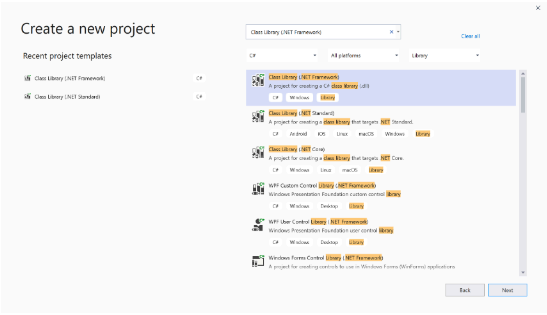
CRESTRON®

Instructions for C# on 4-Series and/or VC-4

1 Create a New Project

In your IDE of choice, simply create a new project of type "Class Library (.NET Framework)" and select version 4.7. Here is what that looks like in some of the popular development environments:

Microsoft Visual Studio 2019 (Windows)



Create a new project

Recent project templates

Class Library (.NET Framework) C#

Class Library (.NET Standard) C#

Search: Class Library (.NET Framework)

Filters: C# · All platforms · Library

Class Library (.NET Framework)
A project for creating a C# class library (.dll)
C# Windows Library

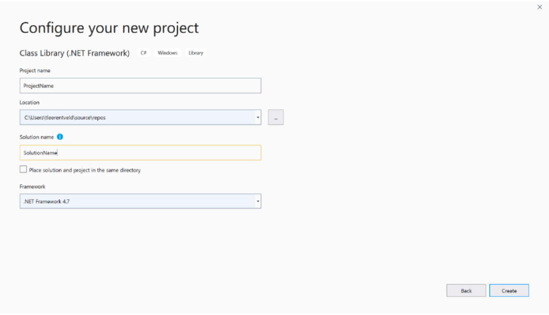
Class Library (.NET Standard)
A project for creating a class library that targets .NET Standard.
C# Android iOS Linux macOS Windows Library

Class Library (.NET Core)
A project for creating a class library that targets .NET Core.
C# Windows Linux macOS Library

WPF Custom Control Library (.NET Framework)
Windows Presentation Foundation custom control library
C# Windows Desktop Library

WPF User Control Library (.NET Framework)
Windows Presentation Foundation user control library
C# Windows Desktop Library

Windows Forms Control Library (.NET Framework)
A project for creating controls to use in Windows Forms (WinForms) applications



Configure your new project

Class Library (.NET Framework) C# Windows Library

Project name: ProjectName

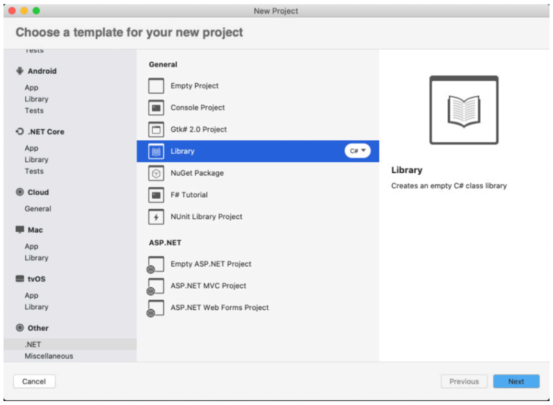
Location: C:\Users\tleerentveld\source\repos

Solution name: SolutionName

☐ Place solution and project in the same directory

Framework: .NET Framework 4.7

Microsoft Visual Studio 2019 (OS X)



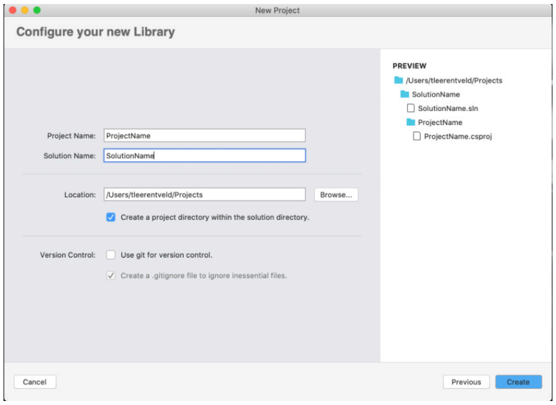
New Project

Choose a template for your new project

General: Empty Project, Console Project, Gtk# 2.0 Project, Library (C#), NuGet Package, F# Tutorial, NUnit Library Project

ASP.NET: Empty ASP.NET Project, ASP.NET MVC Project, ASP.NET Web Forms Project

Library
Creates an empty C# class library



New Project

Configure your new Library

Project Name: ProjectName

Solution Name: SolutionName

Location: /Users/tleerentveld/Projects

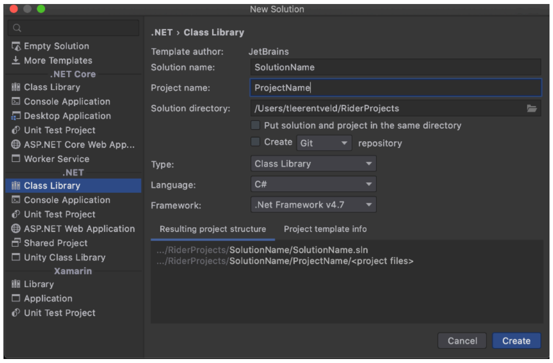
☑ Create a project directory within the solution directory.

Version Control: ☐ Use git for version control.

☑ Create a .gitignore file to ignore inessential files.

PREVIEW
/Users/tleerentveld/Projects
SolutionName
SolutionName.sln
ProjectName
ProjectName.csproj

JetBrains Rider (OS X and Windows)



New Solution

.NET › Class Library

Template author: JetBrains

Solution name: SolutionName

Project name: ProjectName

Solution directory: /Users/tleerentveld/RiderProjects

☐ Put solution and project in the same directory

☐ Create Git repository

Type: Class Library

Language: C#

Framework: .Net Framework v4.7

Resulting project structure

.../RiderProjects/SolutionName/SolutionName.sln
.../RiderProjects/SolutionName/ProjectName/<project files>

Instructions for C# on 4-Series and/or VC-4

2 Add Crestron Package

We have created NuGet Packages to make creating projects for VC-4 and 4-Series easier than ever. Once you have created your solution, and your project, simply right-click on the project, and select "*Manage NuGet Packages*"

Warning: these packages create libraries compatible only with a 4-Series or VC-4.
These projects will not run on a 3-Series Control System.

From there, browse for Crestron, and you'll see three packages. Install the one you need:

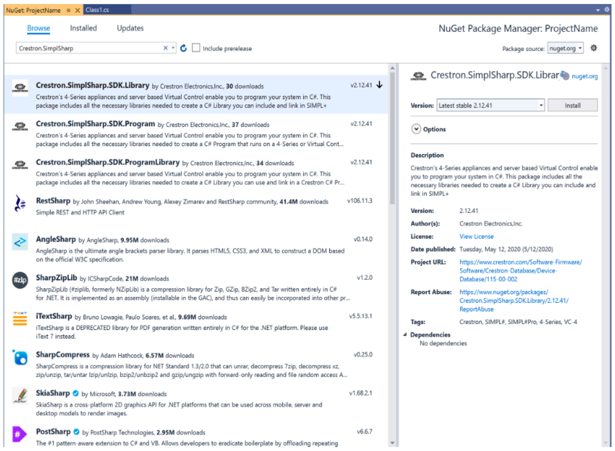
- **Crestron.SimplSharp.SDK.Library**
  This package is required to create a C# Library to be used with SIMPL+ or a Crestron C# Program.
  It will automatically add the required dependencies to your project, and create the CLZ on build.

- **Crestron.SimplSharp.SDK.ProgramLibrary**
  This package is required to create a C# Library that can be used with a Crestron C# Program.
  It will automatically add the required dependencies to your project.

- **Crestron.SimplSharp.SDK.Program**
  This package is required to create a Crestron C# Program.
  It will automatically add the required dependencies to your project, and create the CPZ on build.

Here is what those screens look like in some of the popular development environments:

Microsoft Visual Studio 2019 Windows



Crestron.SimplSharp
☐ Include prerelease
Package source: nuget.org

Crestron.SimplSharp.SDK.Library by Crestron Electronics,Inc., **30** downloads v2.12.41
Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This package includes all the necessary libraries needed to create a C# Library you can include and link in SIMPL+

Crestron.SimplSharp.SDK.Program by Crestron Electronics,Inc., **37** downloads v2.12.41
Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This package includes all the necessary libraries needed to create a C# Program that runs on a 4-Series or Virtual Cont...

Crestron.SimplSharp.SDK.ProgramLibrary by Crestron Electronics,Inc., **34** downloads v2.12.41
Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This package includes all the necessary libraries needed to create a C# Library you can use and link in a Crestron C# Pr...

RestSharp by John Sheehan, Andrew Young, Alexey Zimarev and RestSharp community, **41.4M** downloads v106.11.3
Simple REST and HTTP API Client

AngleSharp by AngleSharp, **9.95M** downloads v0.14.0
AngleSharp is the ultimate angle brackets parser library. It parses HTML5, CSS3, and XML to construct a DOM based on the official W3C specification.

SharpZipLib by ICSharpCode, **21M** downloads v1.2.0
SharpZipLib (#ziplib, formerly NZipLib) is a compression library for Zip, GZip, BZip2, and Tar written entirely in C# for .NET. It is implemented as an assembly (installable in the GAC), and thus can easily be incorporated into other pr...

iTextSharp by Bruno Lowagie, Paulo Soares, et al., **9.69M** downloads v5.5.13.1
iTextSharp is a DEPRECATED library for PDF generation written entirely in C# for the .NET platform. Please use iText 7 instead.

SharpCompress by Adam Hathcock, **6.57M** downloads v0.25.0
SharpCompress is a compression library for NET Standard 1.3/2.0 that can unrar, decompress 7zip, decompress xz, zip/unzip, tar/untar lzip/unlzip, bzip2/unbzip2 and gzip/ungzip with forward-only reading and file random access A...

SkiaSharp by Microsoft, **3.73M** downloads v1.68.2.1
SkiaSharp is a cross-platform 2D graphics API for .NET platforms that can be used across mobile, server and desktop models to render images.

PostSharp by PostSharp Technologies, **2.95M** downloads v6.6.7
The #1 pattern-aware extension to C# and VB. Allows developers to eradicate boilerplate by offloading repeating

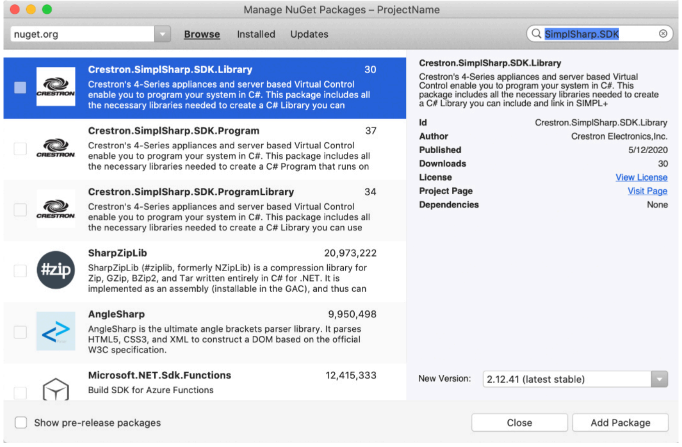
Crestron.SimplSharp.SDK.Librar nuget.org

Version: Latest stable 2.12.41

Description
Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This package includes all the necessary libraries needed to create a C# Library you can include and link in SIMPL+

Version: 2.12.41
Author(s): Crestron Electronics,Inc.
License: View License
Date published: Tuesday, May 12, 2020 (5/12/2020)
Project URL: https://www.crestron.com/Software-Firmware/Software/Crestron-Database/Device-Database/115-00-002
Report Abuse: https://www.nuget.org/packages/Crestron.SimplSharp.SDK.Library/2.12.41/ReportAbuse
Tags: Crestron, SIMPL#, SIMPL#Pro, 4-Series, VC-4
Dependencies
No dependencies

OS X



nuget.org
SimplSharp.SDK

Crestron.SimplSharp.SDK.Library 30
Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This package includes all the necessary libraries needed to create a C# Library you can

Crestron.SimplSharp.SDK.Program 37
Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This package includes all the necessary libraries needed to create a C# Program that runs on

Crestron.SimplSharp.SDK.ProgramLibrary 34
Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This package includes all the necessary libraries needed to create a C# Library you can use

SharpZipLib 20,973,222
SharpZipLib (#ziplib, formerly NZipLib) is a compression library for Zip, GZip, BZip2, and Tar written entirely in C# for .NET. It is implemented as an assembly (installable in the GAC), and thus can

AngleSharp 9,950,498
AngleSharp is the ultimate angle brackets parser library. It parses HTML5, CSS3, and XML to construct a DOM based on the official W3C specification.

Microsoft.NET.Sdk.Functions 12,415,333
Build SDK for Azure Functions

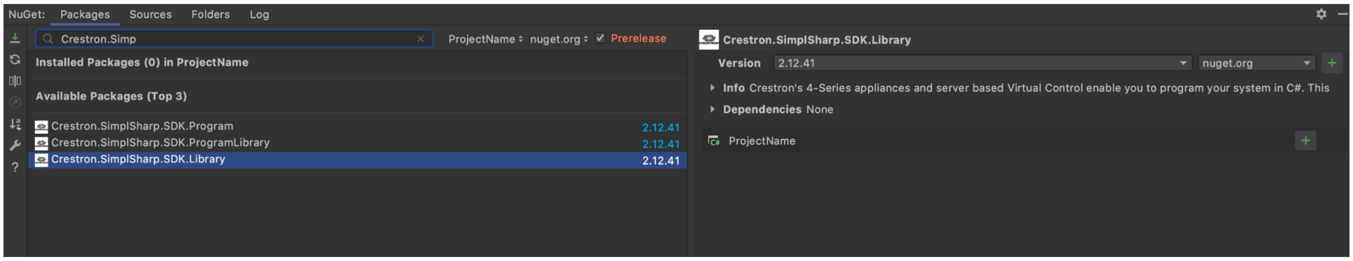
☐ Show pre-release packages

Crestron.SimplSharp.SDK.Library
Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This package includes all the necessary libraries needed to create a C# Library you can include and link in SIMPL+

Id	Crestron.SimplSharp.SDK.Library
Author	Crestron Electronics,Inc.
Published	5/12/2020
Downloads	30
License	View License
Project Page	Visit Page
Dependencies	None

New Version: 2.12.41 (latest stable)

JetBrains Rider (OS X and Windows)



Crestron.Simp
ProjectName nuget.org ☑ Prerelease

Installed Packages (0) in ProjectName

Available Packages (Top 3)

Crestron.SimplSharp.SDK.Program 2.12.41
Crestron.SimplSharp.SDK.ProgramLibrary 2.12.41
Crestron.SimplSharp.SDK.Library 2.12.41

Crestron.SimplSharp.SDK.Library
Version 2.12.41 nuget.org
Info Crestron's 4-Series appliances and server based Virtual Control enable you to program your system in C#. This
Dependencies None
ProjectName

CRESTRON

JetBrains Rider IDE
Debugging C# on 4-Series

1

Setting Up The Control System

Make sure your control system and Rider IDE have the exact same version of code. (Rebuild to be sure).

After you've done a build, use SFTP, and upload the ZIP/CPZ File that the Post-Build step created to the control system. Log into the control system's SSH console and type something like this. You can change the program number and port number to your liking.

```
MC4>PROGLOAD -P:1 -D

MC4>DEBUGPROGRAM -P:1 -Port:50000 -IP:0.0.0.0

MC4>PROGRESET -P:1
```

When you are done debugging, please follow these steps **before detaching your IDE from the processor:**

```
MC4>STOPPROGRAM -P:1

MC4>DEBUGPROGRAM -P:1 -C
```

[Safe to detach your debugger]

```
MC4>PROGRESET -P:1
```

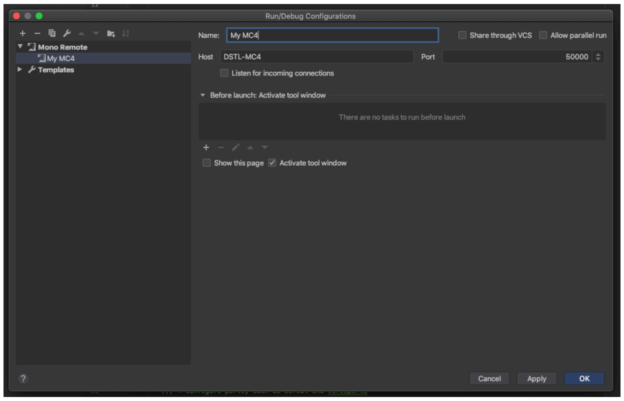
2

Attaching via Rider

In Rider, go to the "Run" menu, and select "Edit Configurations." Click the "+" and select "Mono Remote." Fill in a friendly name, the IP Address / Hostname of your control system, and the port you specified in the console command from section 1.

Click "OK."

Go to "Run" menu, and select "Debug." A window will pop up that lets you select a profile. Pick the profile you just created.



Run/Debug Configurations

Name: My MC4

□ Share through VCS □ Allow parallel run

Host DSTL-MC4 Port 50000

□ Listen for incoming connections

Before launch: Activate tool window

There are no tasks to run before launch

□ Show this page ☑ Activate tool window

Warning

On the 4-Series you cannot attach to a running program. You will always have to restart in debug mode. Once you detach, or stop debugging, you cannot re-attach until you restart the program.

Visual Studio 2017 / 2019 on Windows Debugging C# on 4-Series

1 Prerequisites

In order to debug the 4-Series using Visual Studio 2017 / 2019 on Windows, you need to install a VS Extension called "VSMonoDebugger"

Unfortunately, there is no remote mono debugger available for Visual Studio 2019 on Mac

2 Setting Up The Control System

Make sure your control system and Visual Studio 2019 IDE have the exact same version of code. (Rebuild to be sure).

After you've done a build, use SFTP, and upload the ZIP/CPZ File that the Post-Build step created to the control system. Log into the control system's SSH console and type something like this. You can change the program number and port number to your liking.

```
MC4>PROGLOAD -P:1 -D

MC4>DEBUGPROGRAM -P:1 -Port:50000 -IP:0.0.0.0

MC4>PROGRESET -P:1
```

When you are done debugging, please follow these steps **before detaching your IDE from the processor:**

```
MC4>STOPPROGRAM -P:1

MC4>DEBUGPROGRAM -P:1 -C
```

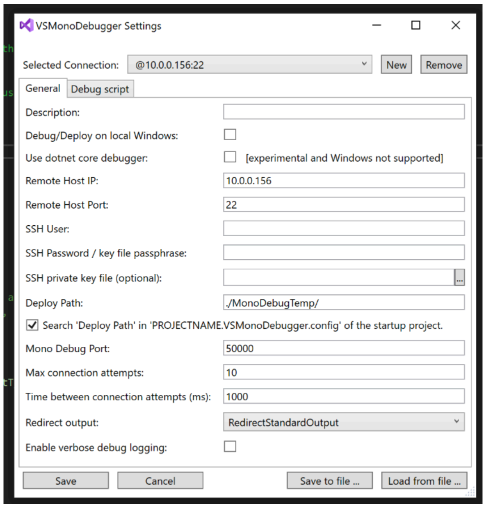
[Safe to detach your debugger]

```
MC4>PROGRESET -P:1
```

3 Attaching via VisualStudio 2017 / 2019

In Visual Studio, go to the "Extensions" Menu, and select "Mono," followed by "Settings." Fill in the Remote Host IP, and the Mono Debug Port. Click "Save."

Once this is done, go to "Extensions," "Mono," and select "Attach to mono debugger." This should allow you to debug your program.


VSMonoDebugger Settings

Selected Connection: @10.0.0.156:22

Description:

Debug/Deploy on local Windows: □

Use dotnet core debugger: □ [experimental and Windows not supported]

Remote Host IP: 10.0.0.156

Remote Host Port: 22

SSH User:

SSH Password / key file passphrase:

SSH private key file (optional):

Deploy Path: ./MonoDebugTemp/

☑ Search 'Deploy Path' in 'PROJECTNAME.VSMonoDebugger.config' of the startup project.

Mono Debug Port: 50000

Max connection attempts: 10

Time between connection attempts (ms): 1000

Redirect output: RedirectStandardOutput

Enable verbose debug logging: □

Warning

On the 4-Series you cannot attach to a running program. You will always have to restart in debug mode. Once you detach, or stop debugging, you cannot re-attach until you restart the program.