**How to create offline chocolatey installation packages**

This is a general guide on how to create an offline chocolatey package for installation. I've altered chocolatey's package template files to the absolute bare minimum necessary to build a package, as there is a lot of extraneous information in the default template. You can get this template package file package at <https://github.com/Michael-Lindauer/Chocolatey-Resources/blob/master/Barebones%20Package%20Template.zip>

**Prerequisites:**

1. Chocolatey server setup is complete
2. Chocolatey is installed on a test client

**Directions:**

1. Download and extract the "Barebones Package Template.zip" onto the **test client's** desktop. Go into this folder and begin renaming/altering the following:
   1. NAME\_OF\_PACKAGE folder - Rename
   2. NAME\_OF\_PACKAGE.nuspec - Rename and open with Notepad. Edit the options inside the <metadata> section. Save and go into the tools folder. Paste your software in the tools folder
   3. chocolateyinstall.ps1 - Edit the options in this file with Powershell\_ISE. silentArgs are currently set for most .msi packages. Consult your choice of software's official documentation for command line installation switches. Or use google and search "how to install <X software> with command line"
   4. chocolateyuninstall.ps1 - Edit and change software installer and its extension. This should only be used if chocolatey is unable to create an uninstaller automatically when the package installs on the client, but it’s worth editing the two lines just to not deal with repacking the package
2. Run Powershell as an Admin, change directory to your now ready package ( CD 'C:\Users\YOUR\_USER\Desktop\Barebones Package Template\YOUR\_PACKAGE' ), and run choco pack. This will create a nupkg in your folder
3. Test if your package will install and uninstall properly by running choco install .\YOUR\_PACKAGE.nupkg -y. This will verify that your installation of the software will work. Fix any errors in your nuspec and ps1 files, delete the old nupkg and repack. Additionally, verify choco uninstall YOUR\_PACKAGE works as well to remove the software
4. Once you've verified your software install/uninstall works, Run choco push .\YOUR\_PACKAGE.nupkg --source="http://YOUR\_REPO/chocolatey" -k="YOUR\_API\_KEY" --force to push your package to your repository. Afterwards, **verify the software will install on the client from your repository** by running choco install YOUR\_PACKAGE -y. If your software is unable to install, verify that you've put the software into the tools folder prior to packing

**Creating updates for existing packages**

1. Repeat steps 1-4 for the updated software, remembering to **increment the version number in the nuspec file.** Once the updated package is packed and uploaded to your repository, run choco upgrade all -y
   1. Chocolatey will look for any incremented versions of packages it has installed on the client that runs the command, looking at all sources by default. You can specify with the --source switch for a specific repo
   2. It is highly recommended to have a Production Repository and a Test Repository
   3. If you intend to deploy this at scale, it is highly recommended to get the C4B (Chocolately for Business) license, as it enables you to rapidly create packages/updates vs the manual build process described in this guide