

Course Outline - VTScadaFest Scripting

This course is designed for experienced VTScada developers who are proficient at creating advanced applications and want to learn how to code in VTScada and maintain and debug such code. This course teaches you how to code using the VTScada programming language (often referred to as VTScada scripting) which is a fully featured object-oriented programming language. **You do not need to take this course to become a VTScada expert.**

Prerequisites: In order to take this advanced programming course, you must have taken both the Operation & Configuration and Advanced Configuration courses. You must be comfortable with coding using an object-oriented programming language (e.g., C++, python, java or javascript) and with basic programming concepts such as loops, conditionals, variables, parameters, data structures (e.g., lists, arrays, dictionaries), encapsulation and inheritance.

Please do not register for this course unless you are already proficient in at least one object-oriented programming language and you are very comfortable building tags, pages, widgets, alarms, and other standard VTScada features.

Day 1

Introduction to VTScada Script

- Introduction to VTScada Script
- Anatomy of a Module – Variables, Parameters, States, Statements and Scripts
- Action Triggers, Values, Validity, and Functions
- Source Debugger
 - Components
 - Static, Scope and Caller Trees
 - Values, Watch Window
 - Breakpoints and conditional breakpoints
- Launched Modules, Called Modules and Subroutines
- Script Code and Widgets
- Script Code and VTScada Pages

Day 2

Data Structures and Making use of Script Code in VTScada

- Data Structures, Arrays and Dictionaries
- Working with Times and Dates
- Worked Example: Making an HTTP Request and parsing the response
- Worked Example: Creating a Custom MQTT encoder / decoder
- Worked Example: Creating a Page Template Widget

Day 3

Advanced Debugging and Extending VTScada's Functionality

- Advanced debugging tools: Profiler, Instance Count, Trace Viewer
- Dump Files and Crash Dumps
- Worked Example: Writing a Custom Report
- Worked Example: Programmatically Starting Tags and altering Parameters
- Worked Example: Creating a Custom Configuration Folder for a Custom Tag Type

Depending on available time and student interest, additional topics may be covered.