

## How to setup your {smartassembly} Custom Web Server?

- 1) First, you need to decide the public address of your Web Service. For sample, it can be <http://www.myWebServer.com/{sa}WebServices/>. In this folder, you will copy the 3 "\*.asmx" files (Login.asmx, Reporting.asmx and UploadReportLogin.asmx).
- 2) You also need to copy {smartassembly}.CustomWebServer.dll into the bin directory of your .NET web server.
- 3) You can verify that the first part of the setup is correctly done, by calling the Web Service from your browser: <http://www.myWebServer.com/{sa}WebServices/UploadReportLogin.asmx>  
You must get the standard Web Service Welcome Page:

### UploadReportLoginService

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [GetServerURL](#)
- [GetVersion](#)

- 4) Then, you need to create a folder on the Web Server, which is not publicly available. This will be the **Application Folder** for the Custom Web Server. The Web Service must have write access to the folder and its sub-folders.

In our sample, it will be `c:\MyCustomWebServer\`. You must edit the Server's **web.config** file to inform the Web Server of this location:

```
<configuration>
...
  <appSettings>
    <add key="SmartAssembly.WebServer.ApplicationFolder" value="c:\MyCustomWebServer\" />
  </appSettings>
...
</configuration>
```

- 5) Copy the license file ("`{smartassembly}.WebServer.license.xml`") in the **Application Folder**:  
`c:\MyCustomWebServer\{smartassembly}.WebServer.license.xml`
- 6) Edit the "`{smartassembly}.WebServer.Settings.xml`" file to configure your Custom Web Server:

```
<SmartAssembly.WebServer.Settings>
  <ServerURL>http://www.myWebServer.com/{sa}WebServices/</ServerURL>
  <SecuredURL>https://www.myWebServer.com/{sa}WebServices/</SecuredURL>
  <RunOnException>c:\MyCustomWebServer\SendEmail.exe</RunOnException>
  <Debug>0</Debug>
</SmartAssembly.WebServer.Settings>
```

**ServerURL:** Enter your own URL for the web server.

**SecuredURL:** Enter the secured URL for the web server. It is usually the same URL than the ServerURL with https instead of http, but it can be different if you share an SSL certificate on a share hosting. If you don't have an SSL connection, you can set the same URL than the ServerURL, but the login will not be secured.

**RunOnException:** (Optional). An application file name. This application will be launched each time the server receives an exception report. It can be used to send an email, update a database, etc...

**Debug:** Set this flag to 1 if you encounter problems setting properly the web-server. This will give more information in case of errors. Set it back to 0 before going public to improve the web service security.

**Warning:** *For performance reasons, this file is read by the Custom Web Server only when the web service is loaded into memory. If you change this file, you will need to re-start IIS to ensure that the Custom Web Server reloads the new version of the settings file.*

- 7) Test your new Web-Service. You will have to rebuild your application with this new setting to ensure that new exceptions will be reported to your new custom web server.
- 8) When you are ready, let us know. We will configure our web server to automatically re-direct reports from previously deployed applications to your web-server. This way, the transition to your new server will be smooth, and you will not lose any report.