ReadMe – MasterPass – C#

The following document outlines the steps needed to configure and run the MasterPassCSharp SDK and Sample Application on a Windows 7 development computer.

# Prerequisites

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

1. Microsoft Visual Studio 2010 Professional SP1
2. [ASP.NET MVC3](http://www.asp.net/mvc/mvc3)

# MasterPass Test App Solution & Settings

The MasterPassDotNet-CS solution contains four projects:

* Common – Contains classes with common features that are used by both the MasterPass SDKs and the OpenAPI. These present methods for making connections to the MasterPass services.
* WalletSDK – A library wrapping MasterCard’s sample code plus other utilities needed to successfully integration with the Wallet API.
* WalletWebContent – A C# MVC 3 web application used to demo/test the SDK library and the Wallet API.
* MasterPassCSharp\_Test – Unit tests to aid in the development of the MasterPassService and SampleApplicationController classes.

Perform the following steps to configure the MasterPassDotNet-CS solution and settings:

1. Launch Visual Studio 2010
2. From within Visual Studio, open the MasterPassCSharp solution file (MasterPassCSharp.sln).

# Important Classes

Please reference the following files for integration code samples:

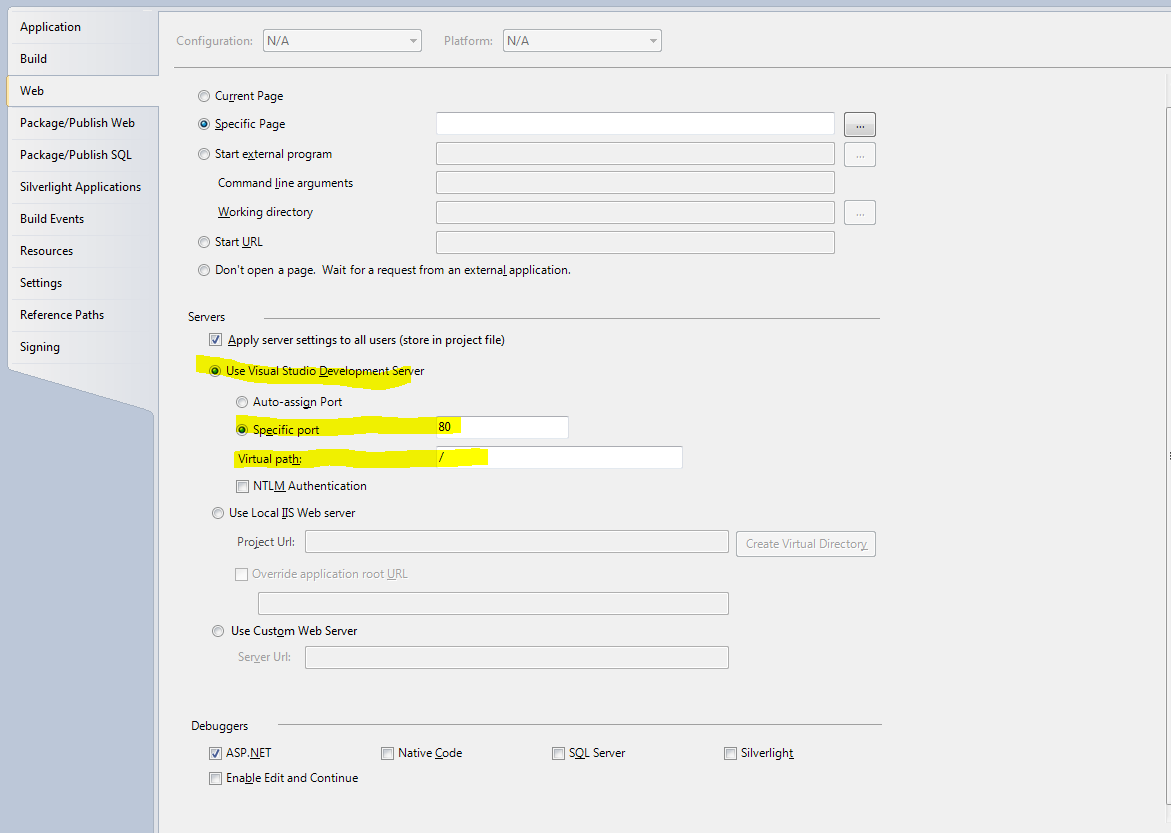
* common\**Connector.cs** – The base methods that interact with the...
* WalletSDK\**MasterPassService.cs** – This extends the Connector class, and defines the main methods that are exposed by the SDK.
* WalletWebContent\Controllers\**SampleApplicationController.cs** – The methods used to present the Wallet API workflow in web pages.

# Using MasterPass Sample Application

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

## Setup & Configuration

1. Now in Visual Studio and the MasterPassCSharp solution, right-click on the WalletWebContent project and select properties.
2. In the properties window, select the Web tab, and ensure that your screen is configured as follows:



You should now be able to run/debug your WalletWebContent application from within Visual Studio and the MasterPassCSharp solution.

## Configuring the HOST File

In order for the MasterPass test application to receive a callback from the Wallet site, 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 YourProjectDomain.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.

## Change the credentials

1. After the solution has loaded, expand the WalletWebContent project and open the Web.config file. You may need to change some of the following appSettings to use the MasterPass Merchant & Project settings provided by the MasterPass Online registration process. The image below is for ILLUSTRATION ONLY:

<!--

Following are the settings used by the Wallet Tester application

which are used to connect/interact with the Wallet API.

-->

<!-- Sandbox Test Project -->

<add key="ConsumerKey" value="cLb0tKkEJhGTITp\_6ltDIibO5Wgbx4rIldeXM\_jRd4b0476c!414f4859446c4a366c726a327474695545332b353049303d" />

<add key="CheckoutIdentifier" value="a4a6x1ywxlkxzhensyvad1hepuouaesuv" />

<add key="CertPassword" value="changeit" />

<add key="CallbackPath" value="/WalletTester/O3\_Callback" />

<add key="CallbackDomain" value="http://projectabc.com" />

<add key="CertPath" value="~/Certs/Sandbox/414f4859446c4a366c726a327474695545332b353049303d.p12

" />

<add key="ApiPrefix" value="sandbox." />

<add key="Auth\_Level\_Basic" value="false" />

*Note: the ApiPrefix setting should be set to an empty string once a project is moved into production.*

1. Copy the MCOpenAPI.p12 keystore file, which you created during your Project Request step, into the appropriate subfolder under WalletWebContent\Certs\. Ensure that your CertPassword setting has been updated with the password that you used during your keystore creation step.
2. Update your Host file to direct the callbackDomain to the localhost.
3. Run the project and see how MasterPass Wallet works!

## Advanced Debugging

Because the Wallet services are secured using HTTPS, it may be difficult to observe your calls (requests/responses) to the MasterCard API. To aid with debugging, you can uncomment the system.diagnostics section provided in the WalletWebContent Web.config file, which will log all calls to the System.Net.trace.log file at the root of the WalletWebCSharp solution.