SSW.Framework.Web.MVC

This project is a library of useful functions for developing MVC websites.

Table of Contents

[Quick guide to using Git 1](#_Toc360194567)

[Distributed Source Control 1](#_Toc360194568)

[Pre requisites 4](#_Toc360194569)

[Solution Projects 4](#_Toc360194570)

[SSW.Framework.Web.Mvc4 4](#_Toc360194571)

[SSW.Framework.Web.Mvc.Examples 4](#_Toc360194572)

[Updating the nuget package 4](#_Toc360194573)

[Deploying the Demo Website 6](#_Toc360194574)

[SSW.Framework.Web.Mvc.Api 7](#_Toc360194575)

[Sync with GitHub 7](#_Toc360194576)

[Setup (do this once) 7](#_Toc360194577)

[Perform a sync 7](#_Toc360194578)

# Quick guide to using Git

As this may be your first attempt at using git, here’s a very quick introduction.

## Distributed Source Control

Git is a *distributed* source control system. That mean that when you work with Git, you have the complete repository installed locally to work on.

When you commit, you commit to your local repository. The advantage here is that advanced operations such as branching and merging that can take a long time with traditional scc systems are very fast with git.

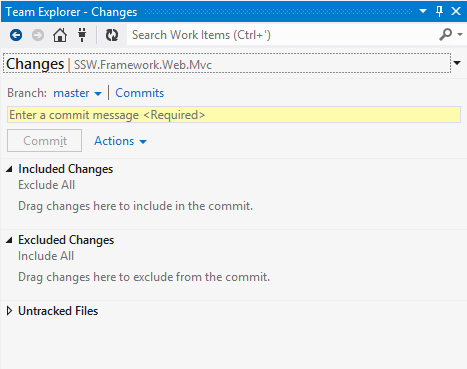
What this essentially means is that when you work with git, there are two operations to “check in” your code to the main TFS Repo: a commit and then a push

Workflow is like this:

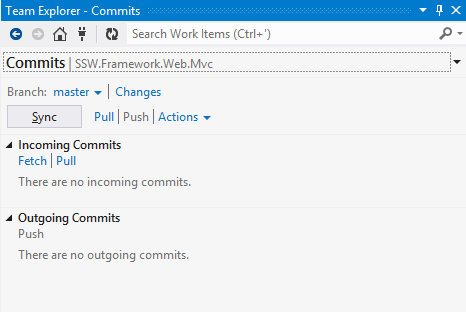
1. Pull gets changes from the TFS repo into your repo (ie Get Lastest)
2. Make some changes to the code
3. Commit will commit your changes to your local repo
4. Push will sync your changes back to the tfs repo

Conflicting changes will be detected in step 4 and you’ll get the ability to use the same merge tool as for TFS source control.

The Sync button in Visual Studio will perform a pull and then a push operation.



**Figure: The Changes view lets you see changed files and commit them to your local repo**



**Figure: the Commits view shows local commits you have made. Use the sync button to sync with the main repo on tfs – this can be considered as equivalent to performing a “Get Latest” followed but a “check-in”.**

# Pre requisites

If you are reading this I assume you already have git installed.

In order to build the generated Api documentation you’ll also need to install sandcastle from here:

<http://shfb.codeplex.com/>

# Solution Projects

This is a quick guide to the projects you’ll find in the solution

## SSW.Framework.Web.Mvc4

This is the core library. It complies to a dll that is then included in a Nuget package.

There are a couple of Powershell scripts:

**package.ps**

This builds a Nuget package to local disk for testing

**publish.ps**

This builds a Nuget package and publishes directly to the Nuget site.

To run this publish, you will need to download and install the nuget\_api.key file from here:

Do not check this file into source control. Git has been set to ignore this file via .gitignore

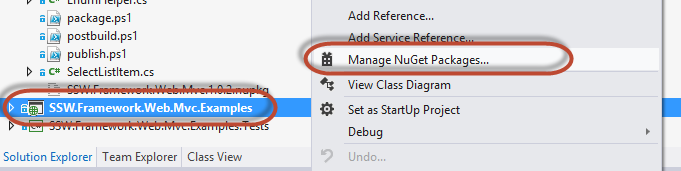
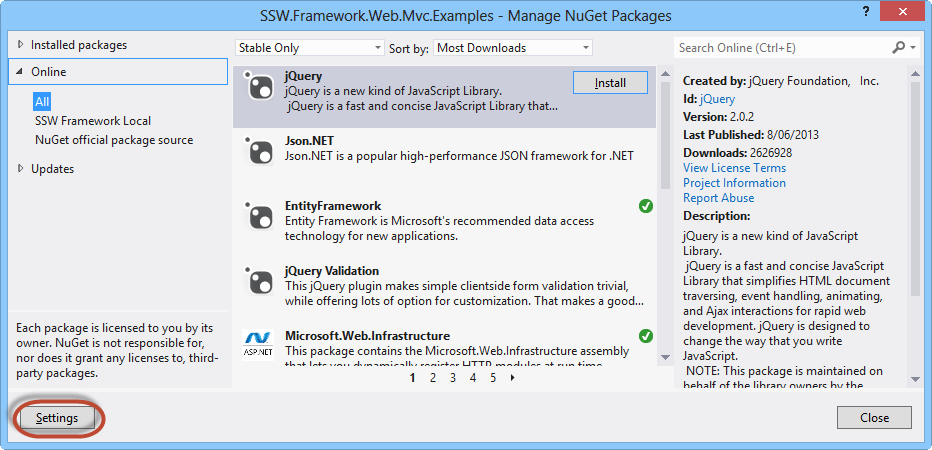
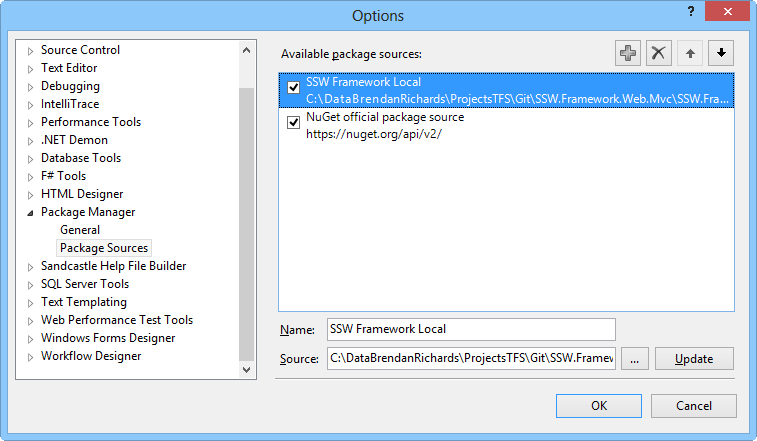
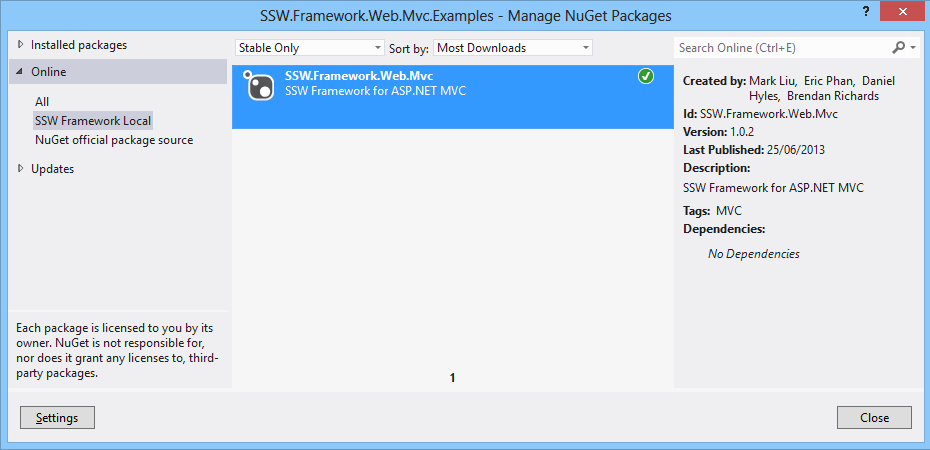
## SSW.Framework.Web.Mvc.Examples

This is a documentation and demo website that includes live demos using the nuget library.

### Updating the nuget package

So that the full SSW.Framework.Web.Mvc package can be tested locally with the Examples site without publishing, the .Mcv4 project is installed as a local Nuget package – not as a simple project reference.

To update the Examples site with the latest Mvc4 package:

1. Build the .Mvc4 project and run package.ps to build the Nuget package
2. The first time you do this, you will need to setup the folder containing the above package as a Nuget source.   
     
   **Figure: Manage NuGet Packages** **Figure: Settings**  
     
     
     
   **Figure: Setup the SSW.Framework.Web.Mvc4 as a package source**
3. 
4. **Now you’ll be able to update the package via the Manage NuGet packages window.**

### Deploying the Demo Website

The demo site has been setup with continuous deployment to azure:

<http://ssw-mvc.azurewebsites.net/>

To Deploy, just push to GitHub (see below) and Azure will do the rest

## SSW.Framework.Web.Mvc.Api

This project uses sandcastle to generate full api documentation for the .Mvc dll.

The documentation is built to a subfolder of the Examples site and is deployed with the examples site. Install sandcastle to run this build.

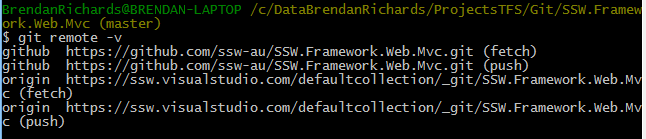
<http://shfb.codeplex.com/>

# Sync with GitHub

Performing a sync with github, publishes all our source code to github and also triggers the Examples site to deploy to azure.

To sync with github perform the following steps:

### Setup (do this once)

1. Create a personal account on GitHub
2. Add permissions for your account to push to the repo repository  
   Login to GitHub using the ssw-au account for this – password details are in the document under:  
   <http://projects.ssw.com.au/tfs/SSW.Framework.Web.Mvc/_layouts/15/start.aspx#/SitePages/Home.aspx>
3. Open git bash (or any git cli) to the repository’s root folder.
4. Create a new “Remote” branch referencing the GitHub repo.  
   $ git remote add github <https://github.com/ssw-au/SSW.Framework.Web.Mvc.git>
5. Run “git remote -v” to check your setup:  
   

### Perform a sync

$ git pull github master

$ get push github master

When prompted, use the authentication details configured above.