Getting Started with the Sedona Application Editor

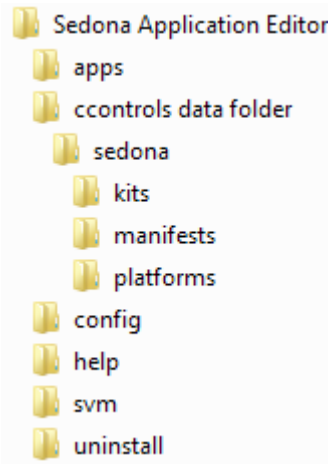
Introduction

Contemporary Controls, a Sedona Framework Community member, has developed the Sedona Application Editor (SAE) that allows system integrators the ability to develop control applications for Sedona devices. Developed by Tridium, Inc., Sedona Framework[tm] is a software environment designed to make it easy to build smart, networked, embedded devices which are well suited for implementing control applications. Using Niagara Workbench or a Sedona tool, such as the Sedona Application Editor, components are assembled onto wire sheets creating applications. This language is ideally suited for graphical representation of control strategies. For those independent system integrators without access to Niagara Workbench, SAE provides the same functionality in a more simplified manner.

A Sedona device is defined as having a Sedona Virtual Machine (SVM) of which Contemporary Controls has developed a series of BACnet/IP Sedona Unitary Controllers in its BAScontrol and BASremote series. These freely-programmable controllers can now be programmed with SAE instead of Niagara Workbench. Those with experience with Niagara Framework will have no problem understanding Sedona Framework. For those without Niagara experience, the graphical representation of components linked on a wire sheet to create applications is intuitive and can be easily learned with a minimum of training. Sedona Framework is available license-free and therefore appropriate that this open technology have an open programming tool such as Sedona Application Editor. Although the tool is delivered with kits that support Contemporary Control's controllers other Sedona suppliers can install their kits in order to use the tool. Sedona components are deployed in kits and the SAE provides not only the Tridium Sedona 1.2 release kits but Contemporary Controls' hardware dependent and hardware independent custom kits. Hardware dependent kits are product specific (links to physical I/O on the controller) while hardware independent kits are freely available to the Sedona Framework Community. It is Contemporary Controls' policy to continually append the latest "kits" information to cover existing and newer controllers.

The tool is freely-available to the controls community and we only ask that it be used by competent system integrators who can translate a given Sequence of Operation (SOO) into a control application.

Installing SAE



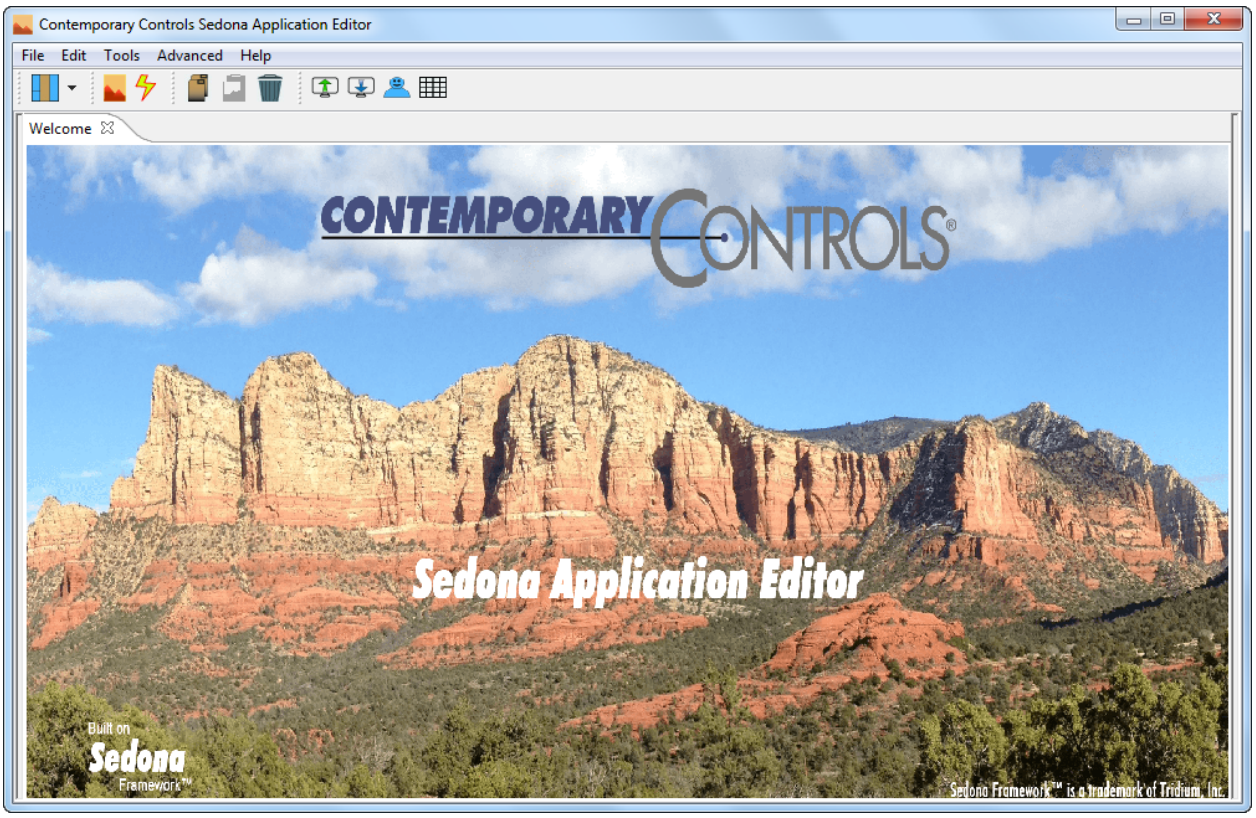
The Java-based (Java Runtime Environment JRE 1.7 or higher) SAE can be downloaded from Contemporary Controls' website after completing a short registration process. Follow the instructions for installing SAE. In addition to the application file SAE there are folders in the SAE folder for Sedona apps, configurations, help files, and Sedona data. The Sedona data folder included with the install is called ccontrols data folder to clarify that the Sedona data has been provided by Contemporary Controls. In this folder is the Sedona Folder where kits, manifests and platforms reside that is needed for Contemporary Controls' Sedona devices. This folder is sometimes called Sedona Home.

Also included in the installation process is an evaluation SVM (SVM-PC) that can run on your PC allowing you the ability to experiment with creating Sedona programs that can execute locally on your PC. In this way you can evaluate the Sedona technology without the need of purchasing a Sedona controller to test.

Starting the SAE and the SVM-PC

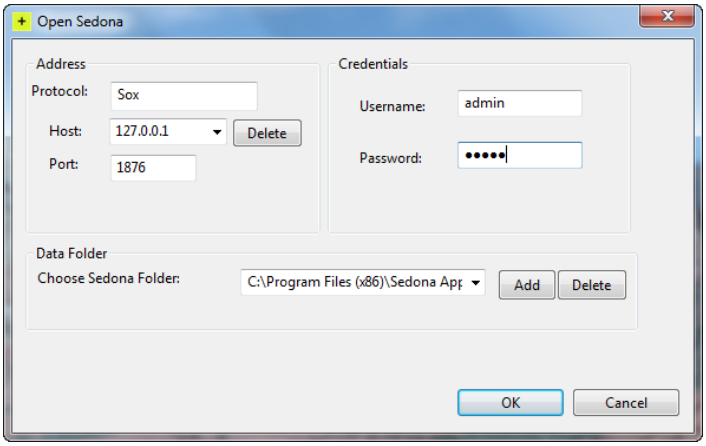
If you want to use the SVM-PC, you can start it by opening up the Start menu and then All Programs. Find the Sedona Application Editor folder and open it. Click on SVM-PC and a window will open. Minimize it but do not close it. Now you can click on the Sedona Application Editor in the same folder.

Once SAE launches you will basically see a title bar, menu bar, tool bar and welcome screen for the editor. Not much useful information is available until you attach to a Sedona device. In this example we will connect to the SVM-PC although you could connect to a Contemporary Controls' controller instead. Make sure that if you want to connect your computer to a controller that your computer and controller share the same network.



Opening a Connection

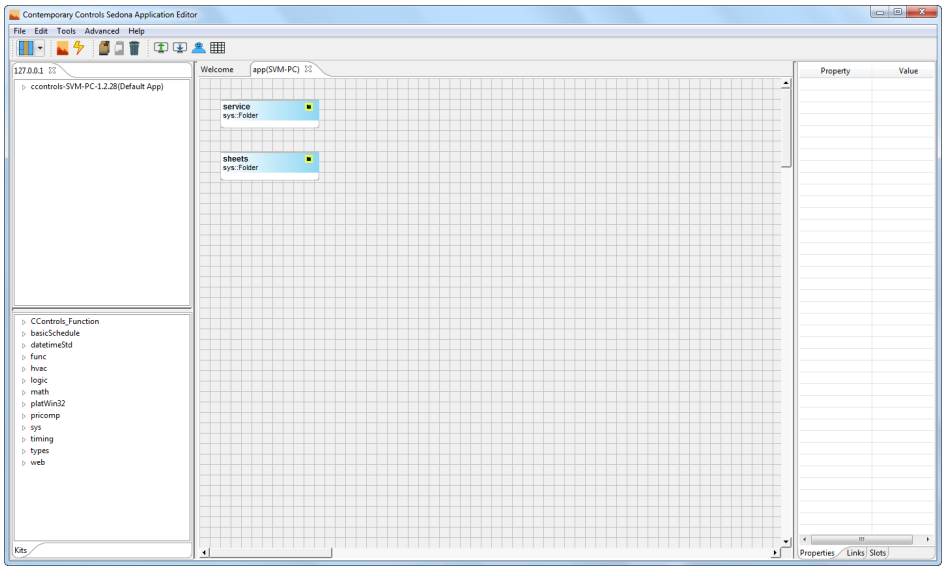
By clicking the Open Connection icon on the tool bar you will open a window for attaching to a Sedona device. Since Sedona devices are IP-based, it is looking for an IP address. Enter IP address under Host. For SVM-PC it is 127.0.0.1. For an external controller use its IP address. There is a drop-down for this and your IP address will be remembered until you delete it. Protocol should remain as Sox and the Port number should remain as 1876.

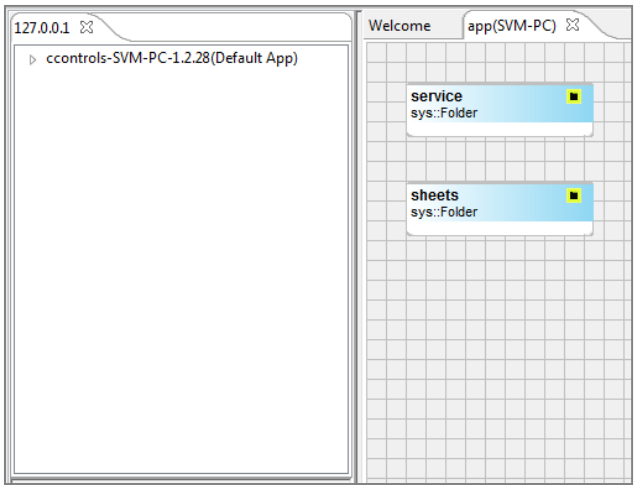


The location of the Data Folder has already been selected for you during installation. If you are seeking an alternate Data Folder (perhaps for a third-party Sedona device), click Add to browse for the Sedona Folder within the computer.

The sub-folder found must be simply named sedona and it should be highlighted in the browse window. Click OK to accept the alternate selection. For the SVM-PC or a default Contemporary Controls' controller, enter the default password which is admin and then click OK. For third-party Sedona devices enter the password. If there is a connection issue you will receive an error message. After a successful login the location of the Sedona Folder will be remembered as well as the IP address of the Sedona device.

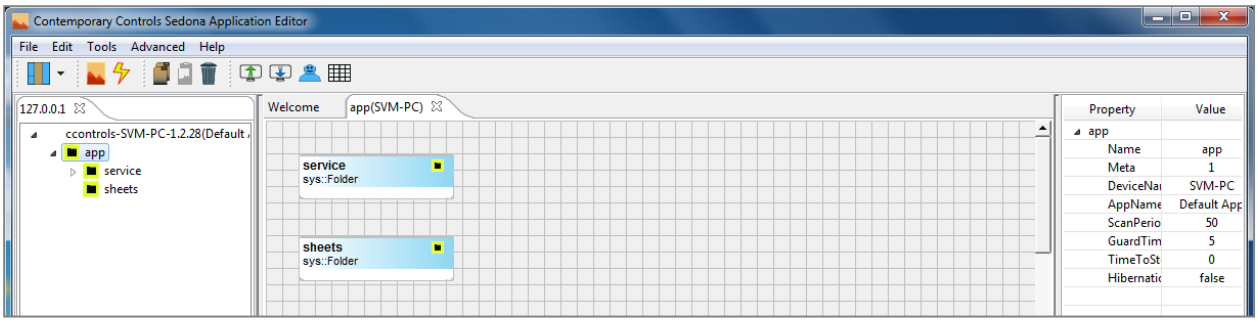
Once you connect to a Sedona device the screen changes significantly showing four windows. In the middle is the Work Area pane where applications are developed. At the upper left corner is the Navigation pane there is a tree-like structure that explains the makeup of the connected Sedona device. Below that is the Kits pane where the installed kits from the connected controller are displayed. Within the various kits are the components that will be used to create applications. On the right is the Properties pane where the attributes of selected components are displayed. The Navigation, Kits and Properties panes can be hidden or displayed using the left-most icon on the tool bar.



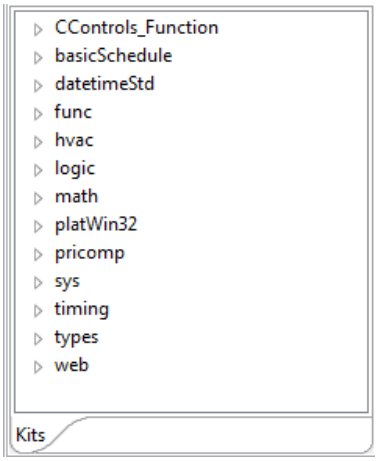


When you first connect to a Sedona device you will be presented with two folder components in the Work Area pane under a tab called app followed by the connected Sedona device name. One is called the service folder and the other the sheets folder. For now, disregard the service folder (never delete it!) which mostly involves background tasks necessary for a functioning Sedona device. We are more interested in creating applications within the sheets folder. The sheets folder can be opened to display a blank wire sheet. This is where components located in kits are dragged onto a wire sheet, having their properties configured, and interconnected with links to form applications. Wire sheets are usually shown as a grid but the grid can be toggled on and off using the grid icon on the tool bar. Additional folders can reside on wire sheets so tabs are used at the top of the Work Area to indicate what wire sheet folder is actively displayed in the Work Area.

In the upper left corner is the Navigation pane. The IP address of the connected Sedona device is shown as a tab. It is possible to have multiple Sedona devices connected resulting in multiple tabs of IP addresses. Just below the tab is the Platform ID, taken from the platform kit, followed by the application name of the connected Sedona device in parenthesis. This Sedona device happens not to have a pre-programmed application so it is using the default name stored in the Sedona device.

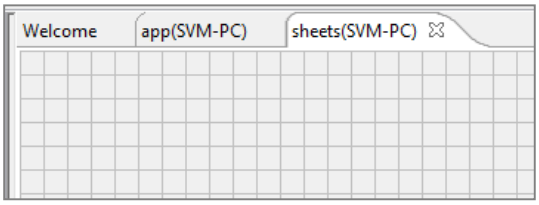


You can expand the Platform ID in the Navigation pane to show app and then service and sheet. If you click on app you will see its properties in the Properties pane at the fair right. The only interesting information for app properties is DeviceName, AppName and ScanPeriod – all of which can be modified from a different screen. We will change these later.



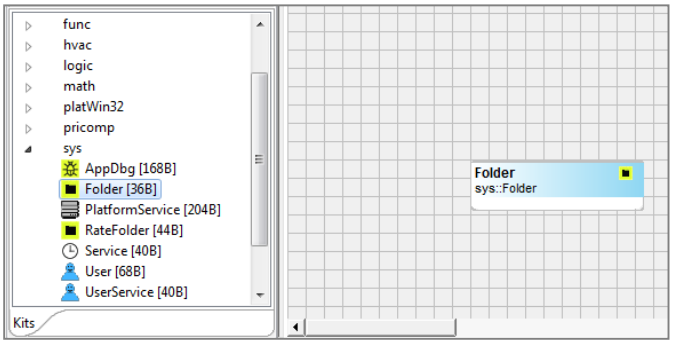
At the bottom left side is the Kits pane which lists the available kits that are present in the connected Sedona device. While all known Tridium and Contemporary Controls' kits are stored in the Sedona Applications Editor, only those loaded in the connected Sedona device are shown. If an unknown Sedona device is connected with no matching kits information in the Sedona Application Editor, a mismatch error will be indicated. Kit names follow a particular format. Kits with no company name are from Tridium and were provided with Sedona release 1.2. It is Contemporary Controls' policy to include them in our products without modification. We call these the standard kits. Tridium kits are hardware independent and can be used on any Sedona 1.2 device. Kits with company names come from Sedona developers who are obligated to identify their custom kits. For Contemporary Controls the indication is CControls. Kits that also carry a model number are deemed hardware dependent kits and can only work with particular hardware and therefore are not portable to other Sedona devices. Kits with just the company name are hardware independent kits and are available for sharing with the Sedona community.

Therefore, when looking at the Kits pane for the SVM-PC you will notice that the first kit listed is a hardware independent kit from Contemporary Controls, while the remaining kits are standard Tridium kits including the platWin32 kit which is a renamed Tridium platform kit.

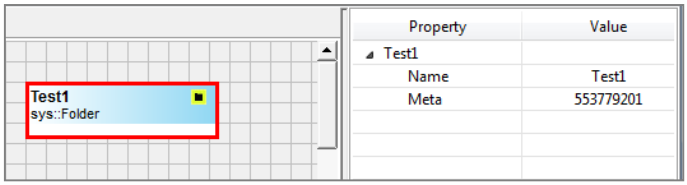


Double-click the Sheets folder in the Work Area pane. Since the connected controller does not have any application installed the wire sheet is blank. Notice the tab at the top called Sheets. We call this sheets folder the main wire sheet because this is where you begin to add components.

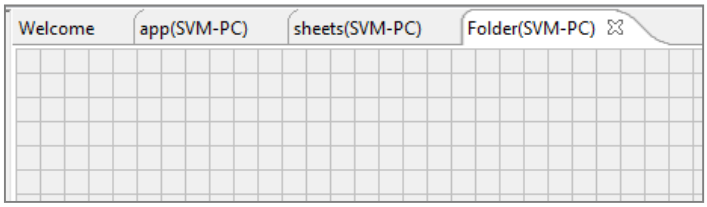
Double-click the Sheets folder in the Work Area pane. Since the connected controller does not have any application installed the wire sheet is blank. Notice the tab at the top called Sheets. We call this sheets folder the main wire sheet because this is where you begin to add components.



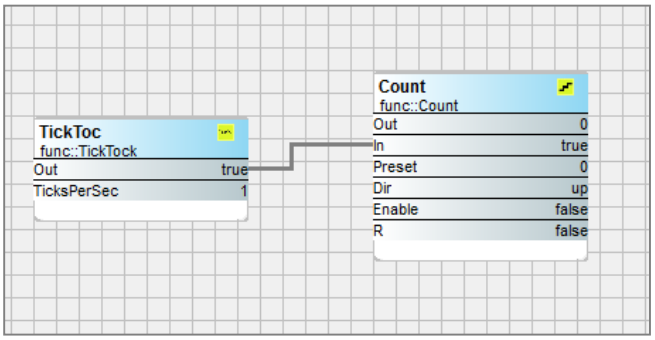
The first component we will put down is a folder that can be found in the sys kit. Expand the sys kit, find the Folder component and drag it onto the wire sheet.



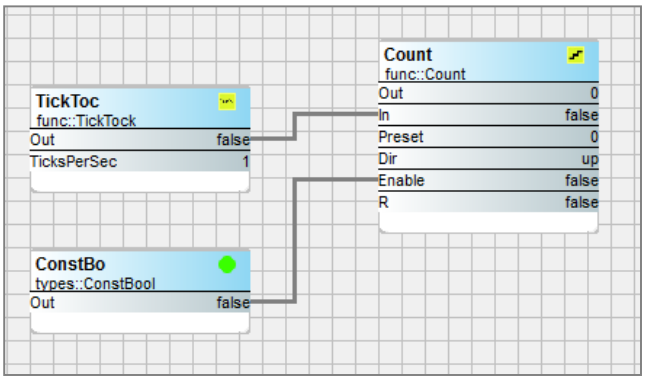
You have an opportunity to rename the component. Select it so that it becomes highlighted with a red border allowing its properties to appear in the Properties pane. Click on the name field in the properties pane and type in Test1. Hit return and the name changes. Note that Sedona only allows you to have up to seven contiguous characters for naming a component and you cannot lead with a number or use special characters. Upper and lower case characters are allowed.



Double-click the component and a new blank wire sheet appears. This shows the contents of your new folder. At the top of the wire sheet is a new tab with the name of the folder you created – Test1. The folder is open so we can add components to this folder.

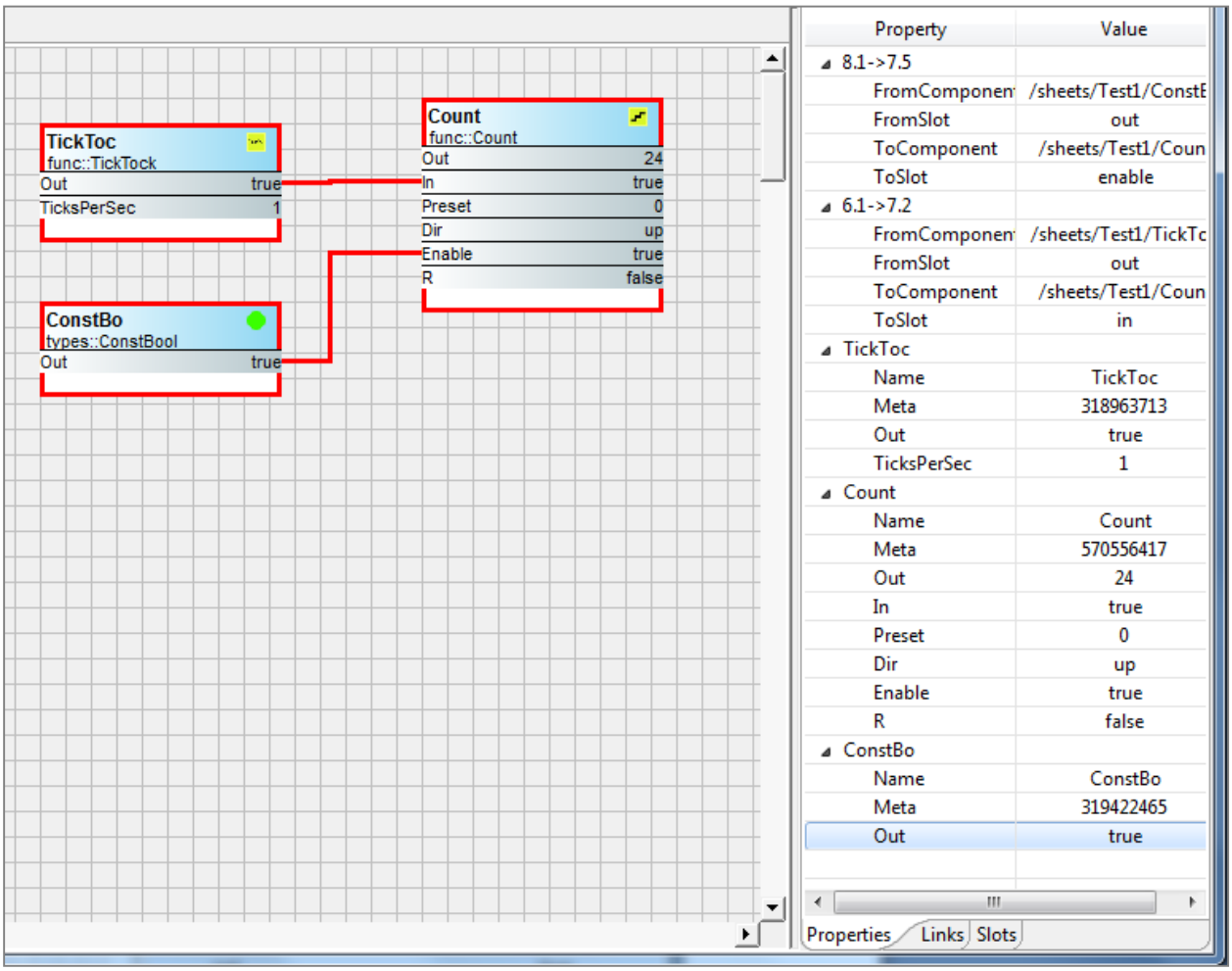


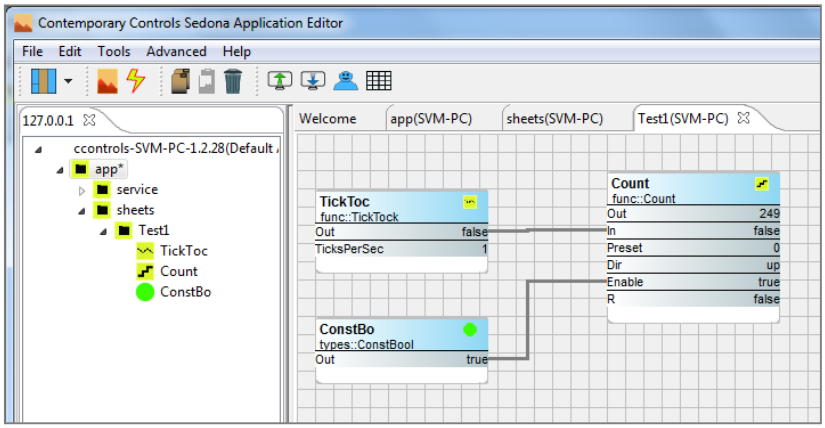
Contract the sys kit and then expand the func kit. Find TickTock and drag it onto the wire sheet. From the same kit select Count and drag it to the right of TickTock. Click on the Out slot of TickToc and drag the cursor to the In of Count to establish a link between the two components. You can move your components anywhere on your wire sheet by clicking on either the header or the footer of the component and dragging it to another location. The middle of the component contains what are called slots. Each slot has a name on the left and a property value on the right. Inputs to slots are usually provided by links attached to the left side of the component and outputs from slots emanate from the right side of the slot.



Contract the func kit and then expand the types kit. Find ConstBool and drag it below TickToc. Notice that the component name gets automatically truncated to seven characters. Connect a link between Out on ConstBo and Enable on the Count component. Notice that Count is not counting. This is because Enable is not True on the counter.

Draw a box around all three components which will highlight all three components while showing their properties in the Properties pane. Go to the properties pane and click on the state of Out for ConstBo. You will be given a drop-down choice. Select True and click on the white space. This sets the ConstBo to True which will enable counting and you will see the effect in the Out slot as the counter increments. To clear the highlighting, simply click on a blank part of the wire sheet. You have now completed your first program.

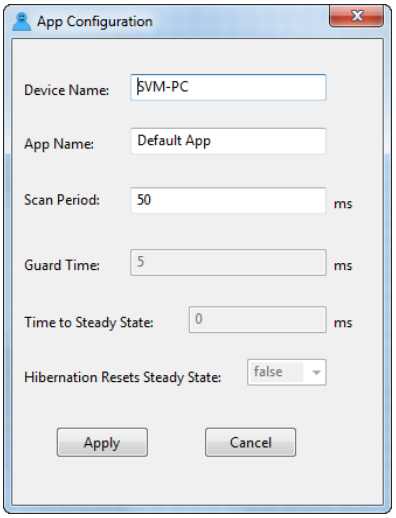




Go to the Navigation pane and expand the sheets and Test1 folders. You will see a different representation of your program. Under Test1 you will see the three components you added to a wire sheet. They are shown in the order that you added them to the wire sheet. In fact, this is the order of execution of the Sedona Virtual Machine. To the right in the Work Area pane is the graphical representation of the program showing the links between various slots on the components. At the top you will see tabs indicating that you are looking at the Test1 wire sheet.

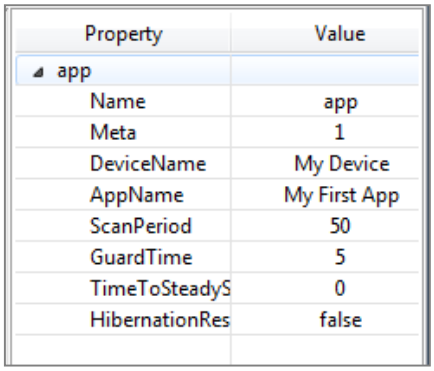
In the Navigation pane next to app is an asterisk (*). This indicates that a change has been made to the program and it has not been saved onto the Sedona device. With Contemporary Controls' Sedona devices, the Sedona application is stored in Flash but executed out of RAM. You need to save to Flash before power is removed from the controller otherwise you will lose your changes. With the SVM-PC there is no Flash memory so you will need to save to your hard disk.

When you are developing an application you are developing it on a live controller where you can instantaneously see the results of your program changes. This is very convenient but changes should be made with care especially if the Sedona device is attached to some physical output points connected to equipment. If you are developing on installed equipment it is best to disconnect any physical output connections or de-energize them until the program is proven. With the SVM-PC this is not an issue because there are no kits supporting physical I/O.



Before we save your program we are going to rename the program from the default name. Go to the menu bar and click on Tools and then App configuration to bring up a configuration window.

Change the default Device Name to My Device and the default App Name to My First App. Leave Scan Time unchanged. Scan time refers to the interval used by the Sedona Virtual Machine in solving its logic. A scan time of 200ms means that the logic is solved 5 times a second. This is typical for Contemporary Controls' controllers. With a SVM-PC and all the resources of a PC available we set the scan time lower. Click Apply.



Property	Value
app	
Name	app
Meta	1
DeviceName	My Device
AppName	My First App
ScanPeriod	50
GuardTime	5
TimeToSteadyS	0
HibernationRes	false

Go back to the Navigation pane and click App and in the Properties pane you should see your edits.

Move your cursor to the Save to Controller icon in the tool bar and click it. You will receive a message that your application program has been successfully saved. It is still running in RAM but an image has been saved in Flash (in the case of the SVM-PC the hard disk) so if you cycle power on your controller your changes will still be intact. The asterisk next to the word App in the Navigation pane should now be removed.

You can also save your program to your PC. Go back to the tool bar and click on Save to PC. You will be prompted to specify a name for your program so simply say My First App although you are free to name it anything you want. Notice that a .sax extension will be applied and that you will be sent to the apps folder within the Sedona Application Editor folder. Click Save and you will see an acknowledgement of a successful save.

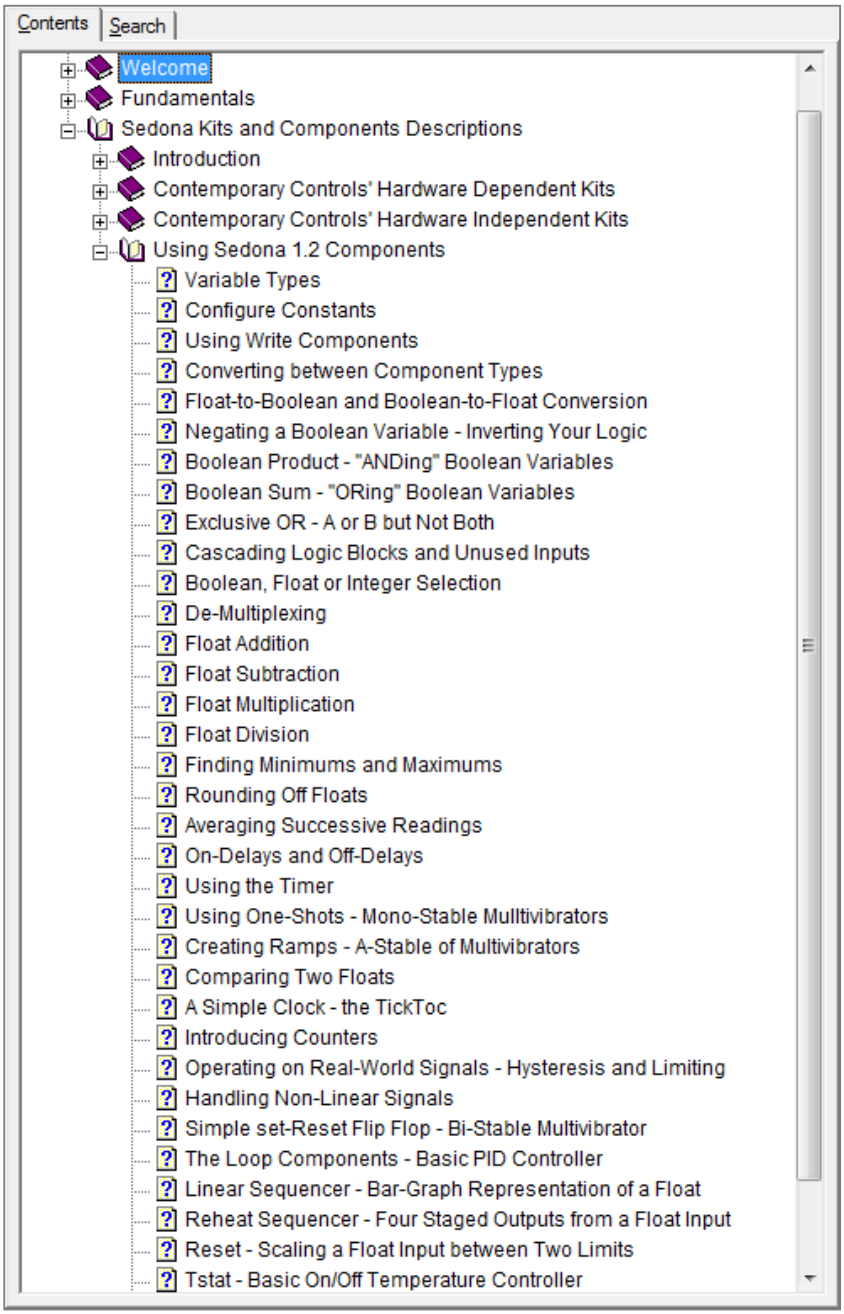
Go back to your wire sheet and draw a rectangle around the Count component to highlight it. Right-click the component and click Delete.

If you cycled power to your controller now, your old program would return because you never saved your changes to the controller. Do not bother to do to this with the SVM-PC.

Go to the tool bar and click on Load from PC. Select your program and then Open. After a time, your saved program will be loaded back in the controller and your controller restarted. You can verify that your program has been restored. In the case SVM-PC, you may be required to restart the SVM-PC yourself.

During the load process, SAE complies the selected program with the .sax extension into a program called app.sab. This program happens to reside in the apps folder. This is the program that executes on your Sedona device. Every time you do a load of a .sax program to the Sedona device, a new app.sab is compiled and the old one replaced.

This concludes the introductory steps to using the Sedona Applications Editor. More information can be found by clicking on Help in the menu bar.


Contents
Search
Welcome
Fundamentals
Sedona Kits and Components Descriptions
Introduction
Contemporary Controls' Hardware Dependent Kits
Contemporary Controls' Hardware Independent Kits
Using Sedona 1.2 Components
Variable Types
Configure Constants
Using Write Components
Converting between Component Types
Float-to-Boolean and Boolean-to-Float Conversion
Negating a Boolean Variable - Inverting Your Logic
Boolean Product - "ANDing" Boolean Variables
Boolean Sum - "ORing" Boolean Variables
Exclusive OR - A or B but Not Both
Cascading Logic Blocks and Unused Inputs
Boolean, Float or Integer Selection
De-Multiplexing
Float Addition
Float Subtraction
Float Multiplication
Float Division
Finding Minimums and Maximums
Rounding Off Floats
Averaging Successive Readings
On-Delays and Off-Delays
Using the Timer
Using One-Shots - Mono-Stable Mulltivibrators
Creating Ramps - A-Stable of Multivibrators
Comparing Two Floats
A Simple Clock - the TickToc
Introducing Counters
Operating on Real-World Signals - Hysteresis and Limiting
Handling Non-Linear Signals
Simple set-Reset Flip Flop - Bi-Stable Multivibrator
The Loop Components - Basic PID Controller
Linear Sequencer - Bar-Graph Representation of a Float
Reheat Sequencer - Four Staged Outputs from a Float Input
Reset - Scaling a Float Input between Two Limits
Tstat - Basic On/Off Temperature Controller