Cheatography

dotnet cli Cheat Sheet

by Olcay Bayram (oba) via cheatography.com/64093/cs/16212/

dotnet new

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

dotnet sln

`dotnet sln list`	List all projects in a solution file
`dotnet sln todo.sln add todo-a pp/ tod o-a pp.c spr oj`	Add a C# project to a solution
`dotnet sln todo.sln remove todo-a pp/ tod o-a pp.c - sproj`	Remove a C# project from a solution
`dotnet sln todo.sln add **/*.c sproj`	Add multiple C# projects to a solution using a globbing pattern

dotnet add

dotnet build

`dotnet build`	Build a project and all of its depend-encies
`dotnet build --conf igu ration Release`	Build a project and its depend-encies using Release config-uration
`dotnet build --runtime ubuntu.16.04-x64`	Build a project and its depend-encies for a specific runtime (in this example, Ubuntu 16.04)

Starting with .NET Core 2.0, you don't have to run `dotnet restore` because it's run implicitly.

dotnet run

`dotnet run`	Run the project in the current directory
`dotnet run --project ./proj ect s/p roj 1/p roj 1. c sproj`	Run the specified project
`dotnet myapp.dll`	Run a framework-dependent app named `myapp.dll`

dotnet clean

dotnet publish

`dotnet publish`	Build a project and all of its depend-encies
`dotnet publish ~/proj ect s/a pp1 sproj`	Build a project and its depend-encies using Release config-uration

The `dotnet publish` command's output is ready for deployment to a hosting system (for example, a server, PC, Mac, laptop) for execution.

dotnet ef

`dotnet add package Newton sof t.Json`	Add *Newton-soft.Json* NuGet package to a project
`dotnet add reference lib1/l ib1.csproj lib2/l ib2.cspr oj`	Add multiple project references to the project in the current directory.
`dotnet add app/ap p.c sproj reference**/*.c sproj`	Add multiple project references using a globbing pattern on Linux/Unix

`dotnet clean`	Clean the output of a project
`dotnet clean --conf igu ration Relea se`	Clean a project built using the Release config-uration

Only the outputs created during the build are cleaned. Both intermediate (*obj*) and final output (*bin*) folders are cleaned.

`dotnet ef migrations add`	Add a migra
`dotnet ef migrations list`	List availa migra
`dotnet ef migrations remove`	Remo the la migra
`dotnet ef migrations script`	Gene a SQ script migra
`dotnet ef database update`	Upda the datab to a speci migra
`dotnet ef database drop`	Drop datab
`dotnet ef dbcontext list`	List availa DbC types
`dotnet ef dbcontext info`	Get inforr ation abou DbC type
`dotnet ef dbcontext scaffold`	Scaff a DbC and e types datab

By **Olcay Bayram** (oba)
cheatography.com/oba/
en.otomatikmuhendis.com

Published 30th June, 2018.
Last updated 30th June, 2018.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Cheatography

dotnet cli Cheat Sheet

by Olcay Bayram (oba) via cheatography.com/64093/cs/16212/

dotnet pack

```
dotnet pack
```

Build the project and create NuGet packages

```
dotnet pack --no-build --output nupkgs
```

Pack the project in the current directory into the *nupkgs* folder and skip the build step

```
dotnet pack /p:Pac kag eVe rsi on= 2.1 .0
```

Set the package version to *2.1.0* with the *Packag-eVersion* MSBuild property

dotnet nuget

dotnet remove (cont)

```
dotnet remove app/ap p.c sproj reference roj
```

etc.

```
dotnet **/*.sp
```

Remove multiple project references using a glob pattern on Unix/Linux

```
dotnet --help
```

Show more detailed documentation online for the command

```
dotnet migrate
```

Migrate a Preview 2 .NET Core project to a .NET Core SDK 1.0 project

```
dotnet msbuild
```

Provides access to a fully functional MSBuild

```
dotnet test
```

Run the tests in the project in the current directory

```
dotnet list reference
```

List the project references for the project in the current directory

Environment variables

`dotnet nuget locals -l all`	Display the paths of all the local cache direc- ories
`dotnet nuget push foo.nupkg -k 4003d7 86- cc3 7-4 004 -bf df- c4f 3 e8 ef9b3a`	Push *foo.nupkg* to the default push source, specifying an API key
`dotnet nuget delete Micros oft.As pNe tCo re.Mvc 1.0 --non- int era ctive`	Delete version 1.0 of package *Micros- oft.As- pNetCo- re.Mvc*, not prompting user for creden- tials or other input

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_C- ANDIDATE_FX

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

dotnet remove

`dotnet remove package Newton sof t.Jso n`	Remove *Newton- soft.Json* NuGet package from a project in the current directory
`dotnet remove reference lib/li b.c spr oj`	Remove a project reference from the current project

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com