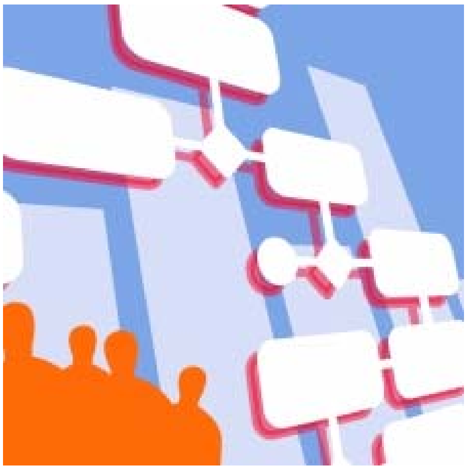
WorkflowGen

WebForms .NET 2 with WorkflowGen.My v2.1

Samples installation

October 30th, 2007



Contents

1 Introduction

1.1 Overview

This setup guide will help you install web form samples that are using the WorkflowGen.My v2.0 assembly.

2 Overview of the samples features

2.1 Sample #1 (VB & C#)

The first sample illustrates the **WorkflowPage using the simple mode**. Here is the list of the principal features covered by this example:

- Automatic double-binding for all the fields;
- Automatic management of required fields
- Automatic fields validation based on the FieldDataType attribute;
- Automatic file uploads management using the FileUpload of the HtmlInputFile control;
- eWorld.UI simple calendar usage for a Date field;
- GridView management
  - Required columns
  - TemplateFields using FieldDataType attribute

2.2 Sample #2 (VB & C#)

The second sample illustrates the **WorkflowPage using the advanced mode.** Here is the list of the principal features covered by this example:

- Double-binding of the web form data for all the fields, using the WorkflowPage BindFormDataToFields and SaveFieldsData functions;
- Management of file uploads using the SaveFileAttachment function;
- Automatic fields validation based on the FieldDataType attribute;
- Advanced GridView using a Business Object and an ObjectDataSource for the binding;
- Easy signing of the web form data using Xml Signatures;

2.3 Sample #3 (VB & C#)

The third example is a more realistic sample of a time sheet web form. Here is the list of the principal features covered by this example:

- Automatic double-binding for all the fields;
- Automatic management of required fields
- Automatic fields validation based on the FieldDataType attribute;
- GridView management
  - Required columns
  - TemplateField with a DropDownList for edition
- Saving the web form as a draft

2.4 Sample #4 (VB)

This example is a simple web form of a project approval process made in Visual Basic.

- Double-binding of the web form data for all the fields, using the WorkflowPage BindFormDataToFields and SaveFieldsData functions;
- Management of file uploads using the SaveFileAttachment function;
- Automatic fields validation based on the FieldDataType attribute;
- Advanced GridView using a Business Object and an ObjectDataSource for the binding;
- Filling DropDownLists from an external data source and making them dependant one from another.

2.5 Sample #5 (VB & C#)

This example is a simple mode example using WorkflowGen.My v2.1 and AJAX enabled pages, using the **ASP.NET 2.0 AJAX Extensions 1.0**, available at http://ajax.asp.net.

Prerequisites: The **ASP.NET 2.0 AJAX Extensions 1.0** *must be installed on the web server before running this sample; otherwise, you will not be able to see this sample.*

- Using the UpdatePanels controls and the ScriptManager of **ASP.NET 2.0 AJAX Extensions 1.0**;
- Separating different parts of the page with different UpdatePanels, so that only the right portions of a page are updated when they need to, without seeing any PostBack on the client side;
- Making two dependant drop down lists AJAX enabled;
- Using the Microsoft Calendar as a DatePicker;
- Making a GridView AJAX enabled;

2.6 Sample #6 (VB & C#)

This example is a very simple web form using the WorkflowFileUpload control.
Prerequisites:

- *WorkflowGen.My v2.1.6 or higher*

2.7 Sample #7 (VB & C#)

This example is using the WorkflowFileUpload control and **ASP.NET 2.0 AJAX Extensions 1.0**, available at http://ajax.asp.net.

Prerequisites:

- *The* **ASP.NET 2.0 AJAX Extensions 1.0** *must be installed on the web server before running this sample; otherwise, you will not be able to see this sample.*
- *WorkflowGen.My v2.1.6 or higher*

3 Installing the samples

3.1 Copy the files

The first step is to copy all the required files at the right place. The package contains the following directory tree:

/wfapps/webforms/samples/
/wfapps/webforms/samples/cs
/wfapps/webforms/samples/cs/sample1 *(simple mode example)*
/wfapps/webforms/samples/cs/sample2 *(advanced mode example)*
/wfapps/webforms/samples/cs/sample3 *(timesheet example)*
/wfapps/webforms/samples/cs/sample5 *(Atlas AJAX example)*
/wfapps/webforms/samples/cs/sample6 *(simple workflowfileupload)*
/wfapps/webforms/samples/cs/sample7 *(workflowfileupload & Atlas AJAX example)*
/wfapps/webforms/samples/vb
/wfapps/webforms/samples/vb/sample1 *(simple mode example)*
/wfapps/webforms/samples/vb/sample2 *(advanced mode example)*
/wfapps/webforms/samples/vb/sample3 *(timesheet example)*
/wfapps/webforms/samples/vb/sample4 (*more sophisticated advanced mode example)*
/wfapps/webforms/samples/vb/sample5 *(Atlas AJAX example)*
/wfapps/webforms/samples/vb/sample6 *(simple workflowfileupload)*
/wfapps/webforms/samples/vb/sample7 *(workflowfileupload & Atlas AJAX example)*

You need to unpack those directories with their contained files at this location:
DISK:\Inetpub\YOURSITE\wfgen

3.2 Create the IIS applications

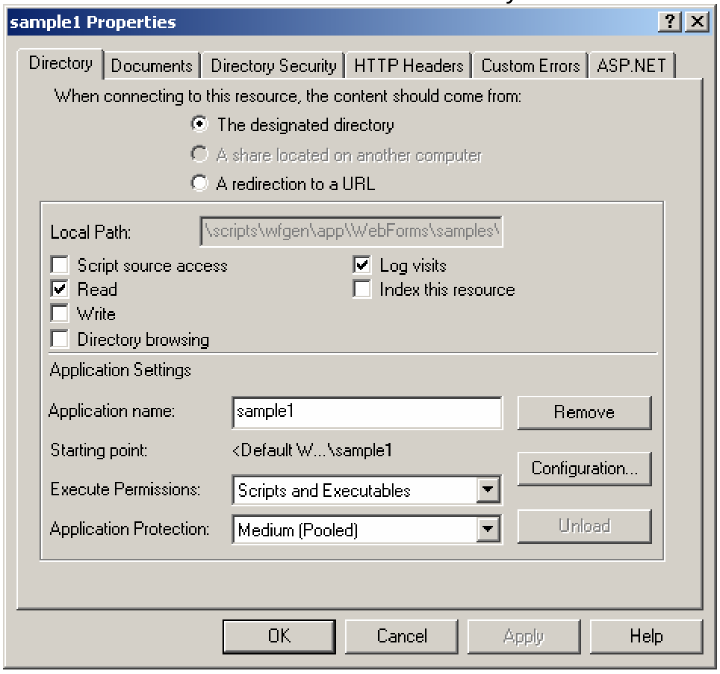
For each sample you want to test, you need to create a different IIS application.

1. Open IIS (Internet Information Services)
2. Navigate to the sample# directory under your WorkflowGen website.
   For C#
   /wfgen/wfapps/webforms/samples/cs/sample#

   For VB.NET
   /wfgen/wfapps/webforms/samples/vb/sample#

3. Right-click on this directory and choose “Properties”
4. Click on the “Create” button next to the “Application name” text box in the Directory tab.

5. Check the “Read” checkbox in the Directory tab.



sample1 Properties

When connecting to this resource, the content should come from:

● The designated directory
○ A share located on another computer
○ A redirection to a URL

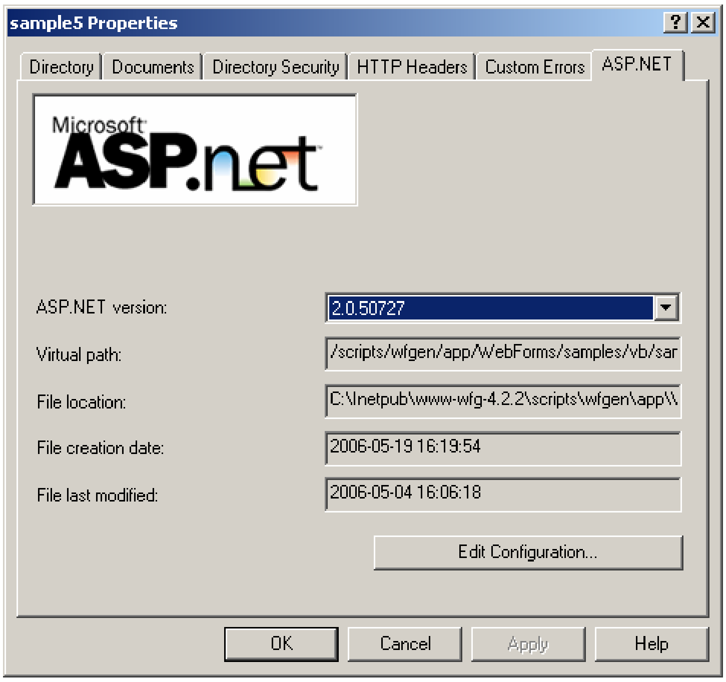
Local Path: \scripts\wfgen\app\WebForms\samples\

□ Script source access
☑ Read
□ Write
□ Directory browsing
☑ Log visits
□ Index this resource

Application Settings

Application name: sample1
Starting point: <Default W...\sample1
Execute Permissions: Scripts and Executables
Application Protection: Medium (Pooled)

6. Click on the “ASP.NET” tab and be sure that the v2.0.xxxxx is selected.



sample5 Properties

ASP.NET version: 2.0.50727
Virtual path: /scripts/wfgen/app/WebForms/samples/vb/sar
File location: C:\Inetpub\www-wfg-4.2.2\scripts\wfgen\app\\
File creation date: 2006-05-19 16:19:54
File last modified: 2006-05-04 16:06:18

7. Click on “OK”
8. Repeat those step for each sample#

3.3 Refering to WorkflowGen.My v2.1 for each sample

The samples do not include the WorkflowGen.My v2.1 assembly. You need to add a reference to this assembly in the samples in order to make them work.

There is two ways you can refer to the assembly: by putting it in the Bin directory under the root of the sample or by putting the WorkflowGen.My assembly in the Global Assembly Cache (GAC) and then referring to it.

3.3.1 Bin directory

- Copy the WorkflowGen.My.dll file in the Bin folder of the project;
- Open Visual Studio, and choose the File menu → Open Web Site;
- Pick your project root folder;
- Right-click on your project name (in the Solution Explorer) and choose “Add reference…” in the contextual menu;
- Click on the “Browse” tabulation;
- Navigate to the “bin” directory you just created and choose the WorkflowGen.My.dll file;
- Click on OK;
- WorkflowGen.My is now referenced in your project.

3.3.2 Using the Global Assembly Cache (GAC)

You can also put this assembly in your GAC in order to have only one centralized reference instead of referring a copy of the assembly in each web form project. Here are the instructions to install the assembly in your GAC:

- Copy WorkflowGen.My.dll to c:\windows\system32;
- Open a command prompt;
- Go to C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\Bin;
- Type this command:
  - gacutil /i c:\windows\system32\WorkflowGen.My.dll
  - Press enter
  - You should now receive a message to confirm that the assembly has been added to the Global Assembly Cache.
- WorkflowGen.My is now in your GAC.

Your project now needs to have a reference to this centralized assembly. You can add this reference in the web.config file of your project.

The “assemblies” node should contain the following nodes:

```
<add assembly="WorkflowGen.My, Version=2.1.1000.xx, Culture=neutral,
PublicKeyToken=df70d2b8f21e0354"/>
```

Where the “xx” should be replaced with the version you currently have of WorkflowGen.My.dll

You can now Rebuild the project, by using the “Build” menu, and by clicking on the “Rebuild Web Site” menu item.

3.4 Install the process in WorkflowGen

The final step is to install the process in WorkflowGen using the XPDL files. For each sample you want to test, create a new process in WorkflowGen using the XPDL files situated at the root of the setup package.

1. Open the administration module of WorkflowGen
2. Create a new Folder named “Samples” or just choose a folder that already exists.
3. Click on “New process”
4. Click on “Import” at the bottom of the form.
5. Click on “Browse” and choose the XPDL file for the sample you want to
6. Click on “Import”
7. Change the process status to “Test”
8. Save the process
9. Click on “Test” and the web form should start.

www.workflowgen.com