

The pipeline

1 host	picks the SDK (global.json, rollForward)
2 restore	NuGet resolves the graph → obj/project.assets.json
3 evaluation	imports, properties, items (six passes)
4 execution	targets run tasks in dependency order
5 outputs	bin/, obj/, .nupkg, publish/

```
dotnet build -bl #
msbuild.binlog
dotnet build -v:diag -tl:off
```

global.json

```
{ "sdk": { "version": "10.0.402",
"rollForward": "latestPatch",
"allowPrerelease": false },
"msbuild-sdks": {
"Microsoft.Build.Traversal":
"4.1.82" },
"test": { "runner":
"Microsoft.Testing.Platform"
} }
```

```
dotnet --list-sdks installed SDKs
dotnet --version the SDK this folder selects
dotnet sdk-check out-of-support or outdated
```

SDK-style project

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>

  <TargetFrameworks>net10.0;net8.0;
netstandard2.0</TargetFrameworks>
  <Nullable>enable</Nullable>
</PropertyGroup>
  <ItemGroup>
    <PackageReference
Include="Humanizer.Core" />
    <ProjectReference
Include="..\Core\Core.csproj" />
  </ItemGroup>
</Project>
```

Directory.Build.props	early import: shared properties
Directory.Build.targets	late import: shared targets/items
Directory.Packages.props	central package versions
.slnx	XML solution (dotnet sln migrate)

MSBuild syntax

\$(Prop)	property
@(Item)	item list; @(l, ',')
	separator
%(Meta)	metadata → batching
@(I->'%(Filename)')	transform
\$([System.IO.Path]::Combine(a,b))	property function
\$([MSBuild]::VersionGreaterThan(...))	intrinsic function
Condition="\$\$(X)"='quote both sides = 'true'	
Include / Exclude	item operations
/ Remove / Update	
<Target Name="Stamp" BeforeTargets="CoreCompile" Inputs="@ (Compile)" Outputs="\$\$(IntermediateOutputPath)stamp">	
<Message Importance="high" Text="\$\$(Version)" />	
</Target>	

MSBuild command line

-t:Clean;Build	targets
-	global property (wins over the file)
p:Configuration=Release	
-restore	restore first (dotnet does it)
-m[:N]	parallel nodes
-graph	static graph build
-bl[:file]	binary log (may contain secrets)
-getProperty:X	print an evaluated value
-pp:out.xml	preprocess: all imports inlined
-check	BuildCheck analyzers
-warnaserror / -nowarn	warning policy
-tl:on off	terminal logger

Classic MSBuild bugs

\$(TargetFramework)	empty in Directory.Build.props
property order	a property read before it is set is empty
no Inputs/Outputs	target always runs
BeforeTargets="Build"	runs once per TFM
Exec unquoted path	breaks on spaces
-p:X on CLI	silently overrides the project

dotnet CLI

dotnet new console -n App	create from template
dotnet restore --locked-mode	fail if lock file differs
dotnet build -c Release	build
dotnet test	VSTest or MTP
dotnet run app.cs	file-based app (#:package, #:property)
dotnet format --verify-no-changes	style gate (exit 2)
dotnet watch	hot reload
dotnet pack -o out	.nupkg (Release by default)
dotnet publish -r linux-x64	RID-specific output
-p:PublishAot=true	Native AOT
-	single file
p:PublishSingleFile=true	
-	OCI image, no Dockerfile
t:PublishContainer	
dotnet sln add /migrate	solutions, .slnx
dotnet workload list	optional SDK components

.NET tools

dotnet tool install -g x	global (~/.dotnet/tools)
dotnet new tool-manifest	local: .config/dotnet-tools.json
dotnet tool restore	restore local tools
dnx pkg@1.2.3 --yes	download and run once (.NET 10)

Package references

```
<PackageReference Include="Serilog" Version="4.3.0" />
<PackageReference Include="MinVer" PrivateAssets="all" />
```

1.0.0 ≥ 1.0.0 (lowest applicable)

[1.0.0] exactly 1.0.0

[1.0,2.0) ≥ 1.0, < 2.0

1.* 1.0.0-* floating

PrivateAssets="all" don't flow to consumers

ExcludeAssets="runtime" compile only

Central Package Management

```
<!-- Directory.Packages.props -->
<ManagePackageVersionsCentrally>true</ManagePackageVersionsCentrally>
<PackageVersion Include="Serilog" Version="4.3.0" />
<GlobalPackageReference Include="MinVer" Version="8.0.0" />
VersionOverride="..." per-project exception
```

CentralPackageTransitivePinningEnabled

RestorePackagesWithLockFile

nuget.config

```
<packageSources><clear />
  <add key="nuget.org"
value="https://api.nuget.org/v3/index.json" />
  <add key="corp"
value="https://pkgs.example/v3/index.json" />
</packageSources>
<packageSourceMapping>
  <packageSource key="nuget.org">
<package pattern="*" /></packageSource>
  <packageSource key="corp"><package pattern="Corp.*" /></packageSource>
</packageSourceMapping>
```

dotnet nuget locals all --list	cache folders
dotnet nuget why proj Pkg	who pulls a package in
dotnet package list --vulnerable	audit report
NuGetAuditMode=all	transitive audit (net10 default)
--no-http-cache	skip the 30-minute HTTP cache

Pack errors

NU5039 / NU5046	README / icon not packed
NU5030 / NU5033	licence file missing / both kinds
NU5048 / NU5125	iconUrl / licenseUrl deprecated
NU5104	stable package, prerelease dep
NU5128	lib/ and dependency groups differ
CP0002	public member removed
PKV006	target framework dropped

Pack a library

```
<PackageId>LedgeLine</PackageId>
<VersionPrefix>1.1.0</VersionPrefix>
<Description>...</Description>
<PackageLicenseExpression>MIT</PackageLicenseExpression>
<PackageReadmeFile>README.md</PackageReadmeFile>
<PackageIcon>icon.png</PackageIcon>
<EnablePackageValidation>true</EnablePackageValidation>
<PackageValidationBaselineVersion>1.0.0</PackageValidationBaselineVersion>
<IncludeSymbols>true</IncludeSymbols>
<SymbolPackageFormat>snupkg</SymbolPackageFormat>
```

dotnet pack -o out --version-suffix beta.1

Package folders

lib/<tfm>/	compile + run
ref/<tfm>/	compile only
runtimes/<rid>/native/	native libraries
build/ buildTransitive/	<Id>.props/targets
analyzers/dotnet/cs/	analyzers, generators
contentFiles/any/any/	content for consumers
tools/	.NET tool payload

Publish

```
dotnet nuget push "out/*.nupkg" -s nuget.org --skip-duplicate
# key from NUGET_API_KEY (NuGet 7.6+)
permissions: { id-token: write } # Trusted Publishing
- uses: NuGet/Login@v1 # id: login, user: profile name
--api-key ${{ steps.login.outputs.NUGET_API_KEY }}
```

API keys	max 30 days since 2026-08-17 409
dotnet nuget delete	unlists on nuget.org
deprecate	nuget.org Manage package: reason, alternative
dotnet nuget sign/verify	author signature + timestamp

Restore errors

NU1101 / NU1102	package / version not found
NU1107 / NU1605	version conflict / downgrade
NU1004	lock file inconsistent
NU1301 / NU1302	source failed / HTTP refused
NU1901–NU1904	known vulnerability

Then → now

packages.config → PackageReference	.sln → .slnx	PackageIconUrl → PackageIcon	PackageLicenseUrl → PackageLicenseExpression	.symbols.nupkg → .snupkg
Source Link package → built into SDK 8+	dotnet add package → dotnet package add	long-lived API keys → Trusted Publishing	VSTest → MTP (Microsoft.Testing.Platform)	dotnet tool install + run → dnx