

ALC

- Automated Logic's WebCTRL[®] is a building automation system that offers an intuitive user interface and powerful control features. Your building can be accessed from anywhere in the world using your favorite browser, eliminating the need for special software on the workstation or tablet.
- It uses a SOAP-like API with a username and password. SOAP is essentially XML in and XML out.

Licensing

- Each ALC connector extension is licensed to a single SkySpark Node Id
- Each ALC connector extension is licensed to a set maximum number of points
- If either of these states goes into fault, the issue must be corrected and SkySpark must be restarted

Setting Up the WebCTRL Service I

- 1) Someone with Admin access needs to create a new Operator with the SOAP Privilege Set (System)
- 2) The Accept button will be on the top left. After clicking this, the user will be live.
- 3) Note that HTTP is not enabled by default (HTTPS is). You can find this setting in SiteBuilder || Configure || Preferences || allow SOAP over HTTP

Setting Up the WebCTRL Service II

WebCTRP	Properties					*	€ 💧 🖉 🖶
Checker C	Operators						
	Edits to this page take effect who	en operator logs in again.					^
	[▲] Operators V	Last Login					
My Settings		Today	Login				
System Settings			Name: Rhysnark				
Operators			Name: akyspank				
Privilege Sets			Login Name: skysparkuser				
Operator Groups			Last Login: Mar 18, 2024 2:36:08 PM				
Categories			Change password				
ACxelerate		6 Days	Force user to change password at login				
Scheduled Reports		Yesterday	Exempt From Password Policy				
Semantics		Today					
Alarm Templates			Use system's logon time of 8:00 (nrcmm)				
Connections			O Use different logoff time of XX:XX (hh:mm). (max	amum 500 hours)			
Modern Contiguration		Yesterday	Do not use automatic logoff for this operator				
License Administration			Starting Location				
Update		Yesterday					
Hierarchical Servers		5 Days					
Client Installs			Privilege				
			Show current privileges only				
		4 Days	System-wide Privilege Sets	Groups	Effective System-wide Privileges		
		Today	C Admin	2 Everybody	Access Brivilanes		
		43 Days	[InterOn Level 1 (Sustem)	Administrators	Z Access Groups		
	Skyspark	Yesterday	InterOp Level 10 (Sustem)		Access Confinitients		
	Skyspark - Zivatech	19 Days	InterOp Level 5 (System)		(E) receipt on any many		
			InterOp Level 8 (System)		Runctional Privilages		
		Testerday	PP Operators (System)	V Scap Group	Maintain System Parameters		
			PP Operators Advanced (System)	view Only	Maintain Schedule Group Members		
	Add Des	ete Report	The second second diversion in the second se		Maintain Categories		
			✓ Soap Privilege Set (Syst(*t))		Maintain Alarm Templates		
			From		Acknowledge Non-Critical Alarms		
			Standard (System) Everybody,		Acknowledge Critical Alarms		
			view Only		Force Normal Non-Critical Alarms		
			View Only (System)		Force Normal Critical Alarms		
					Delete Non-Critical Alarms		
					Delete Critical Alarms		
					Execute Audit Log Report		
					Download Controllers		
					System Shutdown		
					Engineer System		
					Access Commissioning Tools		
					Maintain Graphs and Reports		
					Maintain Connections		
					Remote File Management		
					Remote Data Access - SOAP		
					Do not audit changes made using SOAP (Web services)		
					Manual Commands/Console Operations		
					Manual Commands/File IO		
					Manual Commands/Adv Network		
					Manual Commands/Unrestricted		
					Change My Settings		
	w						Ŧ

ALC Connectors in SkySpark

- Creating an ALC connector:
 - dis: Any Readable Name
 - uri: `https://host` ie `https://alc.net/`
 - alcUsername: This will be provided by your ALC Team
 - alcPassword: This will be provided by your ALC Team
 - actorTimeout: How long a busy connector will wait to respond to a new message
 - The alcPollFreq tag determines the minimum amount of time before the cache can be refreshed and how often the curVal values will be updated (which is from where both polling with intervals and COV come). The default value of 1min is good for detecting COV changes. The curVals do not matter for trend log synching.
 - Of course, these records will have the alcConn and conn tags
- These can be created in the connector app

ALC Connector

			Edit		
	Markers	alcConn conn 🔹]		
1.2					
✓	dis	Alc Conn			
•	uri	https://host/			
•	alcPassword	•••••			
•	alcPollFreq	1	min Select 🔻		
V	alcUsername				
	actorTimeout	0	sec 🌲		
	connLinger	0	sec 🌲		
	connOpenRetryFreq	0	sec 🌲		
	connPingFreq	0	sec 🌲		
	connTuningRef	null	Sele	ct Id	
4	Add Tag			Ok Cancel	

ALC Connector Set Up

• You must set up the API on the ALC side, then SkySpark will be able to talk to it. As long as the uri, username, and password are correct, it should work.

Polling of curVal

- alcPollFreq (duration)
 - How often you want SkySpark to poll the connector for COV
 - The default is 1min
 - It is a tuning parameter
 - Note that this determines how often the curVal in SkySpark changes. You would still need to set how often the data is collected with the hisCollect tags.

ALC Functions Part I

- alcSyncHis sync history from a trend log
 - readAll(alcHis).alcSyncHis(null)
- alcSyncCur force one or more points to sync
 - readAll(alcCur).alcSyncCur
- alcPing send ping to connector to check connection
 - read(alcConn).alcPing
- alcHisRead attempt to do a raw read from ALC without synching (will tell you if point is trended)
 - read(alcConn).alcHisRead(read(alcHis))
 read(alcConn).alcHisRead("trees/geographic/pointAddr", today)
- alcVals return a list of addresses and curVals (good for testing if an address exists)
 - Use the debug marker tag on the connector to return the raw XML
 - read(alcConn).alcVals(readAll(alcCur))
 read(alcConn).alcVals(null, ["pointAddr1", "pointAddr2"])
- alcLearnTool this gets display names, but not actual point addresses it's an ALC point list report
 - read(alcConn).alcLearnTool("trees/geographic")

ALC Functions Part 2

- alcWsdl A WSDL is the instructions for a SOAP command. Defaults to Report. Can also use Eval, Trend, or System.
 - read(alcConn).alcWsdl("Eval")
- alcReport Run any of the following reports. It takes a report and a location.
 - Use the debug marker tag on the connector to return the result in html

~schedule-instance ~effective-schedule ~point-list-report ~locked-value ~network-io ~test-and-balance ~equipment-checkout ~audit-log ~alarms ~alarm-source ~network-status ~module-version ~security-assignment ~alarm-messages ~alarm-actions ~trend-usage ~parameter-mismatch

- readAll(alcConn).alcReport("~point-list-report", "trees/geographic")
- alcChildren used for the builder learn, but get the children nodes for any node
 - Use the debug marker tag on the connector to return the raw XML
 - readAll(alcConn).alcChildren("trees/geographic")

ALC View Functions

- alcDupFinder find points with the same addresses (default true will add dupPoint tag to point)
 - alcDupFinder(true)
- alcMakePointsToBuckets assigns tuning records to ALC curVal points (use alcTuningBuckets view)
 - Number of points per connTuningRec
 - alcMakePointsToBuckets(100)
- alcMakeTuningBuckets makes ALC Tuning records (use alcTuningBuckets view)
 - Number of points per connTuningRec, starting pollFreq, max pollFreq
 - alcMakeTuningBuckets(100, 5min, 9min)
- alcViewStatus see ALC points
 - alcViewStatus()
- alcViewSummary see status of ALC points
 - alcViewSummary()
- alcViewTuning display ALC tuning records
 - alcViewTuning()

Getting Data for ALC Points *Manually or Automatically*

- Collecting current values (from curVal):
 - Each point must have these tags:
 - alcConnRef (ref) which is a ref that points to the ALC Connector
 - alcCur (str) which points to the ALC point object
 - his (marker)
 - cur (marker)
 - hisCollectCov (marker or number) for cov collection OR
 - hisCollectInterval (duration) for polling
- Getting history data from Trend Logs:
 - Each point must have these tags:
 - alcConnRef (ref) which is a ref that points to the ALC Connector
 - alcHis (str) which points to the ALC Trend Log object
 - his (marker)
 - Run the alcSyncHis(timeRange) function
 - generally as a task
 - readAll(alcHis).alcSyncHis(null)

Note: we recommend also using a hisCollectInterval of 24hr when using a hisCollectCov to catch faulty sensors.

Writing to ALC Points from SkySpark Manually or with Automated Tasks

- Writing/Sending current values to a ALC point:
 - Each point must have these tags:
 - alcConnRef (ref) which is a ref that points to the Hubitat Connector
 - alcWrite (string) which points to the Hubitat point object
 - writable (marker)
 - Note: The pointWrite() command is the only writing command that is truly supported. The pointAuto() function is only used to clear the writeVal on SkySpark's end. It does not do anything on the device's end.

Basic Troubleshooting

- Note that it is normal for a connection to close when it has no points in a watch for a given period of time
- For larger systems, you want to set the poll frequency to a higher value to avoid network congestion
- Make sure time zones, kinds, and units are the same in SkySpark and ALC
- ALC seems very sensitive towards buckets. The absolute limit in testing appeared to be around 450 points / bucket.
- Note: If a point kind is Bool, 0 and 1 will automatically be converted to false and true for curVals and trends in SkySpark. This conversion is not done on SkySpark writing back to ALC.

Getting Points with GUI

• We recommend using the Builder App's drag and drop capability to select which points are of interest as there are often many superfluous points

A <mark>▼ Site</mark> Rec Tag View		≔ ★ ⊥ 🗉
New Site New Child 🖸 Open In 💌		Bind Info
▼ 🛄 Wheat Ridge	SequipRef Back Kitchen Light switch	Unbound 🔺
▶ 🖵 1st Floor	○ \$equipRef Energy Meter 2 cost	heat Ridge ElecMeter 2 Cost 2
► 🛄 2nd Floor		at Ridge ElecMeter 2 Current 2
► □ 3rd Floor ► □ Attic		dge ElecMeter 2 Consumpti
► Get Content		Ridge ElecMeter 2 Energy
Generation SequepRef Energy Meter 2 cost		t Ridge ElecMeter 2 Demand 2
► □ Garage \$equipRef Energy Meter 2 current		at Ridge ElecMeter 2 Voltage 2
 Holders \$equipRef Energy Meter 2 energy Main ElecMeter 	SequipRef Front Kitchen Light Wheat	Ridge 1st Floor Front Kitche
▶ □ Outside	○ \$equipRef Front Kitchen Light level	Unbound
► <u>□</u> Weather	○ \$equipRef Front Kitchen Light switch	Unbound
SequipRef Energy Meter Avenue	System current When	eat Ridge Garage Entertain
	○ \$equipRef Garage Entertainment System of	currentH Unbound
	○ \$equipRef Garage Entertainment System of	currentL Unbound
	System energy When the second	eat Ridge Garage Entertain
	System energyDuration Wh	eat Ridge Garage Entertain
	SequipRef Garage Entertainment System	evel Unbound
	System power WH	eat Ridge Garage Entertain