How to Get Started with C# 8.0 on Crestron

Version 1.2 of document Written By Tim Gray

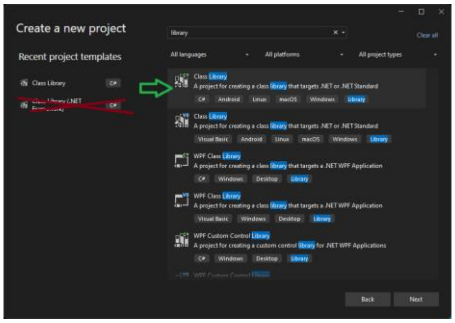
Selecting .NET 8 will get you all the latest and greatest C# features, but there will be no feature difference for Crestron classes between .NET Framework 4.7.2 and .NET 8. However, if you need to develop a SIMPL# library to use in a SIMPL+ module, you can currently only do that with a .NET Framework library. .NET 8 is currently **NOT** supported in SIMPL+ libraries.

One thing that can trip up a C# for Crestron programmer when migrating is that C# 8.0 introduces nullable reference types. This means you must explicitly handle nullability in your code to avoid warnings and potential runtime errors. Do not just blindly slap a "?" on everything to silence the notices in the IDE. Learn how to use this properly. Research Nullable Reference Types (NRT) and their proper use and implementation.

You will have to go over your signatures to decide in each place where you have a reference type whether to leave it non-nullable (string) or make it nullable (string?). Does your method handle null arguments gracefully, or does it immediately throw an exception? If it throws on null, you want to keep it non-nullable to signal that to your callers. Does your method sometimes return null? If so, you want to make the return type nullable to "warn" your method callers about it.

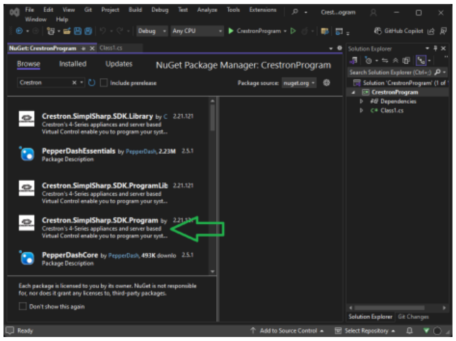
Step 1 – Create the Project for C# 8.0

1. Launch Visual Studio 2022 and ensure you have the latest updates installed to support .NET 8.0
2. Click on 'Create a new project' from the start window.
3. In the project template search box, type 'Class Library' and select 'C# Class Library', do NOT select 'C# Class Library (.NET Framework)' as it targets an older framework.
4. Provide a meaningful project name and choose a location to save your project files.
5. In the Target Framework dropdown, select '.NET 8.0 (Long Term Support)'.


Create a new project
library
Clear all
Recent project templates
Class Library C#
~~Class Library (.NET Framework) C#~~
All languages
All platforms
All project types
Class Library
A project for creating a class library that targets .NET or .NET Standard
C# Android Linux macOS Windows Library
Class Library
A project for creating a class library that targets .NET or .NET Standard
Visual Basic Android Linux macOS Windows Library
WPF Class Library
A project for creating a class library that targets a .NET WPF Application
C# Windows Desktop Library
WPF Class Library
A project for creating a class library that targets a .NET WPF Application
Visual Basic Windows Desktop Library
WPF Custom Control Library
A project for creating a custom control library for .NET WPF Applications
C# Windows Desktop Library
Back
Next

Step 2 – Import Required NuGet Packages

1. In Solution Explorer, click on the solution node at the top.
2. Right-click and choose 'Manage NuGet Packages for Solution'.
3. Navigate to the 'Browse' tab.
4. Type 'Crestron' in the search bar to find relevant packages.
5. Locate and select 'Crestron.SimplSharp.SDK.Program'.
6. Click 'Install' and accept the license agreement when prompted.


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Crest...ogram
Debug
Any CPU
CrestronProgram
GitHub Copilot
NuGet: CrestronProgram
Class1.cs
Browse
Installed
Updates
NuGet Package Manager: CrestronProgram
Crestron
Include prerelease
Package source: nuget.org
Crestron.SimplSharp.SDK.Library by C 2.21.121
Crestron's 4-Series appliances and server based Virtual Control enable you to program your syst...
PepperDashEssentials by PepperDash, 2.23M 2.5.1
Package Description
Crestron.SimplSharp.SDK.ProgramLib 2.21.121
Crestron's 4-Series appliances and server based Virtual Control enable you to program your syst...
Crestron.SimplSharp.SDK.Program by 2.21.121
Crestron's 4-Series appliances and server based Virtual Control enable you to program your syst...
PepperDashCore by PepperDash, 493K downlo 2.5.1
Package Description
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
Don't show this again
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'CrestronProgram' (1 of
CrestronProgram
Dependencies
Class1.cs
Solution Explorer
Git Changes
Ready
Add to Source Control
Select Repository

7. Once installation is complete, the 'Install' button will change to 'Uninstall', indicating success.

Step 3 – Create the ControlSystem Class

1. In Solution Explorer, right-click on the project node.
2. Select 'Add' > 'Class'.
3. Name the new file 'ControlSystem.cs'.
4. Replace the contents of the file with the following bare bones template code:

💡 Tip: You can also copy this code from a working ControlSystem.cs file in an older project or a blank .net framework project that has the full template intact.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Crestron.SimplSharp;
using Crestron.SimplSharpPro;

namespace <PUT YOUR NAMESPACE HERE>
{
    public class ControlSystem : CrestronControlSystem
    {
        public ControlSystem() : base()
        {
            try
            {
                // Initialize Crestron threadpool

Crestron.SimplSharpPro.CrestronThread.Thread.MaxNumberOfUserThreads = 20;
            }
            catch (Exception e)
            {
                ErrorLog.Error("Error in the constructor: {0}", e.Message);
            }
        }

        public override void InitializeSystem()
        {
            try
            {
                // Initialization logic here
            }
            catch (Exception e)
            {
                ErrorLog.Error("Error in InitializeSystem: {0}", e.Message);
            }
        }
    }
}
```