



TRAILHEAD
TECHNOLOGY PARTNERS

.NET CLI, Or

How I Learned to Stop Worrying
and Love the Command Line



Jonathan “J.” Tower



r/dotnet
daigoba66

[Join](#)

...

For discussion: are many .NET developers irrationally afraid of a CLI?

Something I've observed for a long time; a lot of .NET developers seem to be afraid of command line interfaces. I know that not all are, but there is definitely a set of developers that won't use a tool unless it has a GUI. Or worse, won't use a tool unless it integrates with Visual Studio.

Why is this? Is this a bad thing? Especially given how many .NET tools are moving to "CLI first", including those from Microsoft.

I'm not saying that a CLI is *always* better than a GUI. But I think, in my opinion anyway, some things are *easier* from a CLI. It can make you more productive if used smartly. I like using [psake](#) as a task runner. When working with Git, I tend to mix CLI and GUI (but not Visual Studio), depending on what I'm doing with it.

I've also heard and seen several comments about PowerShell as if it has a reputation of being complicated to learn. Though to fair, discoverability of commandlets is terrible. While I use PowerShell all the time, I think I still don't really *know* PowerShell.

What can be done to make developers amicable towards a CLI?

[↑ 16](#)[↓ 63](#)[Share](#)

r/dotnet

[Join](#)

.NET

.NET Community, if you are using C#, VB.NET, F#, or anything running with .NET... you are at the right place!

145K

Members

480

Online

Top 1%

Rank by size ↗

.NET r/dotnet

C# Dev Kit is out for Visual Studio Code



yout.be

96 upvotes · 83 comments

.NET r/dotnet

Announcing .NET 8 RC2



devblogs....

100 upvotes · 30 comments

.NET r/dotnet

...

“The oldest and strongest emotion of mankind is **fear**, and the oldest and strongest kind of fear is **fear of the unknown.**”

- H. P. Lovecraft





What We'll Cover



WHAT IS .NET CLI



WHY CLIS ARE USEFUL
& IMPORTANT



GETTING STARTED
WITH .NET CLI



COMMON
COMMANDS



.NET CLI
FOR POWER USERS

```
1. Sep 15:53 .
2. Sep 15:53 ..
3. Sep 2015 bin -> usr/bi
4. Sep 09:31 boot
5. Sep 15:50 dev
6. Sep 09:32 etc
7. Sep 15:52 home
8. Sep 2015 lib -> usr/u
9. Sep 09:31 lib64 -> us
10. Sep 2015 lost+found
11. Jul 10:01 mnt
12. Aug 22:45 opt
13. Sep 2015 private -> /
14. Sep 15:52 proc
15. Sep 08:15 root
16. Aug 15:37 run
17. Sep 15:50 sbin -> us
18. Sep 2015 sbin
19. Sep 15:51 srv
20. Sep 15:45 sys
21. Aug 15:39 usr
22. Jul 10:25 var
```

Jonathan "J." Tower

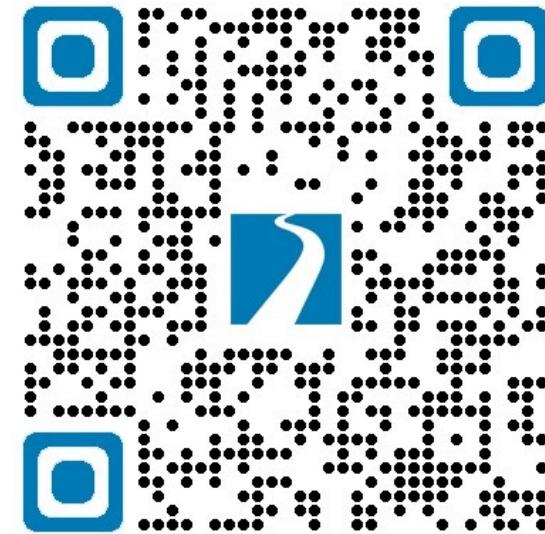
Principal Consultant & Partner



TRAILHEAD
TECHNOLOGY PARTNERS

- 🏆 Microsoft MVP in .NET
- ✉️ jtower@trailheadtechnology.com
- 🌐 trailheadtechnology.com/blog
- ☐ jtowermi
- linkedin jtower

**Free
Consultation**



bit.ly/th-offer

github.com/trailheadtechnology/distributed-monolith

What is the .NET CLI?



What is the .NET CLI?

Command
Line
Interface

What is the .NET CLI?

The .NET Core command-line interface (CLI) is a new cross-platform toolchain for developing .NET applications.

The CLI is a foundation upon which higher-level tools, such as Integrated Development Environments (IDEs), editors, and build orchestrators, can rest.

Source: <https://docs.microsoft.com/en-us/dotnet/core/tools>

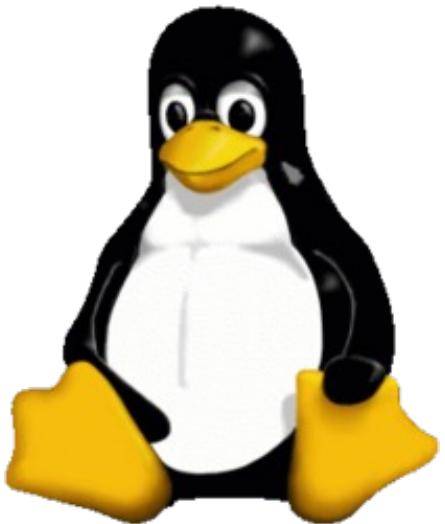
What is the .NET CLI?

The .NET Core [command-line interface](#) (CLI) is a new [cross-platform](#) toolchain for developing [.NET applications](#).

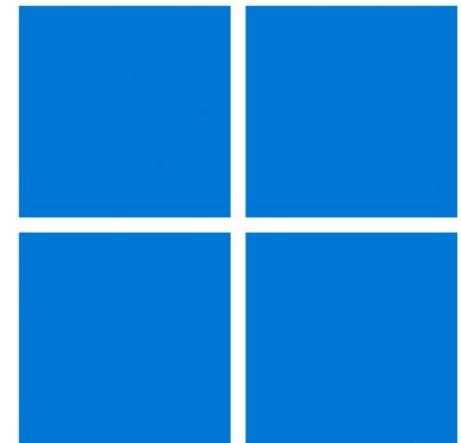
The CLI is a [foundation](#) upon which [higher-level tools](#), such as Integrated Development Environments (IDEs), editors, and build orchestrators, can rest.

Source: <https://docs.microsoft.com/en-us/dotnet/core/tools>

Cross-Platform



Mac OS



.NET / .NET Core, Not .NET Framework



.NET / .NET Core, Not .NET Framework



.NET / .NET Core, Not .NET Framework



Why a CLI?

	CLI	GUI
Memory Use	✓ Less	✗ More
Level of Precision	✓ High	✗ Low
Speed	✓ Faster	✗ Slower
Automation	✓ Yes	✗ No
Cross-Platform Compat	✓ High	✗ Low
Keyboard and Mouse	— Keyboard	— Keyboard and Mouse
Ease of Use	✗ More difficult	✓ Easier
Personalization	✗ Less	✓ More
Discoverability	✗ Harder	✓ Easier
Display of Information	✗ Files and text only	✓ Graphical display
Avoiding Errors	✗ Difficult	✓ Easier

	CLI	GUI
Memory Use	Less	More
Level of Precision	High	Low
Speed	Faster	Slower
Automation	Yes	No
Cross-Platform Compat	High	Low
Keyboard and Mouse	— Keyboard	— Keyboard and Mouse
Ease of Use	More difficult	Easier
Personalization	Less	More
Discoverability	Harder	Easier
Display of Information	Files and text only	Graphical display
Avoiding Errors	Difficult	Easier

What CLIs are Good At



Fast



Precise



Small memory
footprint



Cross-platform



Automation

Common CLI Scenarios



Using .NET
without an IDE



Third-Party IDEs



Using .NET in a
container



Automating .NET
in CI/CD pipelines



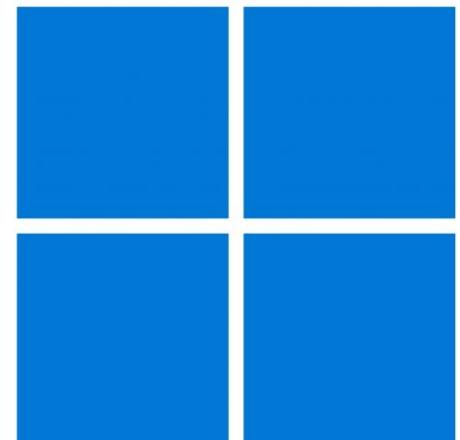
Automating
common tasks

Getting Started

Installing .NET CLI

Installing .NET CLI: Windows

- Install .NET SDK
 - `winget install Microsoft.DotNet.SDK.7`
 - <https://dotnet.microsoft.com/en-us/download/dotnet>
- Install Visual Studio Code & C# Extension



Installing .NET CLI: macOS

- Install .NET SDK
 - <https://dotnet.microsoft.com/en-us/download/dotnet>
- Install Visual Studio Code & C# Extension
- Install Visual Studio for Mac*



Mac OS

*Retirement on August 31, 2024

Installing .NET CLI: Linux

- Install .NET SDK
 - <https://dotnet.microsoft.com/en-us/download/dotnet>
- On some distros
 - `sudo apt-get install -y dotnet7`



Getting Started

Using .NET CLI

Command Structure

```
> dotnet build my_app.csproj --output /build_output
```

Command Structure

```
> dotnet build my_app.csproj --output /build_output
```

Driver

Command

Arguments

Options

Command Structure

```
> build my_app.csproj --output /build_output
```

Driver

dotnet

Command

Arguments

Options

Command Structure

```
> my_app.csproj --output /build_output
```

Driver	Command	Arguments	Options
dotnet	build		

Command Structure

>

--output /build_output

Driver

dotnet

Command

build

Arguments

my_app.csproj

Options

Command Structure

>

Driver

dotnet

Command

build

Arguments

my_app.csproj

Options

--output
/build_output

Command Structure

>

Driver

dotnet

1

Command

build

1

Arguments

my_app.csproj

0:M

Options

--output
/build_output

0:M

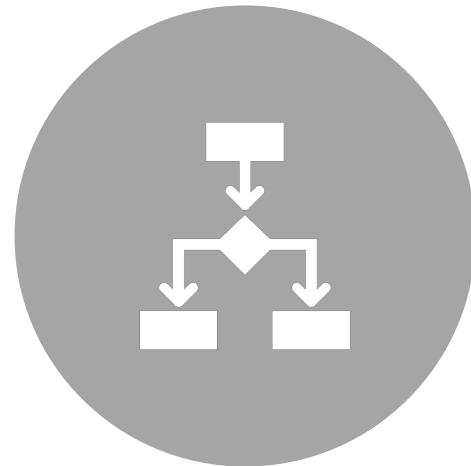
Common Commands

In .NET CLI

Basic Commands



NEW



HELP

DEMO

New Project

Build Commands



BUILD



RUN



CLEAN

DEMO

Build, Run, Clean

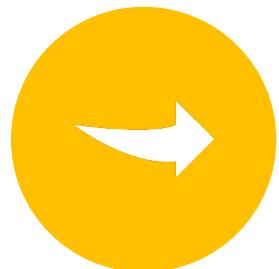
Nuget Commands



NUGET DELETE



NUGET LOCALS



NUGET PUSH



RESTORE

DEMO

Restore

Publishing Commands



PUBLISH



PACK



STORE

DEMO

Publish

Project Commands



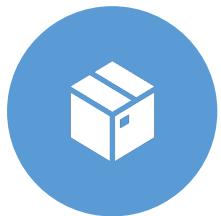
SLN



ADD PACKAGE



ADD
REFERENCE



REMOVE
PACKAGE



REMOVE
REFERENCE



LIST
REFERENCE

DEMO

Modify a Solution

Testing Commands



TEST



VS TEST

DEMO

Running Tests

.NET CLI for Power Users

Power Commands



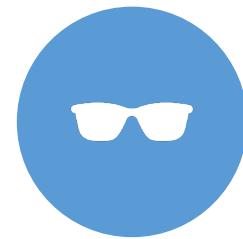
DEV-CERTS



SDK CHECK



WORKLOAD



WATCH



MSBUILD

DEMO

Power CLI Commands

Custom Tools



- Can be installed locally or globally
- Can build your own!
- dotnet tool search
- https://www.nuget.org/packages?pac_kagetype=dotnettool

DEMO

Install a Custom Tool

DEMO

Creating a Custom Tool

Key Takeaways

- CLIs > GUIs at **some things** (like x-plat, automation)
- CLI handles all **common tasks**
- CLI handles many/most **advanced tasks**
- **Custom tooling** capability
- **Familiarity** overcomes fear

Thanks! Questions?

Jonathan "J." Tower

🏆 Microsoft MVP in .NET

✉️ jtower@trailheadtechnology.com

🌐 trailheadtechnology.com/blog

❑ jtowermi

in jtower

github.com/trailheadtechnology/dotnetcli

Free Consultation



bit.ly/th-offer