CONTEMPORARY CONTROLS®

Creating an Open Controller with Sedona Framework™

George Thomas
Contemporary Controls

White Paper
Revised February 15, 2016

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. The Sedona language is a component-oriented programming language similar to Java or C#. By utilizing this language, custom components can be developed and assembled into applications. The Sedona Framework trademark is owned by Tridium, Inc. but can be used by acknowledging the owner. More importantly, the Sedona Framework technology is available to the public under an Academic Free License granted by the licensor – Tridium, Inc. A licensee is allowed the worldwide, royalty-free and non-exclusive use of the technology. For those who want to be part of the Sedona Community, software and documentation is available for download from sedonadev.org. Contemporary Controls is a community member and views this technology as the best hope in creating a truly open controller.

Contemporary Controls defines an open controller as follows:

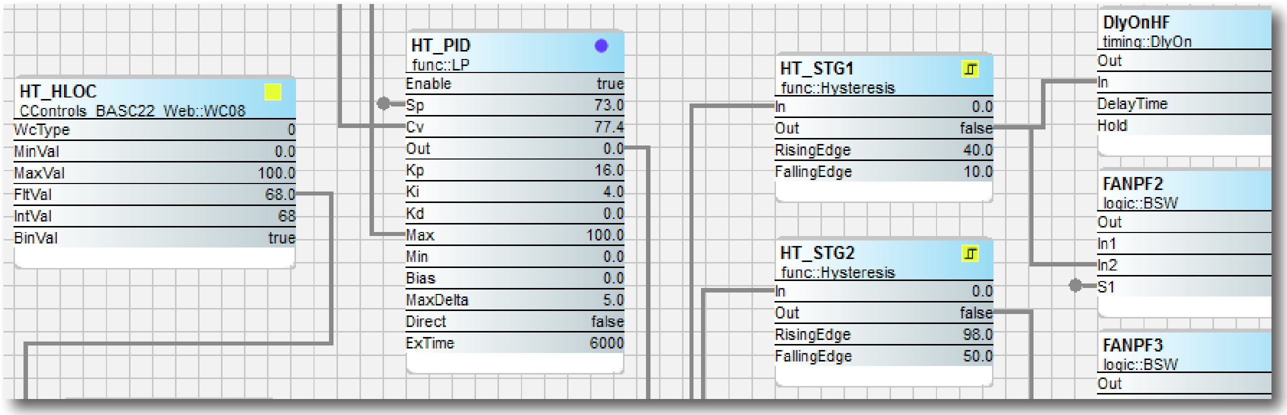
- Utilizes an open protocol for network communications
- Supports an open programming language for implementing control strategies
- Provides a programming tool that is available to systems integrators without restriction
- Fosters a community of developers and integrators that share technology for the public good

Open Protocol for Network Communications

BACnet® is recognized as an open protocol supporting several network variants including IP but it is not a programming language. It defines the way data is represented and communicated over a network and should be used in an open controller. Criticism of BACnet as not being open is only because BACnet compliant controllers may require different programming tools that may not be available to all system integrators. This is an issue BACnet cannot address because it is not involved with programming. However, BACnet has world-wide appeal and BACnet/IP has the additional benefit of providing access to the web-based suite of protocols making data access through web browsers viable.

Open Programming Language for Implementing Control

Sedona Framework is a component-oriented programming language where components are assembled onto wire sheets creating applications (see Figure 1). This language is ideally suited for graphical representation of control strategies. It has a similar look-and-feel to the popular Niagara Framework™ and it is IP-based. 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.



HT_HLOC
CControls_BASC22_Web::WC08

WcType	0
MinVal	0.0
MaxVal	100.0
FltVal	68.0
IntVal	68
BinVal	true

HT_PID
func::LP

Enable	true
Sp	73.0
Cv	77.4
Out	0.0
Kp	16.0
Ki	4.0
Kd	0.0
Max	100.0
Min	0.0
Bias	0.0
MaxDelta	5.0
Direct	false
ExTime	6000

HT_STG1
func::Hysteresis

In	0.0
Out	false
RisingEdge	40.0
FallingEdge	10.0

HT_STG2
func::Hysteresis

In	0.0
Out	false
RisingEdge	98.0
FallingEdge	50.0

DlyOnHF
timing::DlyOn
Out
In
DelayTime
Hold

FANPF2
logic::BSW
Out
In1
In2
S1

FANPF3
logic::BSW
Out

Figure 1—Sedona Wire Sheet with connected components to form applications.

Programming Tool Available to Systems Integrators without Restrictions

To assemble components onto a wire sheet requires a Sedona Tool. Sedona programming tools are available from Tridium, Contemporary Controls or from other sources. Although Tridium's Niagara Workbench is intended for Niagara Framework programming, it can be used as a Sedona programming tool with the installation of Sedona. Workbench is ubiquitous in the building controls industry and available from several OEMs under different brand names. However, it cannot be claimed to be available to all without restriction. Contemporary Controls has developed a Sedona Tool called Sedona Application Editor which is available to all system integrators independent of their access to the Tridium channel (see Figure 2).

Contemporary Controls Sedona Application Editor
File
Edit
Tools
Advanced
Help
Welcome
CONTEMPORARY CONTROLS
Sedona Application Editor
Version 1.0

Figure 2 — Contemporary Controls' Sedona Application Editor can be used as a Sedona Tool if Niagara Workbench is not available.

Community of Developers and Integrators Sharing Technology for the Public Good

Contemporary Controls serves the Sedona community as both a developer and integrator. As a developer, it produces Sedona devices and has created special Sedona components beyond those supplied by Tridium. As an integrator it has assembled components onto wire sheets in creating control strategies and has offered these strategies to its customers as example applications. Components are deployed in kits and some kits are dependent upon the hardware platform and some not. Those custom component kits that are hardware independent are available free to the Sedona community in the spirit of sharing in the development of a truly open controller.

Sedona Workflow Model

Much of the following information comes from the SedonaDev.Org site. The actual process of developing a Sedona device is complex and only understandable by skilled software professionals. We will only provide an overview of the process as we study Figure 3.

The Sedona community consists of device vendors, software developers, system integrators and end users. Contemporary Controls' dual role is that of a device manufacturer and software developer where we produce Sedona-based products to be used by system integrators and end users. The roles of Developer and Integrator differ.

- **Developer** – software professional that develops *components* that are assembled into *modules* and deployed as *kits*.
- **Integrator** – experienced systems integrator using graphical tools places components onto a *wire sheet,* connects *links* between components, *configures* components thus creating an *application*.

Although the developer could probably play both roles, the intent is to develop Sedona devices where a non-programmer, using graphical tools, can implement an application by assembling components onto a wire sheet. Systems integrators understand applications with little time to learn the intricacies of the Sedona programming language, which is left to the developers.

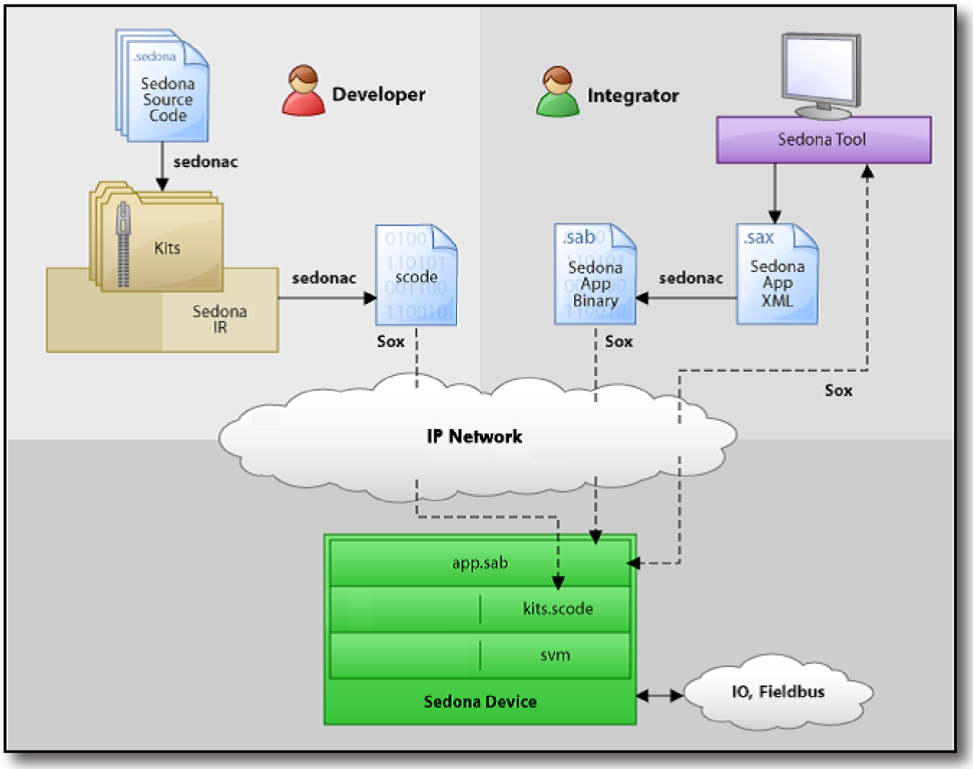


Figure 3—Sedona Workflow Model shows the activities of the developer and integrator.

Using the Sedona language, the developer writes a library of components that are to be compiled into kits. But first, the components are grouped in some logical fashion for meaningful deployment as a kit. Tridium provides examples by grouping logical elements into a logic module or by grouping all mathematical elements into a math module (see Figure 4). Now they can be complied as a Logic kit or the Math kit using the *sedonac* compiler. These pre-defined components are the building blocks used by the integrator when constructing applications. The deployment of components is via kits. Consider a kit to be similar to a Java JAR file or a Windows' DLL.

Once all the kits are defined they must be assembled into one compact file called a *Scode* image the structure of which is dependent upon the target hardware. This *kits.scode* file along with an *app.sab* is what is interpreted by the *Sedona virtual machine (SVM)* once they are placed in the Sedona device.

The Sedona device is shown in Figure 3. The developer must determine the interface to real input/output points and determine a way to communicate to the Sedona programming tool using the Sox protocol. Although Tridium provides a sample SVM, the SVM must be modified to suit the hardware requirements of the target Sedona device. The tradeoffs usually relate to RAM and flash memory space especially when using very small microcontrollers. This was accomplished by Contemporary Controls with its line of Sedona devices. Once the SVM is developed and the Scode image is defined by Contemporary Controls, all that is left is the application which comes from the integrator.

In the Sedona Framework environment, applications can be represented in two forms – a *sax* file or a *sab* file. The sax file is human-readable being that it is structured as an XML file. The sab file is a binary file intended to be interpreted on the Sedona device. Using the sedonac complier, you can move between the two representations of the same application.

Tridium Components and Custom Components

Components are the basic elements for creating applications. Each component type serves a specific function and Tridium provided a head-start by providing a set of commonly needed components (see Figure 4). These components are sorted in several functional groups but it was never the intent of Tridium to create all that were needed to implement an application. The Sedona community was to help. It is the policy of Contemporary Controls to provide the Tridium components – called Tridium's Sedona 1.2 Components – in its Sedona devices without modification. They are easy to recognize because no vendor name is included in the kit name or the component name. The Tridium components are deemed necessary but not sufficient. Custom components need to be created to handle the unique I/O requirements of the Sedona device being developed.

Custom components can be developed by systems integrators or product manufacturers. These components will be tagged with the developer's name and will be deployed in kits with the same developer name. If they only function on particular hardware, they are called product-specific components and should carry both the developer name and the product name. If they function independent of particular hardware, they only need to carry the developer's name. Of course, each component must carry a component name to distinguish its function and a kit name that is somewhat meaningful to distinguish the kit. Contemporary Controls has developed both hardware dependent and hardware independent components all deployed in kits carrying the company name.

Tridium’s Sedona 1.2 Components

The HVAC Group **operations that facilitate control**	**LSeq**	Linear Sequencer — bar graph representation of input value
	ReheatSeq	Reheat sequence — linear sequence up to four outputs
	Reset	Reset — output scales an input range between two limits
	Tstat	Thermostat — on/off temperature controller
The Scheduling Group **scheduling operations based upon time of day**	**DailySc**	Daily Schedule Boolean — two-period Boolean scheduler
	DailyS1	Daily Schedule Float — two-period float scheduler
	DateTime	Time of Day — time, day, month, year
The Function Group **convenient functions for developing control schemes**	**Cmpr**	Comparison math — comparison (<=>) of two floats
	Count	Integer counter — up/down counter with integer output
	Freq	Pulse frequency — calculates the input pulse frequency
	Hysteresis	Hysteresis — setting on/off trip points to an input variable
	IRamp	IRamp — generates a repeating triangular wave with an integer output
	Limiter	Limiter — Restricts output within upper and lower bounds
	Linearize	Linearize — piecewise linearization of a float
	LP	LP — proportional, integral, derivative (PID) loop controller
	Ramp	Ramp — generates a repeating triangular or sawtooth wave with a float output
	SRLatch	Set/Reset Latch — single-bit data storage
	TickTock	Ticking clock — an astable oscillator used as a time base
	UpDn	Float counter — up/down counter with float output
The Priority Group **prioritizing actions of Boolean, Float and Integer variables**	**PrioritizedBool**	Prioritized boolean output — highest of sixteen inputs
	PrioritizedFloat	Prioritized float output — highest of sixteen inputs
	PrioritizedInt	Prioritized integer output — highest of sixteen inputs
The Types Group **variable types and conversion between types**	**B2F**	Binary to float encoder — 16-bit binary to float conversion
	ConstBool	Boolean constant — a predefined Boolean value
	ConstFloat	Float constant — a predefined float variable
	ConstInt	Integer constant — a predefined integer variable
	F2B	Float to binary decoder — float to 16-bit binary conversion
	F2I	Float to integer — float to integer conversion
	I2F	Integer to float — integer to float conversion
	L2F	Long to float — long integer to float conversion
	WriteBool	Write Boolean — setting a writable Boolean value
	WriteFloat	Write Float — setting a writable float value
	WriteInt	Write integer — setting an integer value
The Logic Group **logical operations using Boolean variables**	**ADemux2**	Analog Demux — Single-input, two-output analog de-multiplexer
	And2	Two-input Boolean product — two-input AND gate
	And4	Four-input Boolean product — four-input AND gate
	ASW	Analog switch — selection between two float variables
	ASW4	Analog switch — selection between four floats
	B2P	Binary to pulse — simple mono-stable oscillator (single-shot)
	BSW	Boolean switch — selection between two Boolean variables
	DemuxI2B4	Four-output Demux — integer to Boolean de-multiplexer
	ISW	Integer switch — selection between two integer variables
	Not	Not — inverts the state of a Boolean
	Or2	Two-input Boolean sum — two-input OR gate
	Or4	Four-input Boolean sum — four-input OR gate
	Xor	Two-input exclusive Boolean sum — two-input XOR gate
The Timing Group **time-based components**	**DlyOff**	Off delay timer — time delay from a “true” to “false” transition of the input
	DlyOn	On delay timer — time delay from an “false” to “true” transition of the input
	OneShot	Single Shot — provides an adjustable pulse width to an input transition
	Timer	Timer — countdown timer
The Math Group **math-based components**	**Add2**	Two-input addition — results in the addition of two floats
	Add4	Four-input addition — results in the addition of four floats
	Avg10	Average of 10 — sums the last ten floats while dividing by ten thereby providing a running average
	AvgN	Average of N — sums the last N floats while dividing by N thereby providing a running average
	Div2	Divide two — results in the division of two float variables
	FloatOffset	Float offset — float shifted by a fixed amount
	Max	Maximum selector — selects the greater of two inputs
	Min	Minimum selector — selects the lesser of two inputs
	MinMax	Min/Max detector — records both the maximum and minimum values of a float
	Mul2	Multiply two — results in the multiplication of two floats
	Mul4	Multiply four — results in the multiplication of four floats
	Neg	Negate — changes the sign of a float
	Round	Round — rounds a float to the nearest N places
	Sub2	Subtract two — results in the subtraction of two floats
	Sub4	Subtract four — results in the subtraction of four floats
	TimeAvg	Time average — average value of float over time

Figure 4—Tridium’s Sedona 1.2 components are considered standard components. It is recommended to include them with any Sedona device without modification.

Using the Sedona Tool

For a Sedona Tool we will study the Sedona Application Editor (SAE). Using SAE, the integrator connects to the Sedona device using the SOX protocol which is accomplished with an IP connection. A kits pane shows all the kits present in the Sedona device's Scode image (see Figure 5). Notice that in this example there are both Tridium and custom component kits. The CControls_BASC20_IO and CControls_BASC20_Web kits are product specific kits because they carry both the company name and the product name. The first kit is used to interface physical I/O points to the wire sheet while the other interfaces wire sheet components to web pages. They are to be used only with Contemporary Controls' BASC20 product. The CControls_Function kit is hardware independent and can be used on any Sedona 1.2 compliant device. This kit contains a collection of miscellaneous components that enhance the use of Sedona.

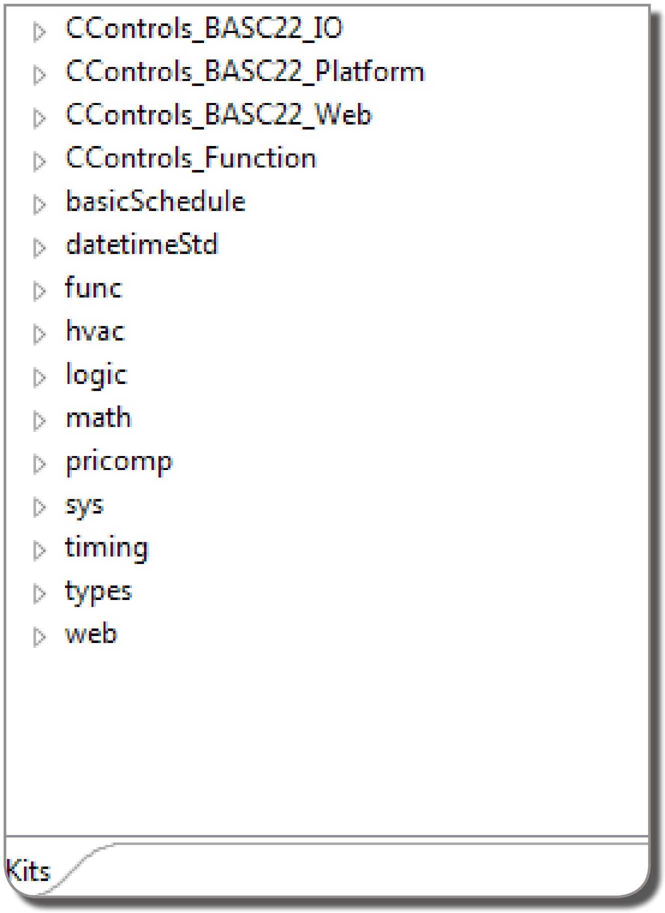

▷ CControls_BASC22_IO
▷ CControls_BASC22_Platform
▷ CControls_BASC22_Web
▷ CControls_Function
▷ basicSchedule
▷ datetimeStd
▷ func
▷ hvac
▷ logic
▷ math
▷ pricomp
▷ sys
▷ timing
▷ types
▷ web

Figure 5—Listing of kits installed in the Sedona device.

Applications can now be created on the Sedona device by expanding the kits in the Kits pane thereby exposing the component building blocks. Components are then dragged onto the wire sheet, configured and then linked with other components to create an application. The application can be tested live on the device with any program changes witnessed immediately. Once the application is finalized, it is saved on the Sedona device in order for all program changes to survive power cycling. It can also be saved as an *app.sax* file using the *Save to PC* function in SAE, which converts the *app.sab* to an *app.sax.* If the *Backup/Restore Tool* in Workbench is used instead, both the *app.sax* and kits.scode files are saved. It may also appear with Workbench that you can backup and restore the SVM. That depends on the hardware implementation of the Sedona device and in most cases this is not possible. The SVM is not intended to be modified once it is shipped as a Sedona device.

There is one important thing to remember with the information in Figure 3. From the figure, it would appear that the kits.scode is developed once and permanently installed in the Sedona device. That depends upon the type of memory in which the kits.scode file is loaded. If it is in ROM there is no hope of change but in flash memory it is possible to provision the file through the Sedona Tool. This provides greater flexibility in that the Sedona Tool can accept additional kits (even from a third-party) that provide components not found in existing kits. Using the *Kit Manager* in the Sedona Tool, these new kits can be added to the Sedona device by recompiling the *app.sax* and kits.scode files and pushing them down to the Sedona device. This can be done by the integrator without an in-depth knowledge of the process. This provisioning capability would allow for the sharing of kits or development of additional kits from those originally shipped with the Sedona device.

A Sedona Device Development Example – the BAScontrol20 as an Open Controller

The BAScontrol20 (BASC20) Sedona Unitary Controller was "built on the Sedona Framework™." This is the proper trademark recognition that should be given to Tridium as the Sedona Framework trademark holder. The BASC20 is a good example of an open controller in that it embodies the attributes of an open controller as defined by Contemporary Controls.

- It is BACnet/IP compliant and incorporates a Sedona Virtual Machine.
- It can be programmed using Niagara Workbench or with a Sedona Tool.
- It is available to any systems integrator without restriction.

If you study the features of this product you will notice design intents beyond just Sedona programming. Sedona only solves the programming aspects of the product and does not address BACnet or web page configuration issues. To be viable in the building controls marketplace the product needed to do more. However, Sedona had enough flexibility in that custom components were able to be created that interface wire sheet data with BACnet clients and web pages. This means that the BASC20 can cooperate with BACnet head-ends and BACnet tools along with being able to be configured or examined using a standard web browser. Remember that Sedona was designed with IP networks in mind.

The BAScontrol20 features are as follows:

- 55Mz ARM7 32-bit processor with 512kB of flash, 128kB RAM and a 16 Mbit serial flash
- 20 Points of I/O
  - Eight universal inputs – thermistor, resistance, analog voltage, binary, and pulse
  - Four binary inputs – voltage-free contacts
  - Four analog outputs – 0-10 VDC
  - Four relay outputs – 30 VAC/VDC 2A or 30VAC 0.5A triacs
- BACnet/IP compliant with a B-ASC device profile
- 16 COVs – 14 binary and 2 analog
- 24 virtual components interface wire sheet data to BACnet clients
- 48 web components interface wire sheet data to web pages
- Real-time clock can be set manually or via a network-time-protocol (NTP) server
- Sedona – 1.2.28 Virtual Machine supports up to a 200 component wire sheet
- Workbench or Sedona Tool programming
- Webpage configuration
- 24 VAC/VDC powered
- DIN-rail mountable

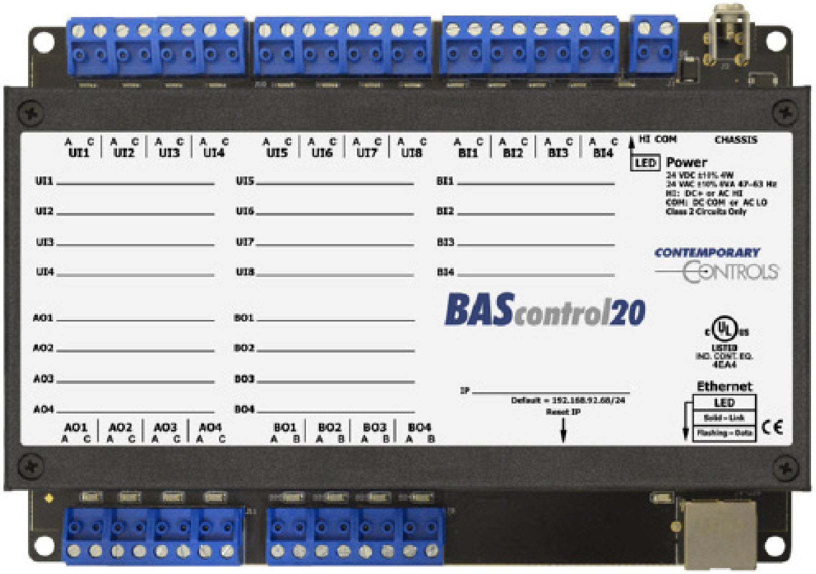


Figure 6—BAScontrol20 Sedona Unitary Controller is BACnet/IP compatible.

By having a 10/100 Mbps Ethernet connection, the BASC20 can easily connect to Workbench or a Sedona Tool for programming or to a web browser for configuration or monitoring. The BASC20 can operate in two modes – freely programmable and remote I/O. As a freely programmable controller a Sedona Tool is needed to input the control logic but if the unit is to be just BACnet/IP compliant remote I/O, only a web browser is needed for channel and device configuration.

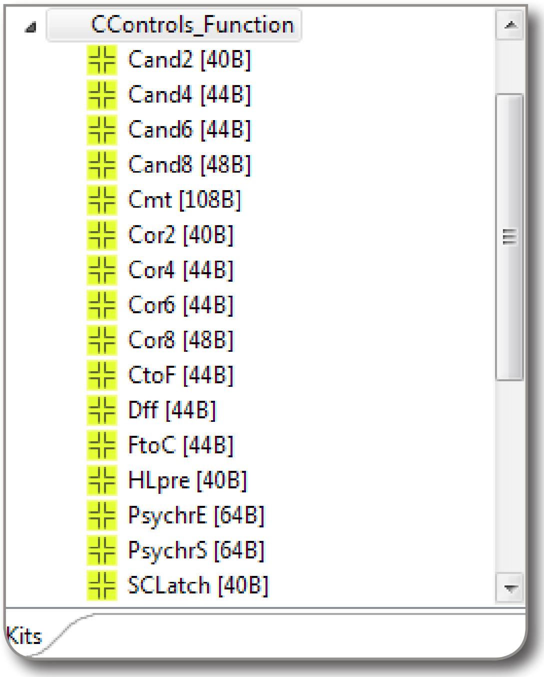
The universal points must be first configured for channel type using a web browser while the remaining real points require no channel configuration. All real points and all 24 virtual points can be configured for BACnet name, type and units. BACnet object instances are pre-defined. It is also necessary to configure the BACnet device parameters with Device Object Name and Device Instance.

The eight universal inputs can be uniquely configured. Type II and Type III 10kΩ thermistor curves as well as a 20kΩ thermistor curve are resident in the unit. Open thermistor conditions can be sensed and an out-of-bounds temperature reading can be assigned. Resistance can range from 1kΩ to 100kΩ. Analog voltage is 0-10 VDC or 0-20 mA using an external resistor. Binary inputs are voltage-free as are the pulse inputs. Pulse inputs only apply to the first four universal points, which have retentive memory for storing pulse accumulations. For passive transducers a pull-up resistor to +12 VDC can be enabled during configuration. High and low threshold points can be assigned.

The BASC20 can be configured for a static IP address or for DHCP. If Internet access exists, the BASC20 can be also be configured for DNS server addresses allowing a domain name to be used when setting time using NTP. The real-time clock can also be set manually and will retain time for up to seven days using capacitor storage. Daylight Savings Time (DST) is supported as well.

Sedona Kits Provide the Personality

The BASC20 utilizes a SVM from the latest release of Sedona – 1.2.28. For short, we call this Sedona 1.2 and the kits that Tridium provides as the Tridium 1.2 kits. Contemporary Controls participated in the Sedona 1.2 certification program. Figure 5 lists all the kits in the BASC20. Of these kits, all those not carrying a company label are from the Tridium 1.2 release. The three with the prefix CControls come from Contemporary Controls. If a model number follows the name it means that the kit is hardware dependent and therefore not portable to another Sedona device. However, the CControls_Function kit is portable and available to the Sedona Community. Figure 7 provides a listing of components in this kit.



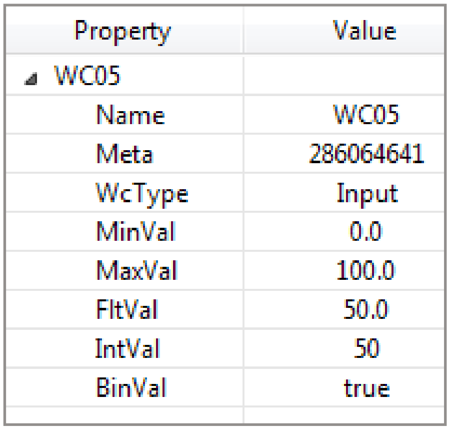
CControls_Function
- Cand2 [40B]
- Cand4 [44B]
- Cand6 [44B]
- Cand8 [48B]
- Cmt [108B]
- Cor2 [40B]
- Cor4 [44B]
- Cor6 [44B]
- Cor8 [48B]
- CtoF [44B]
- Dff [44B]
- FtoC [44B]
- HLpre [40B]
- PsychrE [64B]
- PsychrS [64B]
- SCLatch [40B]

Figure 7—List of components in the CControls _ Function kit.

This function kit is not to be confused with the Func kit from Tridium. It is just a collection of components that Contemporary Controls felt would be useful to integrators. The components include multiple-input AND and OR gates with complementary outputs, plus temperature conversion components and psychrometric components for either SI or English units. It is an example of how a Sedona community member can add value to the Sedona community.

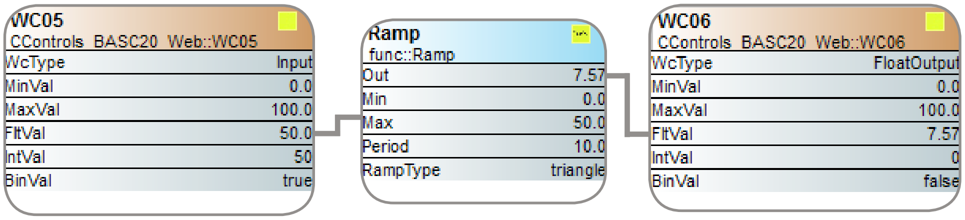
Of the two remaining hardware dependent kits, the CControls_BASC20_Web kit is more interesting in that it is a collection of 48 web components that provide an interface between a Sedona wire sheet and web pages. These components allow a user to provide data, such as setpoints, to the wire sheet without the need of the Sedona Tool. The same component can be configured to supply wire sheet data to a web page. This is a good example of how innovation is supported in an open source project such as Sedona Framework.

In a similar fashion, the CControls_BASC20_IO kit has 24 virtual point components that allow data transfer between BACnet clients and the BASC20 wire sheet. These components can be configured to supply data or to consume data.



Property	Value
WC05	
Name	WC05
Meta	286064641
WcType	Input
MinVal	0.0
MaxVal	100.0
FltVal	50.0
IntVal	50
BinVal	true

The web component is a custom Sedona component from Contemporary Controls. In this example, WC05 is configured using SAE to be a wire sheet input thereby controlling the Ramp output via a web page. WC06 is configured as a wire sheet output displaying the instantaneous output of the Ramp. Notice that WC05 can be configured for limits, which will restrict the range of values that can be inputted from the web page. A similar component exists for BACnet virtual points.



WC05
CControls BASC20 Web::WC05

WcType	Input
MinVal	0.0
MaxVal	100.0
FltVal	50.0
IntVal	50
BinVal	true

Ramp
func::Ramp

Out	7.57
Min	0.0
Max	50.0
Period	10.0
RampType	triangle

WC06
CControls BASC20 Web::WC06

WcType	FloatOutput
MinVal	0.0
MaxVal	100.0
FltVal	7.57
IntVal	0
BinVal	false

Contemporary Controls’ Developed Sedona Components

BAScontrol20 I/O Kit – BAScontrol20 platform specific components

AO1 – AO4	Analog output – analog voltage output point
BI1 – BI4	Binary input – binary input point
BO1 – BO4	Binary output – binary output point
ScanTim	Scan time monitor – records the min, max and average scan times
UI1 – UI4	Universal input – binary, analog voltage, thermistor, resistance or accumulator
UI5 – UI8	Universal input – binary, analog voltage, thermistor or resistance
UC1 – UC4	Retentive universal counters – up/down retentive counters
VT01 – VT08	Retentive virtual points – share retentive wire sheet data with BACnet/IP clients
VT09 – VT24	Virtual points – share wire sheet data with BACnet/IP clients

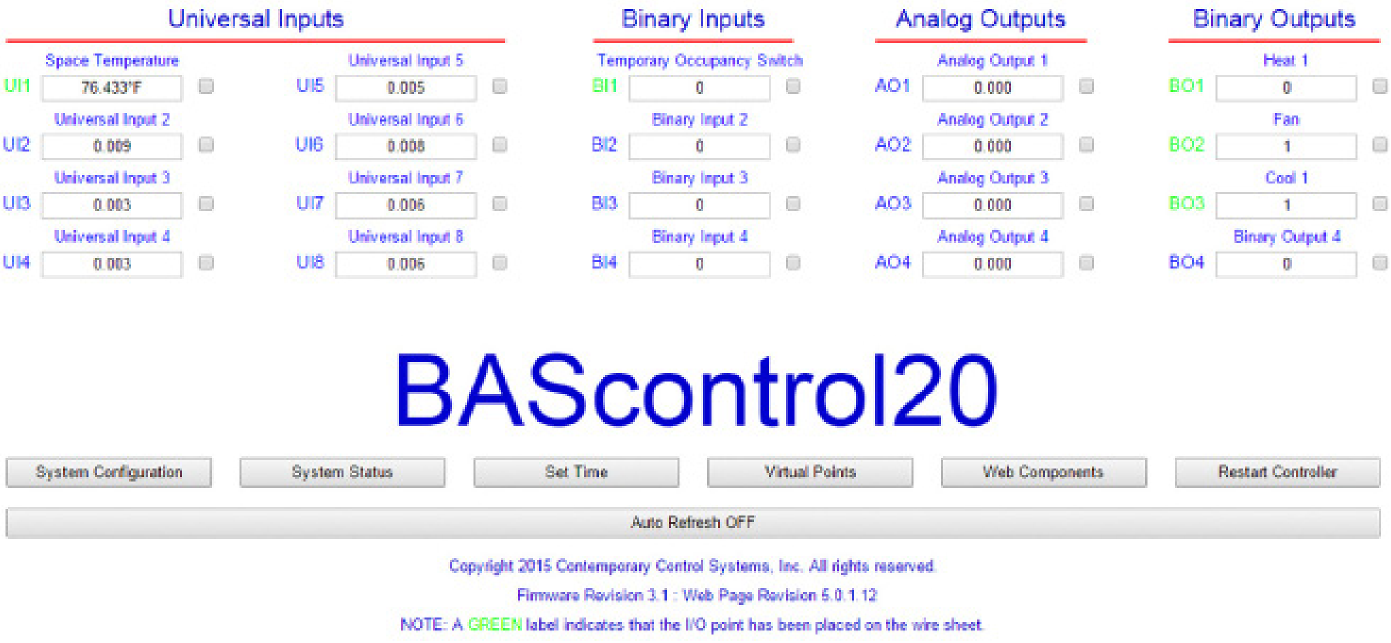
BAScontrol20 Web Kit – BAScontrol20 platform specific components

WC01 – WC48	Web components – share wire sheet data with the BAScontrol20 web pages

Contemporary Controls Function Kit – Common to Sedona 1.2 compliant controllers

Cand2	Two-input Boolean product – two-input AND/NAND gate with complementary outputs
Cand4	Four-input Boolean product – four-input AND/NAND gate with complementary outputs
Cand6	Six-input Boolean product – six-input AND/NAND gate with complementary outputs
Cand8	Eight-input Boolean product – eight-input AND/NAND gate with complementary outputs
Cmt	Comment – comment field up to 64 characters
Cor2	Two-input Boolean sum – two-input OR/NOR gate with complementary outputs
Cor4	Four-input Boolean sum – four-input OR/NOR gate with complementary outputs
Cor6	Six-input Boolean sum – six-input OR/NOR gate with complementary outputs
Cor8	Eight-input Boolean sum – eight-input OR/NOR gate with complementary outputs
CtoF	°C to °F – Celsius to Fahrenheit Temperature Conversion
Dff	“D” Flip-Flop – D-style Edge-triggered Single-bit Storage
FtoC	°F to °C – Fahrenheit to Celsius Temperature Conversion
HLpre	High – Low Preset – defined logical true and false states
PsychrE	Psychrometric Calculator – English Units
PsychrS	Psychrometric Calculator – SI Units
SCLatch	Set/Clear Latch – single-bit level-triggered single-bit data storage

Figure 8—Contemporary Controls has developed two product specific kits and one product independent kit.



Universal Inputs				Binary Inputs		Analog Outputs		Binary Outputs	
	Space Temperature		Universal Input 5		Temporary Occupancy Switch		Analog Output 1		Heat 1
UI1	76.433°F □	UI5	0.005 □	BI1	0 □	AO1	0.000 □	BO1	0 □
	Universal Input 2		Universal Input 6		Binary Input 2		Analog Output 2		Fan
UI2	0.009 □	UI6	0.008 □	BI2	0 □	AO2	0.000 □	BO2	1 □
	Universal Input 3		Universal Input 7		Binary Input 3		Analog Output 3		Cool 1
UI3	0.003 □	UI7	0.006 □	BI3	0 □	AO3	0.000 □	BO3	1 □
	Universal Input 4		Universal Input 8		Binary Input 4		Analog Output 4		Binary Output 4
UI4	0.003 □	UI8	0.006 □	BI4	0 □	AO4	0.000 □	BO4	0 □

BAScontrol20

Copyright 2015 Contemporary Control Systems, Inc. All rights reserved.

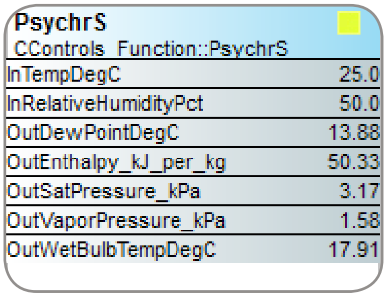
Firmware Revision 3.1 : Web Page Revision 5.0.1.12

NOTE: A GREEN label indicates that the I/O point has been placed on the wire sheet.

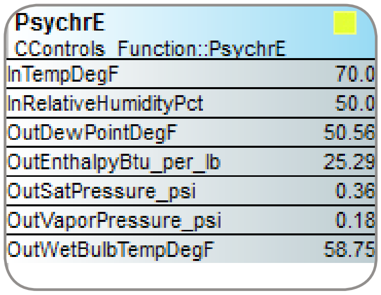
The main web page on the BASC20. All physical I/O points can be viewed with a common web browser. Additional pages exist to view web components and virtual points. From this page access to other pages is possible.

Programmable or Configurable Controller

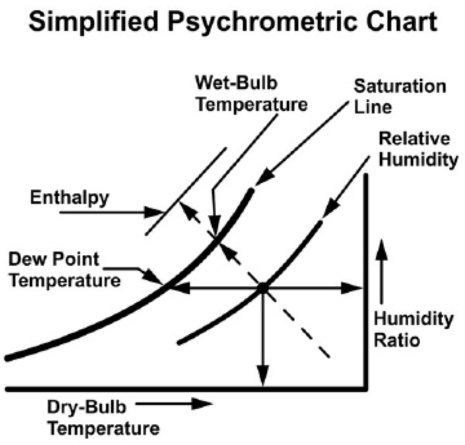
Sedona is inherently freely-programmable using the Sedona Tool but that does not preclude configurable operation with the control application hidden from view. Both web components and virtual point components can be used to configure a pre-designed control application using either a web browser or BACnet tool. The application must first be developed using the Sedona Tool but once the controller is deployed in the field it is not necessary to have the tool at all. Once configured, the controller will automatically boot-up into the application with no operator intervention. If user adjustments are needed, use a web browser, which will be prompted for User Name and Password.



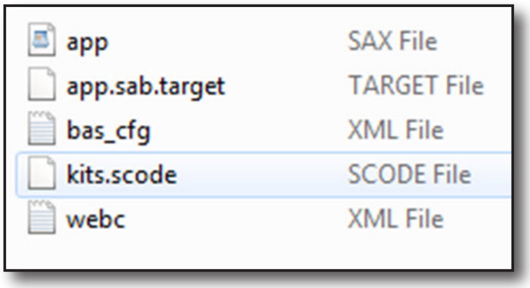
PsychrS CControls_Function::PsychrS	
InTempDegC	25.0
InRelativeHumidityPct	50.0
OutDewPointDegC	13.88
OutEnthalpy_kJ_per_kg	50.33
OutSatPressure_kPa	3.17
OutVaporPressure_kPa	1.58
OutWetBulbTempDegC	17.91



PsychrE CControls_Function::PsychrE	
InTempDegF	70.0
InRelativeHumidityPct	50.0
OutDewPointDegF	50.56
OutEnthalpyBtu_per_lb	25.29
OutSatPressure_psi	0.36
OutVaporPressure_psi	0.18
OutWetBulbTempDegF	58.75



A more complex component is the Psychrometric component demonstrating the range of possibilities when developing custom Sedona components. Input variables are dry-bulb temperature and relative humidity while output variables are dew point, enthalpy, saturation pressure, vapor pressure and wet-bulb temperature. Separate components exist for English and SI units.

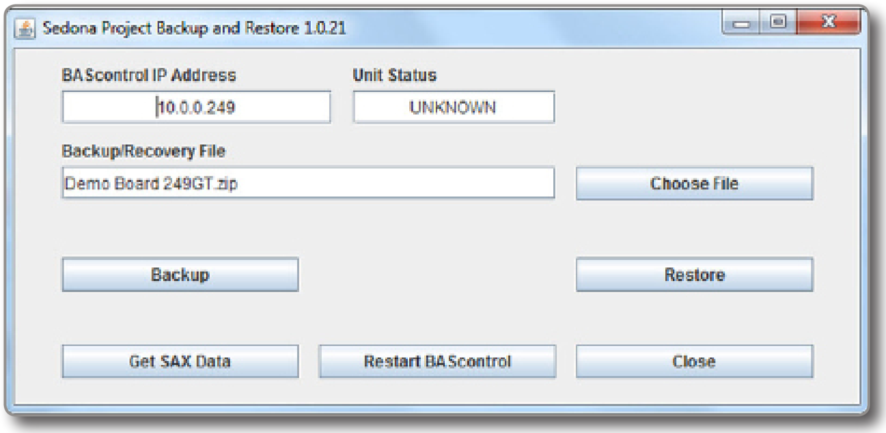


app	SAX File
app.sab.target	TARGET File
bas_cfg	XML File
kits.scode	SCODE File
webc	XML File

Figure 9—Files needed to backup a controller.

Sedona Project Backup and Restore Utility

Although the Workbench tool has a Backup and Restore program it only addresses the Sedona aspects of the controller. To truly backup a BAScontrol20 it is best to use Contemporary Controls' BASbackup – Sedona Project Backup and Restore Utility that is available for free from the Contemporary Controls web site. The list of files in the BAScontrol20 is shown in Figure 9. Two files – the bas_cfg.xml and the webc.xml – are unique to the BAScontrol20 and need to be backed up if everything is to be saved. The first of these two files saves the IP settings and BACnet configurations. The second of the two saves the web component configurations. The SVM is not backed up because is part of the basic firmware of the product itself and does not change. The BASbackup utility handles the complete backup and restore process while saving the complete configuration and wire sheet in one compact zip file.



Sedona Project Backup and Restore 1.0.21

BAScontrol IP Address: 10.0.0.249

Unit Status: UNKNOWN

Backup/Recovery File: Demo Board 249GT.zip

Figure 10—BASbackup main screen.

Summary

Developing a programming language for a controller takes considerable effort but much of this has been done thanks to the initial contribution by Tridium. By having the Sedona Framework technology accessible to the public it can be built upon and improved through the efforts of the Sedona community. Businesses have prospered through the use of open technologies such as the Internet and BACnet and the public has benefited as well. The open controller concept with accessible programming tools is a worthy concept and can be achieved through by a committed Sedona community.

Contemporary Control Systems, Inc
2431 Curtiss Street
Downers Grove, IL 60515
USA
+ 1 630 963 7070
info@ccontrols.com
www.ccontrols.com

Contemporary Controls Ltd
14 Bow Court
Fletchworth Gate
Coventry CV5 6SP
United Kingdom
+ 44 (0) 24 7641 3786
info@ccontrols.co.uk
www.ccontrols.co.eu

Contemporary Controls GmbH
Fuggerstraße 1 B
04158 Leipzig
Germany
+ 49 (0) 341 520359 0
info@ccontrols.de
www.ccontrols.eu

Contemporary Controls (Suzhou) Co. Ltd
11 Huoju Road
Science & Technology Park
New District, Suzhou
PR China 215009
+ 86 512 68095866
info@ccontrols.com.cn
www.ccontrols.com.cn