

dotnet new

<code>dotnet new sln</code>	Solution file
<code>dotnet new console</code>	Console application
<code>dotnet new classlib</code>	Class library
<code>dotnet new mvc</code>	ASP.NET Core Web App (Model-View-Controller)
<code>dotnet new xunit</code>	xUnit test project
<code>dotnet new -l</code>	Obtain a list of the available templates

dotnet sln

<code>dotnet sln list</code>	List all projects in a solution file
<code>dotnet sln todo.sln add</code>	Add a C# project to a solution
<code>dotnet sln todo-app/todo-app.csproj remove</code>	Remove a C# project from a solution
<code>dotnet sln todo.sln add **/*.csproj</code>	Add multiple C# projects to a solution using a globbing pattern

dotnet add

<code>dotnet add package Newtonsoft.Json</code>	Add <i>Newtonsoft.Json</i> NuGet package to a project
<code>dotnet add reference lib1/lib1.csproj lib2/lib2.csproj</code>	Add multiple project references to the project in the current directory
<code>dotnet add reference app/app.csproj **/*.csproj</code>	Add multiple project references using a globbing pattern on Linux/Unix

dotnet build

<code>dotnet build</code>	Build a project and all of its dependencies
<code>dotnet build --configuration Release</code>	Build a project and its dependencies using Release configuration
<code>dotnet build --runtime ubuntu.16.04-x64</code>	Build a project and its dependencies for a specific runtime (in this example, Ubuntu 16.04)
Starting with .NET Core 2.0, you don't have to run <code>dotnet restore</code> because it's run implicitly.	

dotnet run

<code>dotnet run</code>	Run the project in the current directory
<code>dotnet run --project ./projects/proj1/proj1.csproj</code>	Run the specified project
<code>dotnet myapp.dll</code>	Run a framework-dependent app named <i>myapp.dll</i>

dotnet clean

<code>dotnet clean</code>	Clean the output of a project
<code>dotnet clean --configuration Release</code>	Clean a project built using the Release configuration
Only the outputs created during the build are cleaned. Both intermediate (<i>obj</i>) and final output (<i>bin</i>) folders are cleaned.	

dotnet publish

<code>dotnet publish</code>	Publish the project in the current directory
<code>dotnet publish ~/projects/app1/app1.csproj</code>	Publish the application using the specified project file
The <code>dotnet publish</code> command's output is ready for deployment to a hosting system (for example, a server, PC, Mac, laptop) for execution.	

dotnet ef

<code>dotnet ef migrations add</code>	Add a new migration
<code>dotnet ef migrations list</code>	List available migrations
<code>dotnet ef migrations remove</code>	Remove the last migration
<code>dotnet ef migrations script</code>	Generate a SQL script from migrations
<code>dotnet ef database update</code>	Update the database to a specified migration
<code>dotnet ef database drop</code>	Drop the database
<code>dotnet ef dbcontext list</code>	List available DbContext types
<code>dotnet ef dbcontext info</code>	Get information about a DbContext type
<code>dotnet ef dbcontext scaffold</code>	Scaffolds a DbContext and entity types for a database

dotnet pack

<code>dotnet pack</code>	Build the project and create NuGet packages
<code>dotnet pack --no-build -output nupkgs</code>	Pack the project in the current directory into the <i>nupkgs</i> folder and skip the build step
<code>dotnet pack /p:PackageVersion=2.1.0</code>	Set the package version to <i>2.1.0</i> with the <i>PackageVersion</i> MSBuild property

dotnet nuget

<code>dotnet nuget locals -l all</code>	Display the paths of all the local cache directories
<code>dotnet nuget push foo.nupkg -k 4003d786-cc37-4004-bfdf-c4f3e8ef9b3a</code>	Push <i>foo.nupkg</i> to the default push source, specifying an API key
<code>dotnet nuget delete Microsoft.AspNetCore.Mvc 1.0 --non-interactive</code>	Delete version 1.0 of package <i>Microsoft.AspNetCore.Mvc</i> , not prompting user for credentials or other input

dotnet remove

<code>dotnet remove package Newtonsoft.Json</code>	Remove <i>Newtonsoft.Json</i> NuGet package from a project in the current directory
<code>dotnet remove reference lib/lib.csproj</code>	Remove a project reference from the current project
<code>dotnet remove app/app.csproj reference **/*.csproj</code>	Remove multiple project references using a glob pattern on Unix/Linux

etc.

<code>dotnet help</code>	Show more detailed documentation online for the command
<code>dotnet migrate</code>	Migrate a Preview 2 .NET Core project to a .NET Core SDK 1.0 project
<code>dotnet msbuild</code>	Provides access to a fully functional MSBuild
<code>dotnet test</code>	Run the tests in the project in the current directory
<code>dotnet list reference</code>	List the project references for the project in the current directory

Environment variables

DOTNET_PACKAGES

The primary package cache.

DOTNET_SERVICING

Specifies the location of the servicing index to use by the shared host when loading the runtime.

DOTNET_CLI_TELEMETRY_OPTOUT

Specifies whether data about the .NET Core tools usage is collected and sent to Microsoft.

DOTNET_MULTILEVEL_LOOKUP

Specifies whether .NET Core runtime, shared framework, or SDK are resolved from the global location.

DOTNET_ROLL_FORWARD_ON_NO_CANDIDATE_FX

Disables minor version roll forward. For more information, see Roll forward.