

Advanced Program Development (3 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 largely 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 – Introduction to .NET

- a. Moving from VBA to VB.NET
 - i. Language differences
 - ii. Practice: Converting VBA macro to VB.NET
- b. Visual Studio
- c. Object Oriented Programming
 - i. Objects, classes, interfaces
 - ii. Inheritance, abstraction
 - iii. Architectural considerations and design patterns
- d. Resources: MSDN, StackOverflow, and more
 - i. Briefly: VB.NET vs C# (including tools for converting between the two)
- e. Source control and collaboration using Git and GitHub

2. Day Two – Addin Development

- a. Setting up an addin from scratch
 - i. Libraries
 - ii. GUIDs
 - iii. Connecting to SolidWorks
 - iv. DLL registration
- b. Creating menus and buttons
- c. Using property manager pages and user forms
 - i. BackgroundWorker
- d. Debugging in Visual Studio
- e. Deployment
- f. Example – much of the afternoon will be spent creating an example addin that puts together many of the skills we have already learned

3. Day Three – Add-Ins and Stand-Alones

- a. Depending on class interest, we will continue working on addins or begin working on stand-alones
- b. Setting up a stand-alone
- c. Restarting SolidWorks after crash
- d. Logging
- e. Connecting SolidWorks and Excel via stand-alone
- f. Example – much of the afternoon will be spent developing a “bulk processing tool” stand-alone that puts together many of the skills we have already learned