.NET CLI dotnet SDK commands

CONCEPTS

Build, run, test and publish restore NuGet packages first, so a separate restore is rarely needed.

Source .cs + .csproj
↓ restore
Packages obj/project.assets.json
↓ build
Binaries bin/<config>/<tfm>
↓ publish
Deploy folder bin/.../publish

Arguments the app should get go after a literal --: dotnet run -- --help.

CREATE

dotnet new list
List installed templates

dotnet new console -o *<name>*
New console app in a folder

dotnet new classlib -o *<name>*
New class library

dotnet new webapi -o *<name>*
New ASP.NET Core web API

dotnet new xunit -o *<name>*
New xUnit test project

dotnet new sln
New solution file

BUILD AND RUN

dotnet build
Compile the project or solution

dotnet build -c Release
Build in Release configuration

dotnet run
Build and run the project

dotnet run --project *<path>*
Run a project from elsewhere

dotnet run -- *<args>*
Pass arguments to the app

dotnet watch
Rerun or hot reload on change

TEST

dotnet test
Build and run all tests

TEST (cont.)

dotnet test --filter *<expr>*
Run only matching tests

dotnet test --filter "Name~*<text>***"**
Match by test name

dotnet test -l "console;verbosity=detailed"
Show each test result

dotnet test --collect "XPlat Code Coverage"
Collect coverage with Coverlet

PACKAGES AND REFERENCES

dotnet add package *<id>*
Add a NuGet package

dotnet add package *<id>* **-v** *<version>*
Add a specific version

dotnet list package
List referenced packages

dotnet list package --outdated
Show newer versions

SOLUTIONS

dotnet sln add *<csproj>*
Add a project to the solution

PUBLISH

dotnet publish -c Release
Publish for deployment

dotnet publish -r *<rid>* **--self-contained**
Bundle the runtime

TOOLS AND FORMAT

dotnet tool install *<id>*
Install a local (repo) tool

dotnet format
Apply .editorconfig style fixes

dotnet format --verify-no-changes
CI check, fails if changes needed

SDK AND WORKLOADS

dotnet --info
SDK, runtime and OS details

dotnet *<command>* **-h**
Help for any command