

Forum:[Hack Forums](#)

Thread: [Output program without .dll dependency](#)

Sickness - August 15, 2020, 4:50 pm

Hi,

Past hour been looking for a solution to compile my program using netcoreapp.3.1 without any .dll.

I dont use any library. It just outputs the .exe with its .dll, how I can set it to just output the .exe?

Thanks!

Accord - August 15, 2020, 6:00 pm

Maybe this is what your need if the dll that youre talking about are netcore library

<https://docs.microsoft.com/en-us/dotnet/core/deploying/>

Suspect. - August 15, 2020, 6:45 pm

Create Folder Within the Solution Explorer E:\G lib > Right Click on Folder - Add Existing Item : All Files. Add Required .Dlls > After they have been added be sure to Hold CTRL and click all and on the properties in the "Build Action" setting mark them as Embedded Resource also "Copy to Output Directory" Do not copy > Resources > Files > Add Resource > Add Existing File > Make sure to Add the .Dlls from the subfolder we just created firstly as this is where the program gets the reference from.

Then before the InitializeComponent(); call Add the following Code:

```
Code AppDomain.CurrentDomain.AssemblyResolve += (sender, args) =>
{
    var resourceName = new AssemblyName(args.Name).Name + ".dll";
    var resource = Array.Find(GetType().Assembly.GetManifestResourceNames(),
        element => element.EndsWith(resourceName));

    using (var stream =
Assembly.GetExecutingAssembly().GetManifestResourceStream(resource))
    {
        var assemblyData = new byte[stream.Length];
        stream.Read(assemblyData, 0, assemblyData.Length);
        return Assembly.Load(assemblyData);
    }
};
```

This works for mostly every .DLL, Although I know it does not work for WebDriver.dll.
Hopefully this helps. :)

n0\$3ratu\$ - August 15, 2020, 7:19 pm

(Aug 15, 2020 06:45 PM)Suspect. Wrote: [\[CITE\]](#) Create Folder Within the Solution Explorer E:\G lib > Right Click on Folder - Add Existing Item : All Files. Add Required .Dlls > After they have been added be sure to Hold CTRL and click all and on the properties in the "Build Action" setting mark them as Embedded Resource also "Copy to Output Directory" Do not copy > Resources > Files > Add Resource > Add Existing File > Make sure to Add the .Dlls from the subfolder we just created

firstly as this is where the program gets the reference from.

Then before the InitializeComponent(); call Add the following Code:

```
Code AppDomain.CurrentDomain.AssemblyResolve += (sender, args) =>
{
    var resourceName = new AssemblyName(args.Name).Name + ".dll";
    var resource = Array.Find(GetType().Assembly.GetManifestResourceNames(),
        element => element.EndsWith(resourceName));

    using (var stream =
Assembly.GetExecutingAssembly().GetManifestResourceStream(resource))
    {
        var assemblyData = new byte[stream.Length];
        stream.Read(assemblyData, 0, assemblyData.Length);
        return Assembly.Load(assemblyData);
    }
};
```

This works for mostly every .DLL, Although I know it does not work for WebDriver.dll.

Hopefully this helps. :)

This will work.

Good luck.

Freak_OG - August 16, 2020, 1:03 am

What I do is I use dotnetpatcher for this XD just add a new assembly to dotnetpatcher, and go to dependencies -> merging and select all ur dlls, disable obfuscator (enable it if u want) and then click start then u get ur dependencyless exe in the "Protected folder"

Hope I helped :D

Sickness - August 16, 2020, 1:57 am

(Aug 15, 2020 06:45 PM)Suspect. Wrote: [\[CITE\]](#) Create Folder Within the Solution Explorer E:G lib > Right Click on Folder - Add Existing Item : All Files. Add Required .Dlls > After they have been added be sure to Hold CTRL and click all and on the properties in the "Build Action" setting mark them as Embedded Resource also "Copy to Output Directory" Do not copy > Resources > Files > Add Resource > Add Existing File > Make sure to Add the .Dlls from the subfolder we just created firstly as this is where the program gets the reference from.

Then before the InitializeComponent(); call Add the following Code:

```
Code AppDomain.CurrentDomain.AssemblyResolve += (sender, args) =>
{
    var resourceName = new AssemblyName(args.Name).Name + ".dll";
    var resource = Array.Find(GetType().Assembly.GetManifestResourceNames(),
        element => element.EndsWith(resourceName));

    using (var stream =
Assembly.GetExecutingAssembly().GetManifestResourceStream(resource))
    {
        var assemblyData = new byte[stream.Length];
        stream.Read(assemblyData, 0, assemblyData.Length);
        return Assembly.Load(assemblyData);
    }
}
```

};

This works for mostly every .DLL, Although I know it does not work for WebDriver.dll.

Hopefully this helps. :)

I don't think I can use this, I'm using a Console Application so I don't got InitializeComponent

(Aug 16, 2020 01:03 AM)Freak_OG Wrote: [\[CITE\]](#) What I do is I use dotnetpatcher for this XD just add a new assembly to dotnetpatcher, and go to dependencies -> merging and select all ur dlls, disable obfuscator (enable it if u want) and then click start then u get ur dependencyless exe in the "Protected folder"

Hope I helped :D

I'll try that, thanks.

(Aug 15, 2020 06:00 PM)Accord Wrote: [\[CITE\]](#) Maybe this is what your need if the dll that youre talking about are netcore library

<https://docs.microsoft.com/en-us/dotnet/core/deploying/>

tried but still outputting on all options the .exe + the .dll lmao.

Suspect. - August 16, 2020, 3:54 am

(Aug 16, 2020 01:57 AM)Sickness Wrote: [\[CITE\]](#) I don't think I can use this, I'm using a Console Application so I don't got InitializeComponent

I'll try that, thanks.

tried but still outputting on all options the .exe + the .dll lmao.

Ah, no prob. In that case use ILMerge.

Sickness - August 16, 2020, 4:01 am

(Aug 16, 2020 03:54 AM)Suspect. Wrote: [\[CITE\]](#) Ah, no prob. In that case use ILMerge.

No joke I am even having issues with that.

<https://www.inogic.com/blog/2018/06/automate-ilmerge-for-workflowsplugins-to-merge-multiple-assemblies-into-one-in-dynamics-365/>

When compiling I get Unexpected error in the task MergeTask.

n0\$f3ratu\$ - August 16, 2020, 6:06 am

PHP Code

```
<?php
public Program()
{
    AppDomain.CurrentDomain.AssemblyResolve += new ResolveEventHandler(
CurrentDomain_AssemblyResolve);
}

System.Reflection.Assembly CurrentDomain_AssemblyResolve(object sender,
ResolveEventArgs args)
{
    string dllName = args.Name.Contains(',') ? args.Name.Substring(0, args.Name.IndexOf(',')) : args.Name.Replace(".dll", "");

    dllName = dllName.Replace(".", "_");

    if (dllName.EndsWith("_resources")) return null;
```

```

        System.Resources.ResourceManager rm = new System.Resources.ResourceManager(
GetType().Namespace + ".Properties.Resources", System.Reflection.Assembly.
GetExecutingAssembly());

        byte[] bytes = (byte[])rm.GetObject(dllName);

        return System.Reflection.Assembly.Load(bytes);
    }

```

Accord - August 16, 2020, 6:27 am

(Aug 16, 2020 01:57 AM)Sickness Wrote: [\[CITE\]](#) tried but still outputting on all options the .exe + the .dll lmao.

is it only netcore library or it youre using 3rd party library?? im planning to study netcore this month
 You can also tell us what dll youre trying to use, or what config did you use in netcore deployment

Sickness - August 16, 2020, 7:03 am

(Aug 16, 2020 06:27 AM)Accord Wrote: [\[CITE\]](#) is it only netcore library or it youre using 3rd party library?? im planning to study netcore this month

You can also tell us what dll youre trying to use, or what config did you use in netcore deployment
 The project doesn't use any library, it just outputs its code to a .dll and a .exe to open it.

```

Code <PropertyGroup>
  <OutputType>Exe</OutputType>
  <TargetFramework>netcoreapp3.1</TargetFramework>
  <StartupObject></StartupObject>
</PropertyGroup>

```

1>----- Operación Recompilar todo iniciada: proyecto: backupEXE, configuración: Debug Any CPU

```

-----
1>backupEXE ->
C:\Users\Sick\source\repos\GitHub\backupEXE\backupEXE\bin\Debug\netcoreapp3.1\backupEXE.d
ll
===== Recompilar todo: 1 correctos, 0 incorrectos, 0 omitidos =====

```

(Aug 16, 2020 06:06 AM)n0\$3ratu\$ Wrote: [\[CITE\]](#) PHP Code

```

<?php
public Program()
{
    AppDomain.CurrentDomain.AssemblyResolve += new ResolveEventHandler(
CurrentDomain_AssemblyResolve);
}

    System.Reflection.Assembly CurrentDomain_AssemblyResolve(object sender,
ResolveEventArgs args)
    {
        string dllName = args.Name.Contains(',') ? args.Name.Substring(0, args.Name.IndexOf(',')
) : args.Name.Replace(".dll", "");

        dllName = dllName.Replace(".", "_");

        if (dllName.EndsWith("_resources")) return null;

        System.Resources.ResourceManager rm = new System.Resources.ResourceManager(

```

```
GetType().Namespace + ".Properties.Resources", System.Reflection.Assembly.  
GetExecutingAssembly());
```

```
byte[] bytes = (byte[])rm.GetObject(dllName);  
  
return System.Reflection.Assembly.Load(bytes);  
}
```

I don't think this can help me? As I said it's not an external library, it's a .dll that its generated when the program is built, it outputs its code inside a .dll and a .exe to open it.

Accord - August 16, 2020, 7:50 am

(Aug 16, 2020 07:03 AM)Sickness Wrote: [\[CITE\]](#) The project doesn't use any library, it just outputs its code to a .dll and a .exe to open it.

```
Code <PropertyGroup>  
  <OutputType>Exe</OutputType>  
  <TargetFramework>netcoreapp3.1</TargetFramework>  
  <StartupObject></StartupObject>  
</PropertyGroup>
```

Add this to the configuration inside propertygroup

```
<RuntimeIdentifier>win-x64</RuntimeIdentifier>  
<PublishSingleFile>true</PublishSingleFile>
```

it should be like this

```
Code <PropertyGroup>  
  <OutputType>Exe</OutputType>  
  <TargetFramework>netcoreapp3.1</TargetFramework>  
  <StartupObject></StartupObject>  
  <RuntimeIdentifier>win-x64</RuntimeIdentifier>  
  <PublishSingleFile>true</PublishSingleFile>  
</PropertyGroup>
```

n0\$3ratu\$ - August 16, 2020, 7:52 am

(Aug 16, 2020 07:03 AM)Sickness Wrote: [\[CITE\]](#) The project doesn't use any library, it just outputs its code to a .dll and a .exe to open it.

```
Code <PropertyGroup>  
  <OutputType>Exe</OutputType>  
  <TargetFramework>netcoreapp3.1</TargetFramework>  
  <StartupObject></StartupObject>  
</PropertyGroup>
```

```
1>----- Operación Recompilar todo iniciada: proyecto: backupEXE, configuración: Debug Any CPU  
-----
```

```
1>backupEXE ->
```

```
C:\Users\Sick\source\repos\GitHub\backupEXE\backupEXE\bin\Debug\netcoreapp3.1\backupEXE.d  
ll
```

```
===== Recompilar todo: 1 correctos, 0 incorrectos, 0 omitidos =====
```

I don't think this can help me? As I said it's not an external library, it's a .dll that its generated when the program is built, it outputs its code inside a .dll and a .exe to open it.

Ohh I see...

Then (I don't know if you already tried that) why don't you use ilmerge through command line to merge both?

Anyway in the end I don't see why you shouldn't let both as they are.

Sickness - August 16, 2020, 7:54 am

(Aug 16, 2020 07:50 AM)Accord Wrote: [\[CITE\]](#) Add this to the configuration inside propertygroup
<RuntimeIdentifier>win-x64</RuntimeIdentifier>
<PublishSingleFile>true</PublishSingleFile>

it should be like this

```
Code <PropertyGroup>
  <OutputType>Exe</OutputType>
  <TargetFramework>netcoreapp3.1</TargetFramework>
  <StartupObject></StartupObject>
  <RuntimeIdentifier>win-x64</RuntimeIdentifier>
  <PublishSingleFile>true</PublishSingleFile>
</PropertyGroup>
```

Changing that makes a folder called win-x64 where it contains a lot of .dlls, net core one's and exactly the other outputs I mentioned before, same behaviour, cannot be started with the .exe alone.

May someone help me over Discord? ayayaya#1371

Somehow I cannot embed the .exe because the .exe is not a .net assembly or something like that.

```
ILMerge.Merge: Could not load assembly from the location
'C:\Users\Sick\source\repos\GitHub\backupEXE\backupEXE\bin\Debug\netcoreapp3.1\backupEXE.
exe'. Skipping and processing rest of arguments.
  en ILMerging.ILMerge.Merge()
  en ILMerging.ILMerge.Main(String[] args)
```

im really triggered right now.

Accord - August 16, 2020, 8:36 am

(Aug 16, 2020 07:54 AM)Sickness Wrote: [\[CITE\]](#) Changing that makes a folder called win-x64 where it contains a lot of .dlls, net core one's and exactly the other outputs I mentioned before, same behaviour, cannot be started with the .exe alone.

May someone help me over Discord? ayayaya#1371

Somehow I cannot embed the .exe because the .exe is not a .net assembly or something like that.

```
ILMerge.Merge: Could not load assembly from the location
'C:\Users\Sick\source\repos\GitHub\backupEXE\backupEXE\bin\Debug\netcoreapp3.1\backupEXE.
exe'. Skipping and processing rest of arguments.
  en ILMerging.ILMerge.Merge()
  en ILMerging.ILMerge.Main(String[] args)
```

im really triggered right now.

Crazy, i tried those before because its one of netcore features, have you tried warp as alternative to ilmerge??

Sickness - August 16, 2020, 9:09 am

(Aug 16, 2020 08:36 AM)Accord Wrote: [\[CITE\]](#) Crazy, i tried those before because its one of netcore features, have you tried warp as alternative to ilmerge??

Hell yeah I made it work, it went from 100kb total to 1mb.

Thanks buddy!

Suspect. - August 16, 2020, 10:12 am

(Aug 16, 2020 09:09 AM)Sickness Wrote: [\[CITE\]](#) Hell yeah I made it work, it went from 100kb total to 1mb.

Thanks buddy!

Did Warp get the job done?

Sickness - August 16, 2020, 10:16 am

(Aug 16, 2020 10:12 AM)Suspect. Wrote: [\[CITE\]](#) Did Warp get the job done?

Hell yeah, I just coded a GUI that uses Warp to merge files.

<https://github.com/SicknessDev/WarpGUI>

Suspect. - August 16, 2020, 10:18 am

(Aug 16, 2020 10:16 AM)Sickness Wrote: [\[CITE\]](#) Hell yeah, I just coded a GUI that uses Warp to merge files. <https://github.com/SicknessDev/WarpGUI>

Good job bud, keep hold of my solution there it works with Winforms incase you need it again for a winforms proj.

Sickness - August 16, 2020, 10:19 am

(Aug 16, 2020 10:18 AM)Suspect. Wrote: [\[CITE\]](#) Good job bud, keep hold of my solution there it works with Winforms incase you need it again for a winforms proj.

Yeah, will use it on the future, thanks mate.