

User Manual

BAS*control* **Toolset**

CONTEMPORARY **CONTROLS**[®]

Trademarks

BASautomation, Contemporary Controls and CTRLink are registered trademarks of Contemporary Control Systems, Inc. BACnet is a registered trademark of the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. Powered by Sedona Framework is a trademark of Tridium, Inc. Other product names may be trademarks or registered trademarks of their respective companies.

Copyright

© Copyright 2025, by Contemporary Control Systems, Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of:

Contemporary Control Systems, Inc. 2431 Curtiss Street Downers Grove, Illinois 60515 USA	Tel: +1-630-963-7070 Fax: +1-630-963-0109 E-mail: info@ccontrols.com
Contemporary Controls Ltd 14 Bow Court Fletchworth Gate Coventry CV5 6SP UK	Tel: +44 (0)24 7641 3786 Fax: +44 (0)24 7641 3923 E-mail: ccl.info@ccontrols.com
Contemporary Controls (Suzhou) Co. Ltd Room 603, Block A, New Energy Technology Park No.298 Mayun Road, Suzhou New District, PR China 215000	Tel: +86 512 68095866 Fax: +86 512 68093760 E-mail: info@ccontrols.com.cn
Contemporary Controls GmbH Fuggerstraße 1 B 04158 Leipzig Deutschland	Tel: +49 (0)341 520359 0 Fax: +49 (0)341 520359 16 E-mail: cgc.info@ccontrols.com

Disclaimer

Contemporary Control Systems, Inc. reserves the right to make changes in the specifications of the product described within this manual at any time without notice and without obligation of Contemporary Control Systems, Inc. to notify any person of such revision or change.

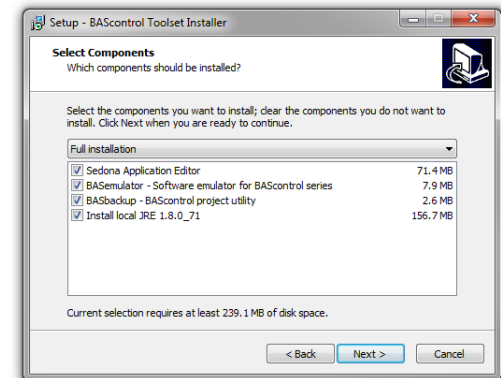
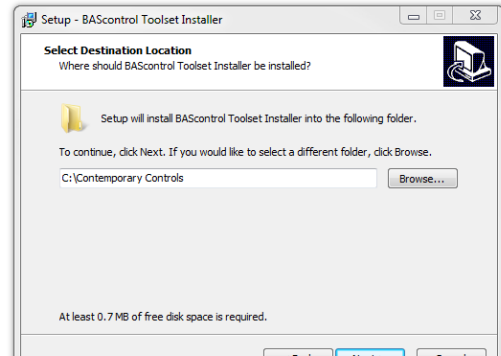
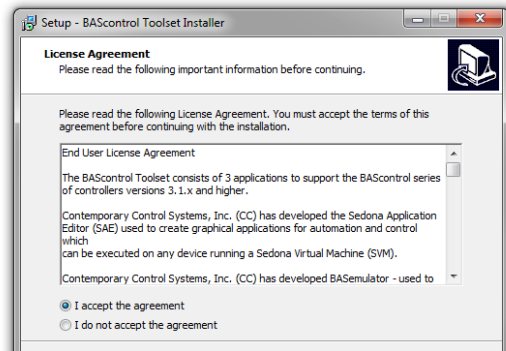
Contents

1	Installing the BAScontrol Toolset.....	4
1.1	Installing the BAScontrol Toolset	4
2	BASemulators – Emulating a BAScontroller on a PC.....	6
2.1	Launching the BASemulator	7
2.2	Using BASbackup and the BASemulator.....	8
2.3	Cloning Programs with BASbackup and the BASemulator	8
3	Using BASbackup – the BAScontrol Project Utility	10
3.1	Installation.....	10
3.2	Backing up a BAScontrol Project	11
3.3	Restoring a Sedona Project	12
3.4	Cloning a Sedona Project.....	14
3.5	Getting SAX Data	16
4	Sedona Application Editor	17
4.1	Introduction.....	17
4.2	Starting the SAE and the BASemulator.....	17
4.3	Opening a Connection	18
5	Sedona Programming Using Niagara 4 Workbench.....	26
6	Understanding Sedona Kits and Kits Manifests	27
6.1	Component Bundle Structure.....	27
6.2	Kits and Manifest Naming	29
6.3	Sedona Tool Database	29
6.4	Installing Component Bundle to Niagara Workbench	29

1 Installing the BAScontrol Toolset

1.1 Installing the BAScontrol Toolset

Thank You for choosing Contemporary Controls' open control system. We will briefly introduce you to the installation of the BAScontrol Toolset.



The BAScontrol Toolset consists of:

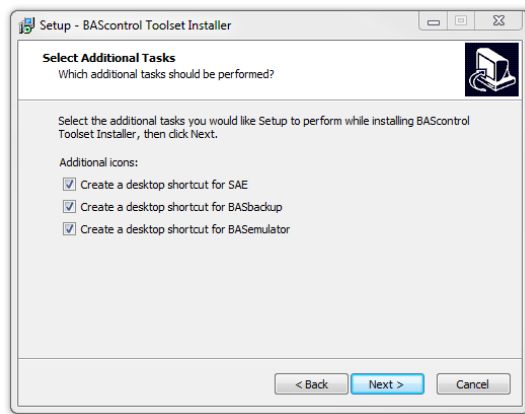
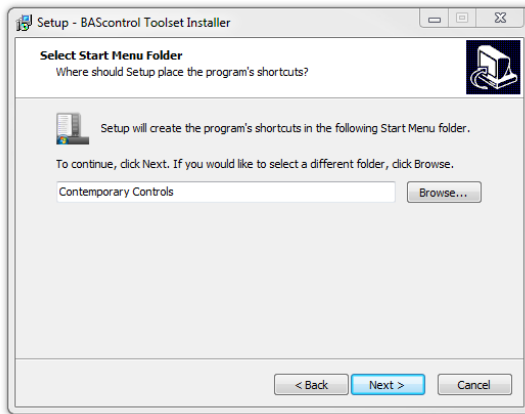
Sedona Application Editor – Sedona graphical editor

BASemulator – software emulation for our series of unitary controllers and edge controllers.

BASbackup – BAScontrol project utility

To install the BAScontrol Toolset, you must run the BAScontrol Toolset installer. The installer is available for free download after a short registration on the Contemporary Controls website. Register and download the installer to begin.

1. Run the installer and accept the End User License Agreement.
2. Choose a location for the installation – default is recommended
3. You will be given the option to choose which applications to install. We recommend that you leave the default - select all and install all of the applications.
4. Chose a name for your Start Menu program folder – Contemporary Controls is the default
5. Select the shortcuts to be created
6. When the installation is complete, you will be given the option to view the BAScontrol Toolset Quick Start guide which will help you get started.



2 BASemulators – Emulating a BAScontroller on a PC

The BASemulator is a handy utility that works in conjunction with the Sedona Application Editor (SAE) and BASbackup – the BAScontrol Project Utility. BASemulator can emulate on a Windows PC the operation of any controller from our series of BACnet-compliant open controllers that address unitary control and edge control. All real, virtual, and web points including Sedona logic can be viewed with SAE as if it was the actual controller. All web pages appear on your browser as if it was the actual controller. This has several advantages.

- Sedona programs can be developed on a PC without the need of the target controller
- Programs can be tested before being transferred to a live controller
- Program development can begin before ordering the actual controller
- Using SAE, you can copy and paste partial or complete Sedona logic between the emulator and a real controller
- Restore files with actual IP addressing and BACnet configuration of real controllers can be created with BASbackup and the BASemulator
- If you are new to Sedona, you can teach yourself Sedona using just SAE, BASbackup and the BASemulator

The BASemulator uses the PC's local address 127.0.0.1 at a specified port (usually 5001) so that is the address to enter into BASbackup when accessing the BASemulator. For SAE, use 127.0.0.1. For either program, these addresses are selectable from a drop-down list. When using BASbackup, the emulator ignores the IP addresses and BACnet device instance settings in the BASbackup file. Although it appears to be using the target's IP address, it always runs the PC's localhost (127.0.0.1). This makes it easy to clone the same application program running on the emulator to multiple real controllers by creating unique IP address and or BACnet settings in the restore file.

SAE, BASbackup and the BASemulator can be downloaded from the Contemporary Controls' website and installed on a PC with shortcuts for launching the three programs. The emulator comes with four Sedona project files that reside in the BASbackup "backups" directory. factoryDefault20 and factoryDefault22 represent the factory default settings of a factory-shipped controller while sample20 and sample22 are Sedona programs that execute outputs for demonstration. These programs use the default factory IP and BACnet settings as follows:

IP address: 192.168.92.168

Subnet mask: 255.255.255.0

Default gateway: 192.168.92.1

DNS1: 8.8.8.8

DNS2: 8.8.4.4

BACnet Device Name: BAScontrol System

BACnet Device Instance: 2749

Although Sedona I/O components exist on the emulator, there are no physical I/O points. However, output action can be observed by viewing the point value on the emulator's main web page or by viewing the execution of the Sedona output components using SAE.

2.1 Launching the BASemulator

To launch the BASemulator, double-click the BASemulator shortcut on your desktop which will bring up a BAScontrol Emulator window. On the top drop-down select the BAScontrol model you want to emulate – BASC20 or BASC22. Next click Start Emulator which will launch your browser showing the BASemulator main web page with the words Emulated BAScontrol20 or Emulated BAScontrol22 displayed depending upon what you selected. If you click on the System Config button on the emulator main web page you should see the default IP and BACnet settings listed above. Close the system configuration window and minimize the browser. Double-click on SAE to launch the editor. Click on the Open Connection icon which will bring up the Open Sedona Connection window. For Host, enter 127.0.0.1 and for login credentials enter admin/admin and then click OK to connect. This should gain you access to the main wiresheet but you will only see two folders – sheets and service. Double-check on sheets folder to view any stored program or to begin programming. To save any program or edits click on the Save to Controller icon which saves your work on the emulator otherwise it will be lost. The next time the emulator is launched your program will be there. Close SAE. To close the emulator, first close your web browser and then click on the Kill Emulator button.

The screenshot displays the 'Emulated BAScontrol22' web interface. It features four columns of status indicators: Universal Inputs (UI1-UI8), Binary Inputs (BI1-BI4), Analog Outputs (AO1-AO4), and Binary Outputs (BO1-BO6). Each indicator consists of a label, a numerical value, and a checkbox. Below these columns, the title 'Emulated BAScontrol22' is prominently displayed in blue. A navigation bar contains buttons for 'System Config', 'System Status', 'Set Time', 'Virtual Points', 'Web Components', and 'Restart Controller'. Below this bar, there is a section for 'Auto Refresh OFF' and two buttons: 'SAE' and 'BASbackup'. At the bottom, a copyright notice reads: 'Copyright 2021 Contemporary Control Systems, Inc. All rights reserved. Firmware Revision 3.1.28-Emulator : Web Page Revision 7.0.6'. A note below the copyright states: 'NOTE: A GREEN label indicates that the I/O point has been placed on the wire sheet'.

Universal Inputs		Binary Inputs		Analog Outputs		Binary Outputs	
UI1	0.000°F	BI1	0	AO1	0.000	BO1	0
UI2	0.000	BI2	0	AO2	0.000	BO2	0
UI3	0.000	BI3	0	AO3	0.000	BO3	0
UI4	0.000	BI4	0	AO4	0.000	BO4	0
UI5	0.000					BO5	0
UI6	0.000					BO6	0
UI7	0.000						
UI8	0.000						

Emulated BAScontrol22

System Config System Status Set Time Virtual Points Web Components Restart Controller

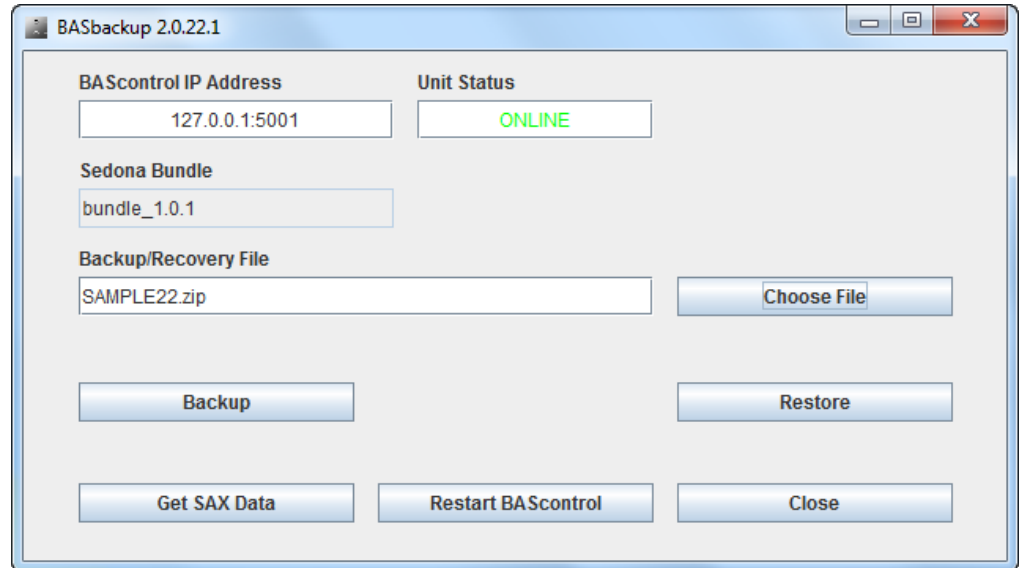
Auto Refresh OFF

SAE BASbackup

Copyright 2021 Contemporary Control Systems, Inc. All rights reserved
Firmware Revision 3.1.28-Emulator : Web Page Revision 7.0.6
NOTE: A GREEN label indicates that the I/O point has been placed on the wire sheet

2.2 Using BASbackup and the BASemulator

Instead of developing programs from scratch on the emulator, programs can be imported, modified and exported using BASbackup. Restart the emulator and select one of the BAScontrol models to emulate. Minimize the web browser. Now click the BASbackup button on the BASemulator which will bring up the BASbackup main

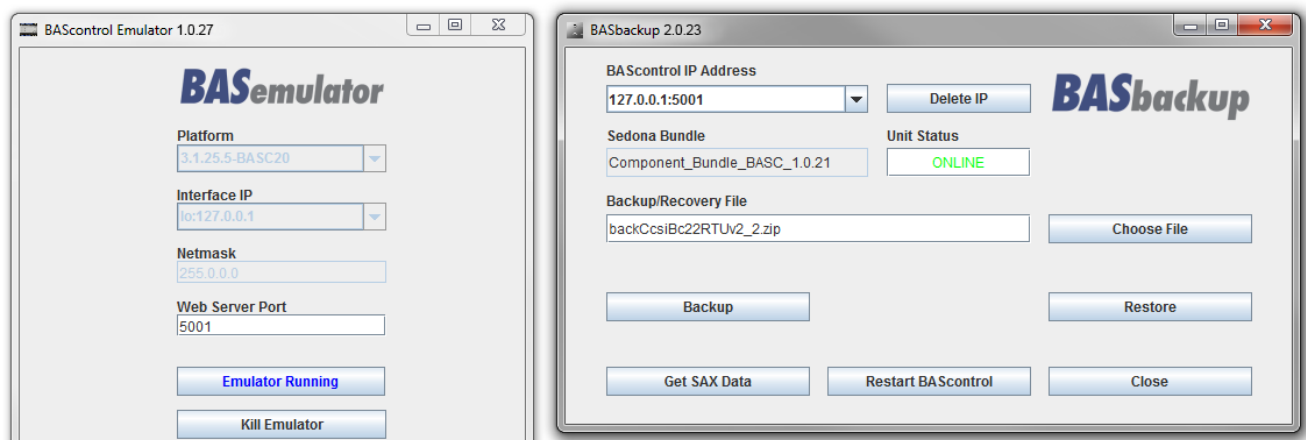


screen. Under BAScontrol IP Address, select 127.0.0.1:5001 or another port if you specified. Click on Choose File and select sample20 or sample22 depending upon which emulation you are using. Click Restore to get you to the Restore Setup page. Notice that the IP and BACnet settings are the same as the default factory settings. Do not make any edits to the settings and click Restore at the bottom. You will get a message that the IP address will change because the restore page IP address is different from the emulator address. Say yes to accepting the change. You will then get a message to restart the controller. Say OK. Remember that the emulator is mimicking a real controller using the default factory settings even though to view the emulator we must use address 127.0.0.1:5001 (or another specified port number). Now maximize your browser so you can see the main web page. Every time you change the program in the emulator, refresh your browser. Click Auto Refresh ON. You should be able to see the binary outputs changing in value which is the result of running the SAMPLE program.

2.3 Cloning Programs with BASbackup and the BASemulator

Assume you want to modify this sample22 program and use it later on a real controller. Start your emulator running the sample22 program. Make your Sedona modifications using SAE and your configuration modifications with your web browser. When you are ready to save your project, click on the BASbackup button on the emulator to launch BASbackup to view its main screen. First you need to make a backup copy of what is on the emulator but you do not want to overwrite the stored sample22 program. Instead, click the Choose File button and enter a new name such as sample22_r1 and click Open. If the name does not exist, BASbackup will make the entry and append the zip extension and it will appear on the Backup main page. Click Backup and the emulator project data will be saved under the new file name. This file becomes the new backup master.

To create a clone copy of this new master but with the proper IP addressing needed for a real controller do the following. On the main BASbackup screen, make no changes to the IP address or the Backup/Recovery entry



It is best to keep a recovery file for each controller in the event a particular controller must be replaced once commissioned. Create your own scheme for matching recovery files with controllers and then click on Restore. You will receive a warning about changing IP addresses so say yes. Enter the default controller credentials (admin / admin). After the restore process, you will be requested to restart the controller. Click OK. Your emulator now has the new settings, and you have a copy of the new settings which BASbackup will use to restore the real controller once it is available.

Restore Setup

IP Address
192.168.92.68

Netmask
255.255.255.0

Gateway
192.168.92.1

DNS1
8.8.8.8

DNS2
8.8.4.4

BACnet Device Instance
2749

BACnet Device Name
BAScontrol System

Recovery File Name

Restore Options

☒ Wire Sheet
☒ Main Configuration
☒ Web Component Configuration

Configuration values in the text boxes will replace values in the Main Configuration file when a recovery operation is performed.

The recovery process generates new files to be uploaded to the controller. If you want to save a backup of the recovery data, choose a file; when the recovery process is complete, a new zip file will be created.

3 Using BASbackup – the BAScontrol Project Utility

The BAScontrol Project Utility program (BASbackup) provides a convenient way of saving a BAScontrol project to a desktop or laptop computer and restoring it to a real or emulated controller. With the BAScontrol device, a project consists of the following files:

- Sedona Framework binary application file (app.sab.target)
- Sedona Framework source application file (app.sax)
- BAScontrol configuration file (bas_cfg.xml)
- BAScontrol SCODE file (kits.scode)
- BAScontrol platform file (platform.xml)
- BAScontrol platform manifest (platformManifest.xml)
- BAScontrol web component file (webc.xml)

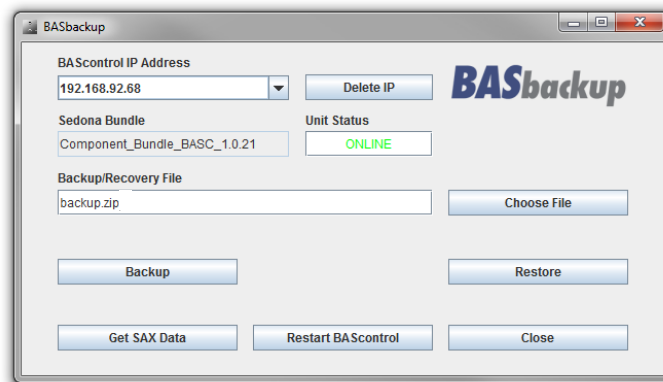
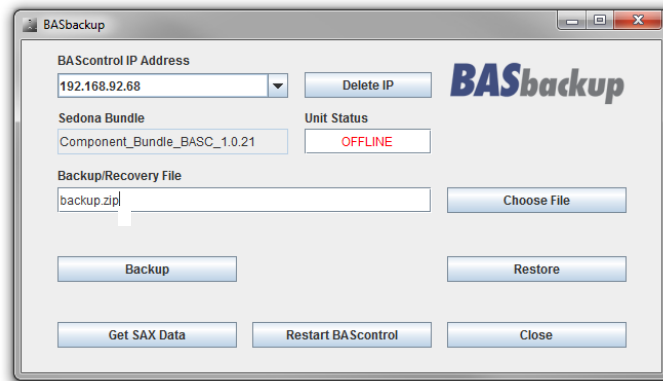
These files are sufficient to completely backup a BAScontrol project without the need of a Sedona tool such as Workbench N4 (after installing our N4 Sedona driver) or Sedona Application Editor. Although Sedona (SAX) files can be backed up and restored using either Workbench N4 or the Sedona Application Editor, the BAScontrol configuration and web component configurations are ignored. Therefore, the BAScontrol Project utility is the best and simplest way to archive BAScontrol projects. The BASbackup program is applicable to BAScontrol20 and BAScontrol22 firmware 3.1.x and later.

3.1 Installation

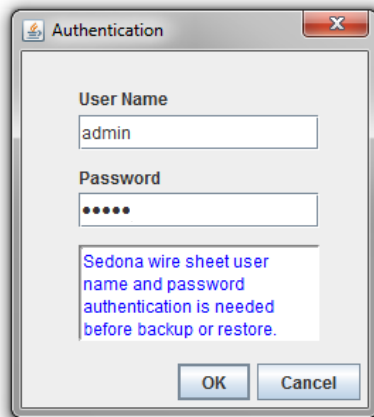
BASbackup along with the Sedona Applications Editor and BASemulator are part of the BAScontrol Toolset installation. Included with the toolset is the latest Sedona Bundle of kits and components. The BAScontrol Toolset Installer can be downloaded from the Contemporary Controls' web site.

3.2 Backing up a BAScontrol Project

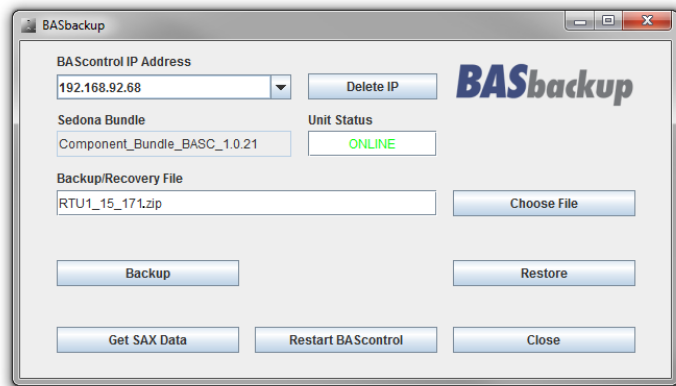
To the left is the default main screen for BASbackup showing the IP address of the BASemulator and a Backup/Recovery File called backup.zip. The BASemulator address will always be present in the BAScontrol IP Address drop-down but you can overwrite it with the desired IP address of the real controller that is to be backed up. Enter the IP address of the controller that is to be backed up in the BAScontrol IP Address field if it is not the BASemulator and then click Enter. This utility program will attempt to communicate with a device at the address entered. Make sure your computer is on the same subnet as the target controller. The target controller must be powered with Sedona enabled and accessible. Once found, the Unit Status field will indicate ONLINE. The utility provides you a backup file location which can be seen in the Backup/Recovery File field. If you want something different you can click on Choose File and select from the files in the Backups folder. It is not necessary that the file exists. Just enter the name of the file you want to create and then click Open. The utility will create the file you named and append



the zip extension as well. If you have a previous file which you want to overwrite, just select it and it will be overwritten. When you click the Choose File button the utility will first look for it in the backups folder but you can use the navigation buttons if the file location is somewhere else on your computer.



Once you are ready, click Backup. An Authorization window will appear from the targeted controller. Enter the User Name and Password of the Sedona application residing in the controller. The default credentials are admin/admin. The User Name and Password for web page access are not necessarily the same as those of the Sedona application. Click OK and the backup should begin.



In our example, we changed the controller IP address and renamed the backup file. You will notice some progress messages followed by a Complete indication if your backup was successful. If so, no error messages will appear. Once the progress messages cease on the main page you can look at the contents of your backup zip file by double-clicking it in the backups folder. You should see the seven files that were identified earlier. You now have a complete project backup of the controller in the backup folder.

3.3 Restoring a Sedona Project

Name	Type
app	SAX File
app.sab.target	TARGET File
bas_cfg	XML File
kits.scode	SCODE File
platform	XML File
platformManifest	XML File
webc	XML File

Restoring a Sedona Project is just as easy. Before you begin the restore process, set the IP address on the main page to the location of the controller to be restored. The controller must be powered up and accessible. You will need to select the appropriate recovery file by first clicking Choose File on the main page and selecting the file.

In this example, we are going to restore the same controller that we just backed up so the main BASbackup screen needs to appear identical to the screen used in the backup process. In this situation, the backup file and recovery file are the same. Instead of clicking on Backup we are going to click on Restore. If you do not see the correct recovery file appearing in the Backup/Recovery File field,

click on Choose File to find it. Once you have the correct recovery file and IP address displayed on the main screen, click the Restore button on the main page to get the Restore Setup screen.

Look at the top of the screen and you will see the following parameters:

- IP Address
- Netmask
- Gateway
- DNS1
- DNS2
- BACnet Device Instance
- BACnet Device Name

These seven parameters will be loaded into the controller being restored. Are these seven parameters that come from the saved zip file what you want? Is the IP address of the target controller the same as shown on Restore Setup screen? They should be the same if you picked the correct backup file. Now study the Restore Options at the bottom. There are some options available before restoring a controller.

Under the Restore Options there are three boxes:

- Wiresheet
- Main Configuration
- Web Component Configuration

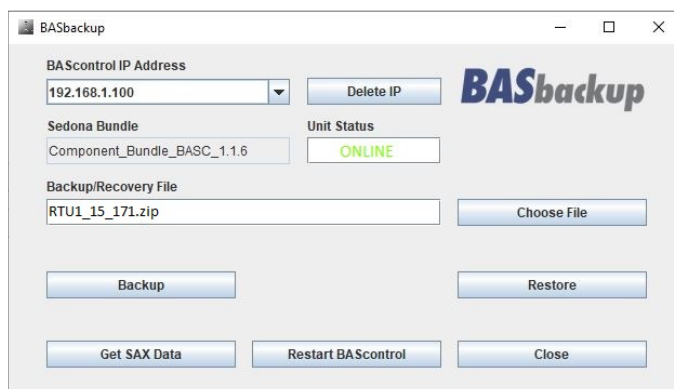
By un-checking boxes you can control what individual files are being restored. Consider this an advanced feature because under normal conditions you want to make sure that all files you saved in the project backup file are going to be reflected in the restored controller.

Just above the Restore Options is a field for a Recovery File Name. By selecting a file using the Choose File button, you will save a copy of the file being restored. Since you are restoring a controller to its saved backup file there is no need to make another copy so leave the field blank. Once you are convinced the settings are correct, click the Restore button.

Once the Restore operation is completed, evidenced by the completion of progress messages, the application program in the target controller continues to run the old program. In order to run the newly restored program, a controller Restart is required. You will be prompted for an immediate restart. You can do it now or later.

If you want to restart the controller later, this can be accomplished by clicking the Restart BAScontrol button on the BASbackup main page or by restarting from the controller's web page. Regardless of what method is used, care should be exercised when restarting the program on an active controller. The controller and application should be in sight of the systems integrator initiating the restart to confirm a safely functioning restart of the application.

3.4 Cloning a Sedona Project



The screenshot shows the BASbackup application window. It has a title bar with the text 'BASbackup'. Inside the window, there are several sections: 'BAScontrol IP Address' with a dropdown menu showing '192.168.1.100' and a 'Delete IP' button; 'Sedona Bundle' with a text field showing 'Component_Bundle_BASC_1.1.6' and a 'Unit Status' label showing 'ONLINE'; 'Backup/Recovery File' with a text field showing 'RTU1_15_171.zip' and a 'Choose File' button. At the bottom, there are three buttons: 'Backup', 'Restore', and 'Restart BAScontrol'. There is also a 'Close' button at the bottom right. The 'BASbackup' logo is visible in the top right corner of the window.

It is also possible to restore a saved project to a different controller that needs to run the identical program stored in the saved project file. However, the IP address, netmask and gateway address stored in the saved project file need to reflect the target controller otherwise there will be an IP conflict. In addition, the BACnet Device Instance and BACnet Device Name must be unique so it must be changed from what is in the saved project file.

On the BASbackup main page, set the IP address of the target controller and choose the backup file that

is to be used for cloning. In this example, we are going to use the same RTU1_15_171.zip for cloning to the 192.168.1.100 controller. Once the information is correct, click Restore which gains you access to the Restore Setup page.

Restore Setup

IP Address
192.168.1.100

Netmask
255.255.250.0

Gateway
192.168.1.1

DNS1
8.8.8.8

DNS2
8.8.4.4

BACnet Device Instance
2749

BACnet Device Name
RTU2_21310

Recovery File Name
RTU2 13 10.zip Choose File

Restore Options

☒ Wire Sheet

☒ Main Configuration

☒ Web Component Configuration

Restore Close

Configuration values in the text boxes will replace values in the Main Configuration file when a recovery operation is performed.

The recovery process generates new files to be uploaded to the controller. If you want to save a backup of the recovery data, choose a file; when the recovery process is complete, a new zip file will be created.

On the Restore Setup page, replace the IP address from the saved file to that of the target controller. The Netmask, Gateway and DNS settings probably do not need to change. The BACnet Device Instance must be changed to something unique as should the BACnet Device Name. Since this program for RTU2 is the same as used by RTU1 we included RTU2 in the BACnet Device Name.

It is a good idea to save the new project settings to a new zip file for easy recovery. In this way we will have a unique backup file for every controller on the job. In our example, the target controller is RTU2 so that is what we are going to enter for the new recovery project name. Click on Choose File and enter the new project file name which will appear as the Recovery File Name. Under Restore Options, we will leave the three Restore Option boxes checked.

This is important to remember. The IP address on the Restore Setup screen will become the target controller's IP address once the controller is restarted. So it is possible to send the Restore Setup data to a controller with a different IP address than the one indicated on the Restore Setup screen. The file is always sent to the IP address indicated on the BASbackup main page. However, once the controller is restarted it will assume the IP address indicated on the Restore Setup screen. This could be very handy when you have several BAScontrollers at factory-default IP addresses that are to be commissioned to specific IP addresses in the field using the same project file.

Just make sure you provide unique BACnet references and IP addresses for each controller. If you have different IP addresses on the Restore Setup screen and the BASbackup main page, you will receive an Alert message asking if you want to proceed. Click Restore.

Once the Restore operation is completed, the application program in the target controller continues to run the old program. In order to run the newly restored program, a controller Restart is required. You will be prompted for an immediate restart. You can do it now or later.

If you want to restart the controller later, this can be accomplished by clicking the Restart BAScontrol button on the BASbackup main page or restarting from the controller's web page. Regardless of what method is used, care should be exercised when restarting the program on an active controller. The controller and application should be in sight of the systems integrator initiating the restart to confirm a safely functioning restart of the application.

When you are finished, you will have cloned a controller in the field but configured it appropriately in terms of IP address and BACnet settings. You also have created a new backup file for project storage.

3.5 Getting SAX Data

SAX File Statistics

☐ Use SAX file from Zip
(backCcsiBc22RTuv2.zip)

☒ Use SAX file on BAScontrol
(10.0.15.170)

Device Name: Bc22_CvRtuV2

App Name: Bc22_CvRtuV2_6

Number of Components: 185

Number of Links: 216

RAM Used: 16280

Complete

Get Data

Close

There is a convenient feature with the BASbackup utility. By having the utility create a SAX file from a SAB file, we can determine useful information about the Sedona program in terms of component count and links including the amount of memory used. This feature applies to both a saved project file or to the controller itself. On the

BASbackup main page, click on Get Sax Data which will bring you to another screen. The default selection will give you data from the saved project file indicated in the Backup/Recovery File field once you click Get Data. If you check the other option and click Get Data, you will see the same information from the controller with the IP address indicated on the BASbackup main page. This is a good way to verify the integrity of the controller memory to that of its backup project file.

4 Sedona Application Editor

4.1 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™ 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 wiresheets 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.

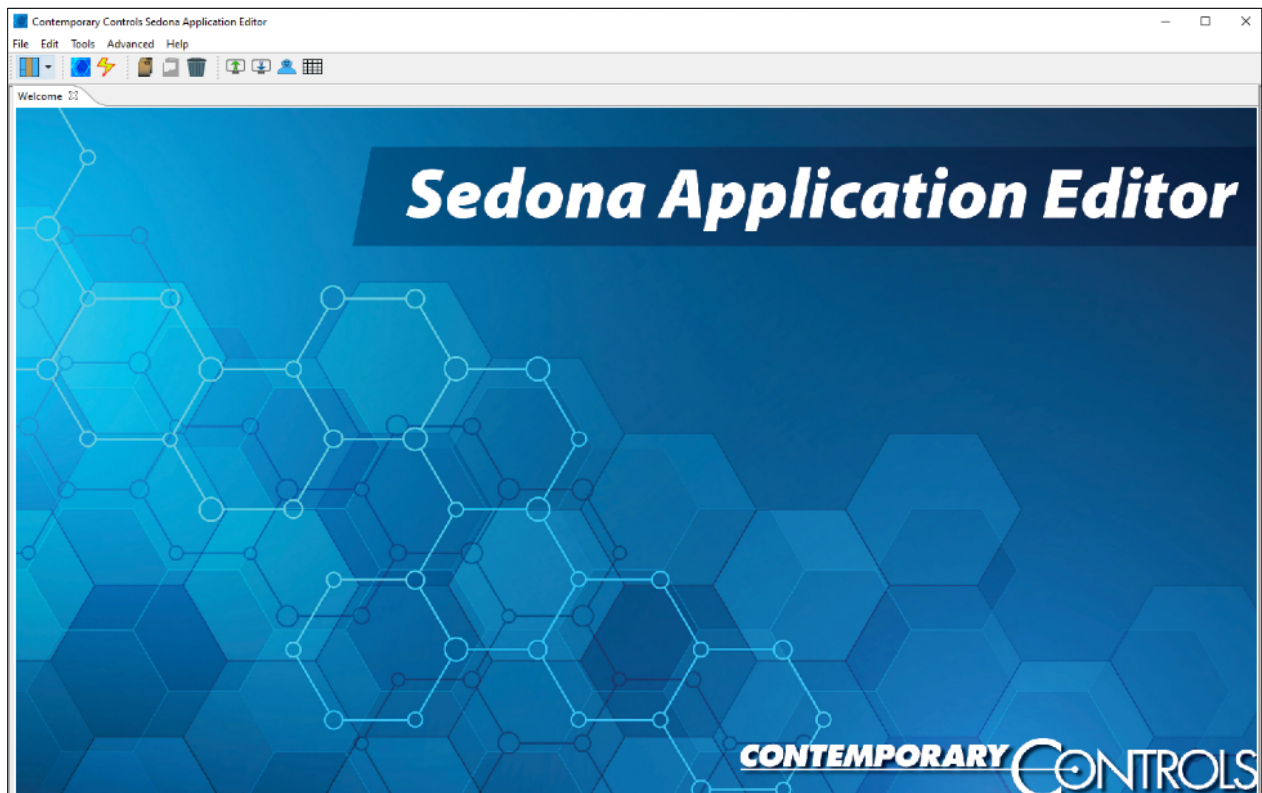
A Sedona device is defined as having a Sedona Virtual Machine (SVM) of which Contemporary Controls has developed two series of BACnet-compliant open controllers that address unitary control and edge control. These freely-programmable controllers can now be programmed with SAE instead of Niagara Workbench. 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 wiresheet 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.

4.2 Starting the SAE and the BASemulator

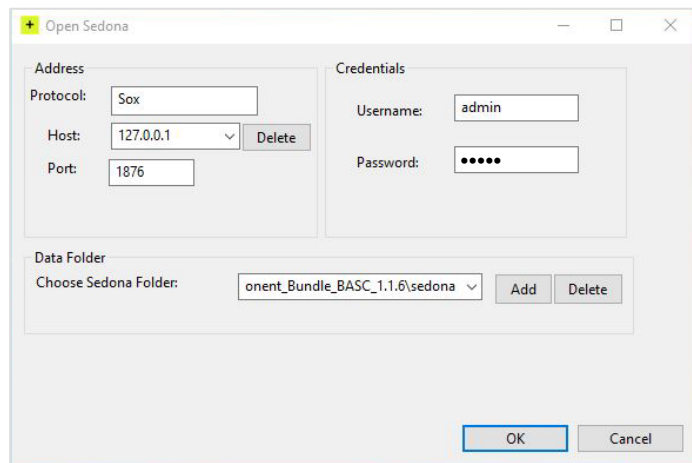
If you want to use the BASemulator, you can start it by opening the Start menu and then All Programs. Find the Sedona Application Editor folder and open it. Click on BASemulator and a window will open. Minimize it but do not close it. Now you can click on the Sedona Application Editor in the same folder.

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 BASemulator 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.



4.3 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 BASemulator 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.



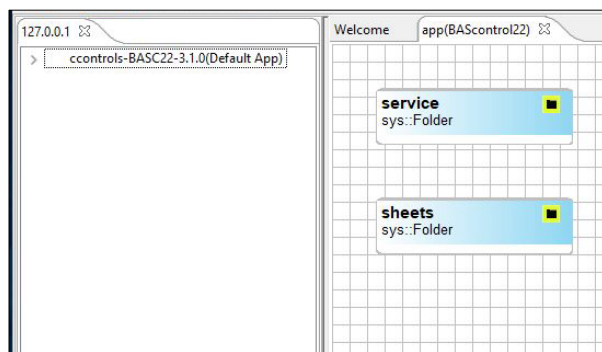
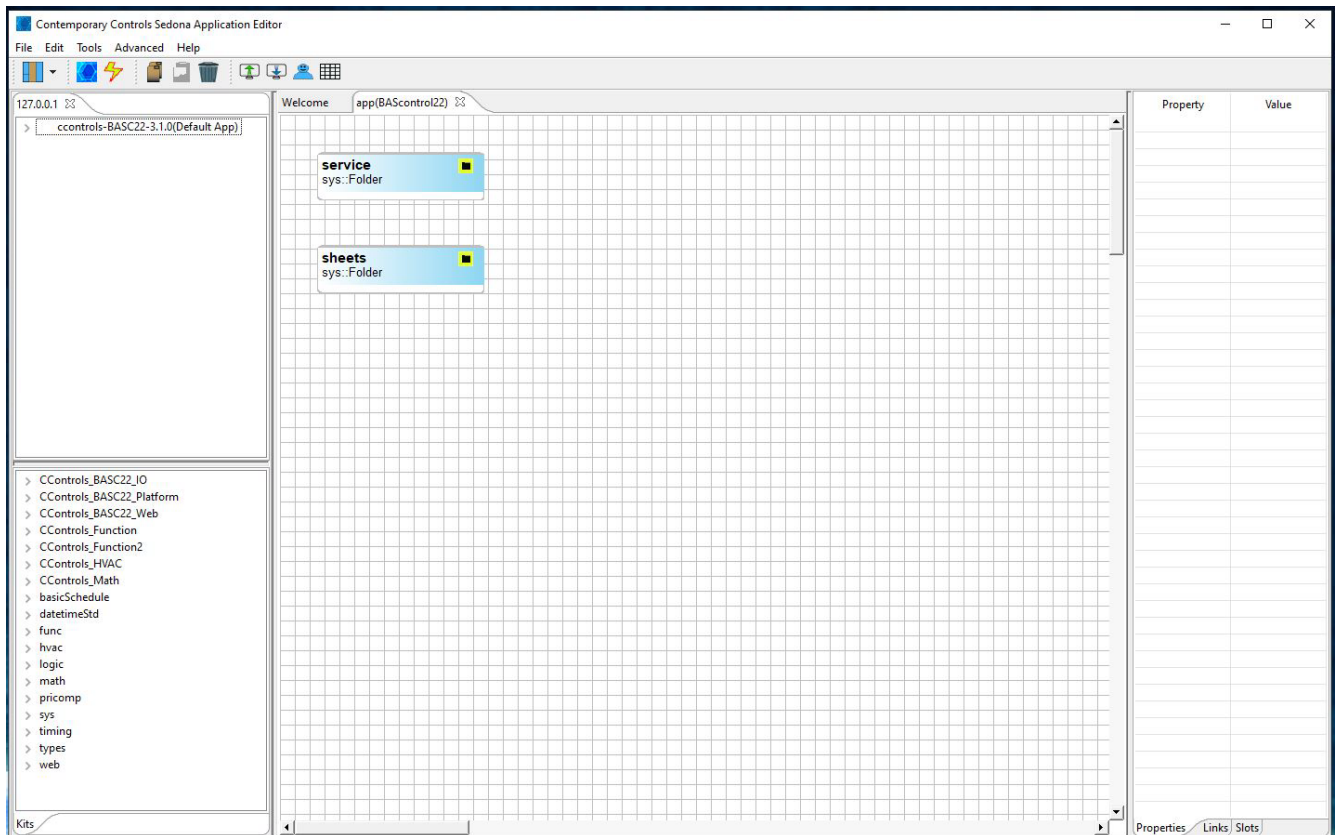
The location of the Data Folder has already been selected for you during installation. If you are seeking an alternate Data Folder (perhaps for a third-party Sedona device), click Add to browse for the Sedona Folder within the computer.

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 BASemulator 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

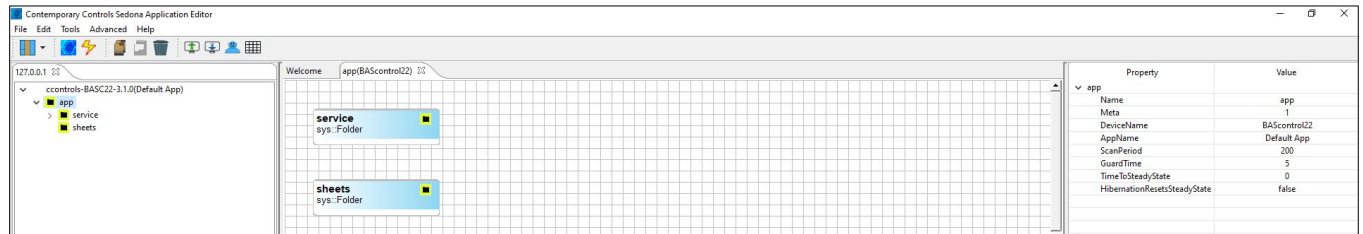
where the installed kits from the connected controller are displayed. Within the various kits are the components that will be used to create applications. On the right is the Properties pane where the attributes of selected components are displayed. The Navigation, Kits and Properties panes can be hidden or displayed using the left-most icon on the tool bar.



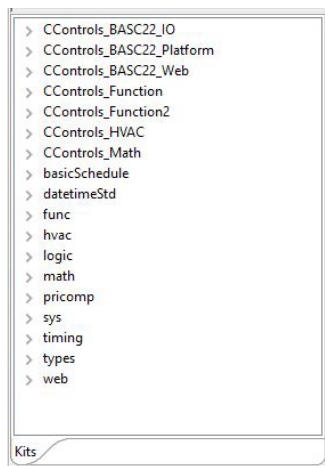
When you first connect to a Sedona device you will be presented with two folder components in the Work Area pane under a tab called app followed by the connected Sedona device name. One is called the service folder and the other the sheets folder. For now, disregard the service folder (never delete it!) 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 wiresheet. This is where components

located in kits are dragged onto a wiresheet, having their properties configured, and interconnected with links to form applications. Wiresheets 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 wiresheets so tabs are used at the top of the Work Area to indicate what wiresheet 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.

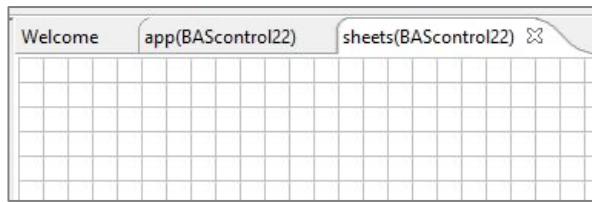


You can expand the Platform ID in the Navigation pane to show app and then service and sheet. If you click on app, you will see its properties in the Properties pane at the fair right. The only interesting information for app properties is DeviceName, AppName and ScanPeriod – all of which can be modified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



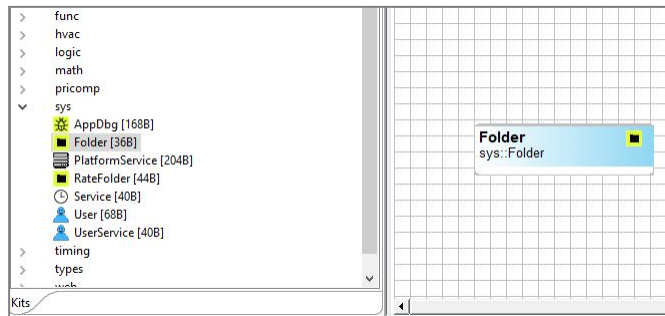
At the bottom left side is the Kits pane which lists the available kits that are present in the connected Sedona device. While all known Tridium and Contemporary Controls' kits are stored in the Sedona Applications Editor, only those loaded in the connected Sedona device are shown. If an unknown Sedona 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 BASemulator 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.

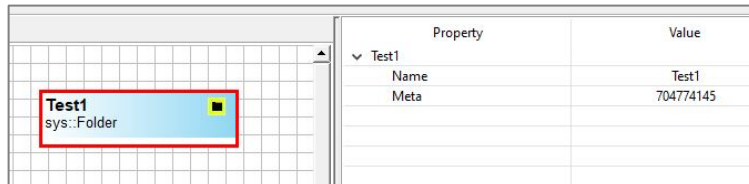


Double-click the Sheets folder in the Work Area pane. Since the connected controller does not have any application installed the wiresheet is blank. Notice the tab at the top called Sheets. We call this sheets folder the main wiresheet because this is where you begin to add components.

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

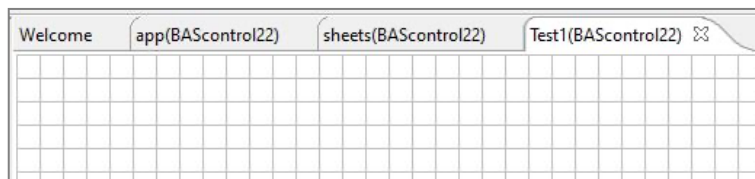


The first component we will put down is a folder that can be found in the sys kit. Expand the sys kit, find the Folder component and drag it onto the wiresheet.

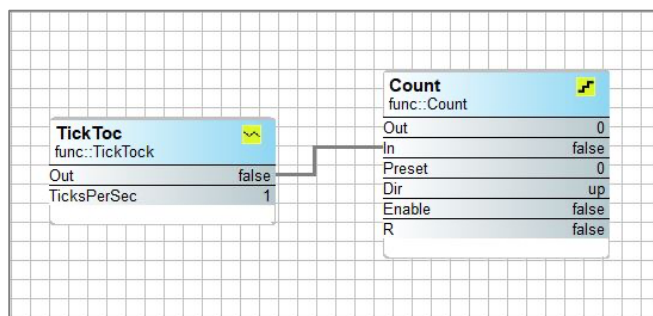


You have an opportunity to rename the component. Select it so that it becomes highlighted with a red border allowing its properties to appear in the Properties pane. Click on the name field in the properties pane and type in Test1. Hit return and the name

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.

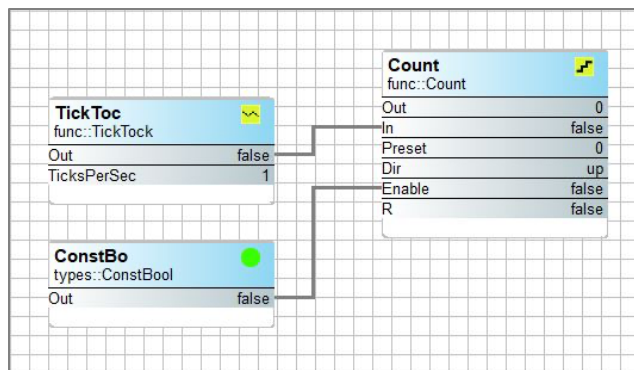


Double-click the component and a new blank wiresheet appears. This shows the contents of your new folder. At the top of the wiresheet is a new tab with the name of the folder you created – Test1. The folder is open so we can add components to this folder.



Contract the sys kit and then expand the func kit. Find TickTock and drag it onto the wiresheet. 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 wiresheet 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.

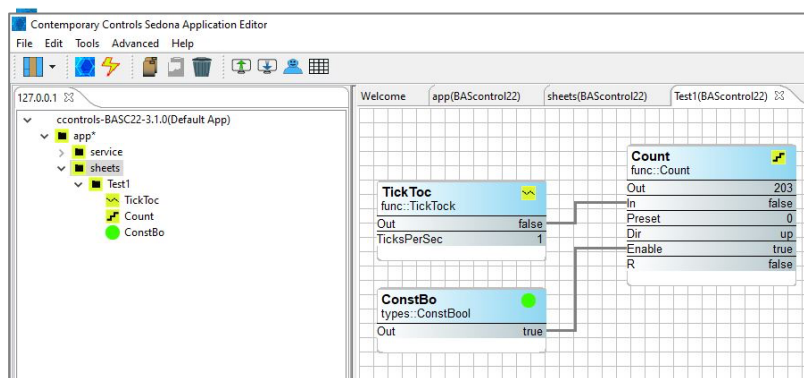


Contract the func kit and then expand the types kit. Find ConstBool and drag it below TickToc. Notice that the component name gets automatically truncated to seven characters. Connect a link between Out on ConstBo and Enable on the Count component. Notice that Count is not counting. This is because Enable 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 wiresheet. You have now completed your first program.

The screenshot shows the Contemporary Controls Sedona Application Editor. On the left, a wiring diagram shows three components: **TickToc** (func::TickTock), **ConstBo** (types::ConstBool), and **Count** (func::Count). The **TickToc** component has an **Out** slot set to **false** and a **TicksPerSec** slot set to **1**. The **ConstBo** component has an **Out** slot set to **true**. The **Count** component has an **Out** slot set to **247**, an **In** slot set to **false**, a **Preset** slot set to **0**, a **Dir** slot set to **up**, an **Enable** slot set to **true**, and an **R** slot set to **false**. On the right, a properties table shows the values for the components.

Property	Value
9.1->8.5	
FromComponent	/sheets/Test1/ConstBo
FromSlot	out
ToComponent	/sheets/Test1/Count
ToSlot	enable
7.1->8.2	
FromComponent	/sheets/Test1/TickToc
FromSlot	out
ToComponent	/sheets/Test1/Count
ToSlot	in
TickToc	
Name	TickToc
Meta	587464705
Out	false
TicksPerSec	1
Count	
Name	Count
Meta	822214657
Out	247
In	false
Preset	0
Dir	up
Enable	true
R	false
ConstBo	
Name	ConstBo
Meta	587857921
Out	true

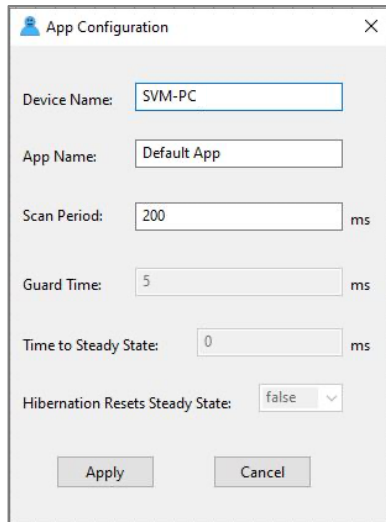


Go to the Navigation pane and expand the sheets and Test1 folders. You will see a different representation of your program. Under Test1 you will see the three components you added to a wiresheet. They are shown in the order that you added them to the wiresheet. In fact, this is the order of execution 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 wiresheet.

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 BASemulator 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 BASemulator this is not an issue because there are no kits supporting physical I/O.



The App Configuration dialog box contains the following fields and controls:

- Device Name: SVM-PC
- App Name: Default App
- Scan Period: 200 ms
- Guard Time: 5 ms
- Time to Steady State: 0 ms
- Hibernation Resets Steady State: false (dropdown menu)
- Buttons: Apply, Cancel

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 BASemulator and all the resources of a PC available we set the scan time lower. Click Apply.

Property	Value
▼ app	
Name	app
Meta	1
DeviceName	SVM-PC
AppName	Default App
ScanPeriod	200
GuardTime	5
TimeToSteadyState	0
HibernationResetsSteadyState	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 BASemulator 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.

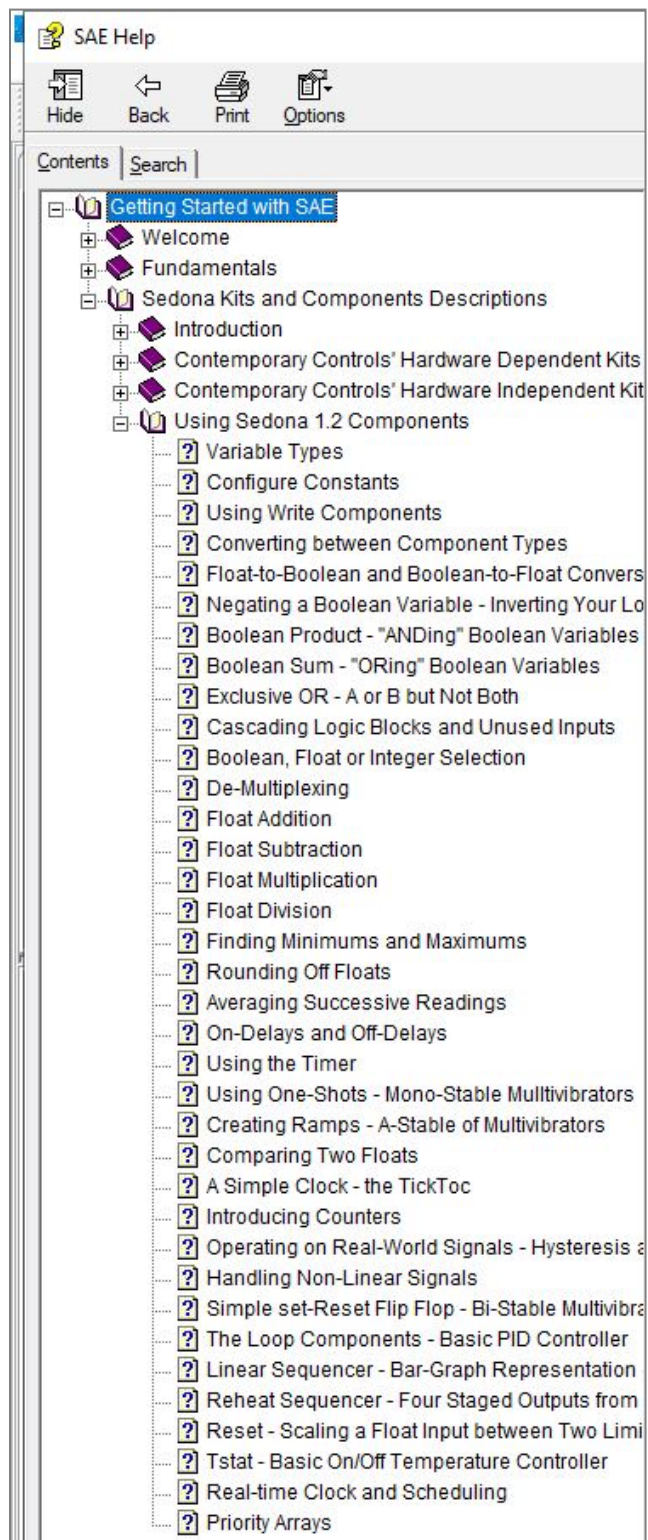
Go back to your wiresheet and draw a rectangle around the Count component to highlight it. Right-click the component and click Delete.

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 BASemulator.

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 BASemulator, you may be required to restart the BASemulator yourself.

During the load process, SAE compiles 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.



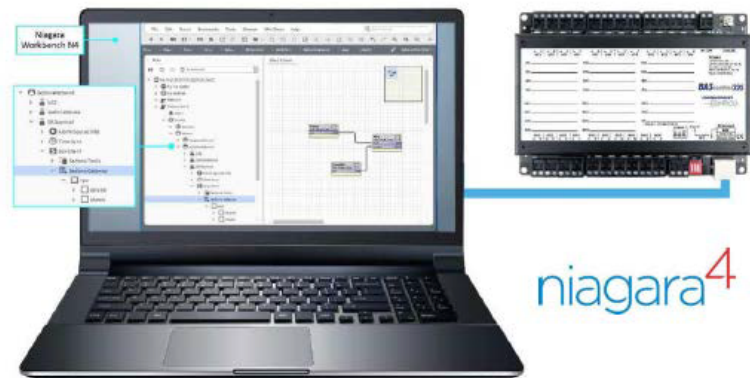
5 Sedona Programming Using Niagara 4 Workbench

When Niagara 3.x transitioned to Niagara 4.x, Sedona programming was lost. Our N4 Sedona Driver allows N4 Workbench (version 4.11 or later) to program our Sedona open controllers.

With the installed N4 Sedona Driver, Niagara 4 can be used to communicate over the SOX protocol to read and write Sedona points and to program the wiresheet on Sedona devices. The Sedona Driver includes features once found in Workbench AX, such as a Sedona Kit Manager and the ability to save and restore SAX files. One difference with the N4 Sedona Driver is that you need to be running a Niagara station which can be an advantage when remote access to a site is required. A VPN connection to the Niagara station at the site provides access to all Sedona controllers connected to the station.

The N4 Sedona Driver is licensed from Sedona community member, Ontrol, and is available for our Sedona controllers at no additional cost.

Visit [N4 Sedona Driver](#) to learn more and to download the driver.



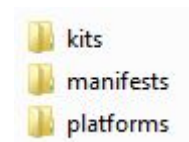
6 Understanding Sedona Kits and Kits Manifests

According to the SedonaDev.org site, a kit is the basic unit of modularity in the Sedona Framework. Kits contain one or more components and it is the components that are assembled onto a wiresheet and linked together to form applications. If a system integrator wants to use a particular component in an application, that component must be part of a kit installed on the target Sedona device. If the kit is missing, in most instances it is possible to load that kit into the Sedona device using the Kit Manager found in a Sedona tool such as Niagara Workbench or Sedona Application Editor (SAE). For the SAE, video 7 in the SAE series: [Introduction to the Kit Manager](#) can be viewed to understand its operation.

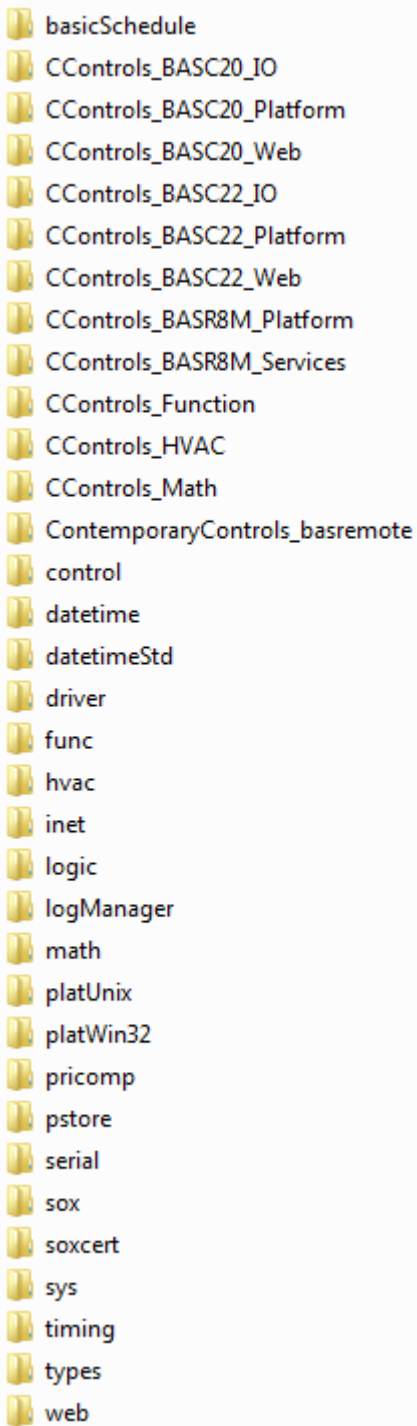
Modularity of kits is important if the Sedona community is to share components among its members. Contemporary Controls is a Sedona community member and offers its hardware-independent kits to Sedona community members in the form of a component bundle. This bundle is to be installed in the member's Sedona tool. This document attempts to explain the concept of kits and kit manifests as they relate to the sharing of hardware-independent components.

6.1 Component Bundle Structure

A component bundle is the term used to identify a single zipped file consisting of kit, manifest and platform files relevant to a particular vendor's Sedona device or devices. It is identified by a Sedona vendor's name along with a version identifier and it is provided by the Sedona vendor so that Sedona tools such as Niagara Workbench or Sedona Application Editor can be used to program the vendors' Sedona devices. In the case of Contemporary Controls (CC), CC has chosen to include information on all its Sedona devices with each release of a single component bundle such as Component_Bundle_BASC_1.0.40. If there is a change to a component within a kit requiring a kit change, CC will append the new kit and manifest files to the component bundle and release the bundle with a change in revision number. Kit and manifest files are not to be deleted – new ones are always appended.



If you extract the component bundle onto your desktop you would see one folder called sedona. This would be what is called sedona home. Expanding this folder will yield just three folders – kits, manifests and platforms.



In terms of portability, the platform folder, or what is called the platform database, can be ignored because platform information is unique to each Sedona vendor and dependent upon native implementations. It is not that this information is unimportant; it is just meaningless to share among community members. What is more of interest to the Sedona community members are the kits and the kit manifests that describe the kits. The kits folder is called the kits database and the manifest folder is called the manifest database. If you expand either the kits or manifests folders, you will see what appears as identical structure of folders – each one devoted to a particular kit. However, the function of the two databases is different. We will first examine the Component_Bundle_BASC_1.0.40.




Kits can be characterized as being Tridium-release, custom hardware-independent, and custom hardware-dependent. The Tridium-release and custom hardware-independent kits can be shared with members of the Sedona community for use with their Sedona implementations. The custom hardware-dependent kits rely upon native hardware implementations and are usually useless to share. You can identify the kit type as follows:

- Tridium-release kits have no vendor name – the vendor is assumed Tridium
- The custom hardware-independent kits have the Sedona vendor name
- The custom hardware-dependent kits have the Sedona vendor name and model reference

Of the custom kits shown on the right, only the CControls_Function, CControls_Math and CControls_HVAC are custom hardware-independent kits and can be shared with the Sedona community. Do not assume that all Sedona devices incorporate all the kits listed – they do not. For example, the BASC20 uses the BASC20 hardware dependent kits and not the BASC22 or BASR8M kits. Not all Sedona devices use the Tridium control kit which was an earlier version that included almost all of the kits on the list. Tridium later broke up the control kit into several kits with more meaningful kit descriptions such as Math, Func and HVAC. Do not confuse the CControls_Math and CControls_HVAC kits with those of Tridium (Math, HVAC). Contemporary Controls created custom components that complement the Tridium components but elected not to merge these components with the Tridium-release components. Contemporary Controls' policy

was not to modify any of the Tridium components unless absolutely necessary in creating Sedona virtual machines or add components to a Tridium kit. All Contemporary Controls' kits have a CControls prefix.

6.2 Kits and Manifest Naming

 basicSchedule-7fdca638-1.0.46.kit	5/14/2010 12:38 PM
 basicSchedule-7fdca638-1.0.47.kit	7/14/2011 7:27 AM
 basicSchedule-7fdca638-1.2.28.kit	1/6/2016 3:43 PM

We will study how the kit and manifest files are named by expanding the basicSchedule kit which is a Tridium-release kit.

In this instance there are three versions of the basicSchedule kit that are available for use.

Each version is identified by kit name, checksum and revision. Up to a four-digit revision number is allowed of the format: major.minor.build [.patch]. Kits must be globally unique. That is why Tridium proposed naming rules for vendors so no naming conflicts occur. During creation of the kits, the checksum is automatically calculated based upon component types within the kit and the number of slots used by the components. However, the revision level is supplied by the developer of the kit. The operation of a component within a kit could be changed with modified code but that may not necessarily change the checksum of the kit. That is when a revision level needs to be incremented by the developer.

In the dateSchedule example provided, a Sedona device can only use one of the three possible kit versions listed or none at all. During Sedona development Tridium released several Sedona builds beginning with Sedona -1.0.31 on up to the current build Sedona 1.2.28. Kits could have incurred modification during the development process which accounts for the different revision levels. Other kits only have one revision level. Generally speaking, the later version is the one to use but older Sedona devices might rely upon earlier versions so that is why it is best to retain earlier versions of the kits in the component bundle.

The kit files (.kit) are not human readable. They are a compact representation of their components so they can be installed in embedded Sedona devices with limited memory. For example, the binary format of a Sedona application (app.sab) does not contain kit, type, or slot definitions, only their numeric ids. So in order to reconstruct the original application from its binary SAB file, we must have some external way to map those ids to the right definitions. Another file is needed called the kit manifest or simply the manifest file. Notice we have a one-to-one matching of a kit folder to a manifest folder. Next, we will look at the basicSchedule manifest folder.

There happens to be only one file in this folder. The naming is similar in that this file has the same checksum but there is no version number. However, this file is an XML file (.xml) and is human readable but tedious to read. The checksum in the manifest folder must match the checksum in the kits folder. With our basicSchedule example, the single manifest entry will work with all three kit versions because the checksums are all the same.

6.3 Sedona Tool Database

A Sedona tool such as Niagara Workbench or Sedona Application Editor that is to connect to a Sedona device must have the same kits and manifests in its database as the ones used on the attached Sedona device or an error message will occur prompting the user to install the missing kits or manifests. Therefore, it is up to the Sedona vendor to provide the latest kits and manifest information conveniently available to the Sedona community especially for newly released products. One way of doing this is with a Component Bundle.

6.4 Installing Component Bundle to Niagara Workbench

Our Sedona controllers can be programmed using N4 Workbench after installing a Sedona driver. See [Application Note – How to Install and Use the N4 Sedona Driver in Niagara 4](#).