3-Series
New Programming Languages

Programming a 3-Series

- **SIMPL Windows and SIMPL+ are not going away!**
- But we need support for more advanced features:
    - Database Connections
    - Floating point
    - XML / HTML parsing
    - Active Directory
    - Runtime exception handling
    - Runtime auto discovery

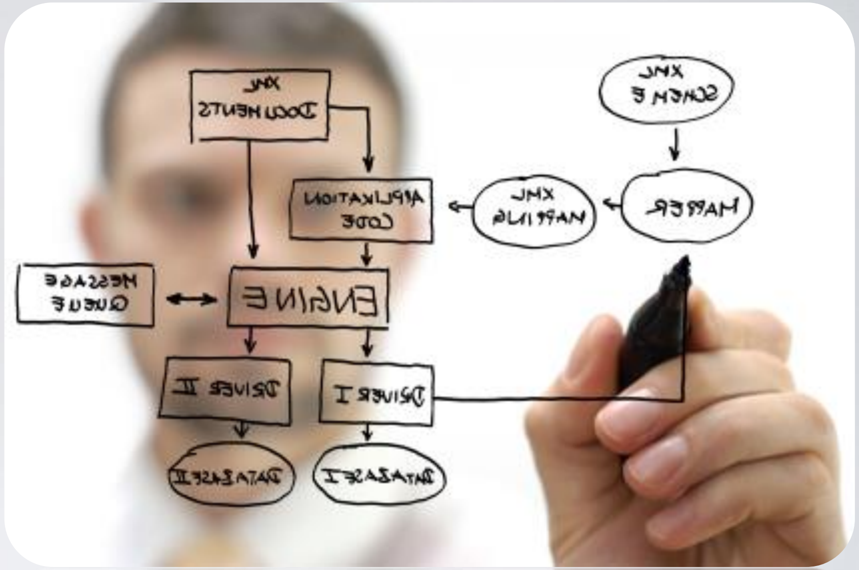
New Language

The new language will be:

- Standards Based
- Flexible and Scalable
- Sandboxed to protect the integrity and reliability of the control system.

Business Benefits of C#

- Existing pool of trained programmers.
- Promotes modularity and code-reuse
- Native Integration with online services
- Hardware Independent Code
- Vast online resources



Crestron's Implementation of C#

SIMPL#

- A Library can be incorporated into SIMPL+
- Use C# features in existing SIMPL Windows Programs
- Source code need not be shared.

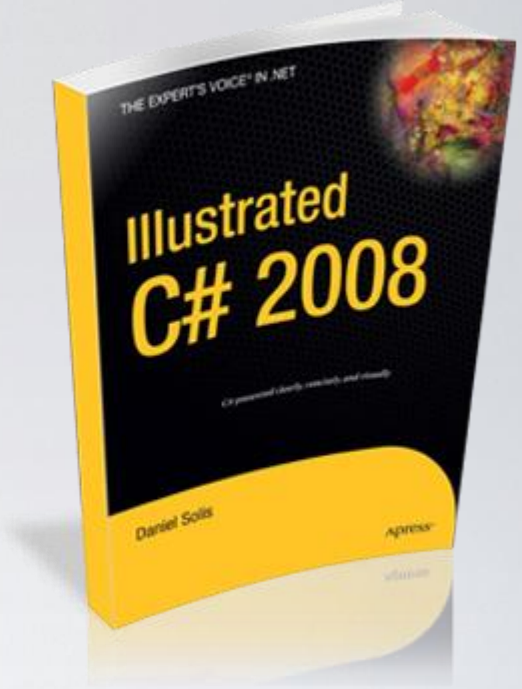
SIMPL# Pro

- Full program written in new language
- Can make use of SIMPL# Libraries.
- One program can run on any 3-series Control System
- Communicate with existing SIMPL Windows Programs

CRESTRON
Integrated by Design

What do you need

- 3-Series Control System
- Visual Studio 2008:
  - Most bang for the buck: Get an MSDN Subscription. ($1,200)
- Working knowledge of C# (Use "Illustrated C# 2008" by Daniel Solis)
- Crestron's Snap-in for Visual Studio 2008



How do I get started

1. Look at your S+ modules that do some of this:
   1. Parse XML
   2. Parse web pages
   3. Heavy duty string parsing
   4. Floating point
2. Take these modules, and convert them to SIMPL#
3. When you’re comfortable with C#, you **may**
   1. Write a SIMPL#Pro program, that is hardware independent.

Glossary

LPZ: 3-Series Program Zip (SIMPL Windows program compiled)
CLZ: SIMPL# Compiled Library (Zipped)
CPZ: SIMPL#Pro Compiled Program (Zipped)
CS : C# File
VCPROJ: Visual Studio Project (can contain multiple CS files)
SLN: Visual Studio Solution (can contain multiple VCPROJ files)
REFERENCE: Link to a CLZ
RESOURCE: Non Program file, to be added to CLZ/CPZ (IR file, XML File, etc.)
NAMESPACE: An abstract container providing context for C# Classes

Snap-In for Visual Studio 2008

Why is there a snap-in:

- Ensure the sandbox is maintained
- Ensure files are compiled in a way that is compatible with the 3-Series processors
- Simplify Single Step Debugging on remote devices.

CRESTRON®
Integrated by Design

SIMPL+ vs SIMPL# Data Types

SIMPL# Datatype	Equivalent SIMPL+ Datatype	Can Declare in SIMPL# Library	Can Declare in SIMPL+ Module	Array Support	
string	STRING	YES	YES	1D	
SimplSharpString	STRING	YES	NO	1D	
int	SIGNED_LONG_INTEGER	YES	YES	1D, 2D	
uint	LONG_INTEGER	YES	YES	1D, 2D	
short	SIGNED_INTEGER	YES	YES	1D, 2D	
ushort	INTEGER	YES	YES	1D, 2D	
struct	STRUCTURE	YES	YES	1D	
class	CLASS	YES	NO	1D	

SIMPL+ Constructs

#USER_SIMPLSHARP_LIBRARY → This is to reference a CLZ file, and access the C# Classes, methods, and properties

(UN)REGISTEREVENT → Create a SIMPL+ Eventhandler for a C# Event
(UN)REGISTERDELEGATE → Create a Function in SIMPL+ that can be executed from C#
CMUTEX → Mutually Exclusive access to a shared resource. Basically a semaphore
CEVENT → Event used for synchronization

CRESTRON®
Integrated by Design

Common Problems Solved

1. Shared memory → Use static variables in SIMPL#. Include the same library in two SIMPL+ modules, and they will share the values.
2. Parse XML → Use the native CrestronXML Namespace and classes to iterate through XML.
3. Download / Parse HTML Websites → Use native HTTPClient to parse through HTML documents.
4. File / String parsing → Use native string functions to find sub-strings, etc.
5. Regular Expressions → Powerful string parsing modules
6. SQL Queries → Connect directly to SQL Servers on the network, and run queries and updates.
7. TCP Server with multiple connections
8. SIMPL# Event executing function in SIMPL+

CRESTRON®
Integrated by Design