Guidelines for Nuget Hosting

IGHS Jaguars Team

Author:	Sathish Ramaraj.
Contributors:	
Project Manager:	
Document Version:	1.0
Initially Created:	09 Oct, 2014
Last Updated:	09 Oct, 2014
Signed Off by:	
Signed Off Date:	

Your Own Local and Remote NuGet Feeds

Some companies restrict which third-party libraries their developers may use. Therefore, they might not want developers to have access to everything in the official NuGet feed, or they might have a set of proprietary libraries they want to make available in addition to the official feed.

In these scenarios, you can set up a custom NuGet feed, and you can configure Visual Studio to offer that feed instead of or in addition to the official feed. A feed can be local (a folder on the local machine or a network folder), or remote (an intranet or internet URL).
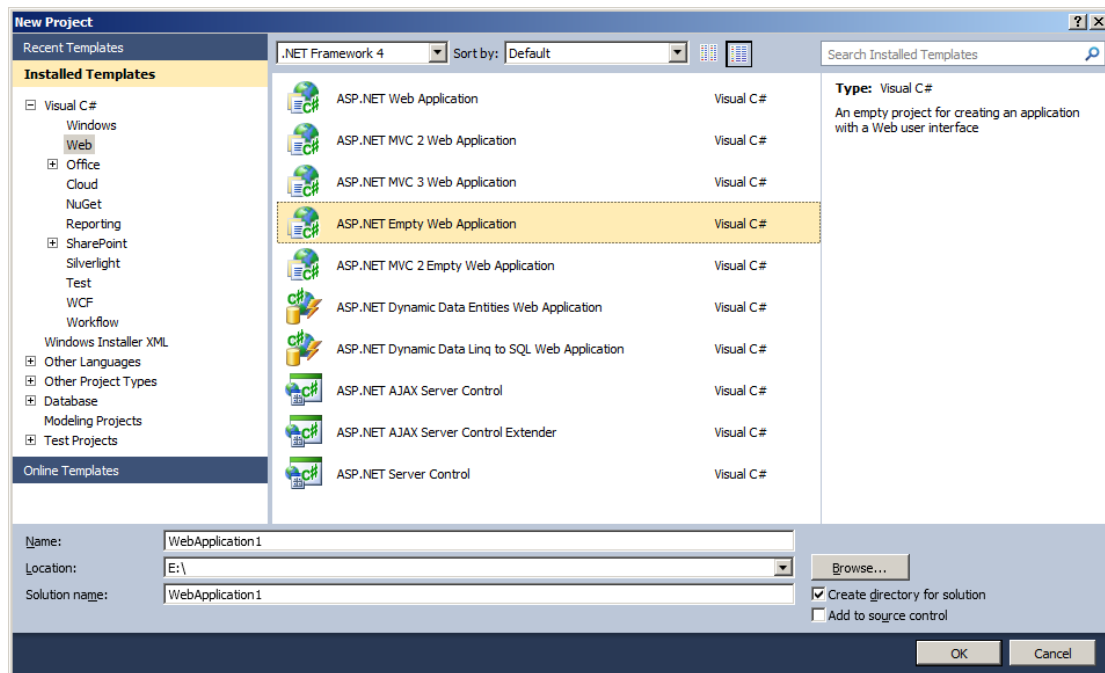
NuGet supports these scenarios with a combination of two features:

1. NuGet can point to any number of different feeds. You don't have to point it to just our feed.
2. NuGet can point to a local folder (or network share) that contains a set of packages.
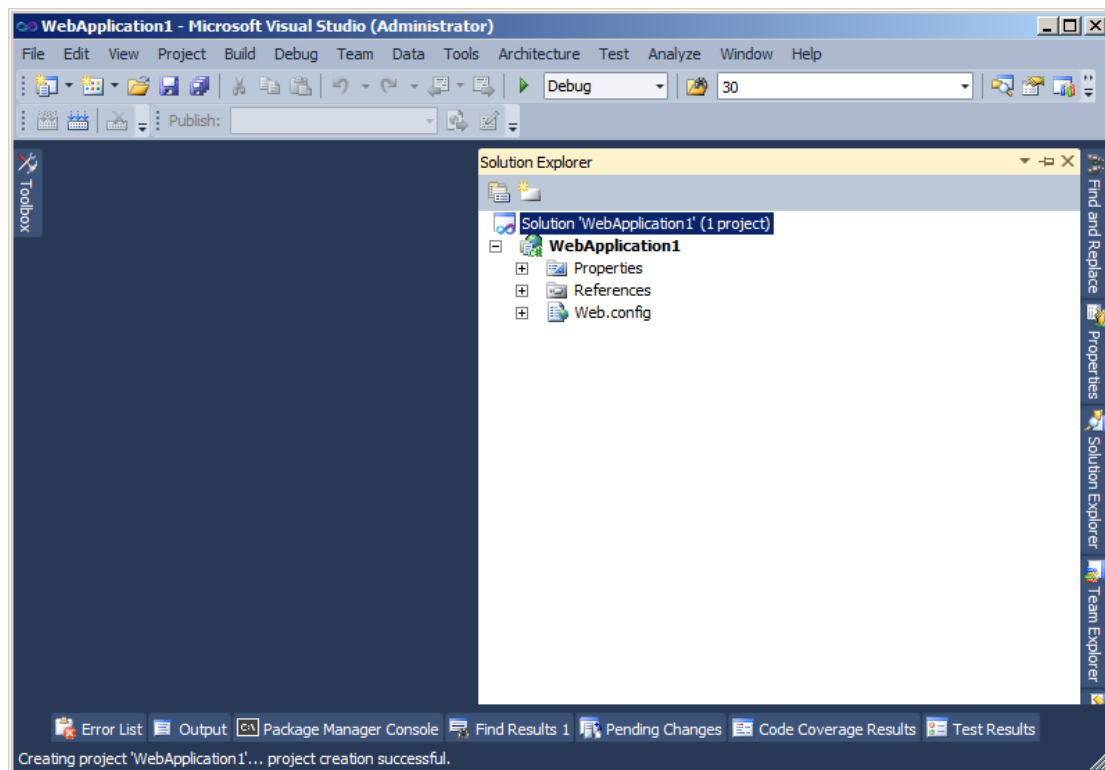
Creating Remote Feeds

Step 1: Create a new Empty Web Application in Visual Studio

Go to the **File | New | Project** menu option (or just hit CTRL + SHIFT + N) which will bring up the new project dialog and select **ASP.NET Empty Web Application** as in the following screenshot. Please note that the steps here including NuGet.Server package are intended for use with C# project only.
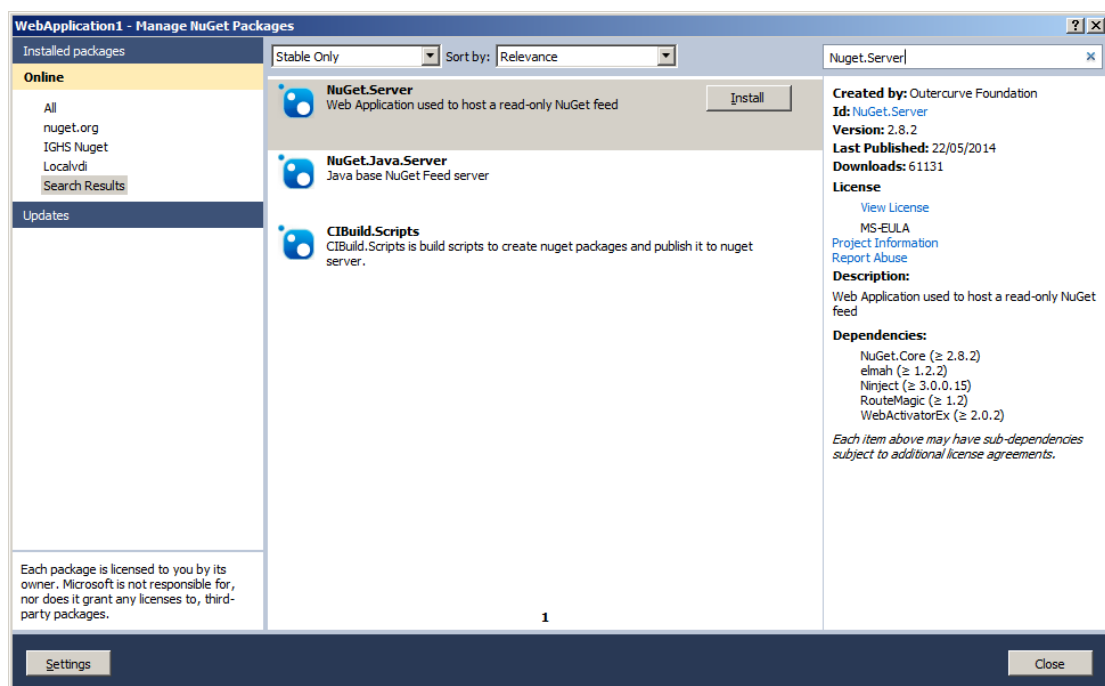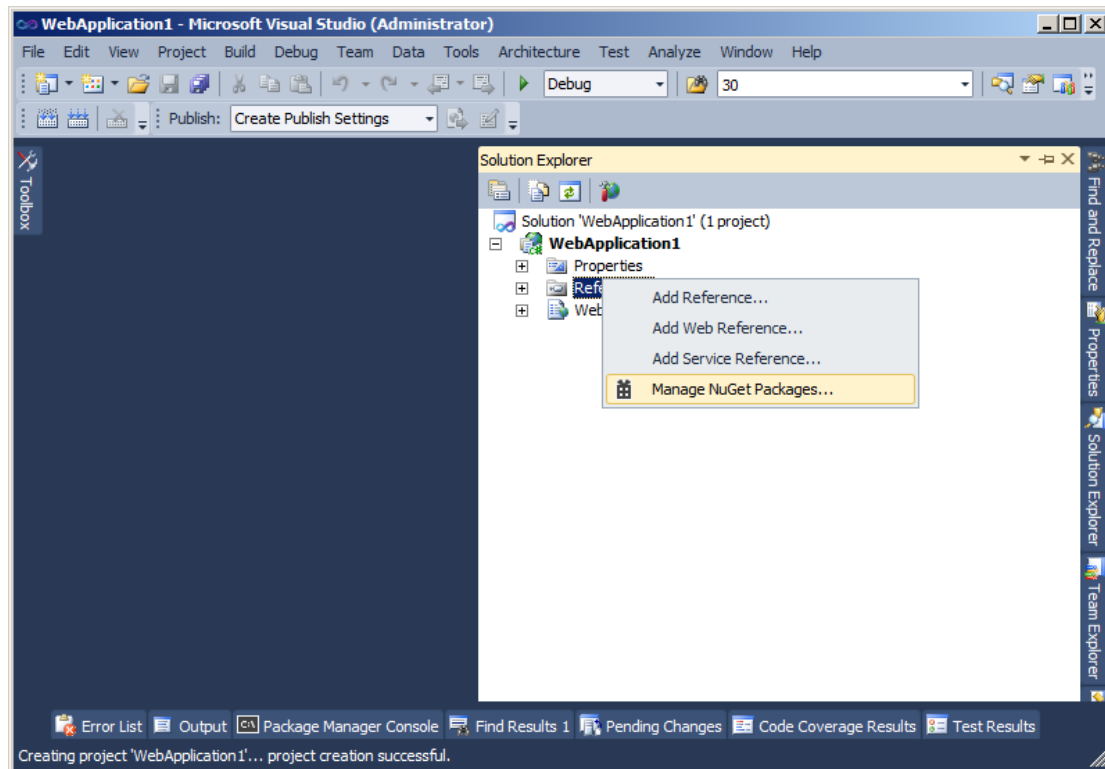
This results in a very empty project template.

Step 2: Install the NuGet.Server Package

Now right click on the References node and select Manage NuGet Packages to launch the NuGet dialog (alternatively, you can use the Package Manager Console instead and type Install-Package NuGet.Server).

Click the **Online** tab and then type **NuGet.Server** in the top right search box. Click **Install** on the **NuGet.Server** package as shown in the following image.
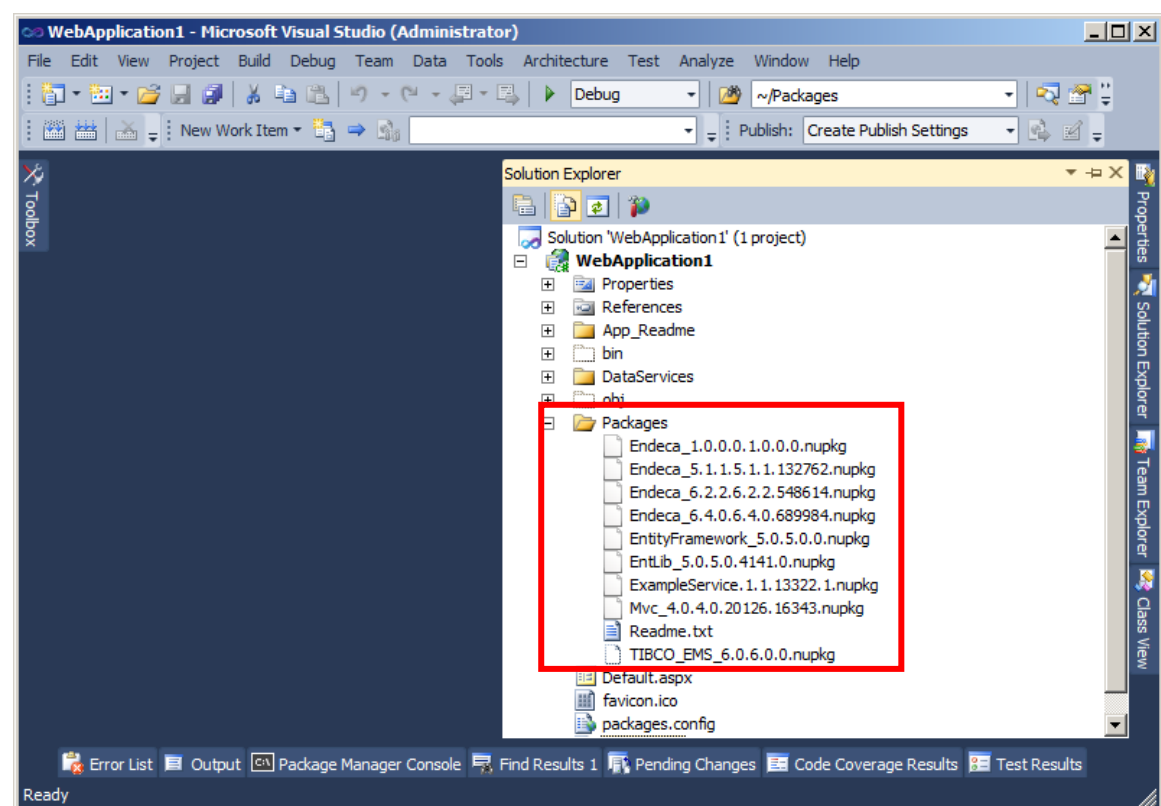
Step 3: Configure the Packages folder

Starting with NuGet.Server, you can configure the folder which contains your packages. The web.config file contains a new appSetting, named packagesPath. When the key is omitted or left blank, the packages folder is the default ~/Packages. You can specify an **absolute path**, or a **virtual path**.

```xml
<appSettings>
    <!-- Set the value here to specify your custom packages folder. -->
    <add key="packagesPath" value="C:\MyPackages" />
</appSettings>
```

Step 4: Add Packages to the Packages folder

The **NuGet.Server** package just converted your empty website into a site that's ready to serve up the OData package feed. Just add packages into the Packages folder

> **ⓘ** **If you want these packages to be published (such as when selecting Build - > Publish from the command prompt or TFS build)**

Starting with NuGet.Server, you can also add and delete packages to the lightweight feed using NuGet.exe. After installing the package, the web.config file will contain a new appSetting, named apiKey. When the key is omitted or left blank, pushing packages to the feed is disabled. Setting the apiKey to a value (ideally a strong password) enables pushing packages using NuGet.exe.

```xml
<appSettings>
    <!--
        Determines if an Api Key is required to push\delete packages from the server.
    -->
    <add key="requireApiKey" value="true" />

    <!-- Set the value here to allow people to push/delete packages from the server.
         NOTE: This is a shared key (password) for all users. -->
    <add key="apiKey" value="" />
</appSettings>
```

If however your server is already secured and \ or you do not require an api key to perform this operation, set the **requireApiKey** value to false.

Step 5: Host to IIS for your new Package Feed!

I can hit CTRL + F5 to run the site and it'll provide some instructions on what to do next.

Welcome to Tesco IGHS NuGet Server.

Please start getting the latest assemblies from NuGet feed instead of referring "Lib" folder.

Tesco IGHS Team is running NuGet.Server v2.5.40416.9020

Click here to view your packages.

Repository URLs

In the package manager settings, add the following URL to the list of Package Sources:

http://ukdbt66build05v.dev.global.tesco.org/Main/nuget

To enable pushing packages to this feed using the nuget command line tool (nuget.exe). Set the api key appSetting in web.config.

nuget push {package file} -s http://ukdbt66build05v.dev.global.tesco.org/Main/ {apikey}

Clicking on "here" shows the Packages feed of URL.

AppStore.Delivery.Contracts

Thursday, September 25, 2014, 1:14:35 PM | Tesco IGHS Team

AppStore.Delivery.WcfClient

Thursday, September 25, 2014, 1:10:59 PM | Tesco IGHS Team

AppStore.Delivery.Contracts

Thursday, September 25, 2014, 1:01:03 PM | Tesco IGHS Team

AppStore.Delivery.WcfClient

Thursday, September 25, 2014, 1:00:40 PM | Tesco IGHS Team

Displaying   100 / 100

● All   100

Sort by:
▼ Date
  Title
  Author

Filter by category:

NuGet.Server.Data...   100