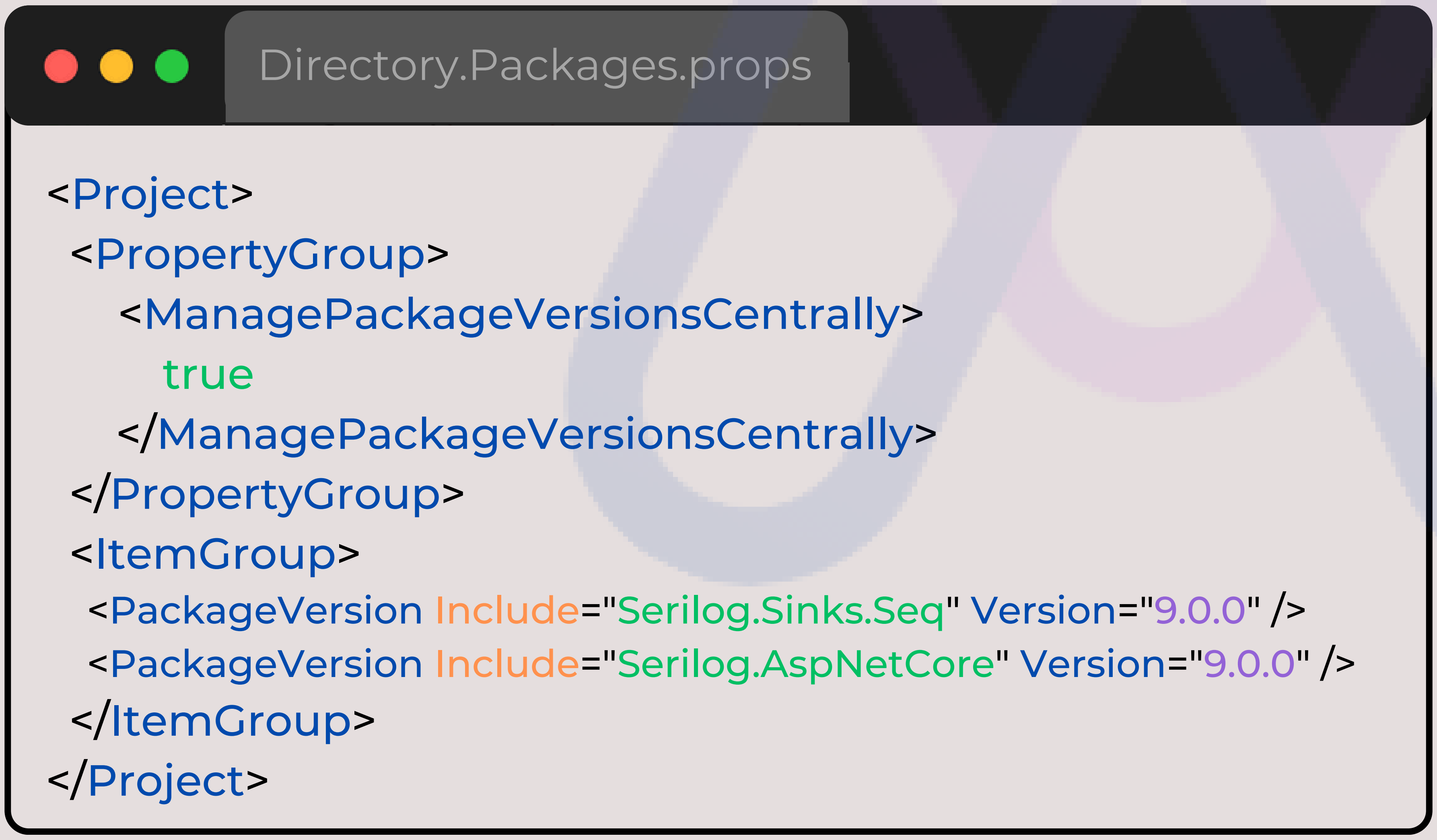
Ajay Patel

.NET

# What is Central Package Management in .NET



Directory.Packages.props

```
<Project>
  <PropertyGroup>
      <ManagePackageVersionsCentrally>
        true
      </ManagePackageVersionsCentrally>
  </PropertyGroup>
  <ItemGroup>
   <PackageVersion Include="Serilog.Sinks.Seq" Version="9.0.0" />
   <PackageVersion Include="Serilog.AspNetCore" Version="9.0.0" />
  </ItemGroup>
</Project>
```

Ajay Patel

## Enabling Central Package Management

### 1. Create a Directory.Packages.props File:

To get started with central package management, **you must create a Directory.Packages.props file** at the root of your solution or repository or any other folder where you want to apply CPM and **set the MSBuild property ManagePackageVersionsCentrally to true.**

Directory.Packages.props

```xml
<Project>
 <PropertyGroup>
    <ManagePackageVersionsCentrally>
      true
    </ManagePackageVersionsCentrally>
 </PropertyGroup>
 <ItemGroup>

 </ItemGroup>
</Project>
```

Ajay Patel

## # For New Project

If you have just started with a new project and no packages have been added yet, that's it, you're all done. Now, whenever you add new packages to your project, their versions will be added to the CPM file.

Directory.Packages.props

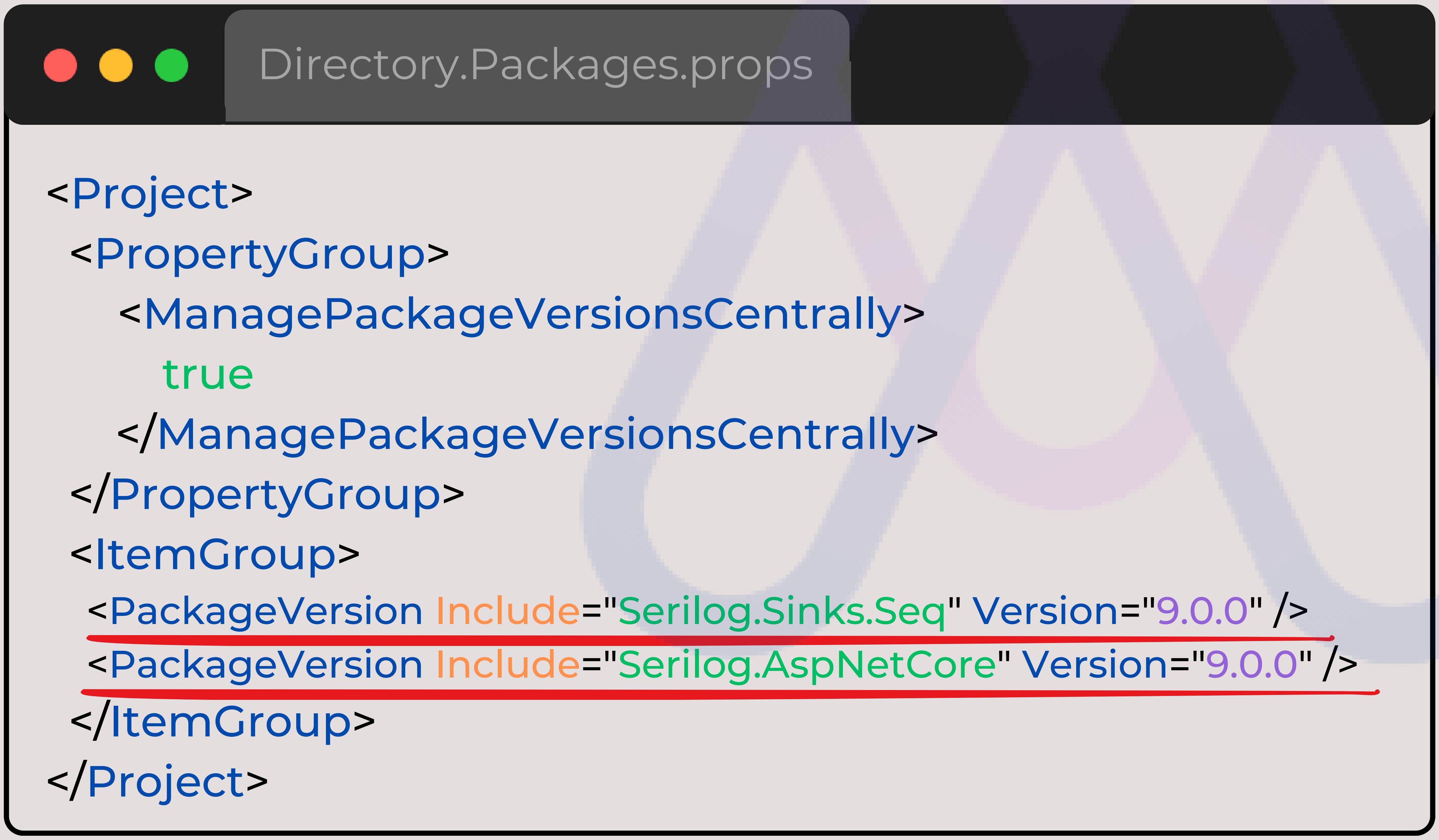
```
<Project>
  <PropertyGroup>
      <ManagePackageVersionsCentrally>
         true
      </ManagePackageVersionsCentrally>
  </PropertyGroup>
  <ItemGroup>
   <PackageVersion Include="Serilog.Sinks.Seq" Version="9.0.0" />
   <PackageVersion Include="Serilog.AspNetCore" Version="9.0.0" />
  </ItemGroup>
</Project>
```

**Ajay Patel**

## # For Existing Project

If you have an existing project and already have a few packages, you have to perform the steps below as an initial setup. After that, any new packages will be added to the CPM file automatically.

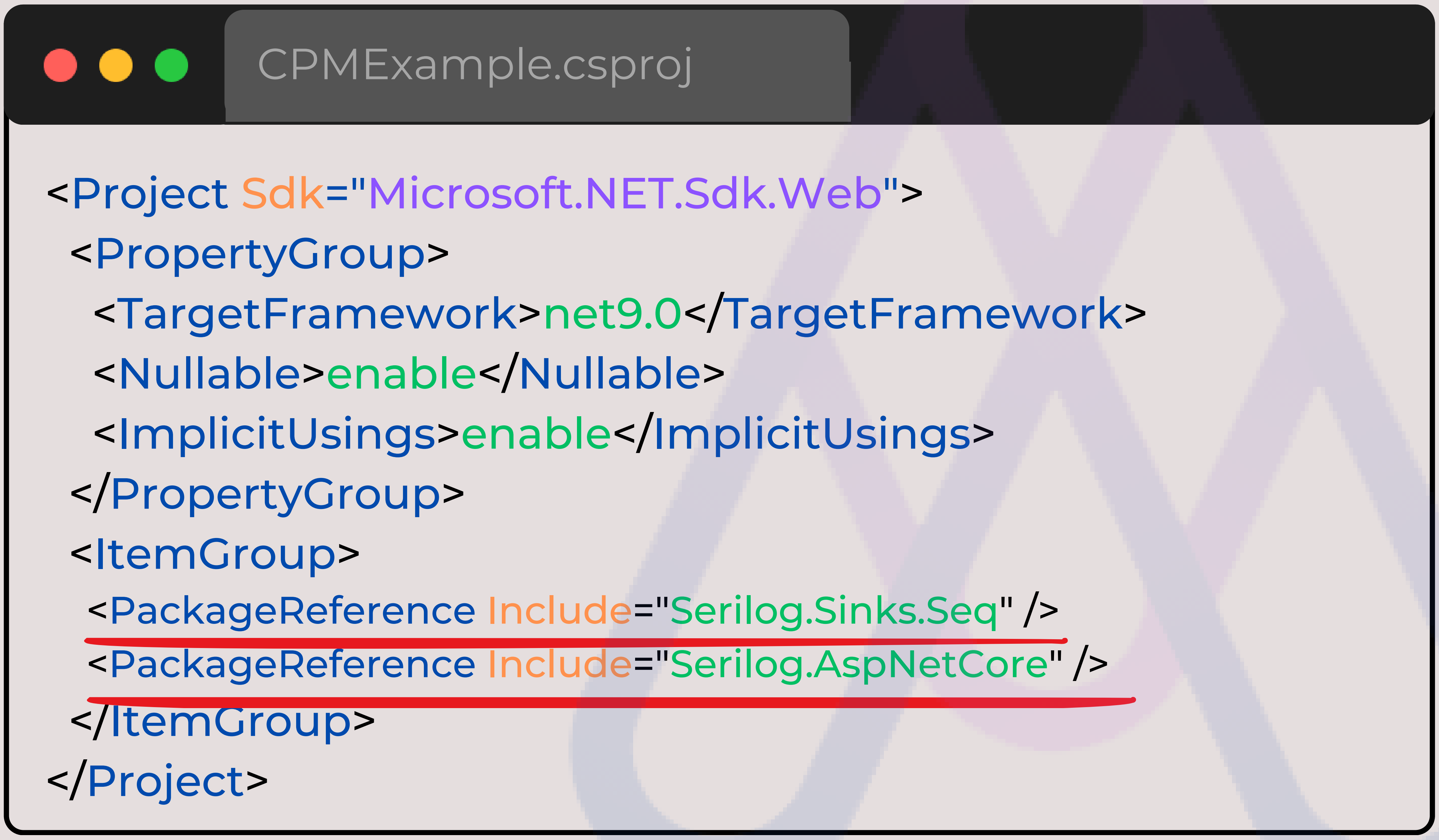
1. **Add existing packages and their versions defined in the .csproj file into the Directory.Packages.props file's PackageVersion element.**



Directory.Packages.props

```
<Project>
 <PropertyGroup>
    <ManagePackageVersionsCentrally>
      true
    </ManagePackageVersionsCentrally>
 </PropertyGroup>
 <ItemGroup>
  <PackageVersion Include="Serilog.Sinks.Seq" Version="9.0.0" />
  <PackageVersion Include="Serilog.AspNetCore" Version="9.0.0" />
 </ItemGroup>
</Project>
```

**2. In your .csproj file, remove the Version attribute from the <PackageReference> elements, and just keep the package name.**



CPMExample.csproj

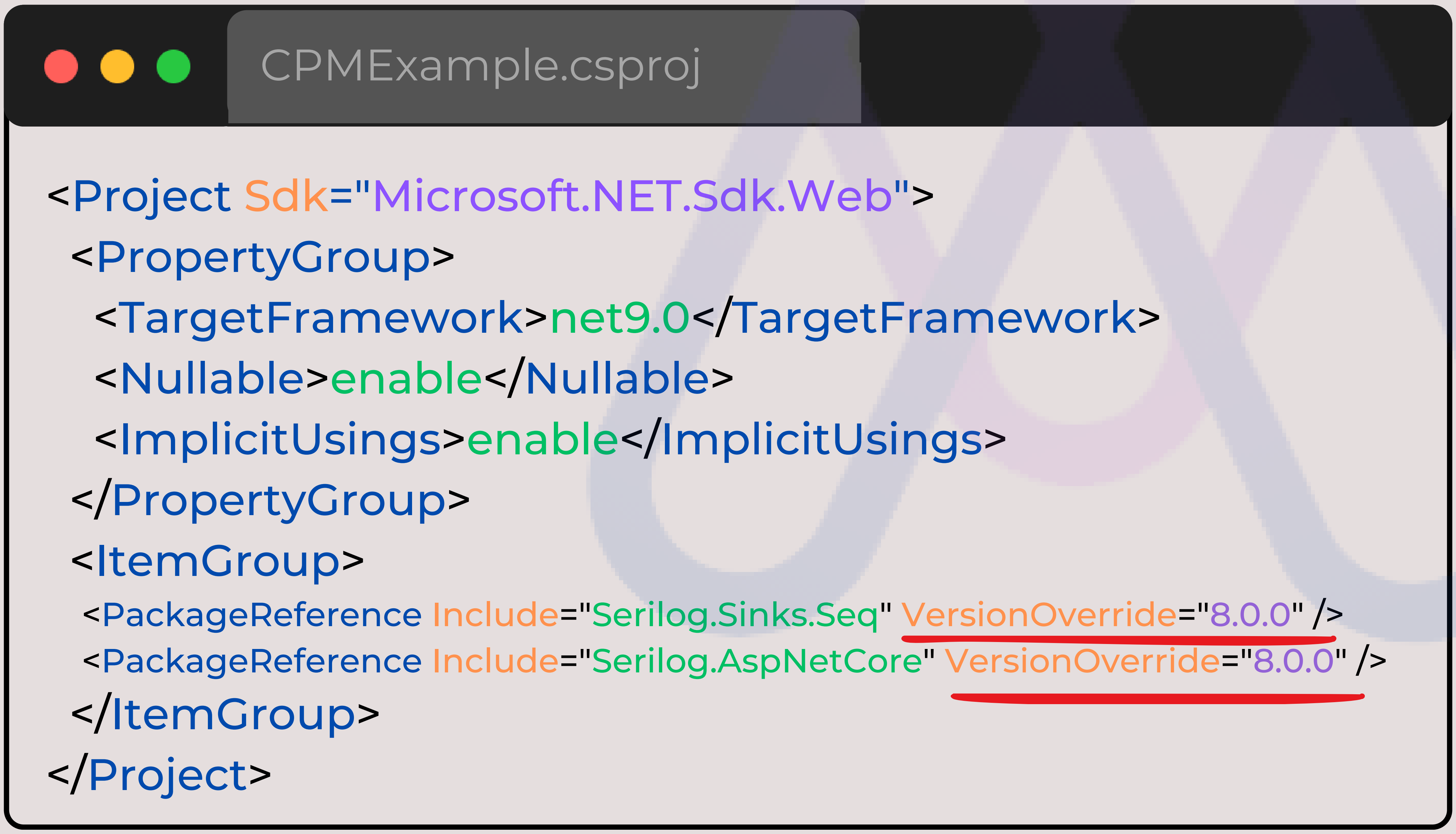
```
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>net9.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>
  <ItemGroup>
   <PackageReference Include="Serilog.Sinks.Seq" />
   <PackageReference Include="Serilog.AspNetCore" />
  </ItemGroup>
</Project>
```

Ajay Patel

**Advanced Scenarios**

## # Overriding package versions

You can override an individual package version by using the VersionOverride property on a <PackageReference /> item. This overrides any <PackageVersion /> defined centrally.



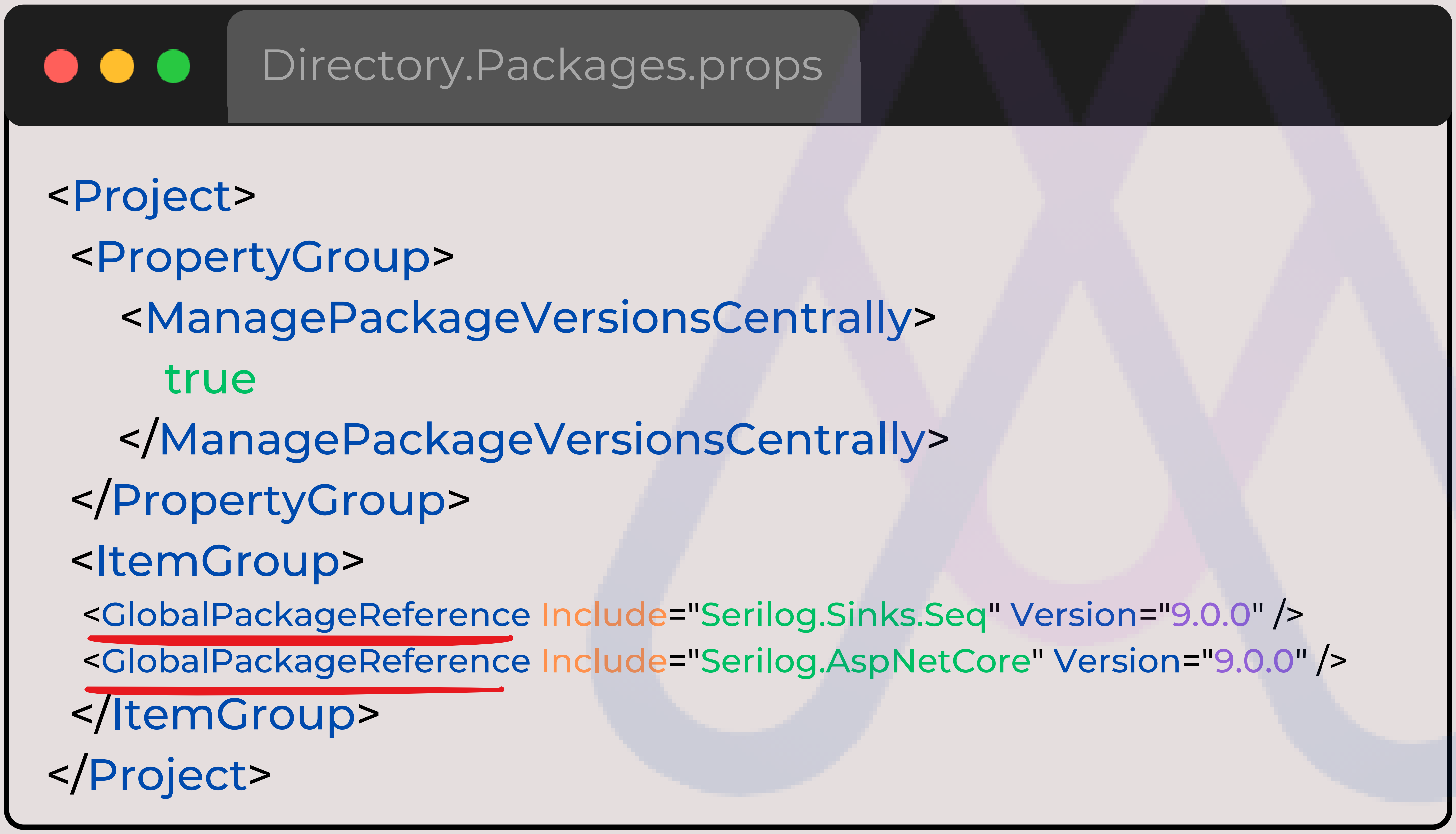
CPMExample.csproj

```xml
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>net9.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>
  <ItemGroup>
   <PackageReference Include="Serilog.Sinks.Seq" VersionOverride="8.0.0" />
   <PackageReference Include="Serilog.AspNetCore" VersionOverride="8.0.0" />
  </ItemGroup>
</Project>
```

Ajay Patel

## # Global Package References

A global package reference is used to specify that a package will be used by every project in a repository. GlobalPackageReference items should be placed in your **Directory.Packages.props** to be used by every project in a solution:



Directory.Packages.props

```
<Project>
  <PropertyGroup>
      <ManagePackageVersionsCentrally>
         true
      </ManagePackageVersionsCentrally>
  </PropertyGroup>
  <ItemGroup>
   <GlobalPackageReference Include="Serilog.Sinks.Seq" Version="9.0.0" />
   <GlobalPackageReference Include="Serilog.AspNetCore" Version="9.0.0" />
  </ItemGroup>
</Project>
```

Ajay Patel

## Additional Information about CPM

- ✓ MSBuild will automatically detect and import Directory.Packages.props file closest to the project.
- ✓ MSBuild will only detect and import file name with Directory.Packages.props. If you have different name of your CPM file you have to manually import it to your .csproj file.
- ✓ You can have multiple Directory.Packages.props file. MSBuild will not automatically import each Directory.Packages.props for you, only the first one closest to the project. You have to manually import other file.

Ajay Patel

.NET

# Knowledge is contagious, let's spread it!

DO YOU LIKE THIS POST?

REPOST IT!

THANKS FOR READING