.NET CLI Commands Reference Guide

What is .NET CLI

- The .NET Command Line Interface (CLI) is a cross-platform toolchain for developing, building, running, and publishing .NET applications using command line.
- It's an integral part of the .NET SDK.
- Provides a comprehensive set of command-line tools for managing .NET projects without requiring an IDE.

> **Key Features:** The .NET CLI enables you to perform essential development tasks through simple, powerful commands.

Installation Verification

Verify your .NET installation and check version information:

```
dotnet --version
# Display the installed .NET version
```

```
dotnet --info
# Display detailed information about .NET installation
```

Project Management Commands

1. Creating Projects and Solutions

```
dotnet new console -n MyConsoleApp
# Create a new console application

dotnet new webapi -n MyDemoWebApi
# Create a new Web API project

dotnet new sln -n MySln
# Create a new solution file
```

2. Solution Management

```
dotnet sln MySln.sln add MyApp/MyApp.csproj
# Add project to solution

dotnet sln MySln.sln remove MyApp/MyApp.csproj
# Remove project from solution
```

3. Dependency Management

```
dotnet restore
# Restore NuGet packages

dotnet add package Newtonsoft.Json
# Add a NuGet package

dotnet remove package Newtonsoft.Json
# Remove a NuGet package

dotnet list package
# List installed packages
```

4. Project References

```
dotnet add MyApp/MyApp.csproj reference MyLib/MyLib.csproj
# Add project reference

dotnet remove MyApp/MyApp.csproj reference MyLib/MyLib.csproj
# Remove project reference

dotnet list reference
# List project references
```

Basic Operations Commands

Build and Clean Operations

```
dotnet clean
# Clean the output of a project (default configuration)

dotnet clean --configuration Release
# Clean a project built using the Release configuration

dotnet restore
# Restore dependencies and tools

dotnet build
# Build the project (Debug configuration by default)

dotnet build --configuration Release
# Build using Release configuration
```

Run and Watch Operations

```
dotnet run
# Run the project

dotnet watch
# Watch for file changes and rebuild

dotnet watch run
# Watch for changes and automatically restart the application

dotnet watch build
# Watch for changes and rebuild automatically
```

Publishing Commands

Basic Publishing

```
dotnet publish
# Basic publish (framework-dependent deployment)

dotnet publish -o ./publish
# Specify output directory

dotnet publish --output ./dist
# Alternative syntax for output directory

dotnet publish -c Release
# Publish with Release configuration (creates .dll file)
```

Platform-Specific Publishing

```
dotnet publish -c Release --runtime win-x64
# Creates Windows-specific executable (.exe)

dotnet publish -c Release --runtime linux-x64
# Creates Linux-specific executable (ELF binaries)

dotnet publish -c Release --runtime osx-x64
# Creates macOS-specific executable (Mach-O)
```

> ### 💡 Pro Tips
>
> - Use `dotnet --help` to get help for any command
> - Add `--self-contained` flag to publish commands for standalone deployments
> - Use `dotnet watch` during development for automatic rebuilds
> - The `-o` flag is short for `--configuration`
> - The `-o` flag is short for `--output`