

Side-by-Side .NET Core Installations

Setup

This describes a setup where .NET Core 3.1 was installed first, then .NET Core 2.2 was installed later.

Since .NET Core 3.1 is the newest version, issuing a command like this:

```
dotnet new console -o MyConsoleApp
```

...creates a console application that targets v3.1. If you look at the .csproj file, you'll see this:

```
<TargetFramework>netcoreapp3.1</TargetFramework>
```

Project Level

There are a couple of ways to target v2.2. Different templates support different methods.

For a console app, simply edit the .csproj to reflect the version you want:

```
<TargetFramework>netcoreapp2.2</TargetFramework>
```

...and then restore:

```
dotnet restore
```

Some templates, like **webapi** and **mvc**, allow you to target an alternate version when you create the project:

```
dotnet new webapi -f netcoreapp2.2 -o MyWebService
```

Framework Level

You can also control the default framework used by the CLI. We'll continue using our example described above, where .NET Core 3.1 was installed first, then .NET Core 2.2 was installed later.

Since 2.2 was installed last, that makes it the default SDK:

```
dotnet --version
```

Result:

```
2.2.108
```

To change the default SDK version, first list your installed versions, and note the version you want to use as the default:

```
dotnet --list-sdks
```

Result:

```
2.2.108  
3.1.301
```

(We'll change our default to 3.1.301)

Create a global.json file. The setting in the global.json file will affect the dotnet cli when it's invoked in any subdirectory under the location of the global.json file, so you'll probably want to create it in a location like \$HOME (for Linux), or C:\ (for Windows):

```
dotnet new globaljson
```

Then, edit the new global.json file, and update the "version" value to reflect the version you wish to be the default:

```
{  
  "sdk": {  
    "version": "3.1.301"  
  }  
}
```

Then, the version reported by the CLI will match:

```
dotnet --version
```

Result:

```
3.1.301
```

More Information

[How to install multiple versions of .NET Core side by side on ubuntu - Stack Overflow](#)

[Side by Side user scoped .NET Core installations on Linux with dotnet-install.sh - Scott Hanselman's Blog](#)

[Switching between .NET Core SDK Versions](#)

[dotnet](#)

From:
<https://kbase.devtoprd.com/> - **Knowledge Base**

Permanent link:
https://kbase.devtoprd.com/doku.php?id=side_by_net_core_installations

Last update: **2024/08/11 18:08**

