SIMPL# plug-in (snap-in) for VS2008

(Quick Start Guide)

Table of Contents

Overview

This document is intended for those, who are using the SIMPL# plug-in to develop libraries and programs in SIMPL#. The language reference and the device library reference can be found from other documentation sources.

Crestron SIMPL# plug-in is an extension to Microsoft Visual Studio 2008 for creating Crestron SIMPL# Lib, SIMPL# Pro and SIMPL# Pro Lib type projects. It allows limiting the environment of Visual Studio to the specifications of SIMPL# as a language and provides the means to use Crestron SIMPL# predefined device libraries.

Prerequisites

SIMPL# plug-in requires a set of software applications and data files to be installed. Its installer checks if these requirements are met and if not, displays corresponding message and prompts the user to install the missing piece.

Here are the requirements, listed in the release notes of the plug-in for:

A. SIMPL#
   a. Microsoft Visual Studio 2008 Professional Edition
   b. Minimum Include4.dat v2.01.18
   c. Minimum Firmware v1.007.0019
   d. Minimum Crestron Toolbox v2.36.541
B. SIMPL# Pro
   a. Microsoft Visual Studio 2008 Professional Edition
   b. Minimum Include4.dat v2.01.037
   c. Minimum Firmware v1.009.0029
   d. Minimum Crestron Toolbox v2.36.715

The include4.dat file is distributed with the Device Database or Crestron Studio. Newer version of firmware may require newer version of include4.dat, so the installation of the corresponding version of Device Database or Crestron Studio has to be verified first.

Installation

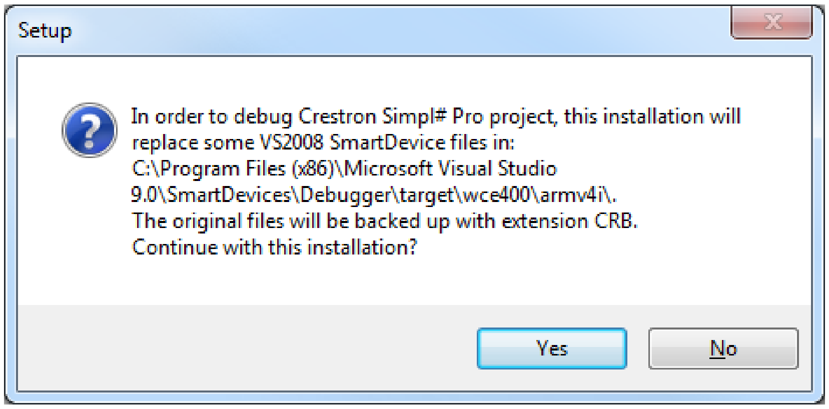
Make sure you have installed and have working Microsoft Visual Studio 2008 with the Microsoft Visual Studio Service Pack 1 installed prior to running the installer.

In the Help | About Microsoft Visual Studio the version should read something like this:

Version 9.0.30729.1 SP

The SP here is the indication you have the Service Pack 1 installed.

When the install is first started, the install program shows a message like:



In order to debug Crestron Simpl# Pro project, this installation will replace some VS2008 SmartDevice files in:
C:\Program Files (x86)\Microsoft Visual Studio 9.0\SmartDevices\Debugger\target\wce400\armv4i\.
The original files will be backed up with extension CRB.
Continue with this installation?

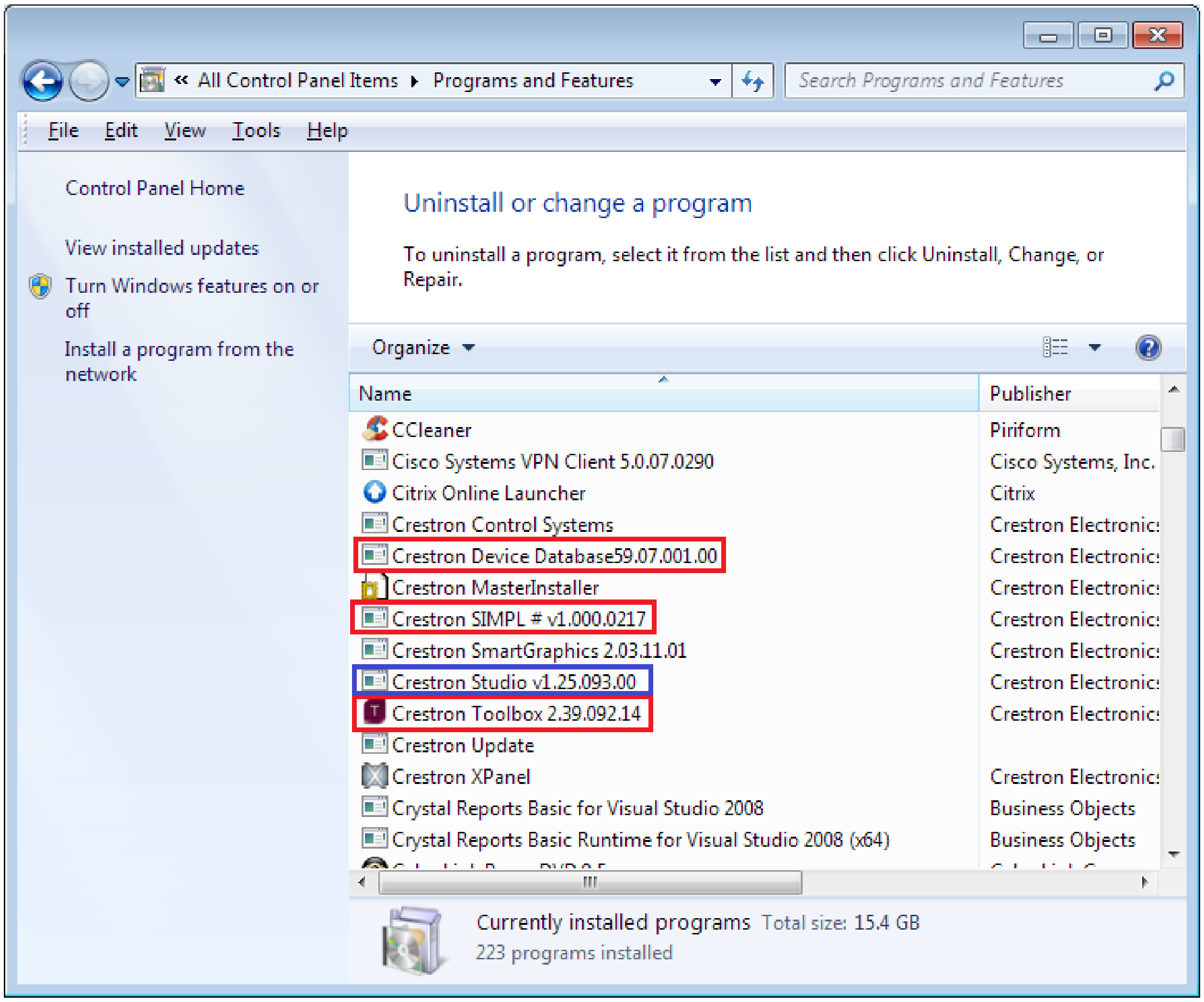
This message stresses on the fact that the installer will change Visual Studio environment by replacing some files and will save a backup of the original files with extension CRB. The user may choose to uninstall the plug-in later and this backup allows restoring the environment to its original state if necessary. Replacing these files during installation of the plug-in is essential for device support in Visual Studio and ensures proper communication with Crestron control systems. The user has to agree to this change explicitly here.

Checking if the software versions are in synch

There are two ways to verify what version of the plug-in is installed in a system.

a. Control Panel

Open Control Panel and select “Programs and Features” in Windows 7 and above or “Add/Remove Programs” in Windows XP. The picture below shows highlighted in red and blue installed software packages – these are the versions we would like to verify. Device Database and Crestron Studio are highlighted with different colors, because either one of them is enough to install, but when both are present, include4.dat file will be the version, determined by the software, which is installed last, even if it did override an existing newer file at the time. It is important to make a note of this fact, because there might be a dependency of the control system firmware version. However for the proper function of the plug-in the version of the include4.dat file should not matter much.


« All Control Panel Items ▸ Programs and Features
Search Programs and Features
File Edit View Tools Help
Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Organize

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Name	Publisher
CCleaner	Piriform
Cisco Systems VPN Client 5.0.07.0290	Cisco Systems, Inc.
Citrix Online Launcher	Citrix
Crestron Control Systems	Crestron Electronic
Crestron Device Database59.07.001.00	Crestron Electronic
Crestron MasterInstaller	Crestron Electronic
Crestron SIMPL # v1.000.0217	Crestron Electronic
Crestron SmartGraphics 2.03.11.01	Crestron Electronic
Crestron Studio v1.25.093.00	Crestron Electronic
Crestron Toolbox 2.39.092.14	Crestron Electronic
Crestron Update	
Crestron XPanel	Crestron Electronic
Crystal Reports Basic for Visual Studio 2008	Business Objects
Crystal Reports Basic Runtime for Visual Studio 2008 (x64)	Business Objects

Currently installed programs Total size: 15.4 GB
223 programs installed

The version of the Crestron SIMPL# plug-in should read “Crestron SIMPL# v.X.XXX.XXXX”. The general rule of thumb is – go with the latest released version.

All the required released software should be available on the Crestron FTP site.

Visual Studio Output window, SIMPL# pane

A special SIMPL# pane also has been added to the output windows, which displays status and current process as well as possible errors during the upload of the project or as a result of a build process. It also displays all the version information for the plug-in as well as such information for some of the important data files and libraries with which the plug-in works, like for example include4.dat file, versions of some of the Crestron SIMPL# libraries, etc.

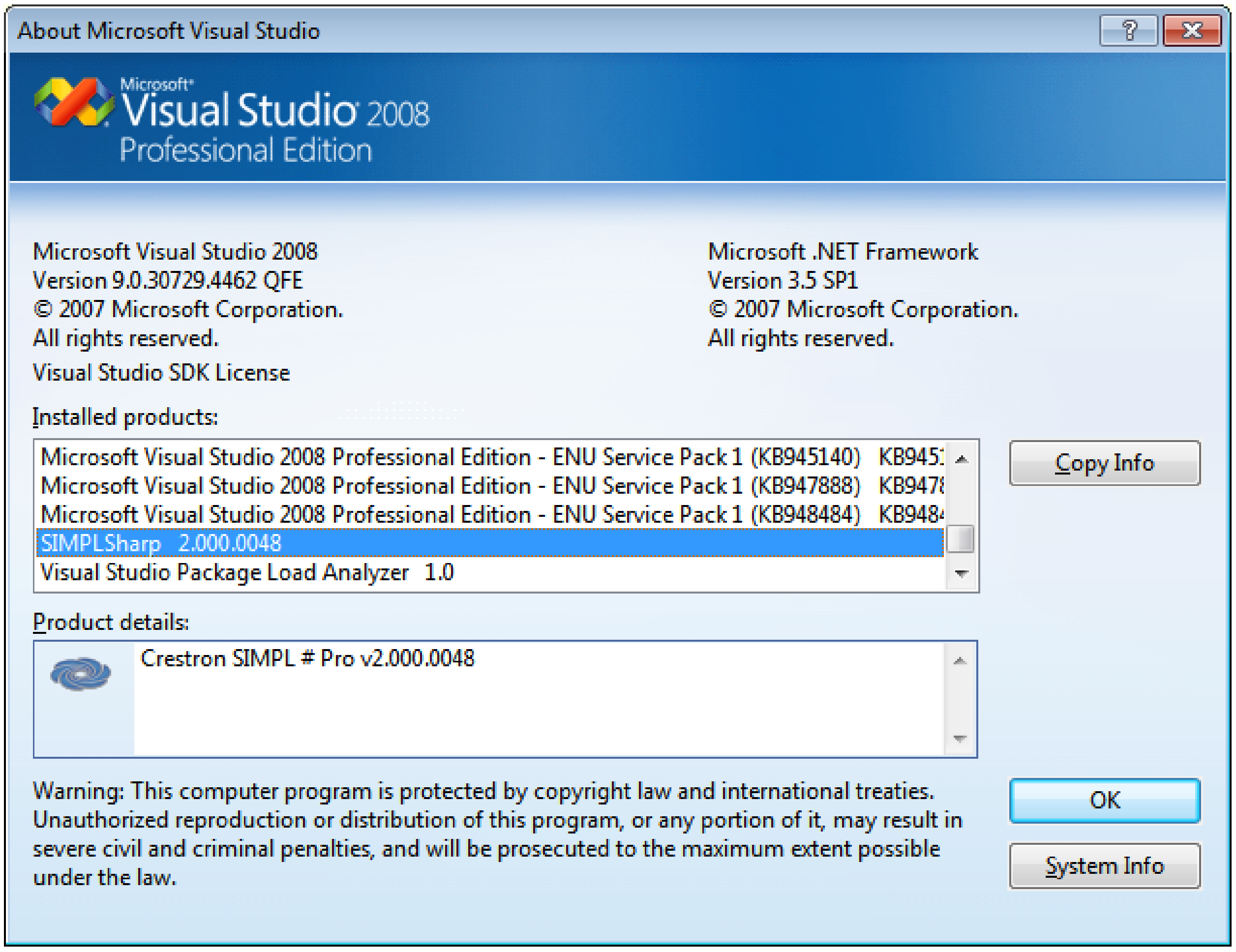
To check plug-in version information is when a new SIMPL#/SIMPL# Pro/SIMPL# Pro Lib project is created in Visual Studio switch to the SIMPL# output window pane by selecting it from the drop-down and similar to the following should be seen:

```
Crestron SIMPL # Initializiation....
Plugin Version: Crestron.SIMPLSharp, Version=2.0.48.0, Culture=neutral,
PublicKeyToken=812d080f93e2de10
Plugin Initialization: Successful
Include4.dat Version: 2.05.003
SDK Initialization: Successful
```

The highlighted numbers are the version of the plugin

b. Visual Studio About box.

Visual Studio about box lists the SIMPL# plugin and when the line for the plugin is highlighted, the version is also displayed in product details, just like on the following screenshot:


About Microsoft Visual Studio
Microsoft Visual Studio 2008 Professional Edition

Microsoft Visual Studio 2008
Version 9.0.30729.4462 QFE
© 2007 Microsoft Corporation.
All rights reserved.
Visual Studio SDK License

Microsoft .NET Framework
Version 3.5 SP1
© 2007 Microsoft Corporation.
All rights reserved.

Installed products:

Microsoft Visual Studio 2008 Professional Edition - ENU Service Pack 1 (KB945140) KB945
Microsoft Visual Studio 2008 Professional Edition - ENU Service Pack 1 (KB947888) KB947
Microsoft Visual Studio 2008 Professional Edition - ENU Service Pack 1 (KB948484) KB948
SIMPLSharp 2.000.0048
Visual Studio Package Load Analyzer 1.0

Product details:

Crestron SIMPL # Pro v2.000.0048

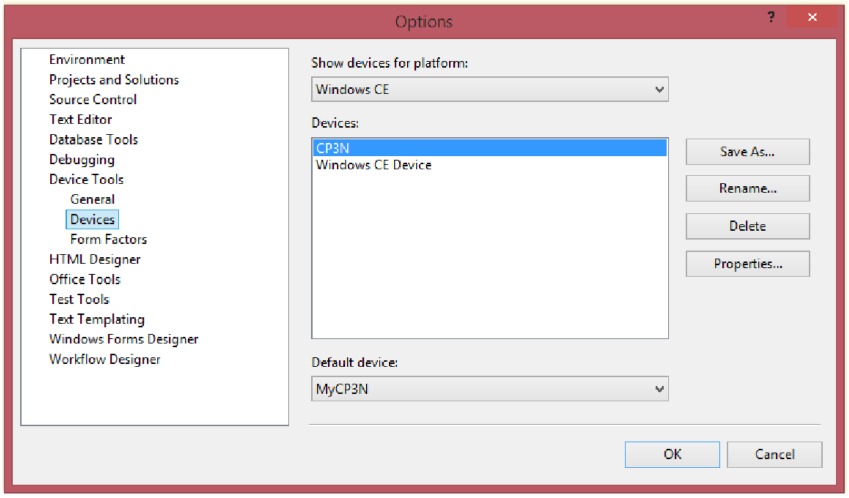
Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

When reporting issues it is recommended, that this version information is included in your report.

Setting up devices (communications)

To properly debug remotely a program on the control system, two types of connections are required and here is how to set them up:

1. Visual Studio TCP/IP connection to a control system.
    a. Create a device, corresponding to your control system with the following steps:
        - In Device Toolbar ("View -> Toolbars -> Device" has to be checked). Click on "Device Options" and in the dialog select "Device Tools -> Devices" as shown below
        - In "Show Devices for Platform" drop-down select "CrestronSDK ARMV7 Device" or "Windows CE"
        - If only one device shows in the list, click "Save As" and name the new one appropriately. Also you may want to select it as a default device.



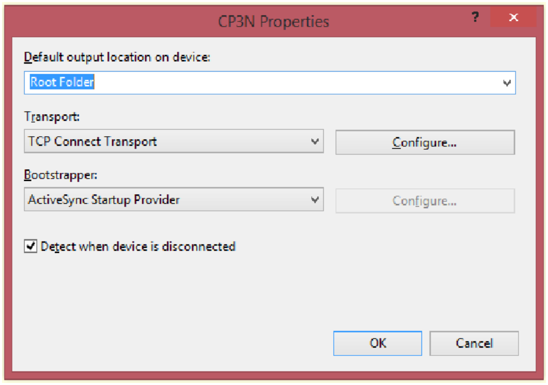
Show devices for platform: Windows CE

Devices: CP3N, Windows CE Device

Default device: MyCP3N

b. Set up the device communication properties.

- Select your device in the list now and click "Properties"
- In the Properties dialog, "Default output location device" select "Root Folder".
- Leave "Transport" as "TCP Connect Transport"
- Leave "Detect when device is disconnected" checked.



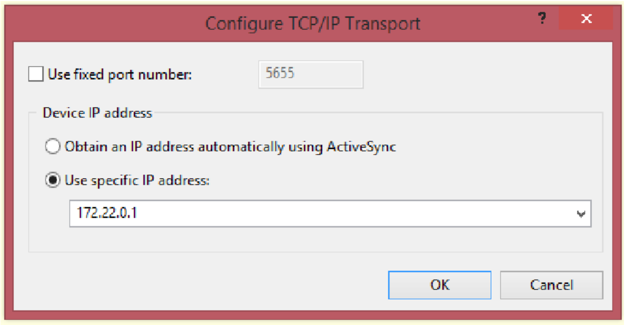
Default output location on device: Root Folder

Transport: TCP Connect Transport

Bootstrapper: ActiveSync Startup Provider

☑ Detect when device is disconnected

- Click Configure and in the TCP/IP configuration dialog select "Use Specific IP Address". Enter the IP of your control system. There is no need to select or check anything else.



☐ Use fixed port number: 5655

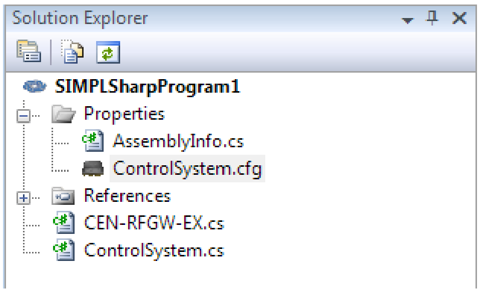
○ Obtain an IP address automatically using ActiveSync

● Use specific IP address: 172.22.0.1

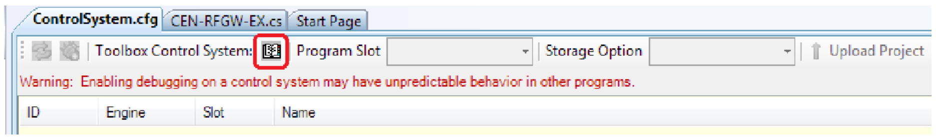
- Click OK and then click OK in the Properties dialog.

- Finally select which device would you want to be your default device to connect to – select your control system, which you named in the one of the steps above.
- Repeat these steps for all control systems you would like to debug remotely in your Visual Studio.

2. Toolbox connection to the corresponding control system.
   a. Making Toolbox connection for your project:
      To use the regular debugging capabilities of the Visual Studio environment it is essential to setup your SIMPL# Pro project to connect properly to your control system. Here is how this is done:
      - Create a new SIMPL# Pro project and name it appropriately.
      - In the solution explorer open Properties section as shown below and double-click on the ControlSystem.cfg to open it:

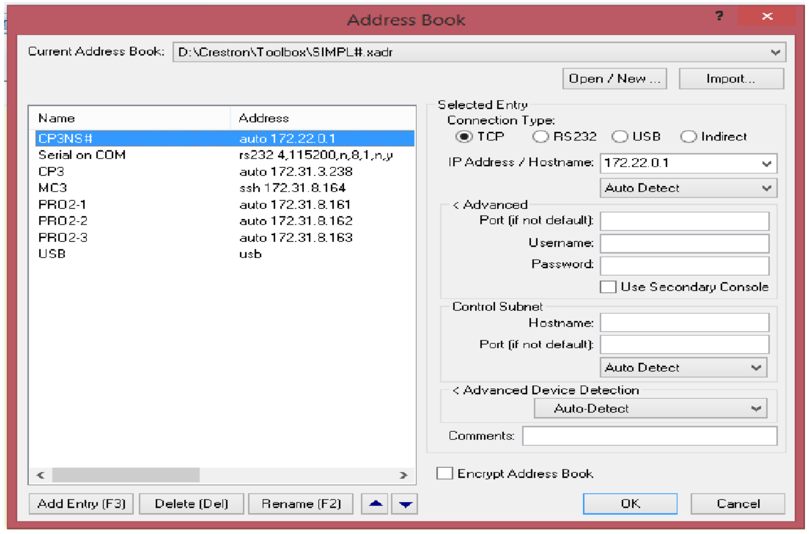


      - In the open ControlSystem.cfg pane across you should see the following:



Warning: Enabling debugging on a control system may have unpredictable behavior in other programs.

      - Click on the highlighted above Address Book button and select your control system, which should have a TCP connection defined, using either IP address or a host name. If you don’t have one, you can define one as you go along in the Address Book dialog:

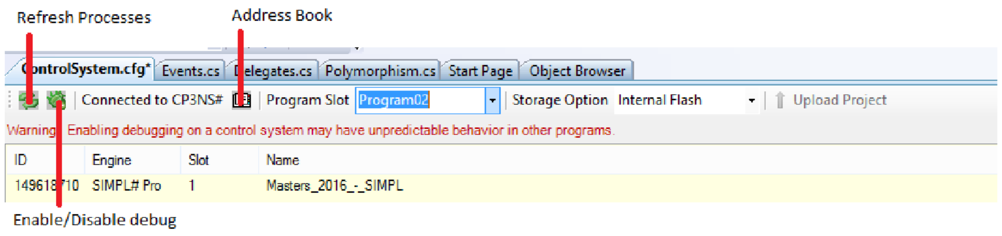

Address Book

Current Address Book: D:\Crestron\Toolbox\SIMPL#.xadr

Name	Address
CP3NS#	auto 172.22.0.1
Serial on COM	rs232 4,115200,n,8,1,n,y
CP3	auto 172.31.3.238
MC3	ssh 172.31.8.164
PRO2-1	auto 172.31.8.161
PRO2-2	auto 172.31.8.162
PRO2-3	auto 172.31.8.163
USB	usb

Selected Entry
Connection Type: [✓ TCP] RS232 USB Indirect
IP Address / Hostname: 172.22.0.1
Auto Detect
Advanced
Port (if not default):
Username:
Password:
Use Secondary Console
Control Subnet
Hostname:
Port (if not default):
Auto Detect
Advanced Device Detection
Auto-Detect
Comments:
Encrypt Address Book

- After you click "OK" the Address Book dialog closes and your project should connect shortly to your control system.
- If connected successfully, you should see ControlSystem.cfg pane change like the following. The picture below also includes little help for the buttons in the ControlSystem.cfg pane.



Connected to CP3NS# Program Slot Program02 Storage Option Internal Flash

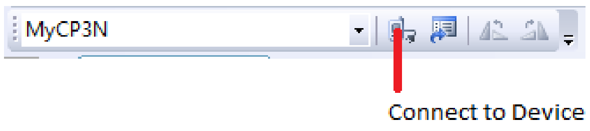
Warning: Enabling debugging on a control system may have unpredictable behavior in other programs

ID	Engine	Slot	Name
14961810	SIMPL# Pro	1	Masters_2016_-_SIMPL

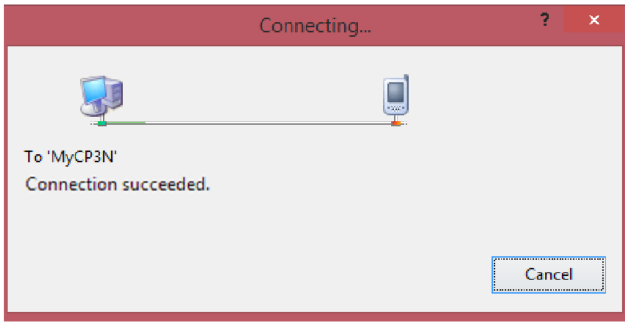
- After successful connection to a control system, it is important to select a program slot, where your program will be uploaded, if single-step debugging is started.
- Storage Option for now has only one choice, so it is set by default to Internal Flash.
- Upload Project button allows to upload and start a program, as if you would do it through other means, say toolbox project upload. It is meant to be used for manual debugging, which will be covered later.

In order to test your Visual Studio connection to the control system you need to enable debugging first (the Enable/Disable button shown above is in status "Debug Enabled" and its balloon help would say "Disable debug" when hovering above it with the mouse pointer).

After both connections are established, **make sure the debugging is on** and you can test the Visual Studio connection by pressing the "Connect to Device" button in the Device toolbar:



If the setup was successful, a confirmation is displayed:



To 'MyCP3N'
Connection succeeded.

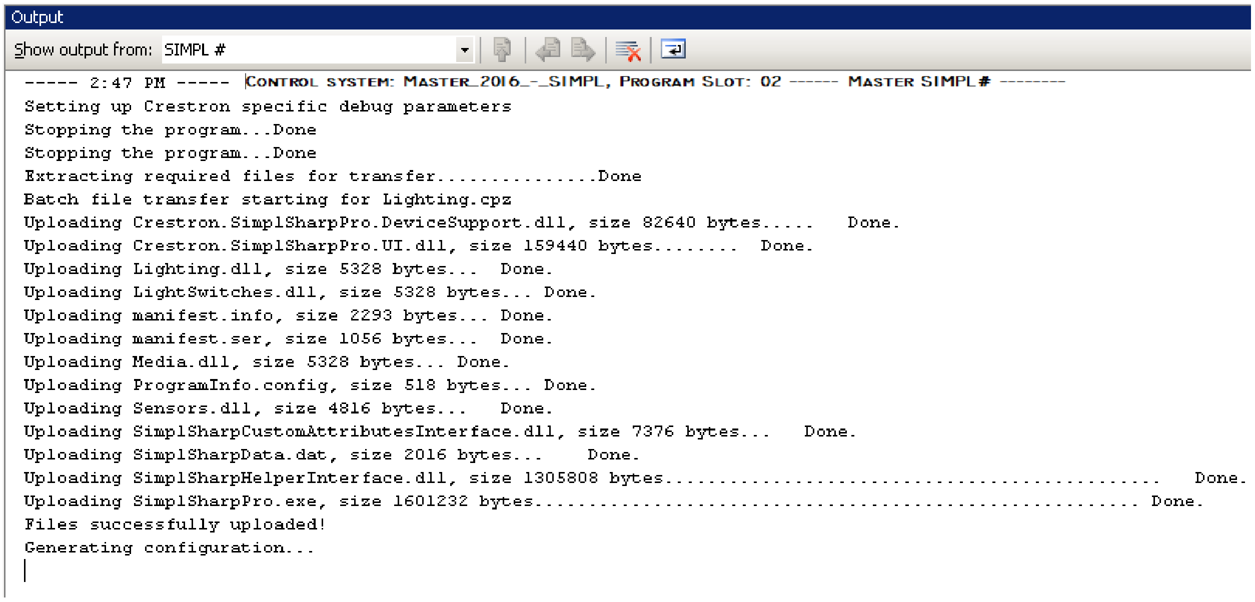
Now the environment is ready for complete SIMPL# software development life cycle.

Single-step (automated) debugging

Single-step debugging feature of the SIMPL# plug-in allows seamlessly debugging remote programs with a click of a button. This process starts by one of the following: clicking on the "Start" green button in the Debug toolbar, selecting "Start" from the Debug menu or pressing F5 key.

SIMPL# plug-in includes a step of automatically deploying the project to the control system and starting the corresponding executable. Since the Crestron control systems are not like the usual embedded devices like smart phones for example, the process requires additional steps to make this work. These steps are provided by the plug-in and because of them deploying and remote debugging of a project on a control system with Visual Studio requires special support.

A progress report also has been added to the SIMPL# pane in the output windows. This report displays status and current process as well as possible errors during the upload of the project or as a result of a build process. Here is an example of the SIMPL# pain content with a successfully deployed project:



Output

Show output from: SIMPL #

```
----- 2:47 PM ----- CONTROL SYSTEM: MASTER_2016_-_SIMPL, PROGRAM SLOT: 02 ------ MASTER SIMPL# --------
Setting up Crestron specific debug parameters
Stopping the program...Done
Stopping the program...Done
Extracting required files for transfer..............Done
Batch file transfer starting for Lighting.cpz
Uploading Crestron.SimplSharpPro.DeviceSupport.dll, size 82640 bytes.....   Done.
Uploading Crestron.SimplSharpPro.UI.dll, size 159440 bytes........   Done.
Uploading Lighting.dll, size 5328 bytes...   Done.
Uploading LightSwitches.dll, size 5328 bytes... Done.
Uploading manifest.info, size 2293 bytes... Done.
Uploading manifest.ser, size 1056 bytes...   Done.
Uploading Media.dll, size 5328 bytes... Done.
Uploading ProgramInfo.config, size 518 bytes... Done.
Uploading Sensors.dll, size 4816 bytes...   Done.
Uploading SimplSharpCustomAttributesInterface.dll, size 7376 bytes...   Done.
Uploading SimplSharpData.dat, size 2016 bytes...    Done.
Uploading SimplSharpHelperInterface.dll, size 1305808 bytes.............................................   Done.
Uploading SimplSharpPro.exe, size 1601232 bytes....................................................... Done.
Files successfully uploaded!
Generating configuration...
```

While the project is uploading, it can be interrupted with Ctrl-Break or through the menu selecting item “Build -> Cancel” like the way the build process is interrupted. The SIMPL# output pane will indicate that the user canceled upload of the project or the upload was interrupted in some way.

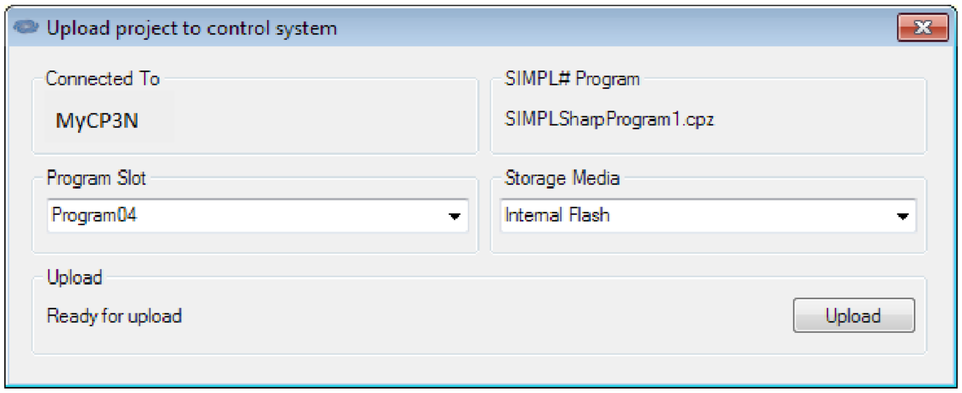
You can debug normally your remote project on your control system like you debug a local project – setting breakpoints even in the constructor of the control system class itself – this allows Visual Studio to load all your libraries and step through the starting process of your project on the control system.

When debug is started, debugging process is automatically enabled on the control system if it was disabled before. When closing the project the debugging is disabled automatically. For all other cases the Enable/Disable debug button can be used to manually set the debug flag.

The ControlSystem.cfg pane shows a list of running processes (if any). It shows the process IDs and program slots as well as the names of the running programs/processes and their engine type.

Manual upload of a program

A “SIMPL# Pro” program can be uploaded manually through the ControSystem.cfg pane by clicking on the “Upload Program” button, which brings up the following upload dialog:



Upload project to control system

Connected To: MyCP3N

SIMPL# Program: SIMPLSharpProgram1.cpz

Program Slot: Program04

Storage Media: Internal Flash

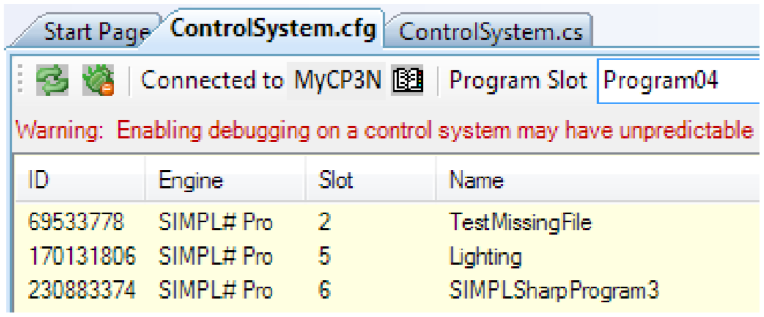
Upload: Ready for upload

It allows uploading of a successfully build project to the selected program slots on the control system. As with the ControlSystem.cfg pane the only destination storage for now is Internal Flash.

After the project is successfully uploaded the program is started automatically. During the upload, the text “Upload” changes to “Cancel”. If the button is pressed, the upload will be canceled, which will also result in a cleanup of the so far uploaded files and of course no program will be started.

Manual debugging

Manual debugging will be covered very briefly here. It is a process of debugging a program or a library, which has been created using the SIMPL# plug-in and uploaded either through outside means like Toolbox for example or through Upload Dialog and is already running on the control system. This requires attaching to the selected process. The process IDs of all running SIMPL# type processes are displayed in the ControlSystem.cfg pane along with their names and program slots:

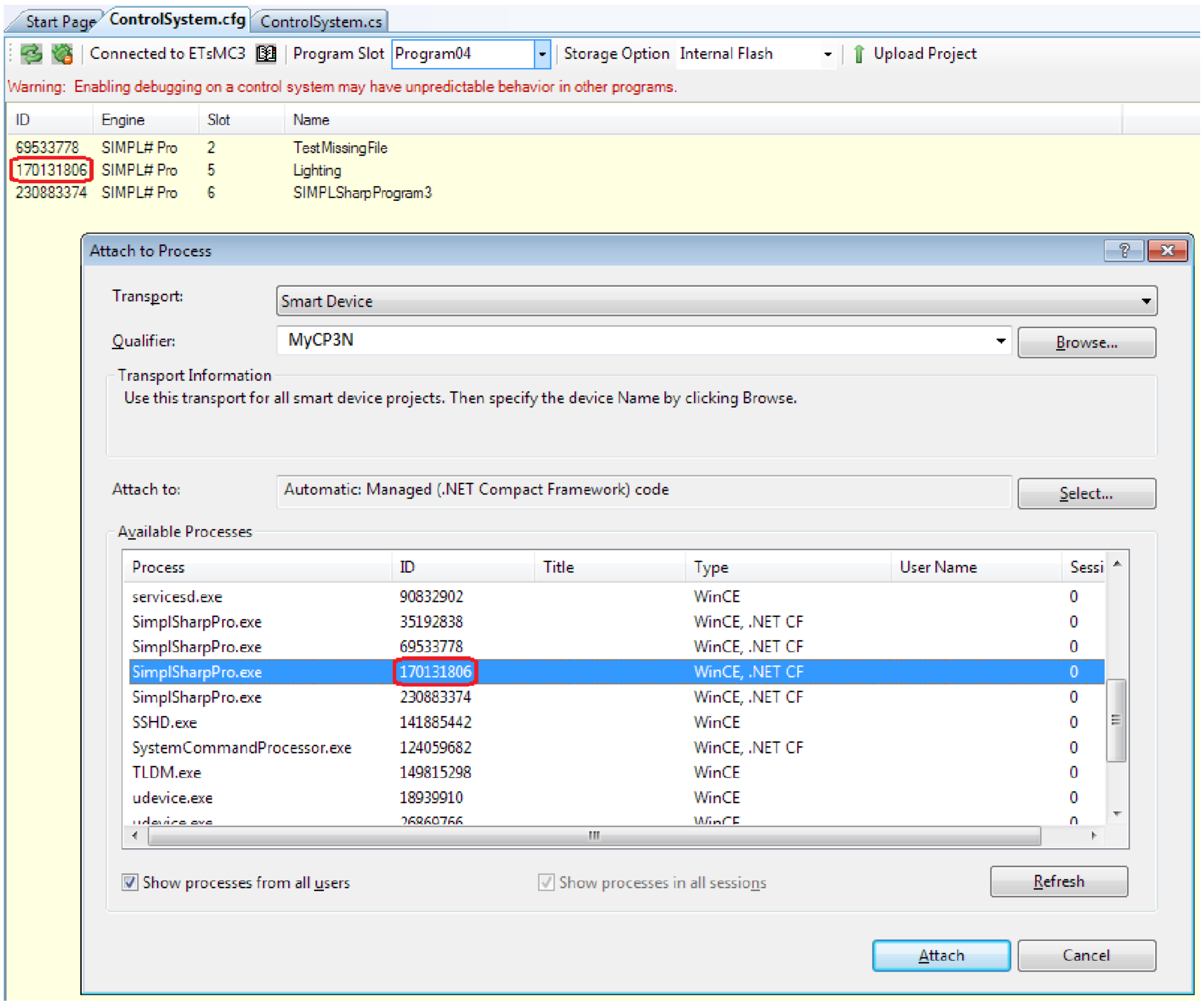


Warning: Enabling debugging on a control system may have unpredictable

ID	Engine	Slot	Name
69533778	SIMPL# Pro	2	TestMissingFile
170131806	SIMPL# Pro	5	Lighting
230883374	SIMPL# Pro	6	SIMPLSharpProgram3

This makes easier to pick a program to attach to. The process comes down to a few steps:

1. Open your SIMPL# Pro Solution and select a project, if you have more than one.
2. Make sure your connections to the selected control system in the project are valid.
3. Select “Debug -> Attach to Process…” from the menu in Visual Studio.
4. The Attach to Process dialog appears on the screen. From the top drop down select “Smart Device” and from the “Qualifiers” drop down select your device, which you configured in “Setting up devices”, section 1.
5. “Available processes” fills up with all the running processes on the control system.
6. Shown on the screen below is a combination of the “Attach to Process” dialog and the ControlSystem.cfg pane. If we choose to attach to our Lighting process, then we can match its ID (170131806 in this case) in both windows, select this line in the dialog and then press “Attach”.



Start Page | ControlSystem.cfg | ControlSystem.cs

Connected to ETsMC3 | Program Slot Program04 | Storage Option Internal Flash | Upload Project

Warning: Enabling debugging on a control system may have unpredictable behavior in other programs.

ID	Engine	Slot	Name
69533778	SIMPL# Pro	2	TestMissingFile
170131806	SIMPL# Pro	5	Lighting
230883374	SIMPL# Pro	6	SIMPLSharpProgram3

Attach to Process

Transport: Smart Device

Qualifier: MyCP3N

Transport Information

Use this transport for all smart device projects. Then specify the device Name by clicking Browse.

Attach to: Automatic: Managed (.NET Compact Framework) code

Available Processes

Process	ID	Title	Type	User Name	Sessi
servicesd.exe	90832902		WinCE		0
SimplSharpPro.exe	35192838		WinCE, .NET CF		0
SimplSharpPro.exe	69533778		WinCE, .NET CF		0
SimplSharpPro.exe	170131806		WinCE, .NET CF		0
SimplSharpPro.exe	230883374		WinCE, .NET CF		0
SSHD.exe	141885442		WinCE		0
SystemCommandProcessor.exe	124059682		WinCE, .NET CF		0
TLDM.exe	149815298		WinCE		0
udevice.exe	18939910		WinCE		0
udevice.exe	26869766		WinCE		0

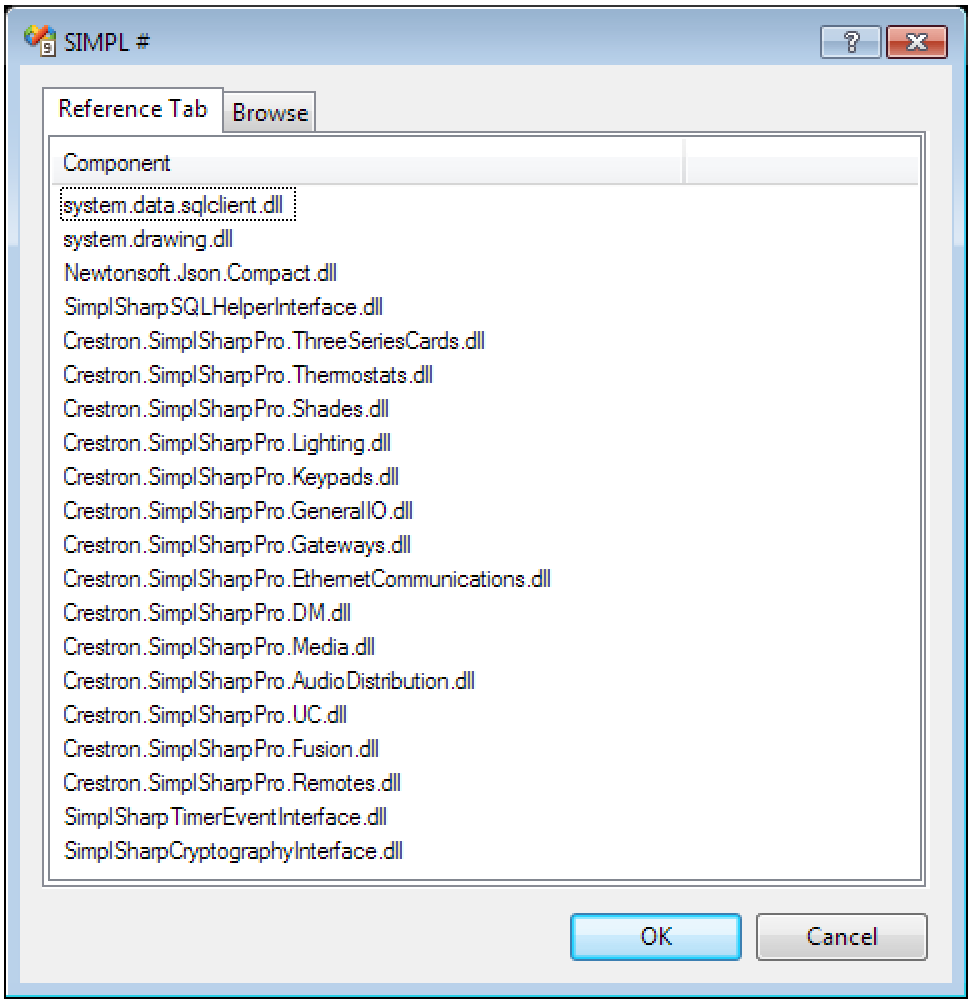
[✓] Show processes from all users [✓] Show processes in all sessions

Assuming you have loaded the source of your library/program and you have set breakpoints where you would like to debug, now you can navigate through the execution steps of the processes to reach your breakpoints.

This process is a good alternative to the single-step debugging, but it would not allow you to set breakpoints early in the execution of the program, for example during the initialization process or the constructor of your control system class.

Using Crestron SIMPL# device libraries

Crestron SIMPL# plug-in provides the means to reference the Crestron device library. To do this, you need to just right-click on the References in Solution explorer under your current project and select “Add Reference”. The dialog has a special section with tab name SIMPL#, which allows adding reference to the pre-build Crestron SIMPL# libraries. It also allows adding reference to user SIMPL# libraries.



system.data.sqlclient.dll
system.drawing.dll
Newtonsoft.Json.Compact.dll
SimplSharpSQLHelperInterface.dll
Crestron.SimplSharpPro.ThreeSeriesCards.dll
Crestron.SimplSharpPro.Thermostats.dll
Crestron.SimplSharpPro.Shades.dll
Crestron.SimplSharpPro.Lighting.dll
Crestron.SimplSharpPro.Keypads.dll
Crestron.SimplSharpPro.GeneralIO.dll
Crestron.SimplSharpPro.Gateways.dll
Crestron.SimplSharpPro.EthernetCommunications.dll
Crestron.SimplSharpPro.DM.dll
Crestron.SimplSharpPro.Media.dll
Crestron.SimplSharpPro.AudioDistribution.dll
Crestron.SimplSharpPro.UC.dll
Crestron.SimplSharpPro.Fusion.dll
Crestron.SimplSharpPro.Remotes.dll
SimplSharpTimerEventInterface.dll
SimplSharpCryptographyInterface.dll

The above shows an example list of Crestron provided libraries that can be used in your program. Please note that every time you add a reference to a library, it will disappear from this list.

Reference for these libraries should be available in a different source, not in this document.

Other sources of SIMPL# documentation

Both the language and the device library reference can be found here:

https://reference.crestron.com

Device Database location for the help file is SIMPLSharpPro.chm.

This can be located in the default director or the one you installed. Something like:

D:\Program Files (x86)\Crestron\Cresdb\Help

Appendix A

Troubleshooting

1. Connections
   a. If the Toolbox connection to your control system does not seem to be working:
      i. Bring up AddressBook and make sure the connection types and addresses look right.
      ii. Bring up Toolbox standalone and make sure the console shows a prompt for the connection in your program.
      iii. If you have multiple address books, pick the right one from the dro-down box in the AddressBook dialog.
   b. If you have trouble with your Visual Studio connection:
      i. Make sure you have defined the right device in the right device platform. The platforms, which can be used to create a device and successfully use it for debugging SIMPL#/SIMPL# Pro projects are “Windows CE” and “CrestronSDK”
      ii. Make sure the connection properties for the selected device are correct and correspond to the connection properties of the device, accessed through the Toolbox connection – usually they have the same IP address or hostname in the Toolbox connection corresponds to the IP address in the Visual Studio device connection properties dialog.
      iii. If the Visual Studio connection fails (given that the debug on the control system is enabled of course), there is a chance the file, associated with Visual Studio devices has been corrupted. This file is located in your Local folder under:

<drive>:\Users\<yourrusernamehere>\AppData\Local\Microsoft\CoreCon\1.0

In fact it is recommended to wipe out the whole folder and let VisualStudio recreate all configuration files in it.

2. Single-step debugging and attaching to processes
   a. Single-step debugging
      i. Make sure you have both connections (VS2008 and Toolbox) as described above, working as expected.
      ii. Check if you have the latest required software both from Crestron and Visual Studio side.
      iii. Make sure VS 2008 is running as Administrator and it has enough privileges to write in your destination folder.
      iv. Check and clean from time to time all temporary folders – there might be a locked file there, which could be associated with loaded libraries, which have not been unloaded from memory.

v. Make sure your control system has the latest released firmware and is working properly.

b. Attaching to running processes

i. Make sure you can connect to the control system through VS 2008 Smart Device toolbar with your custom defined (copied from prototype) control system device definition.

ii. See items ii, iii, iv, v and iv from above.