

Creating packages

To deploy applications using Deployment Manager, you need to create packages and publish them to Deployment Manager Server's package feed. Packages contain database and application data. Deployment Manager deploys NuGet (.nupkg) packages, a ZIP-based file format. You can deploy packages from the web interface.



For more information, see [NuGet Overview](#) (NuGet documentation).

Packages can be made:

- From the command line using [RGPublish.exe](#).
- From within SQL Server Management Studio, using the [SQL Server Management Studio add-in](#) (databases only).
- From within Visual Studio, using the [Visual Studio extension](#) (applications only).
- By a continuously integrated build process.

Learn more about packages and packaging

- [Packaging databases](#)
- [Packaging applications](#)
- [Working with package feeds](#)
- [Automating package creation](#)

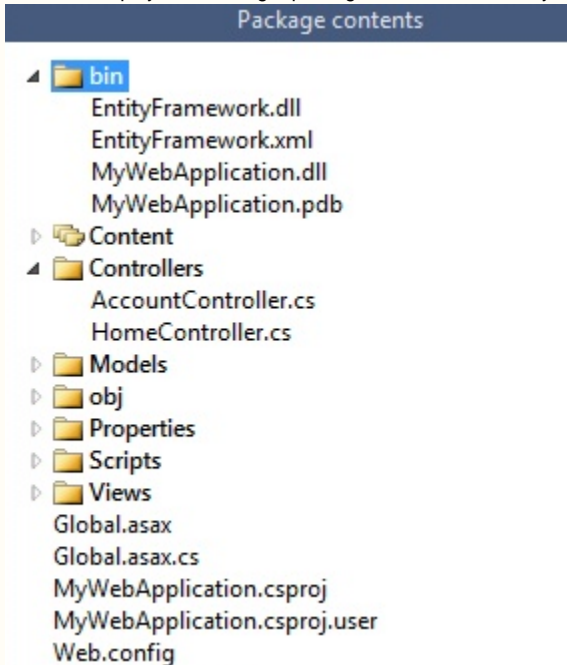
What packages should contain

An application package should contain all of the executables, DLLs, configuration files, installation scripts, and anything else the application needs to run.

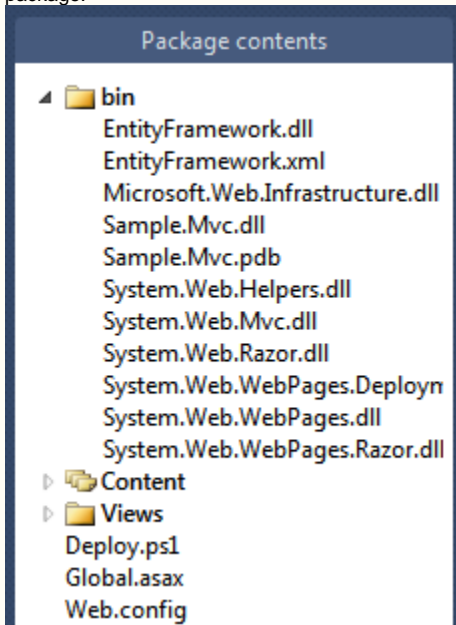


Note

- NuGet encourages you to put compiled binaries into a *lib* folder, scripts into a *tools* folder, and everything else into a *content* folder. However, Deployment Manager packages should look exactly as you want the files to be extracted on disk. For example:



- NuGet packages can't have dependencies. Any libraries you need to use must be included in the standard binaries folder of your package:



PowerShell

Deployment Manager automatically replaces appSettings and connectionStrings, runs XML config transforms, and updates IIS websites during deployment. If you want Deployment Manager to perform further deployment actions, you can [include PowerShell scripts in your package](#).