



Rainer Stropek | time cockpit

NuGet

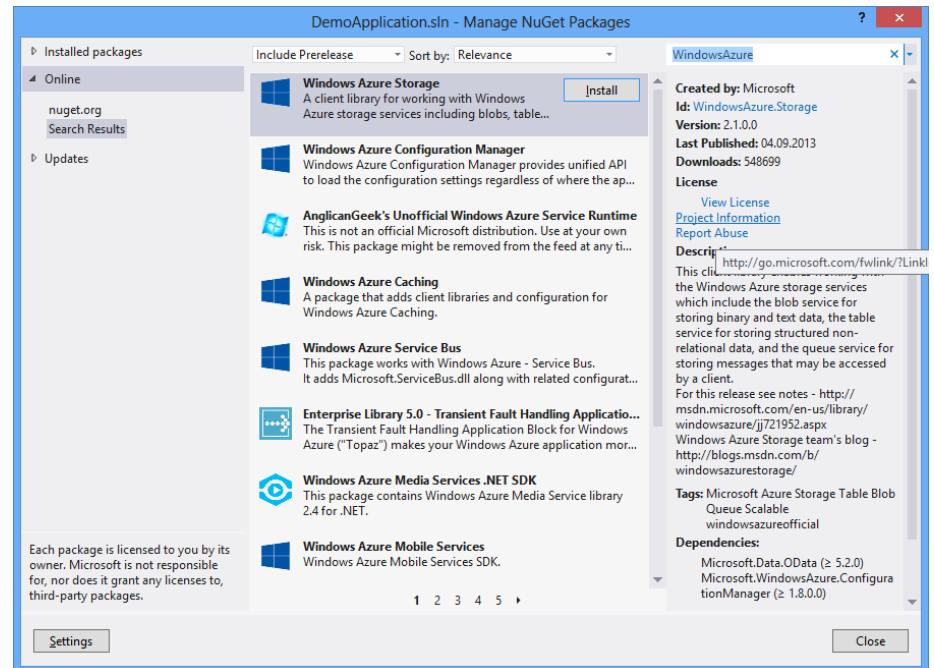
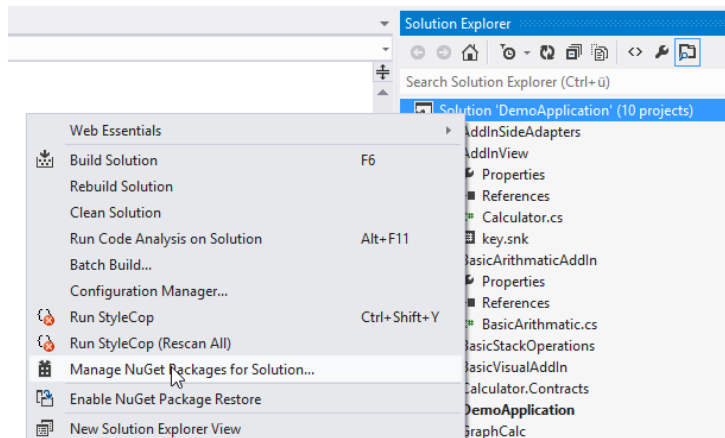
NuGet

Package Manager für die Microsoft-Entwicklungsplattform

Was ist NuGet?

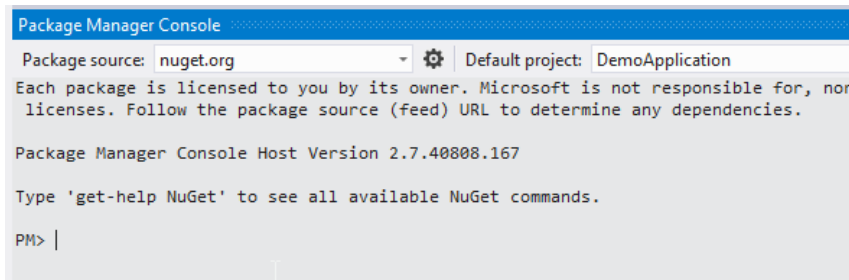
- ▶ Werkzeug zur einfachen Verteilung von Paketen (=Libraries und Tools)
- ▶ Alles Notwendige in einem Paket
 - Binaries für verschiedene Plattformen
 - Optional Symbole und Sourcen
 - Anpassungen am Projekt (z.B. Referenzen, Änderungen am app/web.config)
- ▶ UI-Integration in Visual Studio
 - Ab VS2010, eingeschränkt auch in Mono
 - Express-Editionen werden unterstützt
- ▶ <http://www.nuget.org>
<http://nuget.codeplex.com/> (Sourcecode)

NuGet in Visual Studio



NuGet in Visual Studio

- Package Manager Console
- PowerShell console in Visual Studio
- Automate Visual Studio and NuGet
- [NuGet PowerShell Reference](#)

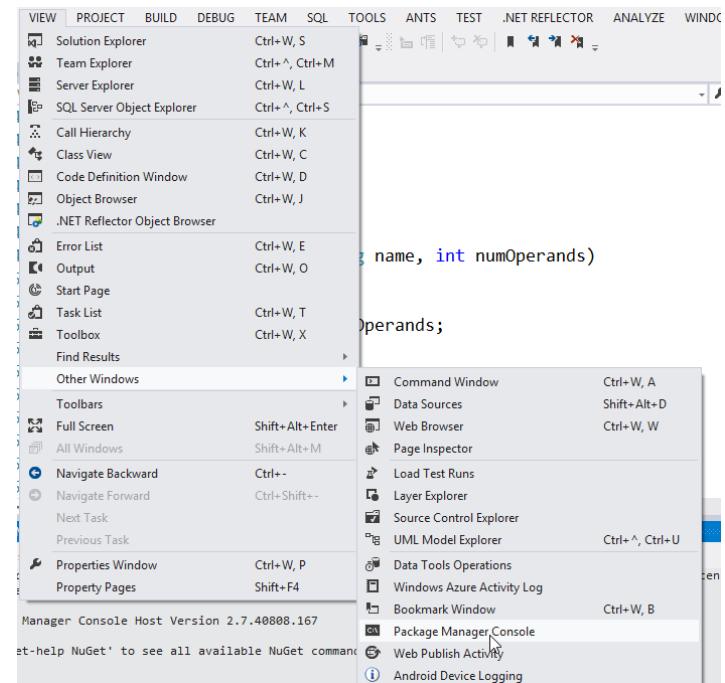


```
Package Manager Console
Package source: nuget.org | Settings | Default project: DemoApplication
Each package is licensed to you by its owner. Microsoft is not responsible for, nor
licenses. Follow the package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 2.7.40808.167

Type 'get-help NuGet' to see all available NuGet commands.

PM> |
```



NuGet Pakete erstellen

- ▶ Kommandozeilentool *nuget.exe*
 - Pakete erstellen (*Pack* Command)
 - Pakete veröffentlichen (*Push, Delete* Command)
 - Paket installieren (*Install, Restore, Update* Command)
 - Generieren eines *nuspec*-Files (*Spec* Command)
 - Wichtig für Buildautomatisierung
 - [Kommandozeilenreferenz](#)
- ▶ NuGet Package Explorer
 - Grafisches UI zur Erstellung/Bearbeitung von NuGet Paketen und *nuspec* Files
 - <http://npe.codeplex.com/>

```
<?xml version="1.0" encoding="utf-16"?>
<package xmlns="http://schemas.microsoft.com/packaging/2012/06/nuspec.xsd">
  <metadata>
    <id>CockpitFramework.Data</id>
    <version>$version$</version>
    <title>Cockpit Framework Data Layer</title>
    <authors>software architects gmbh</authors>
    <owners>software architects gmbh</owners>
    <requireLicenseAcceptance>>false</requireLicenseAcceptance>
    <description>...</description>
    <releaseNotes></releaseNotes>
    <dependencies>
      <group targetFramework=".NETFramework4.0">
        <dependency id="CockpitFramework.Dependencies"
          version="[$version$]" />
      </group>
      <group targetFramework="s15">
        <dependency id="CockpitFramework.Dependencies"
          version="[$version$]" />
      </group>
    </dependencies>
  </metadata>
</package>
```

Example

nuspec File

<dependencies>

```
<group targetFramework=".NETFramework4.0">
  <dependency id="log4net" version="[1.2.11]" />
  <dependency id="Microsoft.SqlServer.Compact.Private"
    version="[4.0.8482.1]" />
  <dependency id="AvalonEdit" version="[4.2.0.8783]" />
  <dependency id="ClosedXML" version="[0.68.1]" />
  <dependency id="DocumentFormat.OpenXml" version="[1.0]" />
  <dependency id="IronPython" version="[2.7.3]" />
  <dependency id="LumenWorks.Framework.IO" version="[1.0.0]" />
  <dependency id="Newtonsoft.Json" version="[5.0.6]" />
  <dependency id="WindowsAzure.Storage" version="[2.0.5.1]" />
  <dependency id="Microsoft.Bcl.Async" version="[1.0.16]" />
</group>

<group targetFramework="s15">
  ...
</group>
</dependencies>
```

Example

nuspec File

Version range syntax

```
1.0 = 1.0 ≤ x
(,1.0] = x ≤ 1.0
(,1.0) = x < 1.0
[1.0] = x == 1.0
(1.0) = invalid
(1.0,) = 1.0 < x
(1.0,2.0) = 1.0 < x < 2.0
[1.0,2.0] = 1.0 ≤ x ≤ 2.0
empty = latest version.
```


Example

nuspec File

```
<files>
  <!-- net4 -->
  <file src=".\$configuration$\TimeCockpit.Common.dll"
    target="lib\net4" />
  <file src=".\$configuration$\TimeCockpit.Data.dll"
    target="lib\net4"/>
  ...

  <!-- s15 -->
  <file src=".\SL\$configuration$\TimeCockpit.Common.dll"
    target="lib\s15" />
  <file src=".\SL\$configuration$\TimeCockpit.Data.dll"
    target="lib\s15" />
  ...

  <!-- include source code for symbols -->
  <file src=".\...\*.cs" target="src\TimeCockpit.Common" />
  <file src=".\...\*.cs" target="src\TimeCockpit.Data" />
</package>
```

Folder Structure

For Details see [NuGet Docs](#)

```
\lib
  \net11
    \MyAssembly.dll
  \net20
    \MyAssembly.dll
  \net40
    \MyAssembly.dll
  \sl40
    \MyAssembly.dll
  \content
    \net11
      \MyContent.txt
    \net20
      \MyContent20.txt
    \net40
    \sl40
      \MySilverlightContent.html
  \tools
    init.ps1
    \net40
      install.ps1
      uninstall.ps1
    \sl40
      install.ps1
      uninstall.ps1
```

Versioning Notes

- ▶ Things to remember
 - NuGet never installs assemblies machine-wide (i.e. not in GAC)
 - You cannot have multiple versions of the same DLL in one AppDomain
- ▶ DLL Hell
 - Policy too loose: Problems with breaking changes
 - Policy too tight: Problems with library having dependencies on other libraries (e.g. ANTLR and ASP.NET MVC, everyone depending on Newtonsoft JSON)
- ▶ For Library Publishers: [SemVer](#)
 - X.Y.Z (Major.Minor.Patch)
 - Rethink your strong naming policies

```
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="mscorlib"
        publicKeyToken="b03f5f7f11d50a3a" culture=""/>
      <bindingRedirect
        oldVersion="0.0.0.0-65535.65535.65535.65535"
        newVersion="1.0.3300.0"/>
    </dependentAssembly>
  </assemblyBinding>
</runtime>
```

Binding Redirects

Note: NuGet can generate
this for you
Add-BindingRedirect Command
See [online reference](#) for details

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="SomePackage"
    version="2.1.0"
    allowedVersions="[2,3)" />
</packages>
```

Versioning

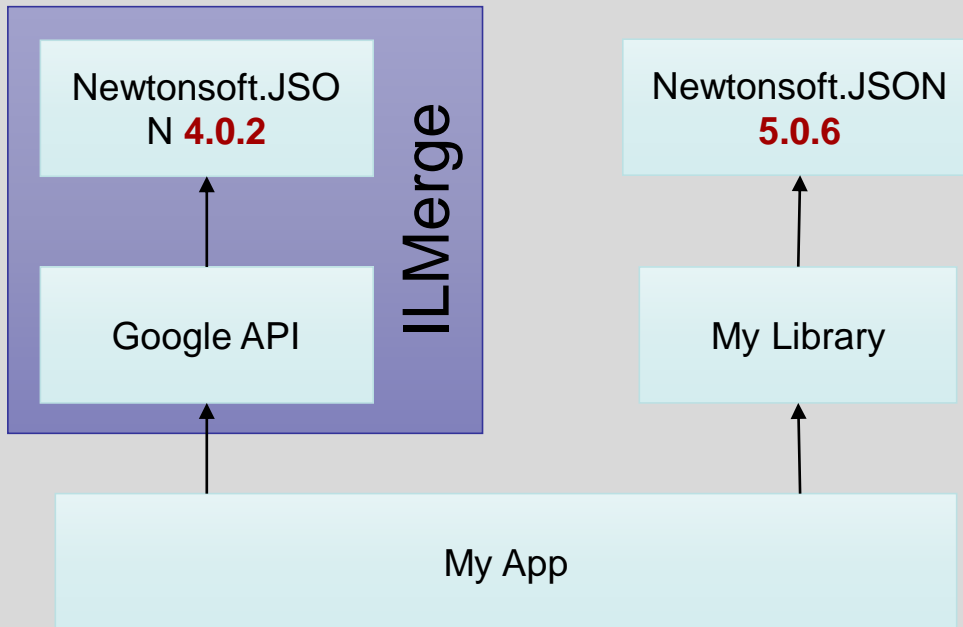
Constraings in packges.config

Manual editing necessary

ILMerge

Solving version conflicts

[Microsoft Download](#)



ILMerge

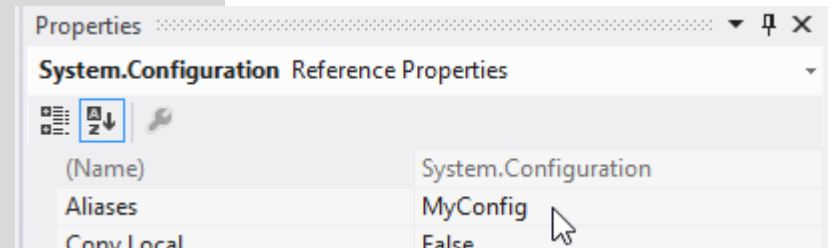
```
"Assemblies\Google.Apis.Authentication.OAuth2.dll"  
"Assemblies\Google.Apis.dll"  
"Assemblies\Google.Apis.Latitude.v1.dll"  
"Assemblies\DotNetOpenAuth.dll" "Assemblies\log4net.dll"  
"Assemblies\Newtonsoft.Json.Net35.dll"  
/out:"c:\temp\Google.Apis.All.dll" /lib:"Lib,,
```

```
extern alias MyConfig;  
using Conf = MyConfig::System.Configuration;  
  
namespace MyTinyMvvmToolkit  
{  
    public class NotificationObject  
    {  
        public void ReadConfiguration()  
        {  
            var setting =  
                Conf.ConfigurationManager.AppSettings["MyDB"];  
        }  
    }  
}
```

ILMerge

Solving version conflicts

C# [extern alias](#)



```
<?xml version="1.0" encoding="utf-16"?>
<package xmlns="http://schemas.microsoft.com/packaging/2012/06/nuspec.xsd">
  <metadata>...</metadata>

  <files>
    <file src="content\app.config.transform"
      target="content\" />
    <file src="content\TimeCockpitInitialization.cs.pp"
      target="content\" />
  </files>
</package>
```

```
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
  </startup>
  <system.data>
    <DbProviderFactories>
      <remove invariant="System.Data.SqlClient.4.0"/>
      <add name="Microsoft SQL Server Compact Data Provider 4.0"
        invariant="System.Data.SqlClient.4.0"
        description=".NET Framework Data Provider for Microsoft SQL Server Compact"
        type="System.Data.SqlClient.SqlCeProviderFactory, System.Data.SqlClient,
        Version=4.0.0.1, Culture=neutral, PublicKeyToken=89845dcd8080cc91"/>
    </DbProviderFactories>
  </system.data>
</configuration>
```

Content Files

New in NuGet 2.6: [XDT](#)


```
...
namespace $rootnamespace$
{
    using System;

    /// <summary>
    /// Class taking care of cockpit framework initialization
    /// </summary>
    public class TimeCockpitInitialization
    {
        ...
    }
}
```

Content Files

[Sourcecode Transformations](#) in .cs.pp
File

Available properties see
[MSDN](#)

User PowerShell scripts to
modify project properties
[NuGet Docs](#)

Nuget in Practice

Nuspec files

- Files

- Dependencies

- Build

Packages in NuGet Explorer

Packages in VS

Demo

Nuget at software architects

Publishing NuGet Packages

- ▶ <http://www.nuget.org>
Public NuGet Feed
- ▶ Create your private feed
 - File system
 - Private NuGet Server
 - For details see [NuGet Help](#)
- ▶ Use a NuGet SaaS like *MyGet*
<http://www.myget.org/>