

Great Migrations

ASP #Include Upgrade Samples

Overview

gmStudio supports three options for upgrading ASP #include files to ASP.NET:

1. Convert #include files Code Classes
2. Convert #include files to web UserControls with associated code behind files
3. Convert some #include files to code classes and others to UserControls

Code Classes

The default ASP translation script upgrades each #include file to a code class. Converting an #include file to a code classes makes sense if the #include files does not contain markup. These “faceless” includes contain only declarations such as common constants, functions, and VBScript classes. Note that when an #include containing markup is converted to a code class, gmStudio separates the markup from the declarations and consolidates the markup into an AuthorMarkup function within the code class. Calling the AuthorMarkup function renders the markup to the HTTP response. The appropriate AuthorMarkup function for each #include is called wherever the #include file was referenced in the original ASP site.

Web UserControls

If a #include file contains a lot of markup then it may be more maintainable to upgrade it to a web UserControl. This feature may be activated by adding few special instructions to the translation script; for example:

```
<Select UseUserControls="on" />
<LoadRuntime dllName="AspGlobalIncludes.dll" />
<Select GlobalSettings="%UserFolder%\GlobalSettings" />
...
<Compile PageSlice... >
...
```

You must use a GlobalIncludes description file when you set UseUserControls=on and the AspGlobalIncludes.dll. In order to use a GlobalIncludes file, you must complete the following preparation steps:

1. Prepare a GlobalIncludes.xml script to generate a GlobalIncludes description file based on a global analysis of the #include file usage throughout the site.
2. Place the GlobalIncludes description content in a GlobalSettings script to create a GlobalSettings file.
3. Reference the GlobalSettings file with the following: <Select GlobalSettings="%UserFolder%\GlobalSettings" />

When UseUserControls=on, gmStudio will declare a web UserControl with codebehind based on the content of each #include. It will then Register a custom tag for each UserControl used a page and insert the customer tag #include file was referenced in the original ASP site.

Sample ASPX page with custom UserControl tags

```
<%@ Register TagPrefix="inc" TagName="t_head" Src="t_head.ascx" %>
<%@ Register TagPrefix="inc" TagName="t_begin" Src="t_begin.ascx" %>
<%@ Register TagPrefix="inc" TagName="t_end" Src="t_end.ascx" %>
...
<html>
...
<inc:t_head ID="_t_head1" runat="server" />
...
<inc:t_begin ID="_t_begin1" runat="server" />
...
<inc:t_end ID="_t_end1" runat="server" />
...
</html>
```

Mixing Code Classes and Web UserControls

If you wish to selectively upgrade some #includes into UserControls while others are upgrade to code classes, you may add Registry commands such as the following to a standard ASP translation script.

```
<Registry type="FixStatus" source="include_file_path" target="UserControl"
/>
```

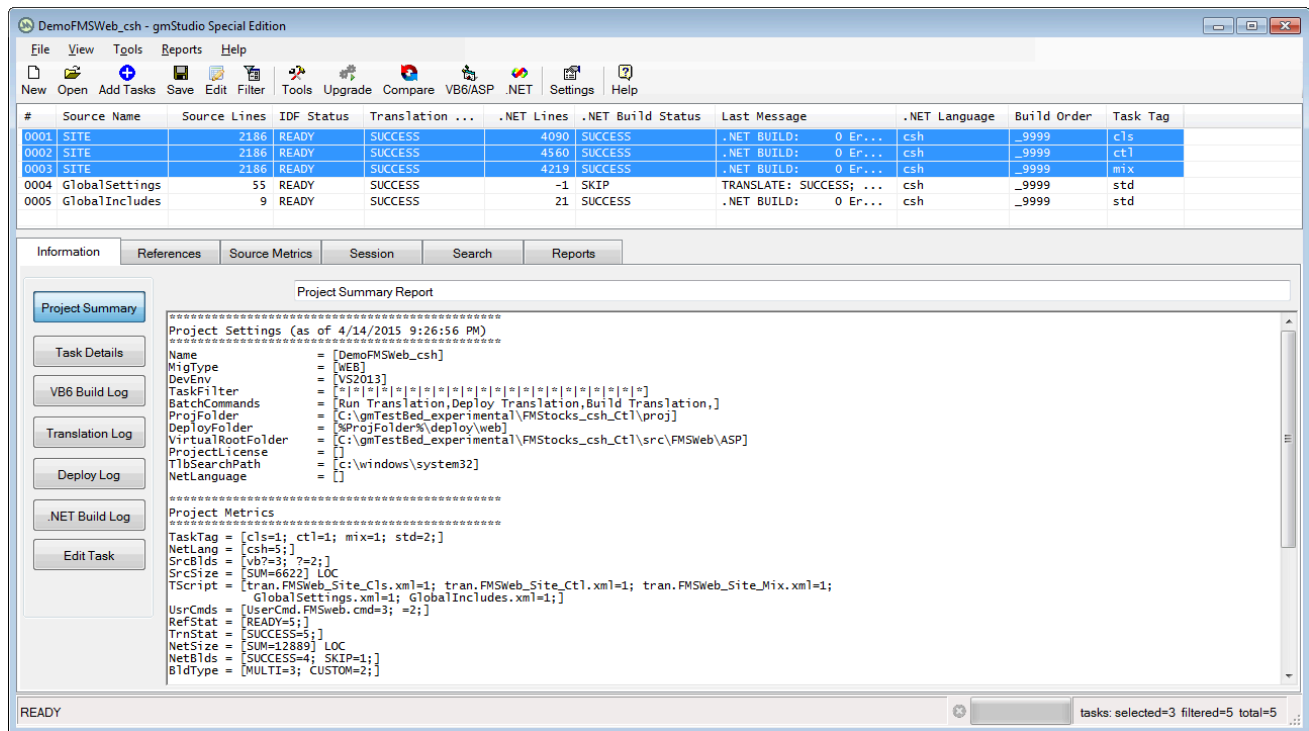
Note

Note: the behavior of <Registry type=FixStatus ... /> is based on an older style of upgrading #includes to UserControls that used the FindControl API to initialize control instances. This older style is deprecated and will be updated to be consistent with the custom tag style used with UseUserControls=on in a future release of the tool.

Sample Translations

The #include files upgrades sample is based on the fmStocks VB6/ASP upgrade sample. You should review [the fmStocks sample documentation](#) for an overview if you are not already

familiar with it.



The sample files are attached in this zip file: [fmStocks_Ctl_01.zip](#). This sample is a draft and will be updated to standard sample format in a future release.

The sample project illustrates the upgrades being done three ways. Specifically, the gmProj file has three different versions of the upgrade, each referencing a different site translation script:

- tran.FMSWeb_Site_Cls.xml: produces an ASP.NET upgrade using Code Classes. The same as the upgrade done for the standard sample.
- tran.FMSWeb_Site_Ctl.xml: produces an ASP.NET upgrade using UserControls.
- tran.FMSWeb_Site_Mix.xml: produces an ASP.NET upgrade using UserControls for all the #includes except strings.asp.

Note: it is somewhat unusual for a gmStudio project to have different upgrade configurations for the same source code, but it makes it easy to compare the results of each configuration by selecting any two upgrade tasks and clicking the compare button. However, beware that all three of these tasks share the same deployment folder and when you deploy the results they will overwrite the content of the deploy folder.

The sample project also contains translation tasks to generate the GlobalIncludes and GlobalSettings files.

Please contact GreatMigrations if you have questions.