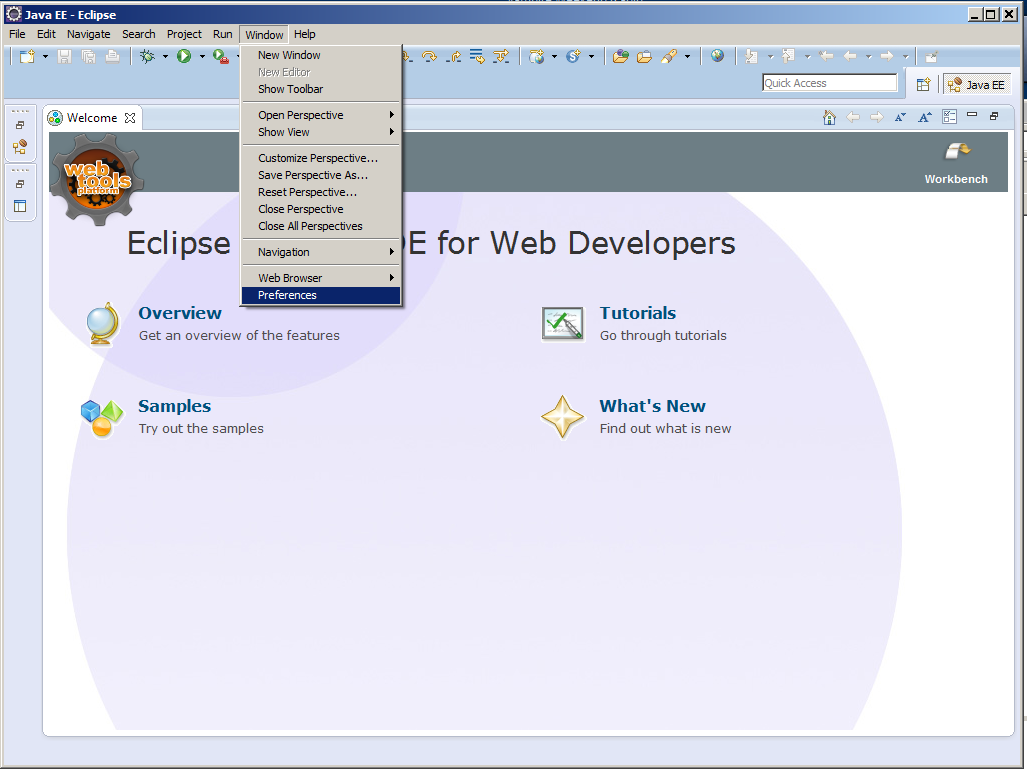
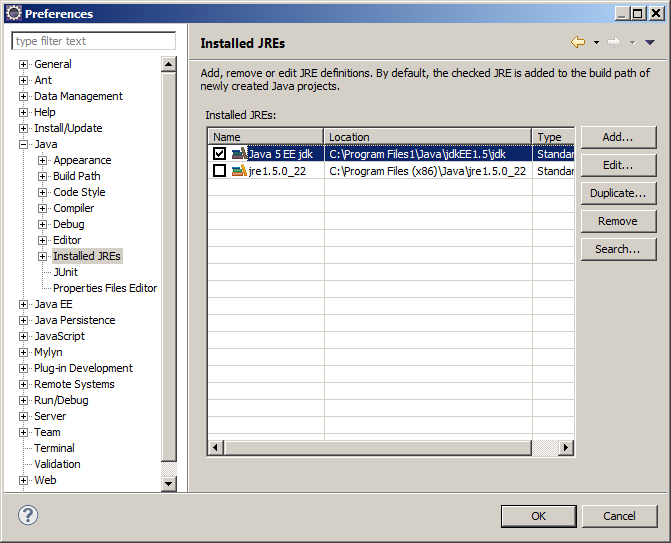
1. Add the following entry at the bottom of the hosts file:

127.0.0.1 projectabc.com

*Note: this domain name must match your MasterPass Online Project’s registered domain name, or the callback will fail.*

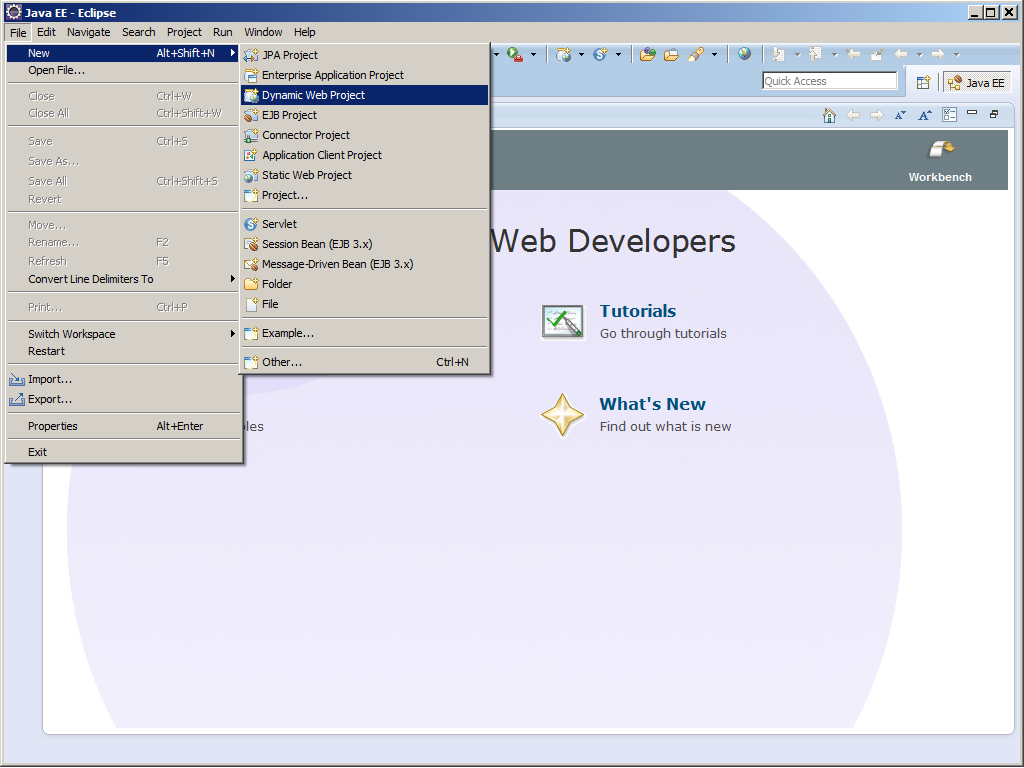
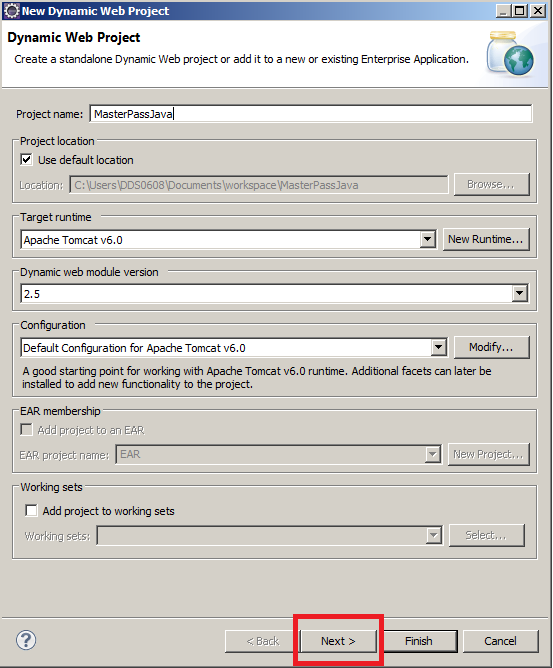
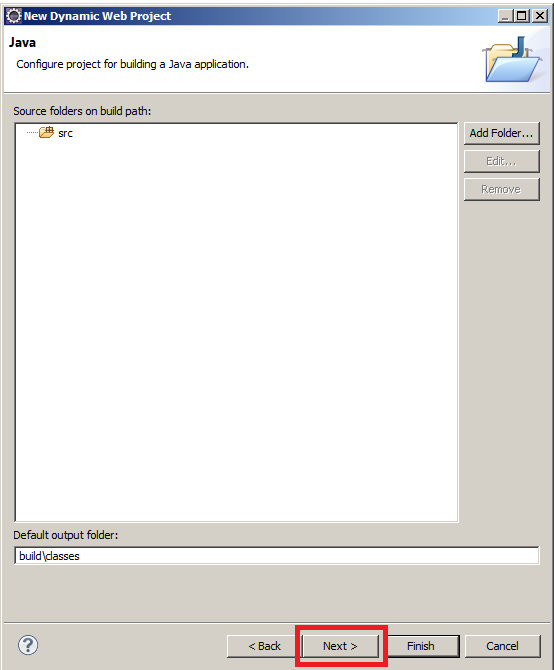
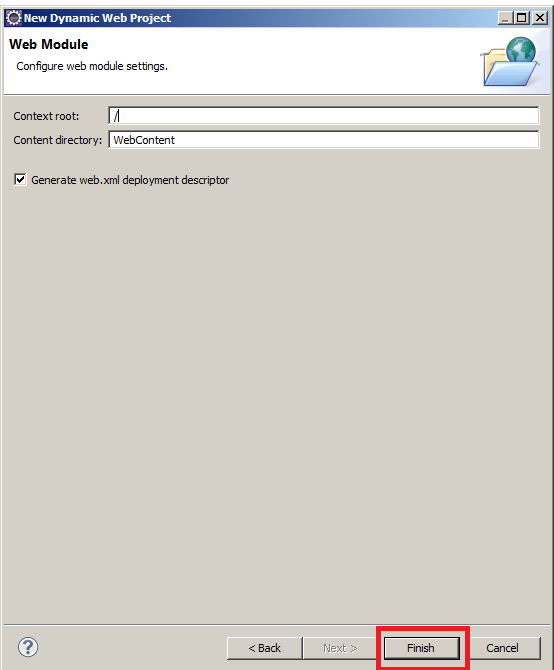
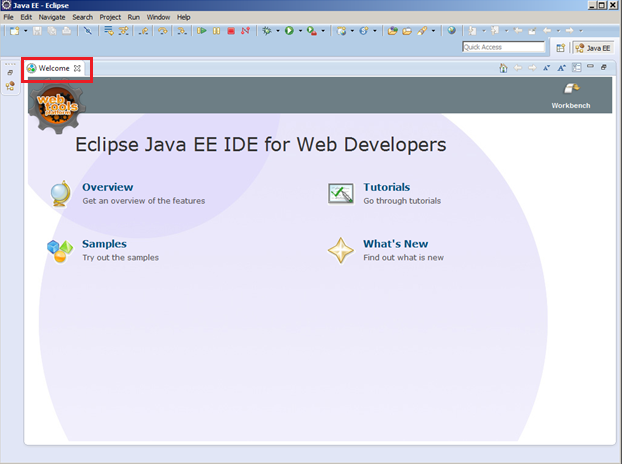
1. Run the MasterPass test application and verify that you are able to receive a callback.

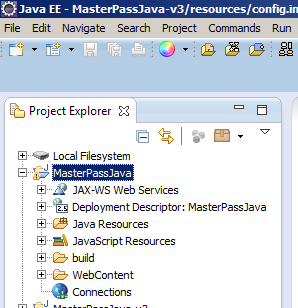
## Configure Eclipse for MasterPass Test Application

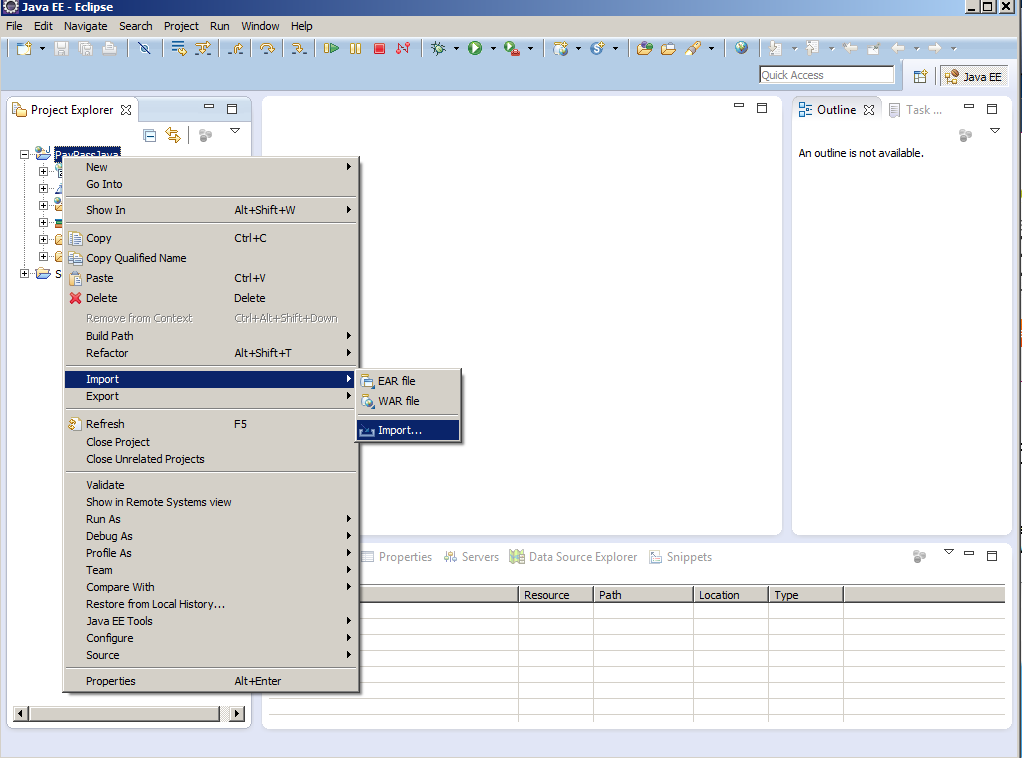
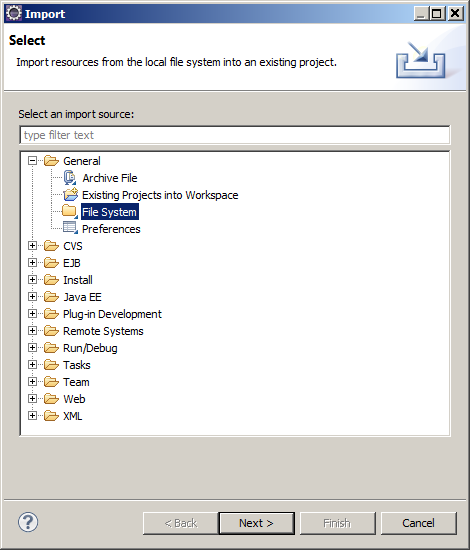
1. Eclipse’s initial screen. Click on Window -> Preferences
2. Select Installed JRE’s. If the Java 1.7 EE JDK (or higher) is not selected, either select or add as needed. (Note: Images are for reference only) 

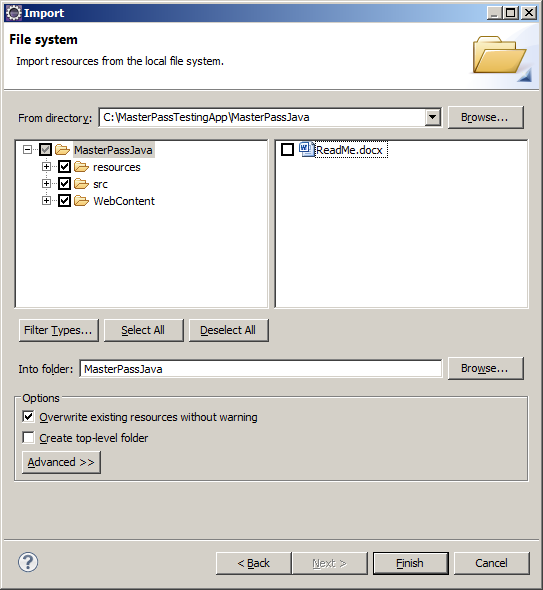
## Importing the MasterPass Test project into Eclipse

Once Eclipse is configured with the appropriate Java environment, it is time to import the MasterPass test project.

1. Create a new project by selecting File -> New -> Dynamic Web Project
2. Enter the Project Name–> MasterPassJava”   
   Ensure that you have the appropriate Tomcat server selected and then click Next 
3. Click Next  
   
4. Replace the Context root with “/” as shown in the screenshot below. Click Finish  
   
5. Close the Welcome window
6. Developer GUI should now appear with the MasterPass project.



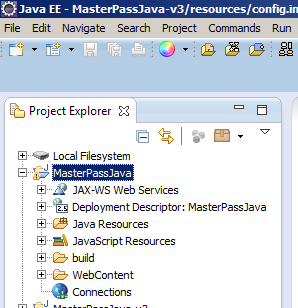
1. Now import the source code from the MasterPass test application into the newly created project. Right click the MasterPass test project, choose Import -> Import…
2. Expand General and choose File System, click Next
3. Click “Browse…” and browse to “C:\Program Files1\MasterPass” (this is the directory that the MasterPass code was unzipped to in a previous step). Check the “Overwrite existing resources without warning”. Check MasterPass checkbox and click Finish.  
   This will complete the import of the source soce into to the project. Next step is adding the dependencies for the project.

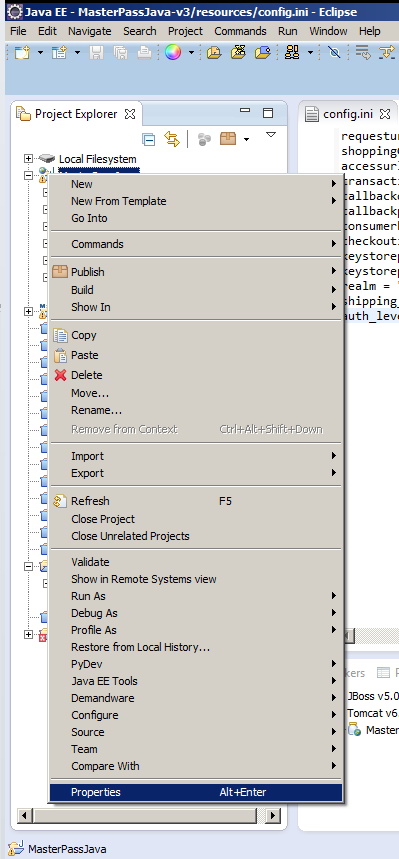
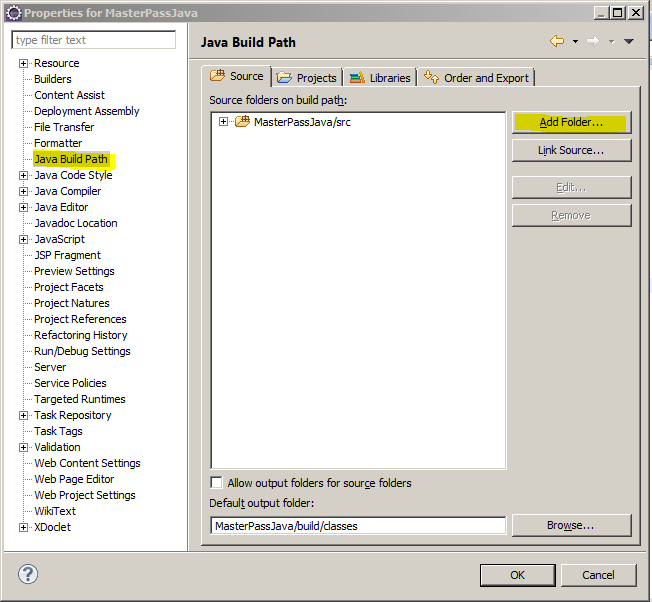
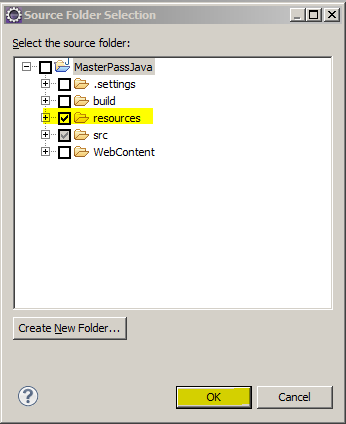
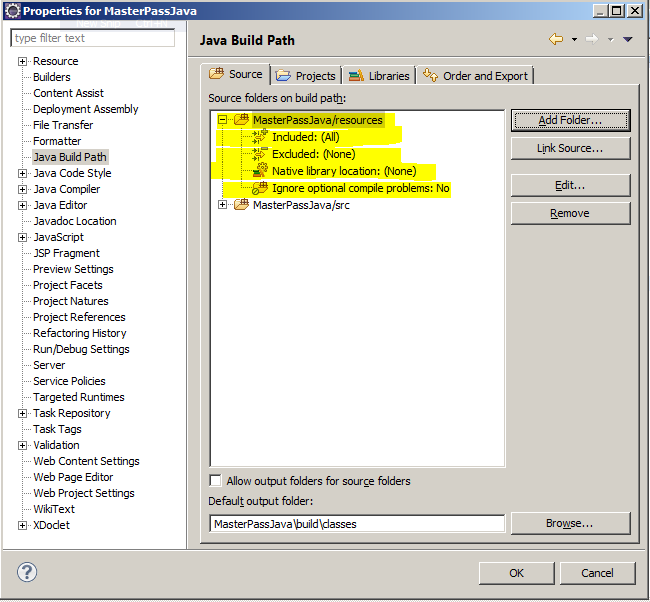
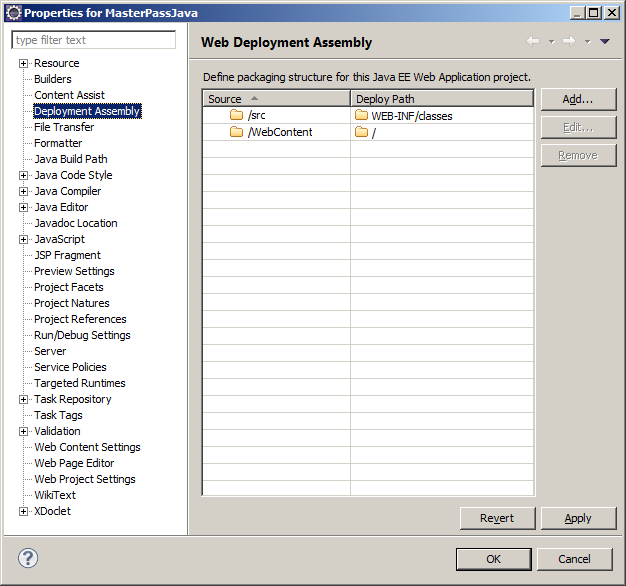
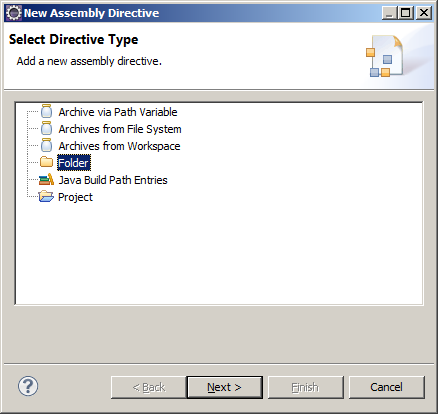
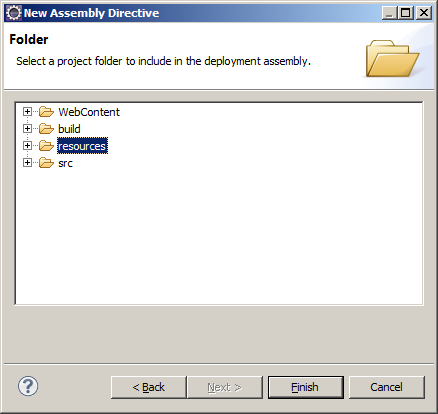


## Adding the project dependencies

This project depends on the Google libraries for its Oauth service calls.

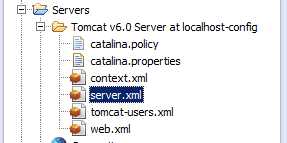
1. Right click on MasterPassJava project

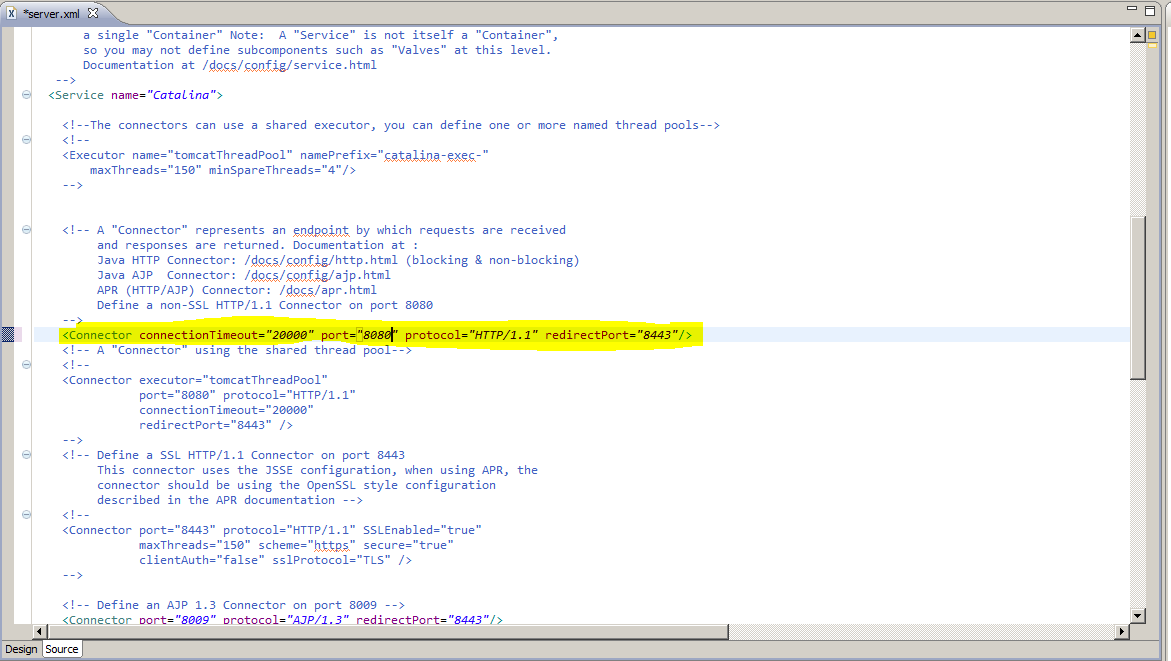


1. Choose Properties  
   
2. Click Java Build Path, and then click Add Folder…  
   
3. Select resources and OK.  
   
4. The Java Build Path should now have the MasterPass/resources folder added  
   
5. Next, click the Deployment Assembly, and then Add… 
6. Click “Folder”, then Next 
7. Select resources folder and then click Finish  
   

## Configure Tomcat to handle requests at port 80

Tomcat must handle requests at port 80 for the merchant callback step.

1. Open Servers, and Tomcat v6.0 Server at localhost-config, then open the server.xml file  
   
2. Change the port number to be 80 instead of 8080. Save and close the file

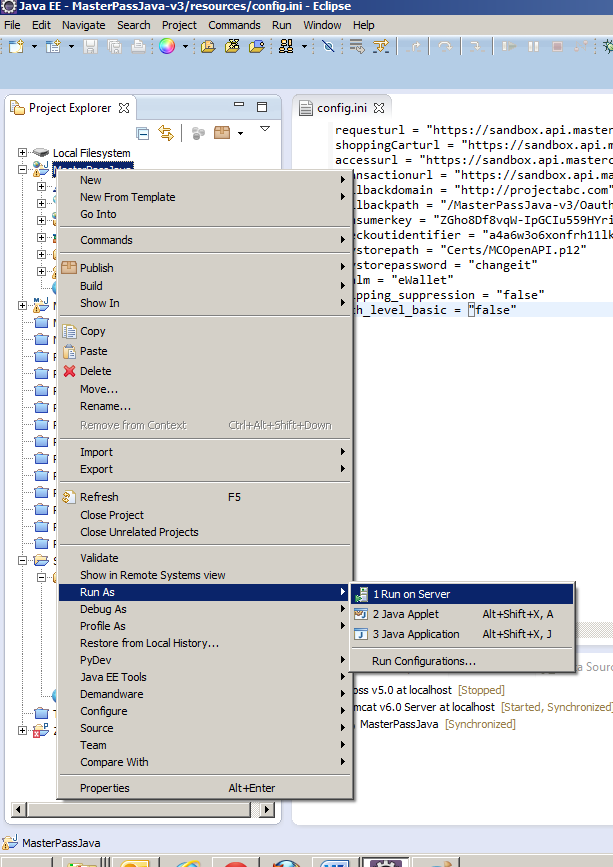
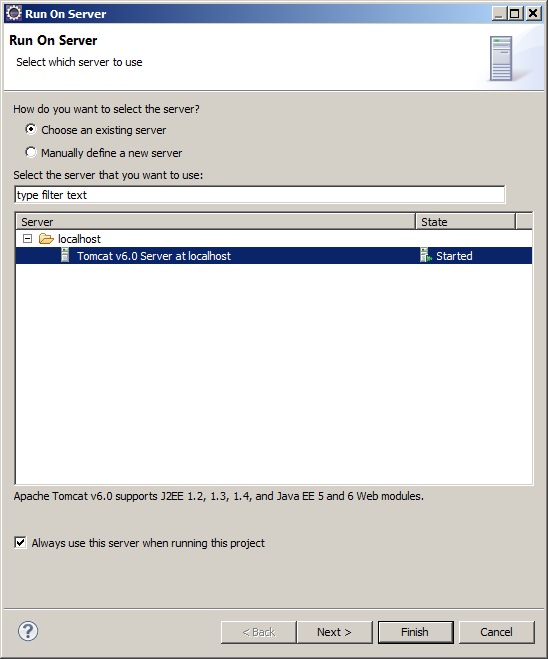
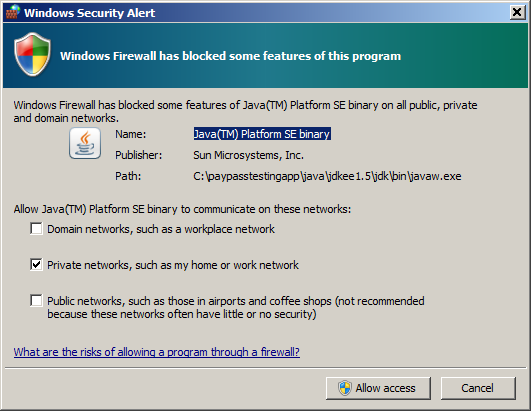


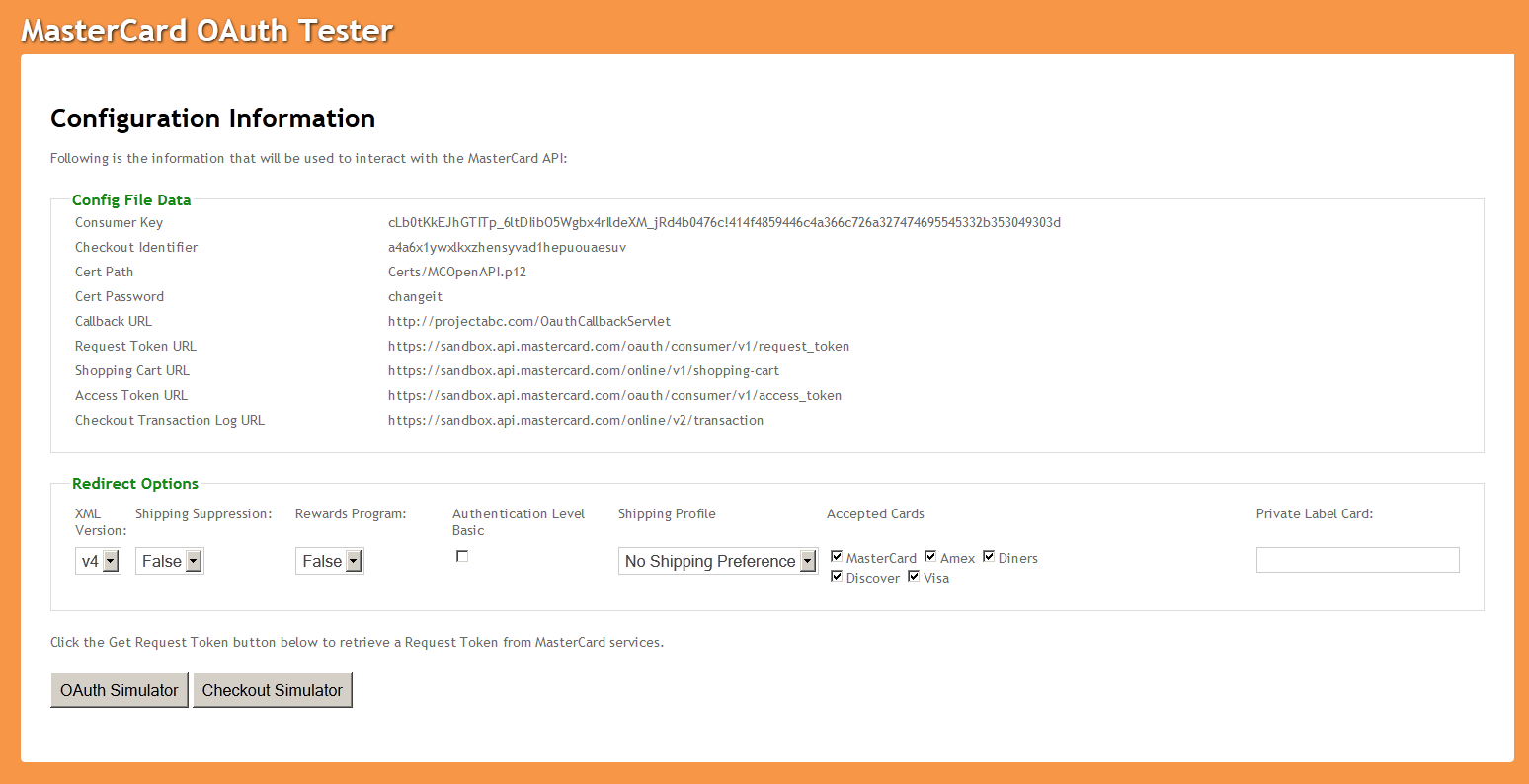
Correct line:

<Connector connectionTimeout=*"20000"* port=*"80"* protocol=*"HTTP/1.1"* redirectPort=*"8443"*/>

## Starting and running the MasterPass Test project.

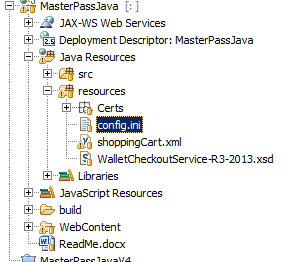
After the source code and dependencies are configured, it is time to start and run the project

1. Right click on MasterPassJava, chose Run As, and pick Run on Server 
2. Click “Always use this server when running this project” and click Finish
3. Allow access to the process 
4. Open a browser and go to address “http://localhost”

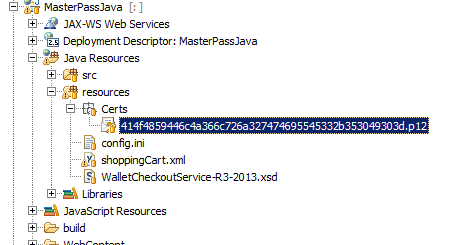


## Change the credentials

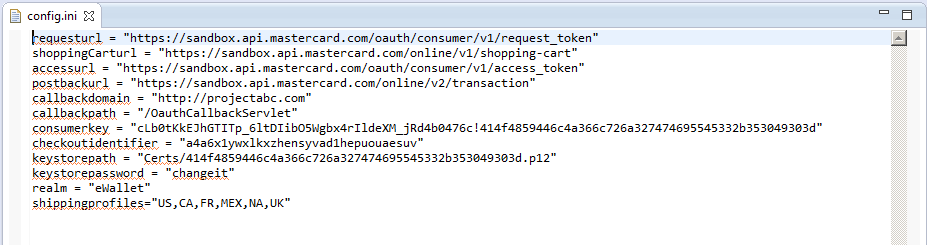
1. Open the config.ini file in the resources folder.



1. Copy your keystore file into the resources folder in Eclipse.



1. Change the callbackdomain, consumerkey, checkoutidentifier, keystorepassword, keystorepath(if different) and shippingProfiles(if used) to your credentials.



Sample:

requesturl = "https://sandbox.api.mastercard.com/oauth/consumer/v1/request\_token"

shoppingCarturl = "https://sandbox.api.mastercard.com/online/v1/shopping-cart"

accessurl = "https://sandbox.api.mastercard.com/oauth/consumer/v1/access\_token"

transactionurl = "https://sandbox.api.mastercard.com/online/v1/transaction"

callbackdomain = "http://projectabc.com"

consumerkey = "cLb0tKkEJhGTITp\_6ltDIibO5Wgbx4rIldeXM\_jRd4b0476c!414f4859446c4a366c726a327474695545332b353049303d"

checkoutidentifier = "a4a6x1ywxlkxzhensyvad1hepuouaesuv"

keystorepath = "resources/Certs/414f4859446c4a366c726a327474695545332b353049303d.p12"

keystorepassword = "changeit"

shippingprofiles="US,CA,FR,MEX,NA,UK"

1. Save your changes to the config.ini file and deploy the changes. Your credentials will now be used to connect to MasterPass web services.
2. Update your Host file to direct the callbackDomain to the localhost.
3. Run the project and see how MasterPass Wallet works!