

Creating a .gclcx To License a User Control

The GCLM utility has a command line switch -lc that will allow you to create a .gclcx file that can provide the runtime license needed to instance the control. There are a few parameters you have to provide.

- 1 - the GUID of the C1License activated on the machine (full list at the bottom of this document)
- 2 - The assembly name of the application that will call or instance the user control
- 3 - The assembly name of the user control

For 1, there is a list at the bottom of this document with all the GUID's for C1 products. For 2, this will "lock" the .gclcx file to a specific application, meaning you will need a custom .gclcx file or entry for all application that will use this User Control. You cannot simply reuse the same one over and over.

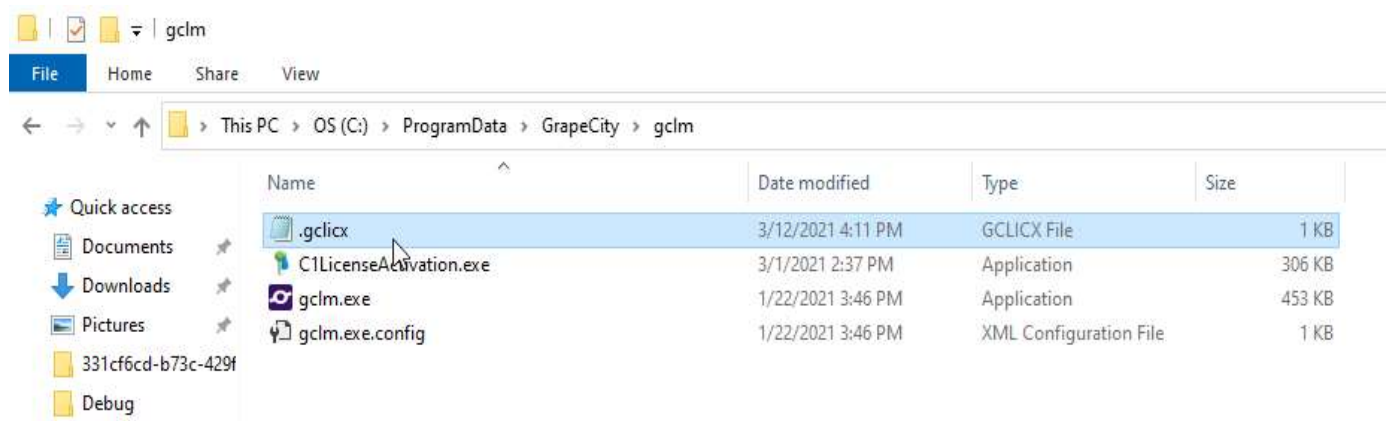
My sample application has the assembly name NewLicenseUserControlTest and the User Control project has the assembly name CustomUserControl and it outputs as a DLL. I have a Studio Enterprise license activated on my dev machine so I am using the GUID "331cf6cd-b73c-429f-ba79-fa2f85eebd68".

Open a command prompt and navigate to the c:\ProgramData\GrapeCity\GCLM folder. Enter the following line into the command prompt but replace the GUID and assembly names with the ones applicable to your project:

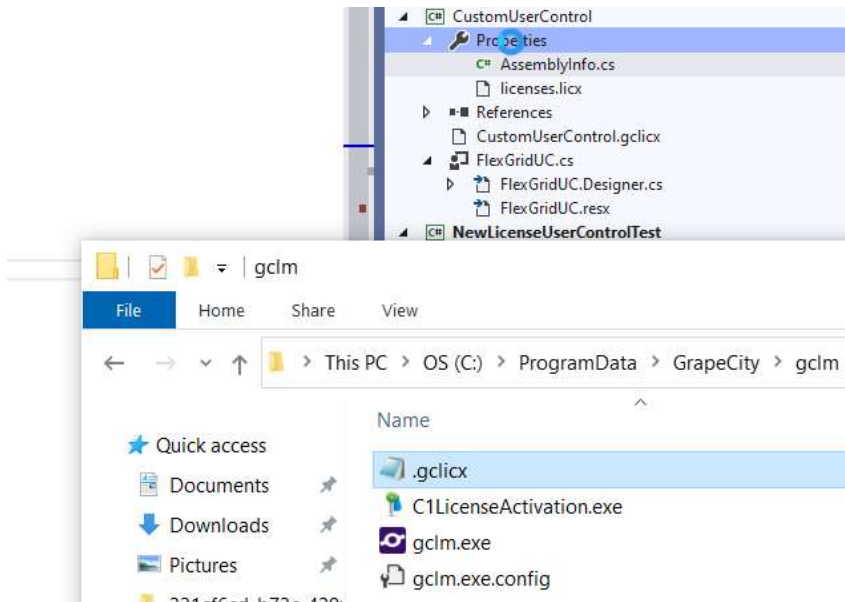
```
gclm.exe "331cf6cd-b73c-429f-ba79-fa2f85eebd68" -lc ./gclcx "NewLicenseUserControlTest.CustomUserControl.dll"
```



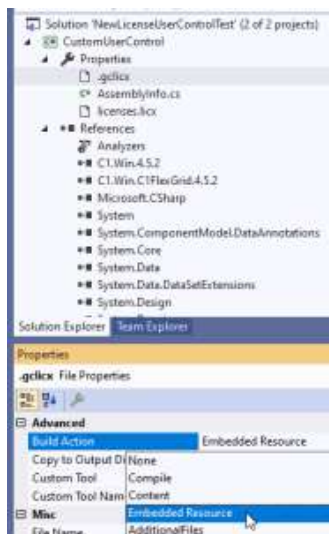
This will produce a .gclcx file in the c:\ProgramData\GrapeCity\GCLM folder



In Visual Studio, navigate to your User Control project and add the .gclcx to the Properties section (you can drag and drop from the file explorer folder or add via Add > Existing Item)



You need to set the Build Action property for the .gclickx file to "Embedded Resource".



Rebuild the application and the project should no longer nag.

GUID references for C1 Products

ComponentOne Xamarin.iOS Edition	c2e333e5-6dae-4aaf-8903-fa1dc779d32a
ComponentOne Xamarin.Android Edition	18dbecd7-edc0-49d0-95d9-bdb9e5e4827f
ComponentOne Studio for Xamarin	de2b5824-e24d-4e7f-86d1-a87c1729993c
ComponentOne UWP Edition	9afa522c-ea0b-47fe-ae14-7d9225612767
ComponentOne Blazor Edition	6631ee67-fec7-45b0-a771-4ec75cd748e3
ComponentOne ASP.NET MVC Edition	839e1737-f256-46ea-b391-50da451c13a4
ComponentOne WPF Edition	c02c28b7-1c24-4109-8eb3-f99c3905f3f1
ComponentOne WinForms Edition	da3d5d14-691f-4908-aa3c-fd3239734232
ComponentOne Studio Enterprise	331cf6cd-b73c-429f-ba79-fa2f85eebd68

You can include a file name before the extension to make identifying and managing the .gclick files easier. For example, this one could be renamed NewLicenseUserControlTest.gclick and it will still work.