

Crestron Code – Definition of Terms

The purpose of this document is to define the components of a Crestron code archive, and to define a variety of terms commonly encountered when working with a Crestron programmer.

Much confusion currently surrounds the non-programmer's understanding of Crestron programming. Standardizing a set of common terms and definitions will enable programmers and non-programmers alike to communicate more effectively about project expectations and deliverables. In particular, independent programmers will be able to provide this documentation to their clients and other project stakeholders.

1. General Terms and Definitions

Source code is any collection of statements or declarations written in some human-readable computer programming language. Source code is the means most often used by programmers to specify the actions to be performed by a computer.

A **Compiler** is a computer program (or set of programs) that transforms source code written in a programming language.

An **Executable** file causes a computer to perform indicated tasks according to encoded instruction. The executable file is not human-readable.

Firmware is fixed, usually small updatable programs and data structures that internally control various subsystems within electronic devices such as processors, touch panels, interface modules, etc.

An **Interface** is any device an end user can utilize to operate and control equipment and perform operations. Interfaces include remote controls; touch panels, and web-based controls.

'Wizard' Programs are special pieces of software, which use a guided process to allow lesser-experienced programmers to develop and program Crestron systems.

2. Crestron Software

Crestron provides a variety of software to program, control, set-up, and troubleshoot the hardware it manufactures. It is important to distinguish between each of these software programs as they are used to develop the software that runs the equipment installed on a jobsite.

- a. **SIMPL Windows** - This is the heart of all Crestron program development, and is used by Crestron hardware to perform all of the 'work' in a system. This software is used to define touch panels, hardware devices, interfaces, user modules, SIMPL+ modules, and infrared interface files. Each of these objects are linked together by the programmer to address the needs of the project. When the SIMPL Windows project is complete it gets compiled into an executable form and uploaded to the processor for execution. When the programmer finishes with the SIMPL Windows Project it should be converted into a project archive. *(The typical file extensions created by SIMPL Windows are .smw and .usm. See glossary for more details.)*
- b. **VTPro-E** – This is the graphical portion of Crestron software. It is used to create the interface users see and touch. Template or custom graphics are used here to create a custom user interface that is loaded to a touch panel or screen. *(The typical file extensions created by VTPro-E are .hex, .vtz and .vtp (see glossary for more details.)*

- c. Crestron Toolbox - This software is used to set-up, configure, and troubleshoot Crestron hardware. An address book file can be created with Toolbox to note how each piece of Crestron hardware can communicate with the system.. (Example: RS-232, and IP Addresses) It is helpful to have the address book file for future service of a system. *(The typical file extension created by Toolbox is .adr. See Glossary for more details.)*
- d. SystemBuilder – This is a “Wizard” based program created by Crestron to allow users to create a complete automation program using a guided process. SystemBuilder uses other Crestron programs, such as SIMPL Windows and VTPro-E behind the scenes to develop the final project files. SystemBuilder creates a series of folders in which the project files are placed after the system is saved and compiled. *(The typical archived file extension created by SystemBuilder is .sba. See glossary for more details.)*
- e. D3Pro – This is a “Wizard” based program created by Crestron to allow users to create a complete lighting automation program using a guided process. D3Pro uses other Crestron programs, such as SIMPL Windows and VTPro-E behind the scenes to develop the final project files. *(The typical archived file extension created by D3Pro is .d3a. See glossary for more details.)*
- f. Deal for Windows– This software is used in conjunction with a Crestron CNX-LIR device to capture the infrared codes from a manufacturer’s remote control. *(The typical file extension created by Deal for Windows is .ir. See glossary for more details.)*
- g. Engraver – This software is used to define and order labels for buttons on keypads and touch panels. *(The typical file extension created by Engraver is .egr. See glossary for more details.)*

3. Project Archive

Crestron project archives are the collection of all the components used to create a Crestron automation program. Once a program has been completed, the Crestron software gives the programmer the ability to ‘archive’ the associated files into a simple package for delivery to the client, long-term storage, and updating by a programmer. Typical files included are:

- a. SIMPL Windows source code files *(see below)*
- b. Crestron or custom user modules *(see below)*
- c. SIMPL+ header and source code files *(see below)*
- d. VTPro-E Files (from ‘wizard’ based programs)
- e. Infrared files *(see below)*

4. Modules

- a. **Crestron User Modules or Custom User Modules** are collections of code written to perform specific functions and can be included within a program. These modules generally control specific hardware devices such as projectors, players or provide a software interface to complex devices. Crestron offers a large number of these modules and programmers may create these modules to address specific interface requirements when the need arises. The benefit from using modules is Source code re-usability. Once a programmer creates a solution, they can compile that solution into a module and re-use it many times.
- b. **SIMPL+** is a programming language developed by Crestron to allow programmers familiar with procedural based languages to write custom modules similar to User Modules. SIMPL+ modules provide the same benefits of source code re-usability.

Note about modules: The Crestron programming platform is modular in nature, meaning that each project can have a code framework (or architecture) with individual devices or functions contained in smaller units that can be “plugged in” to the framework. Frequently, programmers save and reuse several of these smaller units of code that they have created and saved for specific purposes. These smaller units are referred to as “modules,” and there are two types:

- i. **A protected module** means the module has been password protected but still can be used in the code; however, even in the uncompiled source code, the core function of the module itself cannot be changed (except by the original programmer) because it is password protected. Many programmers invest significant resources developing custom modules, and some independent programmers have a significant business in selling these modules to other programmers for use in a specific project. Essentially, depending on where the module originated, the programmer creating the uncompiled source code for a given project may not have complete access to the individual modules within the uncompiled framework.
- ii. **An unprotected module** means that the module can be opened and edited.

5. Infrared Files

Infrared files are a collection of waveforms that represent each button on an infrared remote control. These files are the result of a “Capture” process performed by Crestron or a programmer with a Crestron CNX-LIR. Once these waveforms have been captured, they are included into a SIMPL Windows project and used to control infrared-based devices such as a DVD Player or a Cable TV set top box. Every Crestron dealer or programmer should have a CNX-LIR.

6. Compiled vs. Uncompiled Source Code

Compiled source code is the completed code, which can be reloaded into a processor in the event of a failure. This code is similar to an electronic document that has been saved as a .pdf. The document can be read but not easily edited. Typical compiled file types are:

- SIMPL Windows – [filename].spz
- VTPro-E – [filename].vtz

Uncompiled source code is programming that is saved in a version, which can be modified at a later time. It includes all of the current modules, infrared files, and coding necessary to make changes and updates the system in question. Typical compiled file types are:

- SIMPL Windows – [filename].smw
- VTPro-E – [filename].vtp

A note about custom graphics: Some of the custom graphics seen on touch panels are actually created in Photoshop or some other graphics program. These graphics are then exported to the uncompiled interface file. Depending on how the designer constructs the graphics file, changes to the graphics and artwork must sometimes be made in the original graphic file using the original graphic program. It is important to note that an uncompiled, editable file may contain some custom graphics that are not as easily editable as “uncompiled” would otherwise imply.

Glossary of File Types

[filename].ir – This is a file created by the Deal for Windows software. It is an infrared capture of a manufacturer's remote control buttons.

[filename].d3a - This file is an archive of a D3Pro project. When unzipped, this file contains a variety of folders in which the project resides. This file type can be imported into D3Pro for future modification and editing. It is recommended that only D3Pro edit these files.

[filename].egr - This is a file created by the Engraver software. The file contains all of the detailed information regarding how the engraving of keypad and touch panel buttons must be performed. It also contains the dealer and project information for the ordering of the engraved buttons. It is recommended that only SystemBuilder edit these files.

[filename].sba - This file is an archive of a SystemBuilder project. When unzipped, this file contains a variety of folders in which the project resides. This file type can be imported into SystemBuilder for future modification and editing.

[filename].smw - This is a SIMPL Windows program file detailing the use of a module or programming. When compiled, it generates a .spz file. This file can be edited but will only include .ir files or modules that currently reside in the database.

[filename].spz - This is a SIMPL Windows program file that should be located in the same folder as the same filename .smw file. This is the compiled file for the same name .smw file. **Note:** The .spz file cannot be opened or modified in SIMPL Windows.

[filename].umc – This is a User Module created in SIMPL Windows. It can be opened in SIMPL Windows and modified.

[filename].usp – This is a user module created in SIMPL+. The .usp file is the SIMPL+ source file and can be opened and modified in SIMPL+.

[filename].vtp – This is a file created in VTPro-E. It contains all of the graphics; touch panel configuration, and user interface information. When compiled, it creates a .vtz or .hex file for most touch panels. This file is editable for any future for changes to a touch panel.

[filename].vtz – This is a file created when a .vtp is compiled. It is the file, which is loaded to a touch panel device. **Note:** The .vtz file cannot be opened or modified in VTPro-E.

We need your help! Please contact any members of the board for questions and suggestions!

CAIP Advisory Board Members

Business Development Subcommittee

Richard Blackwell
AV Works
richard.blackwell@avworks.com
(866) 700-7233

Howard Nunes (Board President)
Pepperdash, Inc.
hnunes@pepperdash.com
(781) 910-3222

Micha Thurlow
Pantech Design LLC
mthurlow@pantechdesign.com
(469) 585-0998

CAIP Quality Assurance Subcommittee

Jim Felderman
Control Works
jim@controlworks.com
(440) 449-1100

Steve Greenblatt
Control Concepts
steveg@controlconcepts.net
(201) 797-7900

Matt Grisafe
AV Programming Associates
mgrisafe@avprogramming.com
(760) 632-6996

Dealer Communication Subcommittee (authors of this document)

Bernard Morgan
ICS+
bmorgan@icsplusonline.com
(512) 301-0600

Jeff Coryell
Code Jockeys, Inc.
jeff@codejockeys.net
(813) 864-1010

Michael Landt
Lakeshore Software
mlandt@cfl.rr.com
(386) 547-2202

Crestron Members

Shelly Flynn
Crestron (West)
sflynn@crestron.com
(800) 827-2188

Steve Samson
Crestron (NJ)
ssamson@crestron.com
(800) 237-2041

Rich Sasson
Crestron (NJ)
rsasson@crestron.com
(800) 237-2041