

Advanced Program Development (4 days)

Instructor: Keith Rice

Objective

Students will learn the essential skills of professional SolidWorks add-in and stand-alone development using VB.NET: program setup and architecture, object-oriented programming techniques, using Visual Studio, source code control and collaboration, documentation, deployment, and administration. The contents of this course is borne out of years of real-world, customer-driven experience and has never been published in any form.

A brief look at the differences between VBA and VB.NET is included as a refresher on the first day. Even if you intend to write addins / stand-alones in C#, this course can still benefit you tremendously. The differences between VB.NET and C# are merely syntactical; both languages uses the exact same libraries, which is why many tools exist to automatically convert code from VB.NET to C# and vice versa.

Prerequisites

This course assumes you already have knowledge of the following topics (which I call the Three Pillars of SolidWorks API Programming):

- Visual Basic for Applications – how to create macros involving variables, conditionals, loops, functions, modules, etc.
- SolidWorks API Help – how to navigate to find SolidWorks API calls and interfaces, learn about arguments and syntax, etc.
- SolidWorks API Object Model – what it is and how it relates to SolidWorks API programming (e.g., what accessors are, why they are important, how to find them)

Note: Sufficient understanding of these three topics is easily obtained after watching the first four or five units of our online course, [Automating SolidWorks with VBA](#).

1. Day One

- a. Write the “Config Creator” macro
 - i. Review VBA
 - ii. Review SolidWorks API
- b. Introduction to Visual Studio
- c. Object Oriented Programming
 - i. Objects, classes, interfaces
 - ii. Inheritance, abstraction
 - iii. Shared / static functions, overloads, overrides
 - iv. Classes versus modules
 - v. Architecture
- d. Resources
 - i. MSDN
 - ii. StackOverflow
 - iii. Telerik Converter
 - iv. *Clean Code* by Robert C. Martin
- e. Source control
 - i. Git
 - ii. Sourcetree
 - iii. BitBucket

2. Day Two

- a. Creating a console application
- b. Connecting to SolidWorks
 - i. Activator.CreateInstance()
 - ii. Process.Start()
- c. Creating a Win Form application
- d. Convert VBA “Config Creator” to VB.NET
- e. Creating online repository
- f. About button
- g. Remembering user inputs
- h. Inputs class
- i. Logging
 - i. Use CTLogger and explain need for PDB and XML file
- j. Bulk processing
- k. Restarting SolidWorks after crash
- l. NuGet packages
- m. Build configurations
- n. Embed interop types
- o. Administering upgrades

3. Day Three

- a. Creating up an addin from scratch
- b. Debugging in Visual Studio
 - i. Mention: Setting up a tester addin
- c. Source control
- d. User interface options
 - i. Menus
 - ii. CommandManager tabs
 - iii. Mention
 - 1. Task panes
 - 2. PropertyManager Pages
 - 3. WinForms
 - 4. Context menus
 - 5. Rerouting existing commands
- e. Creating menus and buttons
 - i. Icons
- f. Deployment options
 - i. BAT files
 - ii. Visual Studio Installer Projects Extension
 - iii. Other options
- g. Comparison with SDK
- h. Creating templates

4. Day Four – Additional Topics

- a. Document Manager API
- b. User interfaces
- c. Async / BackgroundWorker
- d. Connecting to Excel
- e. Third-party storage
- f. Hybrid addins
- g. Event notifications
 - i. Overriding existing commands