

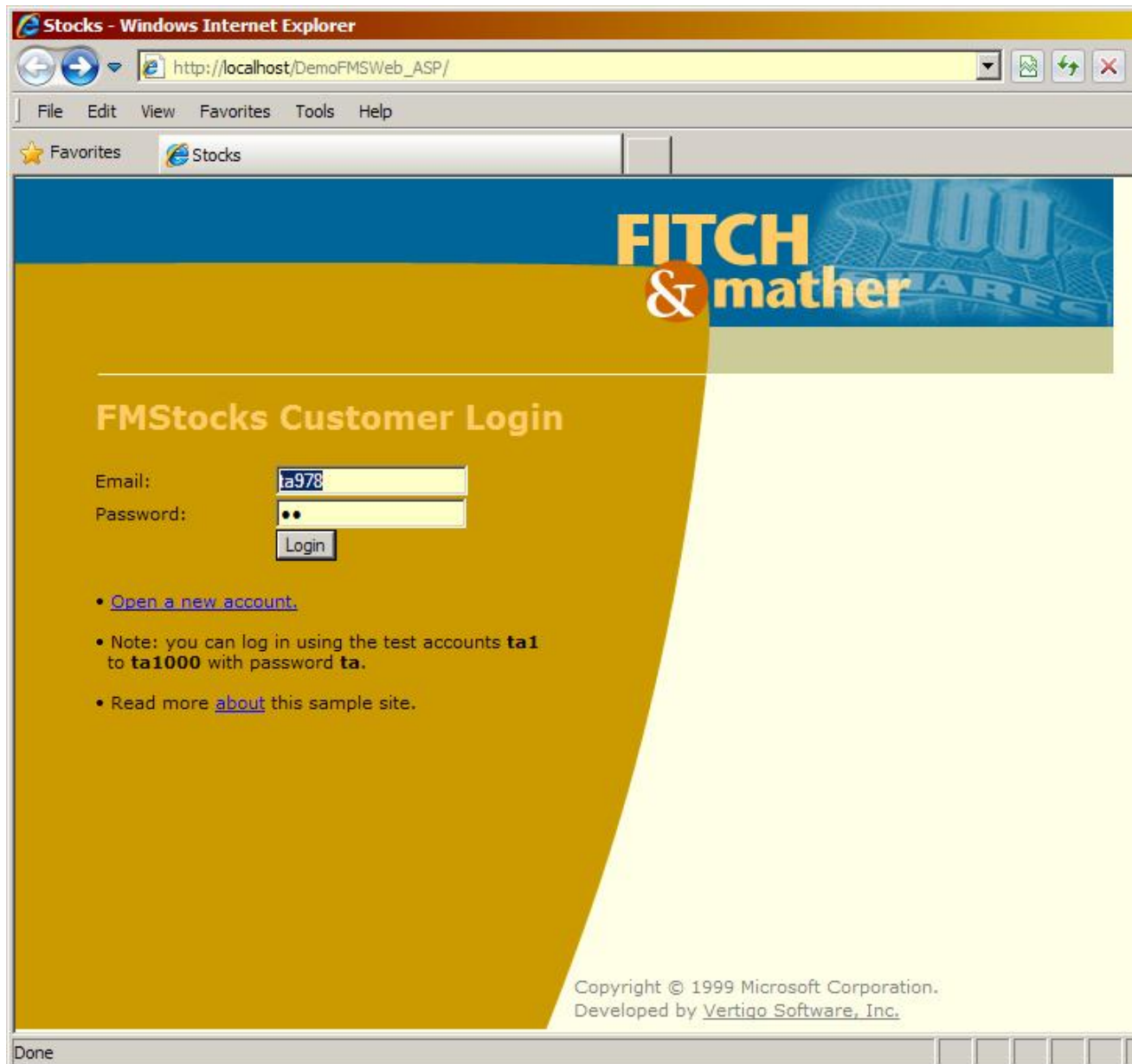
Great Migrations

FMStocks (VB6/ASP)

Overview

The FMStocks sample consists of two sets of projects:

- FMSLib contains six inter-related VB6 libraries and one test executable
- FMSWeb contains thirty-four ASP pages, which reference the six VB6 libraries in FMSLib.



 **Video:** https://www.youtube.com/watch?v=XofC8_QpwK4

The FMStocks sample shows how to migrate interrelated ASP and VB6/COM components. The application was published to demonstrate a 3-tier architecture (ASP, COM+, and SQL Server) in the context of a stock trading site with the following features:

- Portfolio Management
- Research/Buy/Sell Stocks
- On-line Store
- Forms-based Login

FMStocks, the Fitch and Mather Stocks 2000 sample, was created by Vertigo software and distributed through MSDN to demonstrate some of the best practices for building scaleable web applications using ASP and COM+. FMStocks was also featured in the Patterns and Practices guide as an example of how to upgrade VB6 to .NET.

It consists of seven (7) VBPs (2500 LOC)

FMStocks_DB.vbp	(Data Access)
FMSStore_DB.vbp	(Data Access)
FMSStore_Bus.vbp	(Business Objects)
FMStocks_Bus.vbp	(Business Objects)
FMSStore_Events.vbp	(Business Objects)
FMStocks_Ext.vbp	(Business Objects)
FMSTest.vbp	(A Simple Desktop Application Test Harness)

and thirty-four ASP pages (2300 LOC)

ASP Pages
Include files
Pulp (images, css, client-side script files, etc.)

A pivot table of data produced by the Source Structure report for the FMStocks VB6 code is displayed below.

SrcFile	ModName	ModType	Sum of Cnt_Tot	Sum of Cnt_Cod	Sum of Cnt_GUI
FMSSStore_Bus.vbp	Helpers	Module	201	143	0
	Product	Class	72	48	0
	ShoppingCart	Class	188	131	0
FMSSStore_Bus.vbp Total			461	322	0
FMSSStore_DB.vbp	Helpers	Module	201	143	0
	Product	Class	67	45	0
	ShoppingCart	Class	131	88	0
FMSSStore_DB.vbp Total			399	276	0
FMSSStore_Events.vbp	ShoppingCart	Class	19	15	0
FMSSStore_Events.vbp Total			19	15	0
FMSTest.vbp	frmFMSTest	Form	413	285	38
	utils	Module	35	30	0
FMSTest.vbp Total			448	315	38
FMStocks_Bus.vbp	Account	Class	171	107	0
	Broker	Class	340	215	0
	Helpers	Module	201	143	0
	Ticker	Class	143	91	0
	Version	Class	49	40	0
FMStocks_Bus.vbp Total			904	596	0
FMStocks_DB.vbp	Account	Class	134	89	0
	Broker	Class	91	66	0
	DBHelper	Class	371	253	0
	Helpers	Module	193	137	0
	Position	Class	66	47	0
	Ticker	Class	96	64	0
	Tx	Class	84	55	0
	TxNew	Class	42	31	0
	Version	Class	37	31	0
FMStocks_DB.vbp Total			1114	773	0
FMStocks_Ext.vbp	Helpers	Module	201	143	0
	History	Class	45	29	0
FMStocks_Ext.vbp Total			246	172	0
Grand Total			3591	2469	38

A pivot table of data produced by the Migration Set Report for the FMStocks web site is displayed below.

BldType ▼	SrcName ▼	Sum of SrcSize	Count of SrcName
INCLUDE	strings_asp	24	1
	t_begin_asp	67	1
	t_begin_cart_asp	41	1
	t_end_asp	16	1
	t_head_asp	75	1
INCLUDE Total		223	5
PAGE	_about_asp	36	1
	_GetXMLPortfolio_asp	27	1
	_GetXMLTickerHistory_asp	18	1
	_NewAccount_asp	158	1
	_runsql_asp	78	1
	_version_asp	69	1
	401k_asp	43	1
	AccountSummary_asp	41	1
	browse_asp	46	1
	BuyStock_asp	154	1
	cart_asp	99	1
	ChartPortfolio_asp	339	1
	checkout_asp	68	1
	default_asp	168	1
	FindProduct_asp	52	1
	Home_asp	19	1
	Logout_asp	13	1
	news_asp	25	1
	OfflinePortfolio_asp	19	1
	Portfolio_asp	134	1
	SellStock_asp	119	1
	SellStockAction_asp	51	1
	SellStockReceipt_asp	36	1
	store_asp	25	1
	template_asp	21	1
	TickerDetail_asp	108	1
	TickerList_asp	111	1
	ViewProduct_asp	71	1
PAGE Total		2148	28
Grand Total		2371	33

The migration concepts introduced in the context of the ScanTool demo also apply to ASP-to-ASP.NET translations. For the most part, these concepts are all handled the same way for ASP as they are for VB6. In fact, gmStudio compiles both VB6 and ASP codes to the same intermediate semantic model format before analyzing and restructuring them to the target architecture. However, in addition to a programming language change — in this case VB Script to C# or VB.Net — ASP brings several fundamental migration challenges:

- Subtle changes in the semantics of core ASP classes (Request, Response)
- Scoping matters relating to nested and stacked ASP include files
- Weak typing everywhere
- Various obscure differences between the syntax VBScript and Visual Basic

- Different rules for structuring markup and script tags
- Different format and semantics of ASP directives
- Looser coupling to COM, and
- Architectural decisions regarding where and how to use the many powerful new features of ASP.NET

The FMStocks sample illustrates how gmStudio handles many of these matters for you. FMStocks also provides another example of VB to .NET migration because there is a set of VB6 COM objects providing the business and data objects used by the pages.

COM Dependencies

FMStocks depends on the following external COM components that must be installed on your workstation to perform the migration successfully:

File	Description	Upgrade Strategy
scrrun.dll	Scripting, Microsoft Scripting Runtime	Replace with System.IO
msado27.tlb	ADODB, Microsoft ActiveX Data Objects 2.7 Library	Replace with System.Data and System.Data.SqlClient with the help of MigrationSupport.DataLib , a set of classes that implement the ADODB API in C#. MigrationSupport.DataLib is part of the sample <u>runtime library</u> available as source with a Professional Edition License of gmStudio.

Note: The version of FMStocks distributed as a sample here uses

File	Description	Upgrade Strategy
COM instead of COM+.		

Special Features: a Structural Migration

This sample illustrates the **SharedFiles** statement that allows placing a shared code file into a selected .NET translation and referencing its content from that location. In this example Helpers.bas is shared by several of the VB6 projects, but in the generated code, there is only one copy in a selected "host" project. The other projects reference the "host" assembly and reference the Helpers content from there. See GlobalSettings.xml script with the sample and the [gmpSharedFileStatement](#) for more information.

Deployment

The source codes for the FMSTLib VB6 projects are located in the [INSTALLDIR]\samples\source\FMStocks\FMSTLib subdirectory. The source code for the ASP pages is located in the [INSTALLDIR]\samples\source\FMStocks\FMSWeb\asp subdirectory. Four sample Migration Projects are included in the FMStocks example, depending on target language (C# or VB.NET) and treatment of referenced components (Interoped or migrated to .NET), as shown below:

Project[.gmproj]	Description
DemoFMSTLib_csh	All interrelated VB6 DLLs are migrated to C#.NET
DemoFMSWeb_csh	ASP Site and FMSTLib libraries are migrated to C#.NET
DemoFMSTLib_vbn	All interrelated VB6 DLLs are migrated to VB.NET

Project[.gmproj]	Description
DemoFMSWeb_vbn	ASP Site and FMSLib libraries are migrated to VB.NET

The FMStocks projects have the following deployment settings:

The VB6 Projects (COM libraries):

```
ProjectFolder = [INSTALLDIR]\samples\workspace\FMStocks_[lang]
DeployFolder  = %ProjFolder%\deploy\lib
```

The ASP Projects (web site):

```
ProjectFolder = [INSTALLDIR]\samples\workspace\FMStocks_[lang]
DeployFolder  = %ProjFolder%\deploy\web
```

Log files showing all migration steps are created in the workspace\log folders.

```
[INSTALLDIR]\samples\workspace\FMStocks_[lang]\log
```

The .NET project files are deployed to the following folders:

.NET Library Code:

```
[INSTALLDIR]\samples\workspace\FMStocks_[lang]\deploy\lib\  
[project]_std_[lang]
```

ASP.NET Site:

```
[INSTALLDIR]\samples\workspace\FMStocks_[lang]\deploy\web
```

where:

[project] is the name of the migration unit associated with each migration task in the migration project.

[lang] is "csh" or "vbn" depending on target language choice.

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" />
```

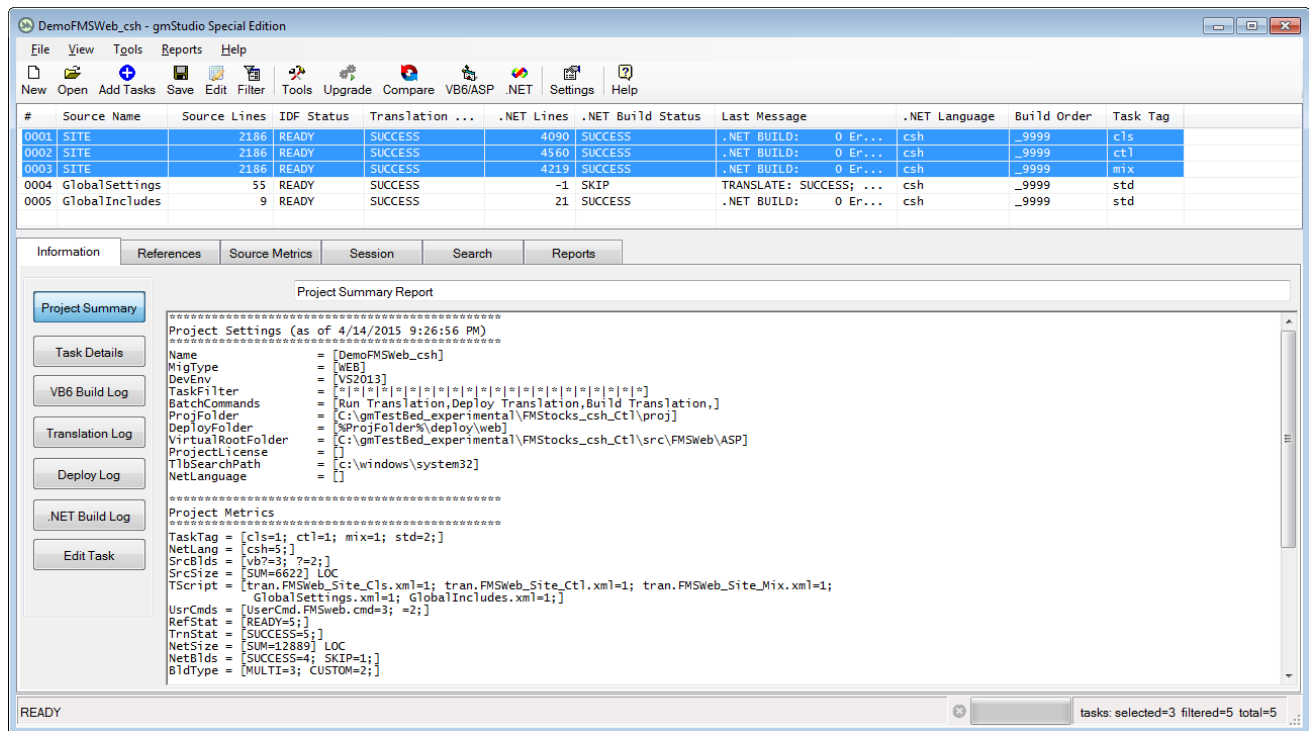
Note

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.

Great Migrations

FMStocks DB

The FMStocks sample uses ADODB to access a small SQL Server Database. A script to create the database is here:

Files

1 FILES

Q

Search attachments...

FILE DETAILS	SIZE	ACTIONS
 stocks_database_sql2012.zip	767.5 KB	VIEW DOWNLOAD

< Prev

1

Next >

