Running the Custom Function Example

Overview

These files can be used to create a WindwardCustomFunctions.dll file containing your own custom functions, to be used with the Windward .NET engine and AutoTag, version 12.5 or newer.

Important Notes

1. Do not change the version of the DLL created using this code.
2. To use, replace the assembly of the same name where you installed the .NET Engine or AutoTag.

Requirements

- Windward Reports .NET Engine
    - Available from http://www.windward.net/support/downloads/
    - Installation Tutorial
- Visual Studio

Tutorial

Step 1: Creating Custom Functions

1. Open Catapult. (Found under Start -> All Programs -> Windward Studios -> Windward Reports .NET Engine -> Catapult - Windward Sample Programs)
2. In the Catapult menu, click C# Custom Functions tab.
3. Click the Visual Studio Project file

   In the file WindwardCustomFunctions.cs, you can write your own custom functions to be used with the .NET Engine or AutoTag. We are provided with three functions here, Pi, square root, and multiply all. To view these functions in AutoTag, continue at step 4. Otherwise skip to step 8.

4. Open Microsoft Word.
5. In the AutoTag tab, open the tag builder.
6. Under the select tab, click equation.
7. In the drop-down menu click Custom to view the three custom functions Pi, square root, and multiply all.
8. Return to the Visual Studio Project File.

In the following steps, we will add a function E() that returns the value of Euler's constant, e.

9. Paste the following code in the third section (below multiply all) to add the function itself:
```
public static double E()
{
    return 2.71828182845904;
}
```
10. Use the `FunctionDescriptionAttribute` custom attribute to provide a description to the function. This will be visible in the Equation dialog in AutoTag.
11. Verify that the indentation is correct; the resulting code should look like this:
```
[FunctionDescription("Returns the value of e, 2.71828182845904,
accurate to 15 digits.")]
public static double E()
{
    return 2.71828182845904;
}
```
12. Click Build, Rebuild Solution. (The build will output a file called WindwardCustomFunctions.dll)
13. In Catapult's .NET Custom Tab, open the source folder.
14. In the source folder, open to WindwardCustomFunctions, bin, Debug. If your configuration was set to Release, open the Release folder. Your new .dll file is in this folder.

Step 2: Installing WindwardCustomFunctions.dll for use with the .NET Engine and AutoTag

Installing these functions depends on your .NET Engine installation options. If you chose to install your .dll files in the Global Assembly Cache (GAC) – an option in the installer – you must register the new file in the Global Assembly Cache. If you did not chose this option, you must copy the files over to the .NET Engine and AutoTag install directories. Choose the option that applies to you below, and follow those instructions.

In the .NET Engine Installation, .dll files were copied to the GAC.

1. Open a Visual Studio Command Prompt. (Found under Start -> All Programs -> Visual Studio -> Visual Studio Tools -> Visual Studio Command Prompt)
2. Type "cd\" to navigate to the C: Directory.
3. Type "cd Program Files (x86)\Windward Studios\Windward Reports .NET Engine\htmlhelp\Examples\DotNetCustomFunctionExample\WindwardCustomFunctions\bin\Debug" to navigate to the folder where your WindwardCustomFunctions.dll file is.
4. Type "gacutil /i WindwardCustomFunctions.dll" to add your new WindwardCustomFunctions.dll file to the GAC

In the .NET Engine Installation, .dll files were NOT copied to the GAC.

1. Navigate to the folder containing your new WindwardCustomFunctions.dll file. (Catapult .NET Custom Tab -> Source Button -> WindwardCustomFunctions -> bin -> Debug)

2. Copy the file WindwardCustomFunctions.dll.
3. Navigate to the install directory of the .NET Engine.
4. Open the folder dll.
5. Delete the current WindwardCustomFunctions.dll file.
6. Paste your new file in its place.
7. If you also have AutoTag installed, navigate to the AutoTag Directory, delete the WindwardCustomFunctions.dll file, and paste your new file in its place.

Step 3: Testing the Installation in AutoTag

1. Close all open windows of Microsoft Office.
2. Open Microsoft Word.
3. Click the AutoTag tab.
4. Open the Tag Builder.
5. Click equation, and choose Custom from the drop-down menu. You should see the new function E() in place.