Publishing .NET Applications (Self-Contained vs Framework-Dependent)

This document explains **step-by-step** how to publish a .NET application using **Framework-Dependent Deployment (FDD)** and **Self-Contained Deployment (SCD)**. It also explains **when to use which mode**, common mistakes, and verification steps.

---

1. Prerequisites

Before publishing, ensure:

1. .NET SDK is installed

```
dotnet --version
```

2. Your project builds successfully

```
dotnet build
```

3. You are inside the **project folder** (where `.csproj` exists)

```
cd ConsoleLearning
```

⚠️ `dotnet publish` must be run from the **project directory**, not the solution directory.

---

2. Key Concepts (Quick Understanding)

Framework-Dependent Deployment (FDD)

- Application **depends on .NET runtime installed on target machine**
- Smaller output size
- Faster publish
- Runtime must be installed separately

Self-Contained Deployment (SCD)

- Application **includes .NET runtime inside output**
- Larger output size
- Runs on machine **without .NET installed**

- OS-specific

---

3. Framework-Dependent Deployment (FDD)

3.1 When to Use FDD

Use FDD when: - Target machine already has .NET installed - Internal/company servers - Docker images with .NET base image - You want smaller binaries

---

3.2 Command to Publish (Framework-Dependent)

```
dotnet publish -c Release
```

This publishes the application using default framework-dependent mode.

3.3 Output Location

After publishing, files are created at:

```
bin/
 └── Release/
      └── net8.0/
           └── publish/
```

3.4 Files Generated (Important)

- `.dll` → Main application
- `.deps.json`
- `.runtimeconfig.json`

To run the app:

```
dotnet ConsoleLearning.dll
```

> ❗ Requires matching .NET runtime installed on the machine

3.5 Verify Framework Dependency

Check runtime config:

```
{
  "framework": {
    "name": "Microsoft.NETCore.App",
    "version": "8.0.0"
  }
}
```

This confirms **framework-dependent mode**.

---

4. Self-Contained Deployment (SCD)

4.1 When to Use SCD

Use SCD when: - Target machine has **no .NET installed** - Client systems - Desktop utilities - Exam / submission / offline delivery

---

4.2 Identify Target Runtime (RID)

Common Runtime Identifiers (RID):

OS	RID
Windows x64	win-x64
Windows x86	win-x86
Linux x64	linux-x64
macOS x64	osx-x64

4.3 Command to Publish (Self-Contained)

```
dotnet publish -c Release -r win-x64 --self-contained true
```

4.4 Output Location

```
bin/
 └── Release/
      └── net8.0/
           └── win-x64/
                └── publish/
```

4.5 Files Generated

- `.exe` (Windows)
- .NET runtime files
- No dependency on external runtime

Run directly:

```
ConsoleLearning.exe
```

✅Works even if .NET is NOT installed

4.6 Verify Self-Contained Mode

Check runtime config file:

```
{
  "includedFrameworks": [
    {
      "name": "Microsoft.NETCore.App",
      "version": "8.0.0"
    }
  ]
}
```

This confirms **self-contained deployment**.

---

5. Optional: Single-File Publishing

5.1 Framework-Dependent Single File

```
dotnet publish -c Release -p:PublishSingleFile=true
```

5.2 Self-Contained Single File

```
dotnet publish -c Release -r win-x64 --self-contained true -
p:PublishSingleFile=true
```

Output: - One `.exe` file - Best for distribution

---

6. Size Comparison (Typical)

Mode	Size
Framework-Dependent	~5–15 MB
Self-Contained	~70–150 MB
Self-Contained Single File	~60–120 MB

---

7. Common Mistakes & Fixes

Mistake 1: Running from solution folder

❌`dotnet publish` fails

✅Fix:

```
cd ConsoleLearning
```

Mistake 2: Wrong RID

❌App doesn't run on target OS

✅Fix: Use correct `-r` value

Mistake 3: Expecting `.exe` in FDD

❌Only `.dll` generated

✅Fix: Use self-contained mode

---

8. Interview-Ready Explanation

> "Framework-dependent deployment relies on a pre-installed .NET runtime and produces smaller outputs, while self-contained deployment packages the runtime with the application, allowing it to run independently on target systems. The choice depends on deployment environment and distribution requirements."

---

9. Recommended Usage (Industry Practice)

Scenario	Recommended Mode
Internal servers	Framework-Dependent
Client machines	Self-Contained
Exam submission	Self-Contained
Docker	Framework-Dependent
Utilities	Self-Contained Single File

---

10. Final Checklist

✔️Build succeeds ✔️Correct project folder ✔️Correct RID ✔️Correct deployment mode ✔️Verified output

---

✅You now know how to publish .NET applications professionally

If you want next: - Publishing ASP.NET Core APIs - Dockerizing published output - CI/CD pipeline publishing - Real company deployment flow

Just say **next**.