

# PACKAGE MANAGEMENT

## Introduction

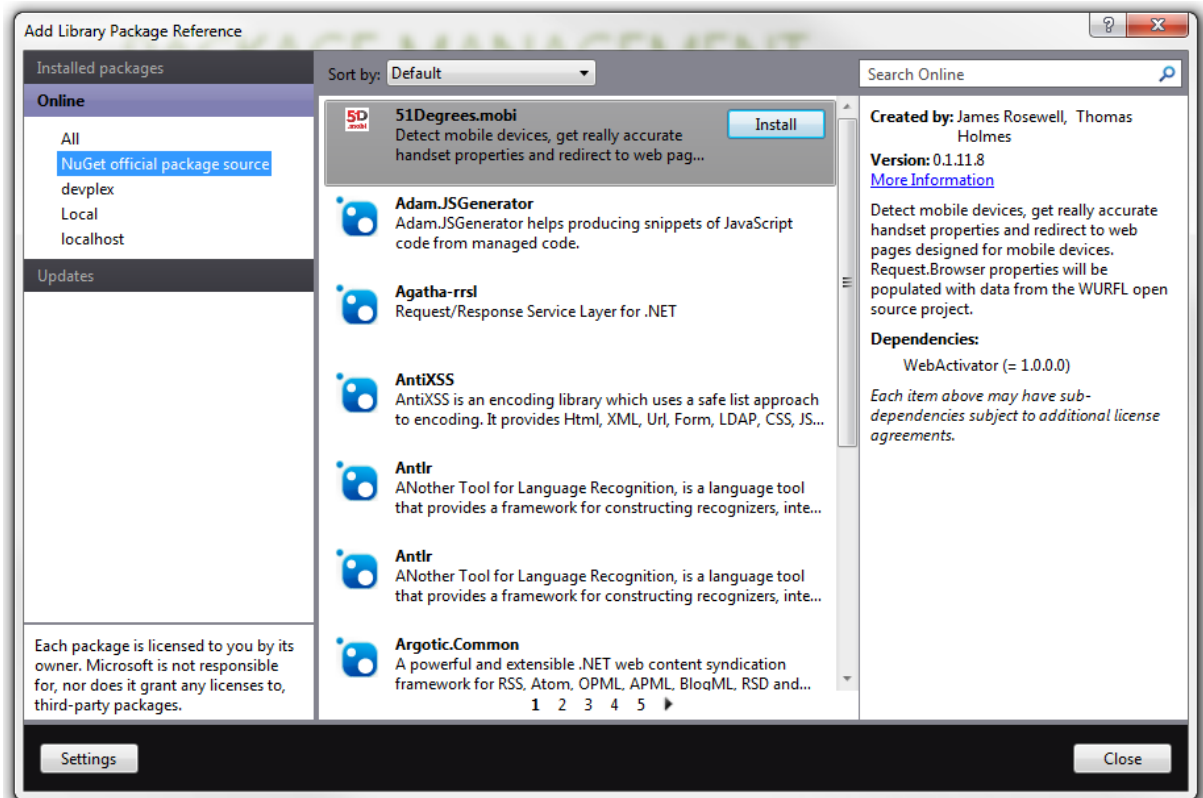
Almost every software project has dependencies to external libraries. During development these packages need to be added and, if a newer version of the external library is released, updated.

## Tooling

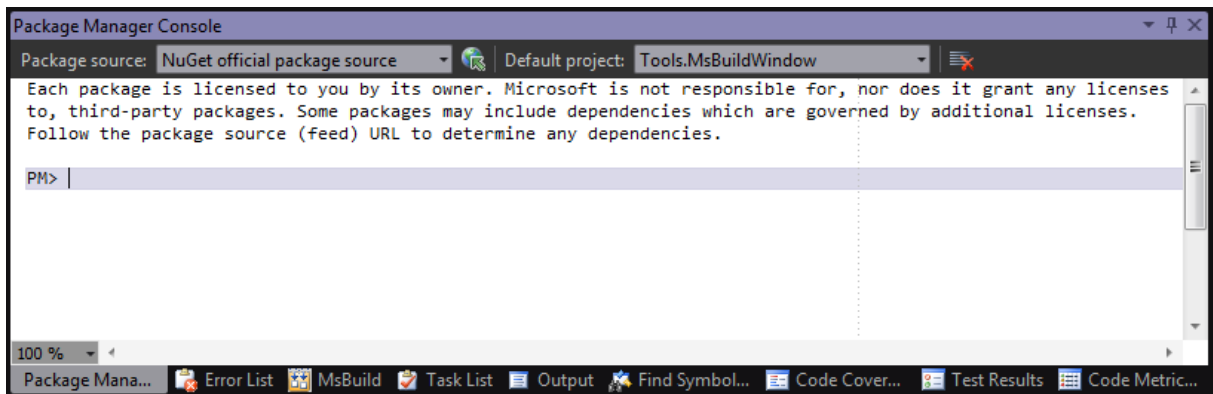
The solution is based on an open source project called NuGet [1] initiated and lead by Microsoft. NuGet provides a Visual Studio Add-in as well as a command line tool and MsBuild tasks. The source for these tools is either a local directory or a repository server – The NuGet Gallery Server.

## Visual Studio Add-In

The NuGet Tools for Visual Studio comes with two flavors. The first is available through a right click on a project and “Add Library Package Reference”:



The second one is a PowerShell window hosted in Visual Studio:



## NuGet PowerShell commands:

Name	Description
List-package	List available packages.
Install-package [packagename]	Installs a package.
Uninstall-package [packagename]	Uninstalls a package.

## Command Line Tool

The command line tools primary intent is to build packages. A package is defined by a \*nuspec-file.

```
<?xml version="1.0" encoding="utf-8"?>
<package xmlns="http://schemas.microsoft.com/packaging/2010/07/nuspec.xsd">
  <metadata>
    <id>devcoach.Xyz</id>
    <version>1.0.0</version>
    <authors>
      <author>devcoach</author>
      <author>Daniel Fisher</author>
    </authors>
    <description>A library to...</description>
    <language>en-US</language>
  </metadata>
  <files src="..\bin\NET40\devcoach.Xyz.dll" target="lib\Net40" />
  <files src="..\bin\NET40\devcoach.Xyz.xml" target="lib\Net40" />
</package>
```

*Listing 1: NuSpec sample*

A package can easily be created by double clicking it when the following registry file is imported:

#### REGEDIT4

```
[HKEY_CLASSES_ROOT\.nuspec]
@="NuGet.Specification"
"Content Type"="text/xml"

[HKEY_CLASSES_ROOT\NuGet.Specification]
"EditFlags"=dword:00010000
"BrowserFlags"=dword:00000008
@=""

[HKEY_CLASSES_ROOT\NuGet.Specification\shell]
@="Create"

[HKEY_CLASSES_ROOT\NuGet.Specification\shell\Create]
@="Create package"

[HKEY_CLASSES_ROOT\NuGet.Specification\shell\Create\command]
@="cmd.exe /k \"%1\" \"C:\\Program Files (x86)\\NuGet\\NuGet.exe\" \"%1\" \"%1\""
```

*Listing 2: Registry File*

To support for instance the install command there is currently only a fork [3] created by devcoach.

## MsBuild Task

The MsBuild task is currently only available through building the source code. The task can be used as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<Project
  DefaultTargets="PackageBuild"
  xmlns="http://schemas.microsoft.com/developer/msbuild/2003">

  <UsingTask
    TaskName="NuPack.MSBuild.NuPack"
    AssemblyFile="NuGet.MSBuild.dll" />

    <Target Name="PackageBuild">

      <NuPack.MSBuild.NuPack
        SpecFile="My.NuSpec"
        PackageDir="..\bin\"
        />

    </Target>

</Project>
```

*Listing 3: MsBuild Task*

# References

[1] NuGet: <http://nuget.codeplex.com>

[2] NuSpec Schema: <http://nuget.codeplex.com/documentation?title=Nuspec%20Format>

[3] NuGet GetCommand Fork: <https://hg01.codeplex.com/forks/lennybacon/getcommandfornuget>