**NuGet Overview**

NuGet is a Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects that use the .NET Framework. When you add a library or tool, NuGet copies files to your solution and automatically makes whatever changes are needed in your project, such as adding references and changing your *app.config* or *web.config* file. When you remove a library, NuGet removes files and reverses whatever changes it made in your project so that no clutter is left.

NuGet provides a quick and easy way to add features to an existing application while ensuring that those features are integrated into source control. After installing a library, you can commit changes to your source control repository, and then another developer can perform a get operation against the repository in order to start working with the new features.

Everything necessary to install a library or tool is bundled into a *package* (a *.nupkg* file). A package includes files to copy to your project and a manifest file that describes the contents of the package and what needs to be done to add or remove the library. Packages are bundled into *feeds* that Visual Studio accesses in order to present lists of available packages. There is an official feed that is the default source for NuGet, and you can contribute to that feed or create your own feeds.

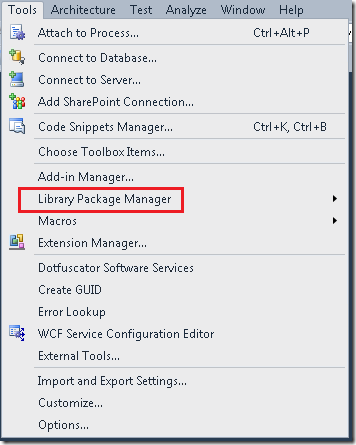
You work with packages by using a dialog box in Visual Studio or by using PowerShell cmdlets in a dedicated Visual Studio window. Both options are accessible from the Visual Studio main menu; you can also open the dialog box from a **Solution Explorer** context menu. The PowerShell cmdlets require PowerShell 2.0. Therefore, NuGet requires one of the following operating systems:

* Windows 7
* Windows Vista SP1
* Windows Server 2008
* Windows Server 2008 R2
* Windows Server 2003 SP2
* Windows XP SP3

NuGet runs in Visual Studio 2010 or Visual Web Developer 2010.

**Installing NuGet**

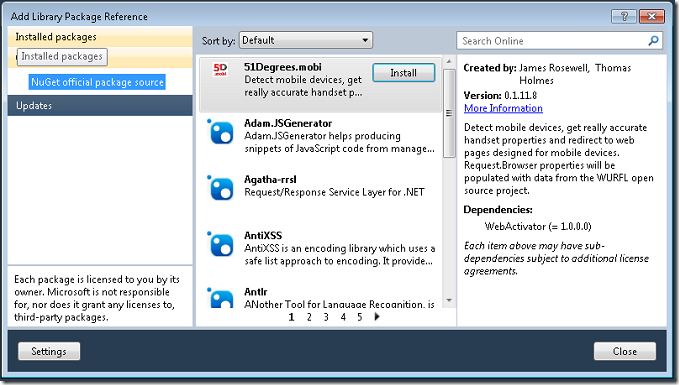
NuGet can be installed and updated using the Visual Studio **Extension Manager**. To check if your copy of Visual Studio already has the NuGet extension, look for **Library Package Manager** in the **Tools** menu of your copy of Visual Studio.

[](http://download.codeplex.com/Project/Download/FileDownload.aspx?ProjectName=nuget&DownloadId=187046)

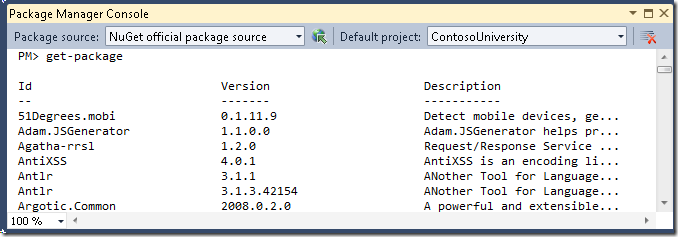
If your copy of Visual Studio does not already have the **Library Package Manager** (NuGet) extension, you can install it using the instructions in [Using the Extension Manager to Install the Library Package Manager (NuGet)](http://nuget.codeplex.com/wikipage?title=Using%20the%20Extension%20Manager%20to%20Install%20the%20Library%20Package%20Manager%20%28NuGet%29). Alternatively, you can install the NuGet Visual Studio extension from the [NuGet downloads page](http://nuget.codeplex.com/releases/view/52017).

**Working with NuGet Packages**

You can add, remove, and update packages using either the **Add Library Package Reference** dialog box or by using PowerShell command-line commands in the **Package Manager Console** window. The following illustration shows the **Add Library Package Reference** dialog box.

[](http://download.codeplex.com/Project/Download/FileDownload.aspx?ProjectName=nuget&DownloadId=187048)

The following illustration shows the **Package Manager Console** window.

[](http://download.codeplex.com/Project/Download/FileDownload.aspx?ProjectName=nuget&DownloadId=189332)

In most cases you can use either method to work with packages. However, some packages create new commands that can only be used in the command-line window. In addition, the **Add Library Package Reference** dialog box can only be used when a solution is open.

For information about how to use the **Add Library Package Reference** dialog box, see the following documents:

* [Finding and Installing a Package Using the Add Library Package Reference Dialog Box](http://nuget.codeplex.com/wikipage?title=Finding%20and%20Installing%20a%20Package%20Using%20the%20Add%20Library%20Package%20Reference%20Dialog%20Box)
* [Removing a Package Using the Add Library Package Reference Dialog Box](http://nuget.codeplex.com/wikipage?title=Removing%20a%20Package%20Using%20the%20Add%20Library%20Package%20Reference%20Dialog%20Box)
* [Updating a Package Using the Add Library Package Reference Dialog Box](http://nuget.codeplex.com/wikipage?title=Updating%20a%20Package%20Using%20the%20Add%20Library%20Package%20Reference%20Dialog%20Box)

For information about how to use the **Package Manager Console**, see the following documents:

* [Finding and Installing a Package Using the Package Manager Console](http://nuget.codeplex.com/wikipage?title=Finding%20and%20Installing%20a%20Package%20Using%20the%20Package%20Manager%20Console)
* [Removing a Package Using the Package Manager Console](http://nuget.codeplex.com/wikipage?title=Removing%20a%20Package%20Using%20the%20Package%20Manager%20Console)
* [Updating a Package Using the Package Manager Console](http://nuget.codeplex.com/wikipage?title=Updating%20a%20Package%20Using%20the%20Package%20Manager%20Console)

For information about how to create NuGet packages yourself, see the following documents:

* [Creating a Package](http://nuget.codeplex.com/documentation?title=Creating%20a%20Package)
* [Contributing Packages to the Official Feed](http://nupackpackages.codeplex.com/)
* [Hosting your Own Local and Remote NuGet Feeds](http://nuget.codeplex.com/wikipage?title=Hosting%20Your%20Own%20Local%20and%20Remote%20NuPack%20Feeds).