How do I update all NuGet packages at once with the dotnet CLI?

Asked 2 years, 10 months ago Active 8 months ago Viewed 4k times

17

I'm trying to update all NuGet packages for a solution in VS Code (using Mac). Is there a way to achieve that in VS code or for a specific project.json file? At the moment I'm going one by one but I would have thought there is either an extension or a feature that does that for you?

nuget visual-studio-code .net-core

★ 3

edited Aug 8 '18 at 13:48
Anthony Mastrean
14.8k 19 85 153

asked Dec 10 '16 at 21:57
Carlos Torrecillas
1,231 1 11 32

7 You should accept an answer below to close the question, while most people in 2019 should use tools like `dotnet-outdated` github.com/jerriep/dotnet-outdated – Lex Li Feb 2 at 21:33

5 Answers

6

Here's a shell script and a powershell script that will do this

```
#!/bin/bash
regex='PackageReference Include="([^"]*)" Version="([^"]*)"'
find . -name "*.*proj" | while read proj
do
  while read line
  do
    if [[ $line =~ $regex ]]
    then
      name="${BASH_REMATCH[1]}"
      version="${BASH_REMATCH[2]}"
      if [[ $version != *-* ]]
      then
        dotnet add $proj package $name
      fi
    fi
  done < $proj
done
```

```
$regex = [regex] 'PackageReference Include="([^"]*)" Version="([^"]*)"'
ForEach ($file in get-childitem . -recurse | where {$_.extension -like "*proj"})
{
  $proj = $file.fullname
  $content = Get-Content $proj
  $match = $regex.Match($content)
  if ($match.Success) {
    $name = $match.Groups[1].Value
    $version = $match.Groups[2].Value
    if ($version -notin "-") {
      iex "dotnet add $proj package $name"
    }
  }
}
```

Should also mention Paket as a fantastic alternative package manager that supports update:

https://fsprojects.github.io/Paket/index.html

```
dotnet tool install paket --tool-path .paket
```

edited Feb 28 at 7:36

answered Sep 18 '18 at 13:36
Jon Canning
1,092 12 13

Looks like that powershell script only returns the first regex match for every file. I will add an alternative solution below. – Rolf Wessels Dec 14 '18 at 12:01

4

Based on Jon Canning's powershell solution. I fixed a small bug where only the first dependency was being updated and not all the dependencies for the project file.

```
$regex = 'PackageReference Include="([^"]*)" Version="([^"]*)"'

ForEach ($file in get-childitem . -recurse | where {$_.extension -like "*proj"})
{
    $packages = Get-Content $file.FullName |
        select-string -pattern $regex -AllMatches |
        ForEach-Object {$_.Matches} |
        ForEach-Object {$_.Groups[1].Value.ToString()}|
        sort -Unique

    ForEach ($package in $packages)
    {
        write-host "Update $file package :$package"  -foreground 'magenta'
        $fullName = $file.FullName
        iex "dotnet add $fullName package $package"
    }
}
```

answered Dec 14 '18 at 12:17

Rolf Wessels

605 7 11

0

I created a cake build task to do the same. See below:

```
Task("Nuget-Update")
    .Does(() =>
{
    var files = GetFiles("./**/*.csproj");
    foreach(var file in files)
    {
        var content = System.IO.File.ReadAllText(file.FullPath);
        var matches = System.Text.RegularExpressions.Regex.Matches(content,
@"PackageReference Include=""([^""]*)"" Version=""([^""]*)""");
        Information($"Updating {matches.Count} reference(s) from {file.GetFilename()}");
        foreach (System.Text.RegularExpressions.Match match in matches) {
            var packageName = match.Groups[1].Value;
            Information($"  Updating package {packageName}");
            var exitCode = StartProcess("cmd.exe",
                new ProcessSettings {
                    Arguments = new ProcessArgumentBuilder()
                        .Append("/C")
                        .Append("dotnet")
                        .Append("add")
                        .Append(file.FullPath)
                        .Append("package")
                        .Append(packageName)
                }
            );
        }
    }
});
```

answered Feb 2 at 21:16

Roemer
1,117 1 14 24

0

Based on Jon Caning answer, I've written this small bash script to add in .bashrc (or just change a bit to keep it in its bash file)

```
function read_solution() {
    echo "Parsing solution $1"

    while IFS='' read -r line || [[ -n "$line" ]]; do
            if [[ $line =~ \"([^\"]*.csproj)\" ]]; then
                    project="${BASH_REMATCH[1]}"

                    read_project "$(echo "$project"|tr '\\' '/')"
            fi
    done < "$1"
}

function read_project() {
    echo "Parsing project $1"
    package_regex='PackageReference Include="([^"]*)" Version="([^"]*)"'

    while IFS='' read -r line || [[ -n "$line" ]]; do
            if [[ $line =~ $package_regex ]]; then
                    name="${BASH_REMATCH[1]}"
                    version="${BASH_REMATCH[2]}"

                    if [[ $version != *-* ]]; then
                            dotnet add "$1" package "$name"
                    fi
            fi
    done < $1
}

function dotnet_update_packages() {
    has_read=0

    if [[ $1 =~ \.sln$ ]]; then
            read_solution "$1"
            return 0
    elif [[ $1 =~ \.csproj$ ]]; then
            read_project "$1"
            return 0
    elif [[ $1 != "" ]]; then
            echo "Invalid file $1"
            return 1
    fi


    for solution in ./*.sln; do
            if [ ! -f ${solution} ]; then
                    continue
            fi

            read_solution "${solution}"
            has_read=1
    done

    if [[ $has_read -eq 1 ]]; then
            return 0
    fi

    for project in ./*.csproj; do
            if [ ! -f ${project} ]; then
                    continue
            fi

            read_project "${project}"
    done
}
export -f dotnet_update_packages
```

To use it, either run it without parameter in a folder with a solution, it would first look for all solution files in the current folder and run for all csproj files referenced in those (it might need to be changed if you work with something else than c#).

If no solution is found, it looks for all csproj files in the current directory and run for those.

You can also pass a .sln or .csproj file as argument:

```
dotnet_update_packages mysolution.sln
dotnet_update_packages myproject.csproj
```

I'm not a bash expert so I'm sure it can be improved

answered Oct 11 '18 at 16:29

Boudin

51 2

-5

You can try **NuGet Package Manager** for VS Code:

https://marketplace.visualstudio.com/items?itemName=jmrog.vscode-nuget-package-manager

answered Aug 31 '18 at 5:28

Lubos Voska
157 1 10

1 This tool does not support what the questioner asked for. – Po-ta-toe Dec 5 '18 at 10:15