Shared Assembly Creation

18 September 2024 09:14

Creating a shared assembly in C# is a great way to learn about .NET assemblies, versioning, and the global assembly cache (GAC). Since you're new to C#, I'll guide you through the process step by step, starting from the basics.

What is a Shared Assembly?

A shared assembly in .NET is an assembly that can be used by multiple applications. It's typically stored in the Global Assembly Cache (GAC) to be accessible to various applications on the system.

Prerequisites

1. **Visual Studio Code (VS Code):** Ensure you have it installed. While Visual Studio Code is a lightweight editor, you'll also need the .NET SDK installed.
2. **.NET SDK:** Download and install the .NET SDK from Microsoft's official website.

Step-by-Step Guide

1. Set Up Your Development Environment

1. **Install .NET SDK:**
   - Ensure you have the .NET SDK installed by running `dotnet --version` in your terminal. If not installed, download it from the .NET website.
2. **Install C# Extension for VS Code:**
   - Open Visual Studio Code.
   - Go to Extensions (Ctrl+Shift+X) and search for "C#". Install the extension by Microsoft.

2. Create a New Project

1. **Open Terminal/Command Prompt:**
   - Navigate to the directory where you want to create your project.
2. **Create a Class Library Project:**
   - Run the following command to create a new class library:

- Run the following command to create a new class library:

```bash
dotnet new classlib -o SharedAssembly
```

- Navigate to the project directory:

```bash
cd SharedAssembly
```

3. **Open the Project in VS Code:**
   - Run `code .` to open the project in Visual Studio Code.

3. Modify the Project

1. **Edit the Class Library:**
   - Open the `Class1.cs` file in VS Code and modify it as needed. For example:

```csharp
using System;

namespace SharedAssembly
{
    public class HelloWorld
    {
        public void Greet()
        {
            Console.WriteLine("Hello from SharedAssembly!");
        }
    }
}
```

2. **Modify `SharedAssembly.csproj` File:**
    - Open the `SharedAssembly.csproj` file and make sure it includes the necessary assembly information:

```xml
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Library</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <AssemblyName>SharedAssembly</AssemblyName>
    <Version>1.0.0.0</Version>
    <FileVersion>1.0.0.0</FileVersion>
    <InformationalVersion>1.0.0</InformationalVersion>
  </PropertyGroup>
</Project>
```

4. Build the Assembly

1. **Build the Project:**
    - Run the following command in your terminal:

```bash
dotnet build
```

    - This will create a `.dll` file in the `bin/Debug/net6.0/` directory.

5. Install the Assembly into the Global Assembly Cache (GAC)

1. **Create a Strong Name Key (optional but recommended):**
   - Generate a strong name key file:

```bash
sn -k MyKey.snk
```

   - Update the `csproj` file to include the key:

```xml
<PropertyGroup>
  <AssemblyName>SharedAssembly</AssemblyName>
  <Version>1.0.0.0</Version>
  <FileVersion>1.0.0.0</FileVersion>
  <InformationalVersion>1.0.0</InformationalVersion>
  <SignAssembly>true</SignAssembly>
  <AssemblyOriginatorKeyFile>MyKey.snk</AssemblyOriginatorKeyFile>
</PropertyGroup>
```

2. **Install the Assembly:**
   - Use the `gacutil` tool to install the assembly into the GAC. Open the Developer Command Prompt for Visual Studio and run:

```bash
gacutil -i path\to\your\assembly\SharedAssembly.dll
```

   - If `gacutil` is not available, you can find it in the Visual Studio Developer Command Prompt or download the Windows SDK.

6. Use the Shared Assembly in Other Projects

1. **Create a New Project to Use the Shared Assembly:**
   - Create a new console application:

```bash
dotnet new console -o UsingSharedAssembly
```

   - Navigate to the new project:

```bash
cd UsingSharedAssembly
```

2. **Add a Reference to the Shared Assembly:**
   - Edit the `.csproj` file of the new project to include a reference to the shared assembly:

```xml
<ItemGroup>
  <Reference Include="SharedAssembly">
    <HintPath>C:\path\to\SharedAssembly.dll</HintPath>
  </Reference>
</ItemGroup>
```

3. **Use the Shared Assembly in Your Code:**
   - In `Program.cs`, use the assembly:

```csharp
using System;
using SharedAssembly;

class Program
{
    static void Main()
    {
        HelloWorld hello = new HelloWorld();
        hello.Greet();
    }
}
```

4. **Run the Application:**
   - Build and run the application:

```bash
dotnet run
```