Version 15 Revision 00
November 2014
SCPT Master List

LONMARK®
SCPT Master List

Contents

SCPTactFbDly

Overview:

Actual position feedback delay. The period for updating the feedback output when the actuator position does not match the requested position

This configuration property sets the period for updating the actual actuator position feedback output from a functional block when the actuator position does not match the requested position as specified by the primary input network variable of the functional block.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*1*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTactFbDly*
Maximum:	*0 0:0:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTactFbDly: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTactFbDly#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopActuator SFPTopenLoopActuator*

SCPTactuatorCharacteristic

Overview:

Actuator Characteristic. This configuration property can be used to provide the characteristic of the actuator

This configuration property can be used to provide the characteristic of the actuator.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*284*
Obsolete:	*no*

Size:	*1*
Programmatic Name:	*SCPTactuatorCharacteristic*
Default:	*DCM_SPEED_CONST*
Neuron C Type:	*SNVT_dev_c_mode*
Formats:	*SCPTactuatorCharacteristic: text("%m")*
Used by:	*SFPThvacValvePositioner*

SCPTactuatorType

Overview:

Actuator label. The identification of the exact actuator type or label

This configuration property is used to identify the exact actuator type or label.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*41*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTactuatorType*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTactuatorType: text("%s", ascii)*
Used by:	*SFPTdamperActuator SFPTfireSmokeDamperActuator SFPThvacValvePositioner*

SCPTahamApplianceModel

Overview:

AHAM Appliance Model. Appliance Model code as defined by the Association of Home Appliance Manufacturers

Appliance Model code as defined by the US-based Association of Home Appliance Manufacturers.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*304*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTahamApplianceModel*
Default:	*AHAM_CLOTHES_WASHER*
Neuron C Type:	*aham_appl_t*
Formats:	*SCPTahamApplianceModel: text("%m")*
Used by:	*SFPTclothesWasherDomestic*

SCPTairTemp1Alrm

Overview:

Air temperature 1 percent alarm. The weighting of the air temp 1 sensor when calculating the air temp alarm

This configuration property sets the weighting of an nviAirTemp1 sensor when calculating the nviAlarmAirTemp alarm. The nviAirTemp2 portion is automatically calculated.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*132*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTairTemp1Alrm*
Minimum:	*0,000*
Maximum:	*100,000*
Default:	*100,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTairTemp1Alrm: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTairTemp1Day

Overview:

Air temperature 1 percent day. The air temperature weighting used during day control

This configuration property indicates the air temperature weighting to be used during Day control. The nviAirTemp2 portion is automatically calculated.

EXAMPLES

nviAirTemp1 measures -25°C and nviAirTemp2 measures -20°C. Required as a regulation parameter is a weighted average consisting of 60% nviAirTemp1 and the rest from nviAirTemp2 (only the nviAirTemp1 share is programmed). This gives the following regulating value:

(0.6 x -25 degrees C) + (0.4 x -20 degrees C) = -23 degrees C

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*126*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTairTemp1Day*
Minimum:	*0,000*
Maximum:	*100,000*
Default:	*100,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTairTemp1Day: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTairTemp1Night

Overview:

Air temperature 1 percent night. The air temperature weighting used during night control

This configuration property indicates the air temperature weighting to be used during Day control. The nviAirTemp2 portion is automatically calculated.

EXAMPLES

nviAirTemp1 measures -25 degrees C and nviAirTemp2 measures -20 degrees C. Required as a regulation parameter is a weighted average consisting of 60% nviAirTemp1 and the rest from nviAirTemp2 (only the nviAirTemp1 share is programmed). This gives the following regulating value

(0.6 x -25 degrees C) + (0.4 x -20 degrees C) = -23 degrees C

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*131*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTairTemp1Night*
Minimum:	*0,000*
Maximum:	*100,000*
Default:	*100,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTairTemp1Night: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTalrmClrT1

Overview:

Alarm clear time 1. The time period that an alarm 1 condition must not exist before it is regarded as a valid cleared alarm

This configuration property is used to determine the time period that an alarm 1 condition must not exist before it is regarded as a valid cleared alarm.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*2*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTalrmClrT1*
Maximum:	*0 0:0:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTalrmClrT1: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTalrmClrT1#LO: text(("%d ", day), time(hour, minute, second,*

	millisecond))
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTalrmClrT2

Overview:

Alarm clear time 2. The time period that an alarm 2 condition must not exist before it is regarded as a valid cleared alarm

This configuration property is used to determine the time period that an alarm 2 condition must not exist before it is regarded as a valid cleared alarm.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*3*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTalrmClrT2*
Maximum:	*0 0:0:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTalrmClrT2: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTalrmClrT2#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTalrmIhbT

Overview:

Alarm output inhibit time. The time period for which alarms are inhibited after an object is enabled or the node is reset

This configuration property is used to determine the time period for which alarms are inhibited after a functional block is enabled or the device is reset or is put on-line.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*4*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTalrmIhbT*
Maximum:	*0 0:0:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTalrmIhbT: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTalrmIhbT#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTalrmSetT1

Overview:

Alarm set time 1. The time period that an alarm 1 condition must exist before it is regarded as a valid alarm

This configuration property is used to determine the time period that an alarm 1 condition must exist before it is regarded as a valid alarm.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*5*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTalrmSetT1*
Maximum:	*0 0:0:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTalrmSetT1: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)*

	SCPTalrmSetT1#LO: text(("%d ", day), time(hour, minute, second, millisecond))
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTalrmSetT2

Overview:

Alarm set time 2. The time period that an alarm 2 condition must exist before it is regarded as a valid alarm

This configuration property is used to determine the time period that an alarm 2 condition must exist before it is regarded as a valid alarm.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*6*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTalrmSetT2*
Maximum:	*0 0:0:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTalrmSetT2: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)*
	SCPTalrmSetT2#LO: text(("%d ", day), time(hour, minute, second, millisecond))
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTareaDuctHeat

Overview:

Heating duct area. Nominal cross-sectional airflow area of the hot or ventilation deck of a dual duct VAV terminal unit

Nominal cross-sectional airflow area of the hot or ventilation deck of a dual duct VAV

terminal unit.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*266*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTareaDductHeat*
Default:	*13,1070*
Neuron C Type:	*SNVT_area*
Formats:	*SCPTareaDductHeat#SI: text("%f")* *SCPTareaDductHeat#US: text("%f", *10.7639+0(0:884))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTaudOutput

Overview:

Audible sound output intensity. Audible sound output intensity specification of the device at 1 meter distant

This configuration property allows the sound output specification in dBA to be read from a device. This information is defined at manufacture time. The audible sound output is defined as the sound output at 1-meter distance from the device, as specified by the manufacturer.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*144*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTaudOutput*
Default:	*0,00*
Neuron C Type:	*SNVT_sound_db*
Formats:	*SCPTaudOutput: text("%f")*

Used by:	*SFPTaudibleFireIndicator*

SCPTaudibleLevel

Overview:

Audible level. The audible level output of the device

The configuration property of type SCPTaudibleLevel may be used to set the audible output of any product.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*228*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTaudibleLevel*
Default:	*0,0 0*
Neuron C Type:	*SNVT_switch*
Formats:	*SCPTaudibleLevel: text("%f %d", value, state)*
Used by:	*SFPTelevatorVoiceAnnouncer*

SCPTautoAnswer

Overview:

Auto answer (boolean) . Enable the automatic call answer function of a device

This configuration property is used to enable or disable the automatic call answer function of a data modem. Auto-answer is enabled if the value is True, otherwise it is disabled.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*177*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTautoAnswer*
Default:	*BOOL_FALSE*

Neuron C Type:	*boolean_t*
Formats:	*SCPTautoAnswer: text("%m")*
Used by:	*SFPTmodemController*

SCPTbaseValue

Overview:

Base value. The base value (where to begin counting)

This configuration property defines the base value for a Utility Data Logger Register functional block. If the value is changed, the register begins to count starting from the given value.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*164*
Obsolete:	*no*
Size:	*6*
Programmatic Name:	*SCPTbaseValue*
Default:	*0 RVU_NONE 0*
Neuron C Type:	*SNVT_reg_val*
Formats:	*SCPTbaseValue: text("%d %m %d", raw, unit, nr_decimals)*
Used by:	*SFPTutilityDataLoggerRegister SFPTutilityMeter*

SCPTbkupSchedule

Overview:

Backup Schedule.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*344*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTbkupSchedule*
Default:	*0 0 0 0*

Neuron C Type:

Structure	
hour_on	Hour ON (Hours) . Time when the luminaire will be switched ON in case of communication failure
	unsigned short Minimum: *0* Maximum: *23* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
minute_on	Minute ON (Minutes) . Time when the luminaire will be switched ON in case of communication failure
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
hour_off	Hour OFF (Hours) . Time when the luminaire will be switched OFF in case of communication failure
	unsigned short Minimum: *0* Maximum: *23* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
minute_off	Minute OFF (Minute) . Time when the luminaire will be switched OFF in case of communication failure
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTbkupSchedule: text("%d %d %d %d", hour_on, minute_on,

	hour_off, minute_off)
Used by:	*SFPToutdoorLuminairController SFPTsmartLuminaireController*

SCPTblockProtectionTime

Overview:

Minimum time for movement. The minimum time in hours for movement to prevent blocking

This configuration property defines the minimum time in hours for movement. This periodic cycling prevents blocking.

Note: Setting disables the protect function.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*251*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTblockProtectionTime*
Default:	*0*
Neuron C Type:	*SNVT_time_hour*
Formats:	*SCPTblockProtectionTime: text("%d")*
Used by:	*SFPThvacValvePositioner*

SCPTbrightness

Overview:

Brightness output. The brightness output of a display device

The configuration property of type SCPT_brightness may be used to set the brightness of a visual display.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*230*
Obsolete:	*no*
Size:	*2*

Programmatic Name:	*SCPTbrightness*
Default:	*0,0 0*
Neuron C Type:	*SNVT_switch*
Formats:	*SCPTbrightness: text("%f %d", value, state)*
Used by:	*SFPTelevatorDirectionLantern SFPTelevatorHallLantern SFPTelevatorPositionIndicator*

SCPTbuildingStaticPressureSetpoint

Overview:

Building static pressure. Setpoint for the default static pressure for the building

This configuration property defines the default building static pressure setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*193*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTbuildingStaticPressureSetpoint*
Default:	*25*
Neuron C Type:	*SNVT_press_p*
Formats:	*SCPTbuildingStaticPressureSetpoint#SI: text("%f")* *SCPTbuildingStaticPressureSetpoint#US: text("%f", *0.0040217+0(0:954))* *SCPTbuildingStaticPressureSetpoint#US_psi: text("%f", *1.4504e-4+0(0:875))*
Used by:	*SFPTdischargeAirController*

SCPTbuttonColor

Overview:

Button color.. Button color configuration for on and off states of a button. May be used to create an array that is used with a SCPTbuttonAction array to specify keypad button behavior.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *312*

Obsolete: *no*

Size: *6*

Programmatic Name: *SCPTbuttonColor*

Default: *0 0 0 0 0 0*

Neuron C Type:

Structure	
on_color	On color.. RGB color for the on state.

Structure	
red	Red.. Red level component of the on state color
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
green	Green.. Green level component of the on state color
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
blue	Blue.. Blue level component of the on state color
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

off_color	Off color.. RGB color for the off state
	Structure
	red: Red.. Red level component of the off state color
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	green: Green.. Green level component of the off state color
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	blue: Blue.. Blue level component of the off state color
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTbuttonColor: text("%d %d %d %d %d %d", on_color.red, on_color.green, on_color.blue, off_color.red, off_color.green, off_color.blue)

Used by:

SFPTisiKeypad

SCPTbuttonHoldAction

Overview:

Button held action.. Button action definition used to create a button held action array, with an entry per button.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *314*

Obsolete: *no*

Size: *2*

Programmatic Name: *SCPTbuttonHoldAction*

Default: *BTA_TOGGLE_STATE 0,0*

Neuron C Type:

Structure	
action	Button action.. Button action for the associated button
	button_action_t
setting	Value.. Value for button actions that require a numeric value.
	Union
	scene_number: Scene number.. Specified scene number for the toggle and set scene actions.
	unsigned short Minimum: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	group_number: Group number. Specified group number for the toggle, enable, and disable group actions.

		unsigned short Minimum: *0* Maximum: *64* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	value	Setting.. Setting level. Specifies an absolute level for the Set Level action. Specifies a relative level for the Increase, Decrease, Move Open, Move Closed, Rotate Open, and Rotate Closed actions.
		unsigned short Minimum: *0* Maximum: *200* Invalid: *255* Scaling (A,B,C): *5, -1, 0* Scaled value: *5 *10 $^{-1}$ *(Raw+0)* Resolution: *0.5*
	angle	(degrees) .
		unsigned short Minimum: *0* Maximum: *180* Invalid: *255* Scaling (A,B,C): *2, 0, 0* Scaled value: *2 *10 ^{0} *(Raw+0)* Resolution: *2*

Formats:

SCPTbuttonHoldAction: text(("%m ", action),((action == 1) ? ("%d ", setting.scene_number) :((action == 3) ? ("%d ", setting.scene_number) :((action == 15) ? ("d ", setting.angle) :((action == 16) ? ("%d ", setting.angle) :((action == 17) ? ("%d ", setting.angle) :((action == 20) ? ("%d ", setting.scene_number) :((action == 34) ? ("%d ", setting.group_number) :((action == 35) ? ("%d ", setting.group_number) :((action == 36) ? ("%d ", setting.group_number) :((action == 255) ? ("UNKNOWN ") :("%f ", setting.value))))))))))))

SCPTbuttonPressAction

Overview:

Button pressed action.. Button action definition used to create a button pressed action array, with an entry per button. This SCPT defines the minimum entries required by the ISI profiles.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *311*

Obsolete: *no*

Size: *2*

Programmatic Name: *SCPTbuttonPressAction*

Default: *BTA_TOGGLE_STATE 0,0*

Neuron C Type:

Structure	
action	Button action.. Button action for the associated button
	button_action_t
setting	Value.. Value for button actions that require a numeric value.
	Union
	scene_number: Scene number.. Specified scene number for the toggle and set scene actions.
	unsigned short Minimum: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	group_number: Group number. Specified group number for the toggle, enable, and disable group actions.

		unsigned short Minimum: *0* Maximum: *64* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	value	Setting.. Setting level. Specifies an absolute level for the Set Level action. Specifies a relative level for the Increase, Decrease, Move Open, Move Closed, Rotate Open, and Rotate Closed actions.
		unsigned short Minimum: *0* Maximum: *200* Invalid: *255* Scaling (A,B,C): *5, -1, 0* Scaled value: *5 *10 $^{-1}$ *(Raw+0)* Resolution: *0.5*
	angle	(degrees) .
		signed short Minimum: *-90* Maximum: *90* Invalid: *127* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTbuttonPressAction: text(("%m ", action),((action == 1) ? ("%d ", setting.scene_number) :((action == 3) ? ("%d ", setting.scene_number) :((action == 15) ? ("d ", setting.angle) :((action == 16) ? ("%d ", setting.angle) :((action == 17) ? ("%d ", setting.angle) :((action == 20) ? ("%d ", setting.scene_number) :((action == 34) ? ("%d ", setting.group_number) :((action == 35) ? ("%d ", setting.group_number) :((action == 36) ? ("%d ", setting.group_number) :((action == 255) ? ("UNKNOWN ") :("%f ", setting.value))))))))))))

Used by: *SFPTisiKeypad*

SCPTbuttonRepeatInterval

Overview:

Button repeat interval (milliseconds) . Time between updates when a button is held down. The updates themselves may be throttled by the application or a SCPTminSendTime CP. Used to create an array used with a SCPTbuttonAction CP array.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*313*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTbuttonRepeatInterval*
Default:	*0*
Neuron C Type:	unsigned long
	Minimum: *0*
	Maximum: *65534*
	Invalid: *65535*
	Scaling (A,B,C): *1, 0, 0*
	Scaled value: $1 * 10^{0} * (Raw+0)$
	Resolution: *1*
Formats:	*SCPTbuttonRepeatInterval: text("%d")*
Used by:	*SFPTisiKeypad*

SCPTbypassTime

Overview:

Bypass time. The maximum amount of time that the controller can be in the bypass (occupancy) mode following the last bypass request. Zero disables the timer.

This configuration property defines the maximum amount of time that a controller can be in the bypass (occupancy) mode following a single bypass request from either a local (hardwired) bypass switch or from another device over the network. Additional bypass requests can restart the timer. Setting this configuration property to zero disables the bypass function and no bypass takes place.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*34*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTbypassTime*
Default:	*0*
Neuron C Type:	*SNVT_time_min*
Formats:	*SCPTbypassTime: text("%f")*
Used by:	*SFPTchilledCeilingController SFPTdischargeAirController SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTsunblindController SFPTunitHeater SFPTunitVentilatorController SFPTwallUnit*

SCPTclOffDelay

Overview:

Controller off delay. The delay after which the controller output is switched off

This configuration property is used to determine the delay after which the controller output is switched off. The controller is only switched off after this delay if the actuator output is 0% and the ambient level exceeds the setpoint value plus the relative hysteresis level.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*85*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTclOffDelay*
Default:	*300,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTclOffDelay: text("%f")*
Used by:	*SFPTconstantLightController*

SCPTclOnDelay

Overview:

Controller on delay. The delay after which the controller output is switched on

This configuration property is used to determine the delay after which the controller output is switched on. The controller is only switched on after this delay if the output was switched off due to a too high ambient level and the ambient level is now lower than the setpoint value minus the relative hysteresis.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*86*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTclOnDelay*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTclOnDelay: text("%f")*
Used by:	*SFPTconstantLightController*

SCPTclockCalibration

Overview:

Clock Calibration (milliseconds per hour) . Corrects clock-cycle variations

Corrects clock-cycle variations.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*300*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTclockCalibration*
Default:	*0*

Neuron C Type:	signed long	
	Minimum:	*-32768*
	Maximum:	*32767*
	Scaling (A,B,C):	*1, 0, 0*
	Scaled value:	*1 *10 ^{0} *(Raw+0)*
	Resolution:	*1*
Formats:	*SCPTclockCalibration: text("%d")*	

SCPTcombFlowCharacteristic

Overview:

Combination-Flow Characteristic. This configuration property can be used to provide the desired system control flow characteristic

This configuration property can be used to provide the desired system control flow characteristic.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*287*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTcombFlowCharacteristic*
Default:	*DCM_SPEED_CONST*
Neuron C Type:	*SNVT_dev_c_mode*
Formats:	*SCPTcombFlowCharacteristic: text("%m")*
Used by:	*SFPThvacValvePositioner*

SCPTcontrolCfg

Overview:

LC Control Configuration.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*382*
Obsolete:	*no*

Size:	*16*
Programmatic Name:	*SCPTcontrolCfg*
Default:	*0,0 0,0 0,0 0 0 0,0 0,00 0,00 0 0 OLC_DEFAULT*
Neuron C Type:	*SNVT_control_cfg*
Formats:	*SCPTcontrolCfg: text("%f %f %f %d %d %f %f %f %d %d %m", defaultLev, rampTm, supplyVoltage, warmupTm, coolDownTm, maxRcvTm, minControlV, maxControlV, clrTime, occupHoldTm, lampType)*
Used by:	*SFPTsmartLuminaireController*

SCPTcontrolPriority

Overview:

Control priority (priority value) . Priority of a control input or output, lower values mean higher priority

This configuration property assigns a group member priority to a controlling device. Low priority values specify low priority, and high priority values specify high priority. In normal control mode the range is 1 to 50, and in alarm control mode the range is 51 to 200. The value zero is used to release control. The range 201 to 255 is reserved.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*171*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTcontrolPriority*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *200* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTcontrolPriority: text("%d")*

SCPTcontrolSignal

Overview:

Control signal (X1, Y1, X2, Y2) . Start and end points (X,Y) for a transition

This configuration property sets control signal preparation for sequencing.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*245*
Obsolete:	*no*
Size:	*8*
Programmatic Name:	*SCPTcontrolSignal*
Minimum:	*0,000 0,000 0,000 0,000*
Default:	*0,000 0,000 0,000 0,000*
Neuron C Type:	

Structure	
x1Value	X1.
	SNVT_lev_percent
y1Value	Y1.
	SNVT_lev_percent
x2Value	X2.
	SNVT_lev_percent
y2Value	Y2.
	SNVT_lev_percent

Formats: *SCPTcontrolSignal: text("%f %f %f %f", x1Value, y1Value, x2Value, y2Value)*

Used by: *SFPThvacValvePositioner*

SCPTcontrolTemperatureWeighting

Overview:

Temperature weighting. Weighting between two temperatures in percent

This configuration property is used as the weighting between discharge temperature and return

temperature for calculation of a confined-space temperature, according to the following relationship:

ConfinedSpaceTemp = (ContWeight/100)*ReturnTemp + (1-ContWeight/100)*DischargeTemp.

The value for this configuration property is typically selected such that the resulting value represents the average temperature in the Confined Space, and is thus specific to the actual design of the Confined Space.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*215*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTcontrolTemperatureWeighting*
Minimum:	*0,000*
Default:	*100,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTcontrolTemperatureWeighting: text("%f")*

SCPTcoolLowerSP

Overview:

Cooling setpoint lower limit. Limits the lower extent of the permitted range for the cooling setpoint

This configuration property defines the lower limit for the cooling setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*76*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTcoolLowerSP*
Minimum:	*10,00*
Maximum:	*35,00*
Default:	*10,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTcoolLowerSP#SI: text("%f", *1+0(0:854))*

	*SCPTcoolLowerSP#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTthermostat*

SCPTcoolSetpt

Overview:

Cooling setpoint. The default setpoint for the leaving chilled water temperature in cooling mode when the default behavior selector is set to zero

This input configuration network variable establishes the default setpoint for the leaving chilled water temperature, when the SCPTdefltBehave configuration property is set equal to zero (0) or is not defined. When the SCPTdefltBehave configuration property is set equal to one (1), the manufacturer-specified values are used instead.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*75*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTcoolSetpt*
Minimum:	*-40,00*
Maximum:	*48,80*
Default:	*7,20*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTcoolSetpt#SI: text("%f", *1+0(0:854))*
	*SCPTcoolSetpt#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTchiller*

SCPTcoolUpperSP

Overview:

Cooling setpoint upper limit. Limits the upper extent of the permitted range for the cooling setpoint

This configuration property defines the upper limit for the cooling setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*77*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTcoolUpperSP*
Minimum:	*10,00*
Maximum:	*35,00*
Default:	*10,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTcoolUpperSP#SI: text("%f", *1+0(0:854))* *SCPTcoolUpperSP#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTthermostat*

SCPTcoolingLockout

Overview:

Cooling lockout. Setpoint for the outdoor air temperature at which cooling will be disabled

This configuration property defines the outdoor air temperature cooling lockout setpoint. When the outdoor air temperature is below this value, mechanical cooling will be disabled.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*209*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTcoolingLockout*
Default:	*10,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTcoolingLockout#SI: text("%f", *1+0(0:854))* *SCPTcoolingLockout#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTcoolingResetEnable

Overview:

Cooling reset enable (boolean) . The cooling reset control is enabled

This configuration property is used to enable or disable the discharge air temperature cooling reset control. A True value enables the reset control.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*211*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTcoolingResetEnable*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*
Formats:	*SCPTcoolingResetEnable: text("%m")*
Used by:	*SFPTdischargeAirController*

SCPTcurrentSenseEnable

Overview:

Current sense enable. Enables current sensing for manual load control

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*359*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTcurrentSenseEnable*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*
Formats:	*SCPTcurrentSenseEnable: text("%m")*

SCPTcutOutValue

Overview:

Cut-out value. The cut-out limit

Indicates the cut out limit to be used for control purposes. If day night control is selected this value is used during day control.

This configuration property defines the upper limit to which the cooling setpoint can be set.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*125*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTcutOutValue*
Default:	*10,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTcutOutValue#SI: text("%f", *1+0(0:854))* *SCPTcutOutValue#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTdayDateIndex

Overview:

Day date index (First day and month, second day and month, time-event index) . One or two dates for matching with a start index to the time-event array

This configuration property provides two dates together with a starting index for a Time-Event list. The indicated Time-Event list will be used when the current date falls between the indicated dates.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*103*
Obsolete:	*no*
Size:	*6*
Programmatic Name:	*SCPTdayDateIndex*
Default:	*0/0,0/0,0*

Neuron C Type:

Structure	
day_1	First day (days) . Day of month
	unsigned short Minimum: *1* Maximum: *31* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
month_1	First month (months) . Month of year
	unsigned short Minimum: *1* Maximum: *12* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
day_2	Second day (days) . Day of month, zero for no date entry
	unsigned short Minimum: *0* Maximum: *31* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

month_2	Second month (months) . Month of year, zero for no date entry
	unsigned short Minimum: *0* Maximum: *12* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
event_mode_index	Event index (index) . Time-event array index
	unsigned long Minimum: *0* Maximum: *65535* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SCPTdayDateIndex: text("%d/%d,%d/%d,%d", day_1, month_1, day_2, month_2, event_mode_index)*

Used by: *SFPTrealTimeBasedScheduler*

SCPTdayNightCntrl

Overview:

Day/night control. Configures the day/night function

This configuration property configures the day/night function.

Note: The bit2 and bit3 fields are reserved and must be set to zero (0). The bit3 through bit15 fields can be used for Manufacturer-defined items. All unused bits should be set to zero (0).

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *121*

Obsolete: *no*

Size:	*2*
Programmatic Name:	*SCPTdayNightCntrl*
Default:	*0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*
Neuron C Type:	*SNVT_state*
Formats:	*SCPTdayNightCntrl: text("%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d", bit0, bit1, bit2, bit3, bit4, bit5, bit6, bit7, bit8, bit9, bit10, bit11, bit12, bit13, bit14, bit15)* *SCPTdayNightCntrl#LO: text("%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d", bit0, bit1, bit2, bit3, bit4, bit5, bit6, bit7, bit8, bit9, bit10, bit11, bit12, bit13, bit14, bit15)*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTdebounce

Overview:

Debounce time. The debouncing time to generate the detection envelope

This configuration property defines the debouncing time for a hardware sensor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*139*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdebounce*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTdebounce: text("%f")*
Used by:	*SFPTisiOccupancySensor SFPToccupancySensor*

SCPTdefInput

Overview:

.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*305*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTdefInput: text("%d")*
Used by:	*SFPTisiLampActuator SFPTisiSunblindActuator*

SCPTdefOutput

Overview:

.

For a sensor functional block, this configuration property determines the position or level that the primary output network variable for the functional block should adopt, when no updates are received from the hardware within the maximum receive time, at power-on or reset, and when an override request is received for the functional block.

For an actuator functional block, this configuration property determines the position or level that the actuator should adopt, when no updates are received by primary input network variable within the maximum receive time, at power-on or reset, and when an override request is received for the functional block.

The override behavior is defined by the SCPTovrBehave and SCPTovrValue configuration properties.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*7*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTdefOutput: text("%d")*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTentryExit SFPThvacValvePositioner SFPTisiKeypad SFPTlampActuator SFPTopenLoopActuator SFPTopenLoopSensor SFPTscheduler*

SCPTdefScale

Overview:

Default speed scale. Default value for the speed scale

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*162*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdefScale*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTdefScale: text("%f")*
Used by:	*SFPTvariableSpeedMotorDrive*

SCPTdefWeekMask

Overview:

Definition week mask. Day type definition for every day of the week

This configuration property sets a default day-type for each day of the week. If there is no schedule set by a SCPTtimeEvent array, this array will be used to determine the daily time event/mode schedule.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*102*
Obsolete:	*no*
Size:	*14*
Programmatic Name:	*SCPTdefWeekMask*
Default:	*0,0,0,0,0,0,0*

Neuron C Type:

Structure	
time_event_index	Start index array (array of 7 index values) .
	unsigned long [7] Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTdefWeekMask: text("%d,%d,%d,%d,%d,%d,%d", time_event_index[0], time_event_index[1], time_event_index[2], time_event_index[3], time_event_index[4], time_event_index[5], time_event_index[6])

SCPTdefWeekMask#LO: text("%d|%d|%d|%d|%d|%d|%d", time_event_index[0], time_event_index[1], time_event_index[2], time_event_index[3], time_event_index[4], time_event_index[5], time_event_index[6])

Used by:

SFPTrealTimeBasedScheduler

SCPTdefaultAutoPanSpeed

Overview:

Default auto-pan speed (% of full level) . The default auto-pan speed for a device

This configuration property sets the default auto-pan speed as a percentage of maximum auto-pan speed.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *176*

Obsolete: *no*

Size: *1*

Programmatic Name: *SCPTdefaultAutoPanSpeed*

Default: *25,2*

Neuron C Type:	unsigned short	
	Minimum:	*0*
	Maximum:	*250*
	Invalid:	*255*
	Scaling (A,B,C):	*4, -1, 0*
	Scaled value:	*4 *10 $^{-1}$ *(Raw+0)*
	Resolution:	*0.40000000596046*
Formats:	*SCPTdefaultAutoPanSpeed: text("%f")*	

SCPTdefaultPanTiltZoomSpeeds

Overview:

Default camera PTZ. The default camera pan, tilt, and zoom speeds

This configuration property is used to configure the default pan, tilt, and zoom speeds for a camera. The speeds are given in percent of maximum speed.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*175*
Obsolete:	*no*
Size:	*6*
Programmatic Name:	*SCPTdefaultPanTiltZoomSpeeds*
Default:	*PAN_STOP 100,0 TILT_STOP 100,0 ZOOM_STOP 100,0*
Neuron C Type:	*SNVT_ptz*
Formats:	*SCPTdefaultPanTiltZoomSpeeds: text("%m %f %m %f %m %f", pan_dir, pan_speed, tilt_dir, tilt_speed, zoom, zoom_speed)*

SCPTdefaultSetting

Overview:

default Setting.

Initially designed for the sunblind's industry, this CP type represents the default values for function, setting, and rotation.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*

Index:	*297*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTdefaultSetting*
Default:	*SET_OFF 0,0 0,00*
Neuron C Type:	*SNVT_setting*
Formats:	*SCPTdefaultSetting: text("%m %f %f", function,setting,rotation)*
Used by:	*SFPTsunblindController*

SCPTdefaultState

Overview:

Default Security State. Provides the default security state upon power cycle or non-detection

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*295*
Obsolete:	*yes*
Size:	*3*
Programmatic Name:	*SCPTdefaultState*
Default:	*SSE_OFF 0*
Neuron C Type:	*SNVT_sec_state*
Formats:	*SCPTdefaultState: text("%m %d", state,identity)*

SCPTdefltBehave

Overview:

Default behavior. Selects which set of values will be used on power-up and communication failure, between the stated default values (0), or manufacturer-specified values (1)

The configuration property determines which set of values will be used on power up and communications failure. The choice is the stated default values or a list of manufacturer specified values.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*71*

Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdefltBehave*
Default:	*0,0 0*
Neuron C Type:	*SNVT_switch*
Formats:	*SCPTdefltBehave: text("%f %d", value, state)*
Used by:	*SFPTchiller SFPThardwiredFireAlarmShutdown* *SFPThardwiredFullVentilation* *SFPThardwiredGasDetectionShutdown* *SFPThardwiredRecirculation* *SFPThardwiredSafetyInstrumentedSystemSIS*

SCPTdefrostCycles

Overview:

Defrost cycles (units) . Number of equally-spaced defrost cycles to perform per day

For internally scheduled defrost initiation, this configuration property provides the number of equally-spaced defrost cycles to perform per day with the first defrost starting at the time defined by a SCPTdefStart value.

Note: The valid range will depend on the maximum defrost period and recovery period. A value of zero will have the effect of disabling internally scheduled defrosts.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*219*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTdefrostCycles*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *255* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTdefrostCycles: text("%d")*

SCPTdefrostDetect

Overview:

Defrost detect temperature differential. Temperature differential of coil refrigerant temperature above discharge air temperature that indicates defrost

This configuration property provides the value for detecting a defrost in a controlled case. If the coil outlet refrigerant temperature rises above the discharge air temperature by this value, then the case is assumed to have gone into a defrost. If the value is set to zero, this feature is disabled. This feature is typically used when hot-gas or electric defrost is controlled externally from the controller, and allows the controller to determine when to switch off the fans.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*225*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdefrostDetect*
Minimum:	*0,00*
Maximum:	*50,00*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_diff_p*
Formats:	*SCPTdefrostDetect#SI: text("%f", *1+0(0:854))*
	*SCPTdefrostDetect#US: text("%f", *1.8+0(0:855))*

SCPTdefrostFanDelay

Overview:

Fan delay after defrost. Delay after refrigeration is resumed after defrost before fan is started

This configuration property provides the delay after refrigeration is resumed after defrost before the fan is started. This is typically used to prevent warm, humid air from being sent to the case after defrost.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*222*
Obsolete:	*no*

Size:	*2*
Programmatic Name:	*SCPTdefrostFanDelay*
Default:	*0*
Neuron C Type:	*SNVT_time_min*
Formats:	*SCPTdefrostFanDelay: text("%d")*

SCPTdefrostHold

Overview:

Defrost hold on sync (boolean) . Enable hold in defrost mode until synchronization signal goes away

This configuration property determines whether a controller should remain in defrost mode until the Defrost Signal network variable input (nviDefSignal) goes away. This can be used to synchronize a collection of controllers coming out of defrost.

A value of BOOL_FALSE specifies that synchronization should be ignored; a value of BOOL_TRUE specifies that the controller should provide synchronization. Any other value, including BOOL_NUL, is interpreted as BOOL_FALSE.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*224*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTdefrostHold*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*
Formats:	*SCPTdefrostHold: text("%m")*

SCPTdefrostInternalSchedule

Overview:

Defrost internal schedule. Enable the internal scheduling of defrost

This configuration property controls internally scheduled defrosts as defined by other configuration properties (SCPTdefStart and SCPTdefNumber). This property does not disable externally scheduled defrosts initiated from the Defrost Signal network variable input (nviDefSignal).

A value of BOOL_TRUE specifies internally scheduled defrosts and BOOL_FALSE specifies that defrosts should not be internally scheduled. Any other value, including BOOL_NUL, is interpreted as BOOL_FALSE.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*217*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTdefrostInternalSchedule*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*
Formats:	*SCPTdefrostInternalSchedule: text("%m")*

SCPTdefrostMode

Overview:

Defrost mode. The type of defrost to perform

This configuration property indicates the type of defrost to perform.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*106*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTdefrostMode*
Default:	*DFM_MODE_FORCED*
Neuron C Type:	*SNVT_defr_mode*
Formats:	*SCPTdefrostMode: text("%m")*
Used by:	*SFPTrefrigDisplayCaseControllerDefrost*

SCPTdefrostRecoveryTime

Overview:

Defrost recovery time. Time allowed after defrost finishes for temperature to be within normal limits

This configuration property sets the time that a case controller will allow after the defrost cycle has finished for the case to return to within its normal operating band. If the case controller does not manage to get back within its normal operating limits within this time, then an over-temperature alarm will be generated. This configuration property is similar to the existing SCPThighLimDefrDly configuration property, except that it has units of minutes rather than seconds.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*223*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdefrostRecoveryTime*
Default:	*0*
Neuron C Type:	*SNVT_time_min*
Formats:	*SCPTdefrostRecoveryTime: text("%d")*

SCPTdefrostStart

Overview:

Defrost start time. Start time for the first daily defrost

This configuration property sets the time at which the first daily defrost will be scheduled for an internally scheduled defrost initiation. Only the hour and minute fields are used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*218*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTdefrostStart*
Minimum:	*0:0*
Maximum:	*23:59*
Default:	*0:0*
Neuron C Type:	*SNVT_time_stamp*
Formats:	*SCPTdefrostStart: text("%d:%d", hour, minute)*

SCPTdefrostStart#LO: text(time(hour, minute))

SCPTdelayTime

Overview:

Delay time, default to scene. The delay time, default to scene

This configuration property sets the delay time for a scene. It is possible to learn scenes with different delay times by changing this configuration property value for each scene before the learn current command is used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*96*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdelayTime*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTdelayTime: text("%f")*
Used by:	*SFPTentryExit SFPTsceneController SFPTscheduler*

SCPTdeltaNight

Overview:

Delta night. The value to be added to the cut-out value to get the cut-out limit during night control

This configuration property sets the value to be added to the SCPTcutOutValue configuration property to get the cut out limit to be used during night control.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*134*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdeltaNight*
Default:	*0,00*

Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTdeltaNight#SI: text("%f", *1+0(0:854))* *SCPTdeltaNight#US: text("%f", *1.8+0(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTdevListDesc

Overview:

Device list entry description.. Human readable description for an entry in the device list

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*322*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTdevListDesc*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTdevListDesc: text("%s", ascii)*
Used by:	*SFPTdeviceMonitor*

SCPTdevListEntry

Overview:

Device list entry.. Device list entry containing the address of the device to be monitored

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*323*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTdevListEntry*
Default:	*0 UNKNOWN*
Neuron C Type:	Union
	address_type

	address_type_t
sn	Device address as subnet/node address.. This structure is filled out in case the device address is given as subnet/node address
	Structure

subnet	Destination subnet.. Specifies the destination subnet number (1-255)
	unsigned short Minimum: *1* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
unused	Unused.. Set to 0
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
node	Destination node.. Specifies the destination node number (1-127)
	bitfield Signed: *no* Width: *7* Offset: *1* Minimum: *1* Maximum: *127* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

ni	Device address as unique node ID address.. This structure is filled out in case the device address is given as unique node ID address
	Structure
	subnet: Destination subnet.. Specifies the destination subnet number (1-255) or 0 if the destination subnet is unknown
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	nid: .
	unsigned char [6] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTdevListEntry: text(("%m", address_type),((address_type == 1) ? ("%d/%d", sn.subnet, sn.node) :((address_type == 2) ? ("%d/%x-%x-%x-%x-%x-%x", ni.subnet, ni.nid[0], ni.nid[1], ni.nid[2], ni.nid[3], ni.nid[4], ni.nid[5]) :("UNKNOWN"))))

Used by:

SFPTdeviceMonitor

SCPTdevMajVer

Overview:

Device major version number. The major version number for the device

This configuration property sets the major version number for a device.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index:	*165*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTdevMajVer*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTdevMajVer: text("%d")*
Used by:	*SFPTnodeObject*

SCPTdevMinVer

Overview:

Device minor version number. The minor version number for the device

This configuration property sets the minor version number for the device.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*166*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTdevMinVer*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTdevMinVer: text("%d")*
Used by:	*SFPTnodeObject*

SCPTdeviceControlMode

Overview:

Device control mode. Normal default operating device control mode

This configuration property sets the device control mode to be used for the normal operating mode when a remote network pressure or flow sensor is not bound to a controller and the internal speed, pressure or flow feedback signal is used by the controller.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*238*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTdeviceControlMode*
Default:	*DCM_SPEED_CONST*
Neuron C Type:	*SNVT_dev_c_mode*
Formats:	*SCPTdeviceControlMode: text("%m")*
Used by:	*SFPTpumpController*

SCPTdeviceGroupID

Overview:

Group ID (ID number) . A logical group control ID for the device

This configuration property sets a unique logical group control ID for a device. The default number is zero (0), the invalid value. This is to avoid any conflict between devices that have been given a valid number, and devices that have yet to be given a valid number. Choosing a number within the valid range for a default would cause the logical group control ID numbering to be non-unique.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*172*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdeviceGroupID*

Default:	*0*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *65535* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTdeviceGroupID: text("%d")*

SCPTdeviceOutSelection

Overview:

Device Output Selection. This selects the Output which will be used on an OLC

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*347*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTdeviceOutSelection*
Default:	*OLC_DEFAULT*
Neuron C Type:	*olc_select_t*
Formats:	*SCPTdeviceOutSelection: text("%m")*
Used by:	*SFPToutdoorLuminairController*

SCPTdialString

Overview:

Dial string. Telephone number string used in dialing, including characters used for control

This configuration property sets a telephone number (including characters used for control) used for dialing a data modem. The telephone number is a nul-terminated string with optional ASCII control characters.

Support for the following ASCII control characters is required:

- Digits 0-9 plus * and # – telephone dialing digits.

Support for the following ASCII control characters is specified but optional (if the corresponding function is to be implemented, it must be done so using the following specified character):

- Letter T (upper or lower case, first character in string only) – Tone dial.
- Letter P (upper or lower case, first character in string only) – Pulse dial.
- Comma (,) – pause dialing for 2 seconds for each comma encountered.
- Letter W (upper or lower case) – Pause dialing until a dial tone is detected.
- Exclamation Point (!) – Hookflash, go on-hook for 0.5 seconds, then back off-hook.

Support of any other characters/functions in the dial string is considered optional. If an optional character is encountered in the string which is not supported by a particular implementation (including but not limited to punctuation such as space, dash, left and right parentheses), that character must be ignored, rather than causing an error condition.

The default value shall be an empty string.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*178*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTdialString*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTdialString: text("%s", ascii)*
Used by:	*SFPTtelephoneDirectory*

SCPTdiffNight

Overview:

Difference night. The value to be added to the cut-out value to get the cut-in limit during night control

This configuration property indicates the value to be added to the SCPTcutOutValue configuration property to get the cut in limit if cut in/out control is selected. If modulating thermostat control is selected, the target temperature is the SCPTcutOutValue limit + SCPTdiffNight / 2.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*122*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdiffNight*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTdiffNight#SI: text("%f", *1+0(0:854))* *SCPTdiffNight#US: text("%f", *1.8+0(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTdiffTempSetpoint

Overview:

Differential temperature. Setpoint for differential temperature for economizer enable

This configuration property sets the differential between entering (mixed) air temperature and entering condenser water temperature to enable economizer operation.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*201*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdiffTempSetpoint*
Default:	*5,00*
Neuron C Type:	*SNVT_temp_diff_p*
Formats:	*SCPTdiffTempSetpoint#SI: text("%f", *1+0(0:854))* *SCPTdiffTempSetpoint#US: text("%f", *1.8+0(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTdiffValue

Overview:

Difference value. The value to be added to the cut-out value to get the cut-in limit

This configuration property sets the value to be added to the SCPTcutOutValue configuration property to get the cut in limit if cut in/out control is selected. If modulating thermostat control is selected, the target temperature is the SCPTcutOutValue limit + SCPTdiffValue / 2.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*130*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdiffValue*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTdiffValue#SI: text("%f", *1+0(0:854))* *SCPTdiffValue#US: text("%f", *1.8+0(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTdirection

Overview:

Direction / Safety position. The actuator sense of rotation and safety position; bit 0 set => counterclockwise, bit 1 set => damper open

This configuration property sets the direction of rotation and safety position for an actuator. Because improper usage may cause a non-functional device, the device manufacturer can disable write access.

bit0: Actuator direction.
0 : actuator runs clockwise to open control device (top view)
1 : actuator runs counterclockwise open control device (top view)

bit1, **bit2**: Safety position.
0,0 : safety position: control device closed
1,0 : safety position: control device open
0,1 : safety position: manufacturer-defined position
1,2 : safety position: hold current position

bit2–**bit15**: Reserved.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*44*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdirection*
Default:	*0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*
Neuron C Type:	*SNVT_state*
Formats:	*SCPTdirection: text("%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d", bit0, bit1, bit2, bit3, bit4, bit5, bit6, bit7, bit8, bit9, bit10, bit11, bit12, bit13, bit14, bit15)* *SCPTdirection#LO: text("%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d|%d", bit0, bit1, bit2, bit3, bit4, bit5, bit6, bit7, bit8, bit9, bit10, bit11, bit12, bit13, bit14, bit15)*
Used by:	*SFPTdamperActuator SFPTfireSmokeDamperActuator*

SCPTdischargeAirCoolingSetpoint

Overview:

Discharge air cooling setpoint. Default cooling setpoint for discharge air

This configuration property sets the default discharge-air cooling setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*183*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdischargeAirCoolingSetpoint*
Default:	*13,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTdischargeAirCoolingSetpoint#SI: text("%f", *1+0(0:854))* *SCPTdischargeAirCoolingSetpoint#US: text("%f", *1.8+32(0:855))*

Used by:	*SFPTdischargeAirController*

SCPTdischargeAirDewpointSetpoint

Overview:

Discharge air dewpoint. Setpoint for the default discharge air dewpoint

This configuration property sets the default discharge-air dew-point setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*204*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdischargeAirDewpointSetpoint*
Default:	*10,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTdischargeAirDewpointSetpoint#SI: text("%f", *1+0(0:854))*
	*SCPTdischargeAirDewpointSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTdischargeAirHeatingSetpoint

Overview:

Discharge air heating setpoint. Default heating setpoint for discharge air

This configuration property sets the default discharge-air heating setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*184*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdischargeAirHeatingSetpoint*
Default:	*10,00*

Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTdischargeAirHeatingSetpoint#SI: text("%f", *1+0(0:854))* *SCPTdischargeAirHeatingSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTdrainDelay

Overview:

Drain delay. The delay to use after the defrost has terminated

This configuration property sets the delay to use after defrost has terminated. This is the first step in the start after defrost sequence.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*108*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdrainDelay*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTdrainDelay: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerDefrost*

SCPTdriveT

Overview:

Drive time. Time to be taken by the actuator to move from one extreme to the other

This configuration property sets the time to be taken by an actuator to move from one extreme to the other.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*8*

Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTdriveT*
Maximum:	*0 17:59:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTdriveT: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTdriveT#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopActuator SFPTopenLoopActuator*

SCPTdriveTime

Overview:

Drive time. The transition time for a full 100% stroke (change from one extreme to the other)

This configuration property sets the transition time for a full stroke (100%) of an actuator. Because improper usage may cause a non-functional device, the device manufacturer can disable write access. This configuration property does not affect airflow control actuators.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*45*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTdriveTime*
Default:	*150,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTdriveTime: text("%f")*
Used by:	*SFPTdamperActuator SFPTfireSmokeDamperActuator SFPThvacValvePositioner*

SCPTductArea

Overview:

Duct area or size. The duct area used to calculate the air flow, relevant only for VAV actuators / controllers

This configuration property sets the area or size of a duct. It is used to calculate air flow through the duct and is typically used by VAV actuators and controllers. Because improper usage may cause a non-functional device, the device manufacturer can disable write access.

The invalid value indicates that the box is not set up correctly.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*46*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTductArea*
Default:	*13,1070*
Neuron C Type:	*SNVT_area*
Formats:	*SCPTductArea: text("%f")*
Used by:	*SFPTairVelocitySensor SFPTdamperActuator SFPTdischargeAirController SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTvariableAirVolume*

SCPTductStaticPressureLimit

Overview:

Duct static pressure limit. The duct static pressure limit for equipment protection

This configuration property sets the duct static pressure limit for equipment protection.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*192*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTductStaticPressureLimit*

Default:	*1500*
Neuron C Type:	*SNVT_press_p*
Formats:	*SCPTductStaticPressureLimit#SI: text("%f")* *SCPTductStaticPressureLimit#US: text("%f", *0.0040217+0(0:954))* *SCPTductStaticPressureLimit#US_psi: text("%f", *1.4504e-4+0(0:875))*
Used by:	*SFPTdischargeAirController*

SCPTductStaticPressureSetpoint

Overview:

Duct static pressure. Setpoint for default duct static pressure

This configuration property sets a default duct static pressure setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*189*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTductStaticPressureSetpoint*
Default:	*500*
Neuron C Type:	*SNVT_press_p*
Formats:	*SCPTductStaticPressureSetpoint#SI: text("%f")* *SCPTductStaticPressureSetpoint#US: text("%f", *0.0040217+0(0:954))* *SCPTductStaticPressureSetpoint#US_psi: text("%f", *1.4504e-4+0(0:875))*
Used by:	*SFPTdischargeAirController*

SCPTeffectivePeriod

Overview:

Effective period. Time period during which a functional block is effective.

This configuration property defines the time period during which a functional block is effective. The effective period is defined by a start date and an end date. If the start date is undefined it means any date up to and including the end date. If the end date is undefined it means any date from the start date. If both are undefined, it means the functional block is always effective.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*272*
Obsolete:	*no*
Size:	*8*
Programmatic Name:	*SCPTeffectivePeriod*
Default:	*-1 1 1 -1 1 1*

Neuron C Type:

Structure	
start	Starting date. Starting date of the effective period

	Structure	
	year	Year. Starting year
		signed long Minimum: *-1* Maximum: *3000* Invalid: *-1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	month	Month. Starting month
		unsigned short Minimum: *1* Maximum: *12* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	day	Day. Starting day
		unsigned short Minimum: *1* Maximum: *31* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
end	Ending date. Ending date of the effective period	

	Structure	
	year	Ending year. Ending year of the effective period
		signed long Minimum: *-1* Maximum: *3000* Invalid: *-1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	month	Ending month. Ending month of the effective period
		unsigned short Minimum: *1* Maximum: *12* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	day	Ending day. Ending day of the effective period
		unsigned short Minimum: *1* Maximum: *31* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SCPTeffectivePeriod: text("%d %d %d %d %d %d", start.year, start.month, start.day, end.year, end.month, end.day)*

Used by: *SFPTcalendar SFPTscheduler*

SCPTemergCnfg

Overview:

.

Mode that a device has to be brought to when an emergency request state is pending.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*258*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTemergCnfg: text("%d")*
Used by:	*SFPTentryExit*

SCPTemergencyPosition

Overview:

Emergency position. Position in percent of full scale (open) for emergency operation

This configuration property sets the position of an actuator for emergency operation.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*250*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTemergencyPosition*
Minimum:	*0,000*
Maximum:	*100,000*
Default:	*100,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTemergencyPosition: text("%f")*
Used by:	*SFPThvacValvePositioner*

SCPTenableStatusMsg

Overview:

Enable Status Message.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*348*
Obsolete:	*no*
Size:	*5*
Programmatic Name:	*SCPTenableStatusMsg*
Default:	*0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0*
Neuron C Type:	

Structure	
lamp_current_high	Lamp current high.
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
lamp_current_low	Lamp current low.
	bitfield Signed: *no* Width: *1* Offset: *1* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
main_current_high	Main current high.

	bitfield Signed: *no* Width: *1* Offset: *2* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
main_current_low	Main current low.
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
lamp_voltage_high	Lamp voltage high (DALI Bit5).
	bitfield Signed: *no* Width: *1* Offset: *4* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
lamp_voltage_low	Lamp voltage low (DALI Bit5).

	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
main_voltage_high	Main voltage high.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
main_voltage_low	Main voltage low.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
power_factor_low	Power factor low.

	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
OLC_temp_high	Temperature high (DALI Bit 4).
	bitfield Signed: *no* Width: *1* Offset: *1* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
power_high	Power high.
	bitfield Signed: *no* Width: *1* Offset: *2* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
power_low	Power low.

	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
relay_failure	Relay failure.
	bitfield Signed: *no* Width: *1* Offset: *4* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
cap_failure	Capacity failure (DALI Bit 7).
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
lamp_failure	Lamp failure (DALI Bit 6).

	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
ballast_failure	Ballast failure.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
inter_com_failure	Internal communication failure.
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
exter_com_failure	External communication failure.

	bitfield Signed: *no* Width: *1* Offset: *1* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
main_volt_below_spec	Main voltage below performance specification (DALI Bit 1).
	bitfield Signed: *no* Width: *1* Offset: *2* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
lamp_restart_count	Lamp restart retry counter / cycling failure (DALI Bit 2).
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
fading_ready	Fading ready (DALI Bit 3).

	bitfield Signed: *no* Width: *1* Offset: *4* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
ballast_temp_high	Ballast temperature too high.
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digi_in_A	digital input A active.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
digi_in_B	digital input B active.

	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_25_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_26_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *1* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_27_res	Reserve.

	bitfield Signed: *no* Width: *1* Offset: *2* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_28_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_29_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *4* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_30_res	Reserve.

	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_31_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_32_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_33_res	Reserve.

	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_34_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *1* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_35_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *2* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_36_res	Reserve.

	bitfield Signed: *no* Width: *1* Offset: *3* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_37_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *4* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_38_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_39_res	Reserve.

	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
bit_40_res	Reserve.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTenableStatusMsg: text("%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d", lamp_current_high, lamp_current_low, main_current_high, main_current_low, lamp_voltage_high, lamp_voltage_low, main_voltage_high, main_voltage_low, power_factor_low, OLC_temp_high, power_high, power_low, relay_failure, cap_failure, lamp_failure, ballast_failure, inter_com_failure, exter_com_failure, main_volt_below_spec, lamp_restart_count, fading_ready, ballast_temp_high, digi_in_A, digi_in_B)

Used by:

SFPToutdoorLuminairController

SCPTenergyCntInit

Overview:

Energy counter initialization. The initial value of the energy counter for the associated network variable

This configuration property sets the initial value of the energy counter for the associated

output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*137*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTenergyCntInit*
Default:	*0*
Neuron C Type:	*SNVT_elec_kwh*
Formats:	*SCPTenergyCntInit: text("%d")*
Used by:	*SFPTisiLampActuator SFPTlampActuator*

SCPTexhaustEnablePosition

Overview:

Exhaust enable position. Setpoint for the exhaust-enable outdoor air damper position

This configuration property sets the exhaust enable outdoor-air damper-position setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*202*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTexhaustEnablePosition*
Default:	*25,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTexhaustEnablePosition: text("%f")*
Used by:	*SFPTdischargeAirController*

SCPTfadeTime

Overview:

Fade time, default to scene. The desired time to fade to zero

This configuration property sets the fade time for a scene. It is used if the fade time is set to 0 or if the learn current functionality defined by the SNVT_setting network variable type is used. It is possible to learn scenes with different fade times by changing this configuration property for each scene before the learn current command is used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*95*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTfadeTime*
Minimum:	*0,0*
Default:	*1,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTfadeTime: text("%f")*
Used by:	*SFPTsceneController*

SCPTfanDifferentialSetpoint

Overview:

Fan differential. Setpoint for the percent capacity differential between the supply and return fans

This configuration property sets the default setpoint for the percent capacity difference between the supply and return fans. The return fan will be controlled to maintain this differential in capacity below the supply fan.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*195*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTfanDifferentialSetpoint*
Default:	*10,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTfanDifferentialSetpoint: text("%f")*

Used by: *SFPTdischargeAirController*

SCPTfanInEnable

Overview:

Fan-in enable.. Enables fan-in of multiple data sources. When True, the application examines the source address of each input value and uses it to determine the data source of the update.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*328*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTfanInEnable*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *1* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTfanInEnable: text("%d")*
Used by:	*SFPTdataLogger*

SCPTfanOperation

Overview:

Fan operation.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*260*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTfanOperation*
Default:	*HVF_CONTINUOUS*

Neuron C Type:	*fan_operation_t*
Formats:	*SCPTfanOperation: text("%m")*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTfieldCalib

Overview:

Field calibration. Used by the light sensor to self calibrate the hardware

This configuration property specifies the ambient lux value for a light sensor. It is typically used by a light sensor to self calibrate the light sensing hardware. It is typically set to the ambient lux value measured with an external lux meter. The light sensor can then adjust its reflection factor to give exactly the same ambient light output value.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*90*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTfieldCalib*
Default:	*0*
Neuron C Type:	*SNVT_lux*
Formats:	*SCPTfieldCalib: text("%d")*
Used by:	*SFPTlightSensor*

SCPTfireIndicate

Overview:

Fire indicator device type. Describes the fire indicator device

This configuration property sets the fire indicator device.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*153*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTfireIndicate*
Default:	*FN_UNDEFINED*
Neuron C Type:	*SNVT_fire_indcte*
Formats:	*SCPTfireIndicate: text("%m")*
Used by:	*SFPTaudibleFireIndicator SFPTuniversalFireIndicator SFPTvisibleFireIndicator*

SCPTfireInitType

Overview:

Fire initiator type identifier. The fire initiator type identifier, entered into the device at installation and/or configuration time

This configuration property sets the fire initiator type identifier.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*38*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTfireInitType*
Default:	*FI_UNDEFINED*
Neuron C Type:	*SNVT_fire_init*
Formats:	*SCPTfireInitType: text("%m")*
Used by:	*SFPTuniversalFireInitiator*

SCPTfireTxt1

Overview:

Fire text information. Text information relevant to fire conditions. A '>' at end of string indicates presence of fire text 2.

This configuration property sets text information relevant to fire conditions. A “>” character at the end of the text string indicates the presence of additional text in an SCPTfireTxt2 configuration property.

The ascii field contains a nul-terminated string of up to 30 characters (making a total of 31 characters). The default value is manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*149*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTfireTxt1*
Minimum:	*00*31*
Maximum:	*FF*30-00*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTfireTxt1: text("%s", ascii)*
Used by:	*SFPTaudibleFireIndicator SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator SFPTuniversalFireIndicator SFPTuniversalFireInitiator SFPTvisibleFireIndicator*

SCPTfireTxt2

Overview:

Fire text information, continuation. Continuation text information relevant to fire conditions. A '>' at end of string indicates presence of fire text 3.

This configuration property sets text information relevant to fire conditions. It is typically used with the SCPTfireTxt1 configuration property. A “>” character at the end of the text string indicates the presence of additional text in an SCPTfireTxt3 configuration property.

The ascii field contains a nul-terminated string of up to 30 characters (making a total of 31 characters). The default value is manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*150*
Obsolete:	*no*

Size:	*31*
Programmatic Name:	*SCPTfireTxt2*
Minimum:	*00*31*
Maximum:	*FF*30-00*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTfireTxt2: text("%s", ascii)*
Used by:	*SFPTaudibleFireIndicator SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator SFPTuniversalFireIndicator SFPTuniversalFireInitiator SFPTvisibleFireIndicator*

SCPTfireTxt3

Overview:

Fire text information, second continuation. Second continuation text information relevant to fire conditions

This configuration property sets text information relevant to fire conditions. It is typically used with the SCPTfireTxt1 and SCPTfireTxt2 configuration properties.

The ascii field contains a nul-terminated string of up to 30 characters (making a total of 31 characters). The default value is manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*151*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTfireTxt3*
Minimum:	*00*31*
Maximum:	*FF*30-00*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTfireTxt3: text("%s", ascii)*
Used by:	*SFPTaudibleFireIndicator SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator SFPTuniversalFireIndicator*

SCPTflashFreq

Overview:

Flash rate specification. Flash rate specification for visible indication (strobe) device

This configuration property sets the flash rate for a visible indication (strobe) device. This information is defined at manufacture time.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*145*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTflashFreq*
Minimum:	*0,1*
Maximum:	*5,0*
Default:	*1,0*
Neuron C Type:	*SNVT_freq_hz*
Formats:	*SCPTflashFreq: text("%f")*
Used by:	*SFPTvisibleFireIndicator*

SCPTfreeCoolPosition

Overview:

Free cooling valve position. Valve position in percent open for free cooling HVAC mode

This configuration property sets the valve position for the Free Cooling HVAC mode.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*247*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTfreeCoolPosition*
Minimum:	*0,000*

Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTfreeCoolPosition: text("%f")*
Used by:	*SFPThvacValvePositioner*

SCPTgain

Overview:

Gain. This parameter is used to calibrate the external hardware.

This configuration property sets the multiplication factor for raw data readings from sensor hardware. It is used to calibrate readings from sensor hardware. It applies when the readings are linear and do not require a translation table. The gain is applied before any specified offset is applied. The gain consists of a 16-bit multiplier and a 16-bit divisor. An Invalid Value is represented by a divisor of zero (0).

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*31*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTgain*
Default:	*1/1*
Neuron C Type:	*SNVT_muldiv*
Formats:	*SCPTgain: text("%d/%d", multiplier, divisor)*
Used by:	*SFPTairVelocitySensor SFPTclosedLoopSensor SFPTglobalSolarRadiation SFPTopenLoopSensor*

SCPTgainVAV

Overview:

VAV gain. The gain of the VAV controller object

This configuration property sets the gain of a VAV controller.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*66*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTgainVAV*
Default:	*1,0000*
Neuron C Type:	*SNVT_multiplier*
Formats:	*SCPTgainVAV: text("%f")*
Used by:	*SFPTvariableAirVolume*

SCPTgainVAVHeat

Overview:

VAV sensor constant. Calibration constant used to calculate airflow

Calibration constant used to calculate airflow.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*268*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTgainVAVHeat*
Default:	*1,0000*
Neuron C Type:	*SNVT_multiplier*
Formats:	*SCPTgainVAVHeat: text("%f")*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTgeoLocation

Overview:

Geographic Location.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*350*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTgeoLocation*
Default:	*0,0000000 0,0000000 0*
Neuron C Type:	*SNVT_geo_loc*
Formats:	*SCPTgeoLocation: text("%f %f %f %s", longitude, latitude, elevation, name)*
Used by:	*SFPTnodeObject SFPTsmartLuminaireController*

SCPTheatLowerSP

Overview:

Heating setpoint lower limit. Limits the lower extent of the permitted range for the heating setpoint

This configuration property sets the lower limit for the heating setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*79*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTheatLowerSP*
Minimum:	*10,00*
Maximum:	*35,00*
Default:	*10,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTheatLowerSP#SI: text("%f", *1+0(0:854))* *SCPTheatLowerSP#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTthermostat*

SCPTheatSetpt

Overview:

Heating setpoint. The default setpoint for the leaving water temperature in heating mode when the default behavior selector is set to zero

This configuration property sets the default setpoint for the leaving water temperature when in heating mode. If the SCPTdefltBehave configuration property is set to one, the manufacturer-specified values are used instead.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*78*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTheatSetpt*
Minimum:	*-40,00*
Maximum:	*93,00*
Default:	*35,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTheatSetpt#SI: text("%f", *1+0(0:854))* *SCPTheatSetpt#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTboilerController SFPTchiller*

SCPTheatUpperSP

Overview:

Heating setpoint upper limit. Limits the upper extent of the permitted range for the heating setpoint

This configuration property sets the upper limit for the heating setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*80*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTheatUpperSP*
Minimum:	*10,00*
Maximum:	*35,00*
Default:	*35,00*

Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTheatUpperSP#SI: text("%f", *1+0(0:854))*
	*SCPTheatUpperSP#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTthermostat*

SCPTheatingLockout

Overview:

Heating lockout. Setpoint for the outdoor air temperature at which heating will be disabled

This configuration property sets the outdoor air temperature heating lockout setpoint. When the outdoor air temperature is above this value, heating will be disabled.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*210*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTheatingLockout*
Default:	*20,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTheatingLockout#SI: text("%f", *1+0(0:854))*
	*SCPTheatingLockout#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTheatingResetEnable

Overview:

Heating reset enable (boolean) . The heating reset control is enabled

This configuration property enables or disables discharge air temperature heating reset control. A True value enables the discharge air temperature heating reset control.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*212*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTheatingResetEnable*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*
Formats:	*SCPTheatingResetEnable: text("%m")*
Used by:	*SFPTdischargeAirController*

SCPThighLimDefrDly

Overview:

High limit defrost delay. The time limit before high air temp alarm during pull-down

This configuration property sets the time limit before a high nvoAlarmAirTemp alarm during pull-down. This value applies until the actual nvoAlarmAirTemp value has dropped below the SCPThighLimDefrDly value. Thereafter, the SCPThighLimit2 value applies.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*133*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPThighLimDefrDly*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPThighLimDefrDly: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPThighLimDly

Overview:

High limit delay. The time limit during normal operation before the alarm air temp high alarm is recognized

This configuration property sets the time limit before a high nvoAlarmAirTemp alarm during normal operation.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*124*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPThighLimDly*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPThighLimDly: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPThighLimTemp

Overview:

High limit temperature. The high alarm set point for the alarm air temp network variable

This configuration property sets the high alarm set point. It is typically used for an nvoAlarmAirTemp output. When there is night setback operation, the alarm limit is raised by the SCPTdeltaNight value.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*123*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPThighLimTemp*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPThighLimTemp#SI: text("%f", *1+0(0:854))* *SCPThighLimTemp#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPThighLimit1

Overview:

.

This configuration property sets the first alarm high limit against which the primary network output network variable value is tested for alarm conditions. The data type is the same as the value field of the output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*9*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPThighLimit1: text("%d")*
Used by:	*SFPTchannelMonitor SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor SFPTunitHeater SFPTutilityDataLoggerRegister SFPTwallUnit*

SCPThighLimit1Enable

Overview:

High limit 1 Enable. Controls whether high limit 1 is in effect

Controls whether high limit 1 is in effect.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*302*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPThighLimit1Enable*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*

Formats:	*SCPThighLimit1Enable: text("%m")*
Used by:	*SFPTchannelMonitor*

SCPThighLimit2

Overview:

.

This configuration property sets the second alarm high limit against which the primary output network variable value is tested for alarm conditions. The data type is the same as the value field of the output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*10*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPThighLimit2: text("%d")*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPThighLimit2Enable

Overview:

High limit 2 Enable. Controls whether high limit 2 is in effect

Controls whether high limit 2 is in effect.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*303*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPThighLimit2Enable*
Default:	*BOOL_FALSE*

Neuron C Type:	*boolean_t*
Formats:	*SCPThighLimit2Enable: text("%m")*

SCPTholdTime

Overview:

Hold time. Hold time for occupied state after there is no occupancy detected

This configuration property sets the hold time for the occupied state. The hardware input to an occupancy sensor must indicate occupancy for longer than this time before the output value of the occupancy sensor device is changed to indicate occupancy. The purpose of the delay is to avoid unnecessary network traffic when the sensor is not detecting presence continuously. This is useful when passive infrared detectors are used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*91*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTholdTime*
Minimum:	*1,0*
Default:	*30,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTholdTime: text("%f")*
Used by:	*SFPTentryExit SFPTisiOccupancySensor SFPToccupancyController*

SCPThumSetpt

Overview:

Humidity high limit setpoint. High limit humidity setpoint for the controlled space. A zero value disables

This configuration property sets the high limit humidity setpoint for a controlled space. The controller dehumidification functions in response to this limit are manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*

Index:	*36*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPThumSetpt*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPThumSetpt: text("%f")*
Used by:	*SFPTdischargeAirController SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTunitVentilatorController*

SCPThvacMode

Overview:

HVAC mode. The default operating mode of the device when the default behavior selector is set to zero

This configuration property sets the default operating mode of a chiller. If the SCPTdefltBehave configuration property is set to one, the manufacturer-specified values are used instead.

Restricted Range

Value	Identifier	Notes
1	HVAC_HEAT	Heating mode
3	HVAC_COOL	Cooling mode
10	HVAC_FREE_COOL	Cooling with compressor not running
11	HVAC_ICE	Ice-making mode
-1 (0xFF)	HVAC_NUL	Value not available

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*74*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPThvacMode*
Default:	*HVAC_COOL*
Neuron C Type:	*SNVT_hvac_mode*

Formats:	*SCPThvacMode: text("%m")*
Used by:	*SFPTchiller*

SCPThvacType

Overview:

HVAC unit type. The type of HVAC equipment being controlled

This configuration property sets the type of HVAC equipment that is being controlled.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*169*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPThvacType*
Default:	*HVT_GENERIC*
Neuron C Type:	*SNVT_hvac_type*
Formats:	*SCPThvacType: text("%m")*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPThystHigh1

Overview:

.

This configuration property sets the hysteresis level for the value field of the SCPThighLimit1 comparison threshold. The data type must be the same as the value field of the output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*11*

Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPThystHigh1: text("%d")*
Used by:	*SFPTchannelMonitor SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPThystHigh2

Overview:

.

This configuration property sets the hysteresis level for the value field of the SCPThighLimit2 comparison threshold. The data type must be the same as the value field of the output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*12*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPThystHigh2: text("%d")*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPThystLow1

Overview:

.

This configuration property sets the hysteresis level for the value field of the SCPTlowLimit1 comparison threshold. The data type must be the same as the value field of the output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*13*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPThystLow1: text("%d")*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPThystLow2

Overview:

.

This configuration property sets the hysteresis level for the value field of the SCPTlowLimit2 comparison threshold. The data type must be the same as the value field of the output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*14*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPThystLow2: text("%d")*
Used by:	*SFPTchannelMonitor SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTidentity

Overview:

Sensor Identity. A number from 0 to 65535 identifying a sensor by number

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*294*

Obsolete:	*yes*
Size:	*2*
Programmatic Name:	*SCPTidentity*
Default:	*0*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTidentity: text("%d")*

SCPTifaceDesc

Overview:

Interface description.. Human readable description of the interface the functional block is assigned to

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*318*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTifaceDesc*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTifaceDesc: text("%s", ascii)*
Used by:	*SFPTchannelMonitor*

SCPTinFbDly

Overview:

Input value feedback delay. The time period after the last update in a succession of changes to the input, before the feedback output is updated

This configuration property sets the time period between the last update in a succession of changes to the primary input network variable of a functional block and the corresponding feedback output network variable being updated.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*15*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTinFbDly*
Maximum:	*0 0:0:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTinFbDly: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTinFbDly#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopActuator SFPTisiKeypad SFPTisiLampActuator SFPTisiOccupancySensor SFPTisiSunblindActuator SFPTlampActuator*

SCPTinjDelay

Overview:

Injection delay. The delay to use after the defrost has terminated

This configuration property sets the delay to use after defrost has terminated. This is the second step in the start after defrost sequence.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*109*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTinjDelay*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTinjDelay: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerDefrost*

SCPTinstallDate

Overview:

Installation date. The date of installation for the device

This configuration property sets the date of installation for a device.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*146*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTinstallDate*
Default:	*0/0/0 0:0:0*
Neuron C Type:	*SNVT_time_stamp*
Formats:	*SCPTinstallDate: text("%d/%d/%d %d:%d:%d", year, month, day, hour, minute, second)* *SCPTinstallDate#LO: text(date(year, month, day), (" "), time(hour, minute, second))*
Used by:	*SFPTaudibleFireIndicator SFPTfireSmokeDamperActuator SFPThvacValvePositioner SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator SFPTuniversalFireIndicator SFPTvisibleFireIndicator*

SCPTinstalledLevel

Overview:

Installed level (floor number) . The floor or level on which the device is installed

The configuration property sets the installation location of any device used within an elevator control system. It is compatible with the standard elevator notation of identifying each floor (landing) and each front and rear car door opening with a unique level index.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*232*

Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTinstalledLevel*
Default:	*0*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTinstalledLevel: text("%d")*
Used by:	*SFPTelevatorArrivalGong SFPTelevatorHallLantern*

SCPTinvrtOut

Overview:

Invert output. This parameter indicates to invert the active polarity, if the value is nonzero (ON).

This configuration property is used to invert the active polarity of an output network variable. This enables the use of either normally closed or normally open contacts. A value other than ST_OFF specifies that the output value should be inverted.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*16*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTinvrtOut*
Default:	*ST_OFF*
Neuron C Type:	*SNVT_lev_disc*
Formats:	*SCPTinvrtOut: text("%m")*
Used by:	*SFPTclosedLoopSensor SFPTopenLoopSensor SFPTuniversalFireInitiator*

SCPTlampPower

Overview:

Lamp Power.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*346*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTlampPower*
Default:	*0,0*
Neuron C Type:	*SNVT_power*
Formats:	*SCPTlampPower: text("%f")*
Used by:	*SFPToutdoorLuminairController*

SCPTlightingGroupEnable

Overview:

Lighting group enable. Bit masks to enable or disable up to 64 ISI lighting groups. Group 0 is not used. Groups may also be enabled or disabled using a SNVT_switch_2 update.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*342*
Obsolete:	*no*
Size:	*8*
Programmatic Name:	*SCPTlightingGroupEnable*
Default:	*0 0 0 0 0 0 0 0*

Neuron C Type:

Structure	
flags	.
	unsigned short [8] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SCPTlightingGroupEnable: text("%d %d %d %d %d %d %d %d",*

flags[0], flags[1], flags[2], flags[3], flags[4], flags[5], flags[6], flags[7])

Used by: *SFPTisiKeypad SFPTisiLampActuator SFPTisiSunblindActuator SFPTsmartLuminaireController*

SCPTlightingGroupMembership

Overview:

Lighting group membership. Bit masks to specify membership in up to 64 ISI lighting groups. Group 0 is not used.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *361*

Obsolete: *no*

Size: *8*

Programmatic Name: *SCPTlightingGroupMembership*

Default: *0 0 0 0 0 0 0 0*

Neuron C Type:

Structure	
flags	.
	unsigned short [8] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SCPTlightingGroupMembership: text("%d %d %d %d %d %d %d %d", flags[0], flags[1], flags[2], flags[3], flags[4], flags[5], flags[6], flags[7])*

Used by: *SFPTsmartLuminaireController*

SCPTlimitCO2

Overview:

CO2 limit. CO2 threshold limit, controller to maintain concentration below this limit

This configuration property sets the carbon dioxide (CO_2) threshold value. If the sensed CO_2 value exceeds this limit, then the controller will reduce the CO_2 concentration.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*42*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTlimitCO2*
Default:	*0*
Neuron C Type:	*SNVT_ppm*
Formats:	*SCPTlimitCO2: text("%d")*
Used by:	*SFPTroofTopUnit SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTunitVentilatorController*

SCPTlimitChlrCap

Overview:

Chiller capacity limit. The default value for the capacity limit of the chiller when the default behavior selector is set to zero

This configuration property sets the default value for the capacity limit of a chiller. If the SCPTdefltBehave configuration property is set one, the manufacturer-specified values are used instead. The capacity limit value is not the nominal capacity limit of the chiller.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*81*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTlimitChlrCap*
Minimum:	*0,000*
Maximum:	*160,000*
Default:	*100,000*
Neuron C Type:	*SNVT_lev_percent*

Formats:	*SCPTlimitChlrCap: text("%f")*
Used by:	*SFPTchiller*

SCPTlimits

Overview:

LC Limits.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*383*
Obsolete:	*no*
Size:	*15*
Programmatic Name:	*SCPTlimits*
Default:	*0,0 0,0 0,0 0,0 0,00000 0,0 0,00 0,0 0,0 0*
Neuron C Type:	*SNVT_fault_limits*
Formats:	*SCPTlimits: text("%f %f %f %f %f %f %f %f %f %d", powerFault, voltageFault, lampVoltageFault, lampCurrentFault, pfLow, rcvHb, highTemp, lampFailFault, lampVoltage, lampCurrent)*
Used by:	*SFPTsmartLuminaireController*

SCPTlinkPowerDetectEnable

Overview:

Link Power Detection Enabled.. Determines, whether link power detection is enabled. If yes, nvoLinkPower indicates existence of link power voltage.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*320*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTlinkPowerDetectEnable*
Default:	*0 0*
Neuron C Type:	*SNVT_switch*

Formats:	*SCPTlinkPowerDetectEnable: text("%d %d", value,state)*
Used by:	*SFPTchannelMonitor*

SCPTloadControlOffset

Overview:

Load control offsets (percent) . Offsets to be used during standby (unoccupied state but home, or sleep mode) and demand-response modes

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*362*
Obsolete:	*no*
Size:	*6*
Programmatic Name:	*SCPTloadControlOffset*
Default:	*0,000 0 0,000 0*
Neuron C Type:	*SNVT_load_offsets*
Formats:	*SCPTloadControlOffset: text("%f %d %f %d", standby_offset, standby_rotation, demand_response_offset, demand_response_rotation)*
Used by:	*SFPTisiLampActuator SFPTisiSunblindActuator*

SCPTlocation

Overview:

Location. Provides descriptive physical location information related to the object.

This configuration property sets descriptive physical location information for the associated functional block or device. It provides a more detailed description of the device that can be provided by the Neuron Chip's 6-byte location string.

A SCPTlocation configuration property that applies to the Node Object functional block is used to identify the subsystem containing the device. This allows network recovery tools to recover subsystem information from a device. The subsystem may be a simple location name, or may be a hierarchical subsystem name. If a hierarchical subsystem name is specified, the subsystem hierarchy components must be separated by periods (“.”). For example, a device may have a Node Object SCPTlocation value of “Bldg 1.Floor 2.Rm 29”, representing the Bldg 1/Floor 2/Rm 29 subsystem. Periods must not otherwise be used in a SCPTlocation

value that applies to a Node Object functional block. Other characters that cannot be used in a subsystem name are the backslash ("\"), colon (":"), forward slash ("/"), or double-quote characters. For very large networks, subsystem numbers may be used instead of subsystem names, for example: "1.2.29". This allows deeply nested hierarchies to fit within the 31 character limit for SCPTlocation.

The ascii field contains a nul-terminated string of up to 30 characters (making a total of 31 characters). The default value is nul string (all zeroes).

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*17*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTlocation*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTlocation: text("%s", ascii)*
Used by:	*SFPTaudibleFireIndicator SFPTautomaticTransferSwitch SFPTboilerController SFPTchilledCeilingController SFPTchiller SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTclothesWasherDomestic SFPTconstantLightController SFPTdamperActuator SFPTdischargeAirController SFPTelevatorArrivalGong SFPTelevatorDirectionLantern SFPTelevatorFireSystemsPort SFPTelevatorHallLantern SFPTelevatorPositionIndicator SFPTelevatorVoiceAnnouncer SFPTentryExit SFPTfanCoilUnit SFPTfireSmokeDamperActuator SFPTgeneratorSet SFPThardwiredFireAlarmShutdown SFPThardwiredFullVentilation SFPThardwiredGasDetectionShutdown SFPThardwiredRecirculation SFPThardwiredSafetyInstrumentedSystemSIS SFPTheatPump SFPThvacValvePositioner SFPTidentifierSensor SFPTisiKeypad SFPTisiLampActuator SFPTisiOccupancySensor SFPTisiSunblindActuator SFPTlampActuator SFPTlightingPanelController SFPTlightSensor SFPTnodeObject SFPToccupancyController SFPToccupancySensor SFPTopenLoopActuator SFPTopenLoopSensor SFPTpartitionWallController SFPTpressureSensor SFPTpullStationFireInitiator SFPTpumpController SFPTrailcarAudioController SFPTrailcarAudioSensor SFPTrefrigDisplayCaseControllerDefrost SFPTrefrigDisplayCaseControllerEvaporator*

SFPTrefrigDisplayCaseControllerThermostat SFPTroofTopUnit SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTscenePanel SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTspaceComfortController SFPTsunblindActuator SFPTsunblindController SFPTswitch SFPTthermalFireInitiator SFPTthermostat SFPTunitVentilatorController SFPTuniversalFireIndicator SFPTuniversalFireInitiator SFPTutilityMeter SFPTvariableAirVolume SFPTvariableSpeedMotorDrive SFPTvisibleFireIndicator

SCPTlogAlarmThreshold

Overview:

Data log alarm threshold. (percent) . Specifies the log level required to trigger an alarm condition for the data logger.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*339*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTlogAlarmThreshold*
Default:	*95,5*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *200* Invalid: *255* Scaling (A,B,C): *5, -1, 0* Scaled value: *5 *10 $^{-1}$ *(Raw+0)* Resolution: *0.5*
Formats:	*SCPTlogAlarmThreshold: text("%f")*
Used by:	*SFPTdataLogger*

SCPTlogCapacity

Overview:

Data log capacity (bytes) . Specifies the total capacity of all data logs on a device. The size of each data log is specified by its cpLogSize value. The value is specified in bytes.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*324*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTlogCapacity*
Default:	*0*
Neuron C Type:	s32_type Minimum: *-2147483648* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTlogCapacity: text("%d")*
Used by:	*SFPTdataLogger*

SCPTlogFileHeader

Overview:

Data log header.. Describes contents of a data log.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*338*
Obsolete:	*no*
Size:	*20*
Programmatic Name:	*SCPTlogFileHeader*
Default:	*0 v0.0 0 0 0 0,00 0 0,00*
Neuron C Type:	

Structure	
file_type	Data log file type.. Constant identifying a data log.

	unsigned long Minimum: *2049* Maximum: *2049* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
major_version_number	Data log file format major version number..
	unsigned short Minimum: *1* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
minor_version_number	Data log minor version number..
	unsigned short Minimum: *0* Maximum: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
log_number	Data log number.. Index of the data log functional block that received this update.
	unsigned long Minimum: *0* Maximum: *65534* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
record_count	Data log record count. (seconds) . Number of records in data log.

	s32_type Minimum: *0* Maximum: *2147483646* Invalid: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
start_time	Data log start time. (seconds) .
	SNVT_time_stamp_p
end_time	Data log end time. (seconds) .
	SNVT_time_stamp_p

Formats:

SCPTlogFileHeader: text("%d v%d.%d %d %d %d %f %d %f", file_type, major_version_number, minor_version_number, log_number, record_count, start_time.second, start_time.hundredths, end_time.second, end_time.hundredths)

SCPTlogHighLimit

Overview:

.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*330*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTlogHighLimit: text("%d")*
Used by:	*SFPTdataLogger*

SCPTlogLowLimit

Overview:

.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*331*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTlogLowLimit: text("%d")*
Used by:	*SFPTdataLogger*

SCPTlogMinDeltaTime

Overview:

Data log minimum delta time.. Minimum amount of time between logged values. This is used to throttle data entry into a data log. When a data value is logged, a subsequent update to the data value is not logged until the time specified by this value has elapsed. If additional updates are received during this time, the older values are discarded and are not stored in the data log. Time of receipt is ignored if the value of this configuration property is zero or invalid.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*333*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTlogMinDeltaTime*
Default:	*0*
Neuron C Type:	s32_type Minimum: *-2147483648* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTlogMinDeltaTime: text("%d")*
Used by:	*SFPTdataLogger*

SCPTlogMinDeltaValue

Overview:

.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*334*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTlogMinDeltaValue: text("%d")*
Used by:	*SFPTdataLogger*

SCPTlogNotificationThreshold

Overview:

Data log notification threshold. (percent) . Specifies the percentage change in log level required to trigger an update to the Data Log Status (nvoStatus) output.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*325*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTlogNotificationThreshold*
Default:	*45,0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *200* Invalid: *255* Scaling (A,B,C): *5, -1, 0* Scaled value: *5 *10 $^{-1}$ *(Raw+0)* Resolution: *0.5*
Formats:	*SCPTlogNotificationThreshold: text("%f")*
Used by:	*SFPTdataLogger*

SCPTlogRecord

Overview:

Log record.. Documents the format of a data log record. Not used as a CP. Data logs have variable sized records--unused fields are not present.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *337*

Obsolete: *no*

Size: *16*

Programmatic Name: *SCPTlogRecord*

Default: *0 0 0 0 0 0 0 0,00 SINGLE_INPUT NO_FANIN 0 NO_POINT_STATUS*

Neuron C Type:

Structure	
timestamp_type	Timestamp type.. Specifies whether or not a timestamp is included, and if it is incljuded, specifies the format of the timestamp. Contents defined by timestamp_t.
	bitfield Signed: *no* Width: *2* Offset: *0* Minimum: *0* Maximum: *3* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
record_type	Log record type.. Specifies the format of a data log record. Contents defined by log_record_t.
	bitfield Signed: *no* Width: *3* Offset: *2* Minimum: *0* Maximum: *3* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
multiple_input	Multiple input NV flag.. Set to 1 if the functional block has multiple input NVs.

	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
fan_in	Data log fan-in flag.. Set to one if this input NV receives data from multiple data sources.
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
point_status	Data log point status flag.. Set to one if this data log record includes point status information.
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
data_length	Data log record length. (bytes) . Number of bytes in the data portion of a data log record. Set to 0 if the record does not contain a data value.

	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
time	Timestamp..
	Union timestamp: Timestamp (seconds) . Full timestamp. Only present if timestamp_type is TS_FULL. *SNVT_time_stamp_p* offset_stamp: Offset timestamp (seconds) . Offset since last full timestamp. Only present if timestamp_type is TS_OFFSET. unsigned long Minimum: *0* Maximum: *65534* Invalid: *65535* Scaling (A,B,C): *1, -2, 0* Scaled value: *1 *10 $^{-2}$ *(Raw+0)* Resolution: *0.0099999997764826*
member_index	Data log member index.. Functional block member number for the network variable input that received this update. Only present if mulitple_input is 1.
	unsigned short Minimum: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
data_source_index	Data log data source index.. Index into the cpSourceAddress array. Only present if fan_in is 1.

	signed long Minimum: *0* Maximum: *32767* Invalid: *-1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
data	Data log record data.. Data field for a data log record. Contents depend on record_type value. Unused bytes are not included.
	Union
	log_status: Data log status.. Changed value for data log status. Only present if record_type is LR_LOG_STATUS.
	log_status_t
	value: Data log data value.. Data log value. Size depends on size logged value and is defined by data_length. Only present if record_type is LR_DATA.
	s32_type Minimum: *-2147483648* Maximum: *2147483647* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	old_time: Data log old time (seconds) . Previous time after a time change. Only present if record_type is LR_TIME_CHANGE.
	SNVT_time_stamp_p
point_status_value	Data log point status.. Optional status for a data point value. Only present if point_status is 1.
	point_status_t

Formats:

SCPTlogRecord: text(("%d %d %d %d %d %d ", timestamp_type, record_type, multiple_input, fan_in, point_status, data_length),((timestamp_type == 0) ? ("%d %f ",

time.timestamp.second, time.timestamp.hundredths) :((timestamp_type == 1) ? ("%f ", time.offset_stamp) :((timestamp_type == 2) ? ("NO_TIMESTAMP ") : ("UNKNOWN "))), ((multiple_input == 1) ? ("%d ", member_index) :("SINGLE_INPUT ")),((fan_in == 1) ? ("%d ", data_source_index) :("NO_FANIN ")),((record_type == 0) ? ("%d ", data.value) : ((record_type == 1) ? ("%m ", data.log_status) :((record_type == 2) ? ("%d %f ", data.old_time.second, data.old_time.hundredths) :("UNKNOWN "))), ((point_status == 1) ? ("%m ", point_status_value) :("NO_POINT_STATUS")))

SCPTlogRequest

Overview:

Data log access request.. Data log access request message format. Not used as a CP. This request has a variable size--the timestamp field is only included with the get next record request.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*340*
Obsolete:	*no*
Size:	*8*
Programmatic Name:	*SCPTlogRequest*
Default:	*0 0 0 0,00*

Neuron C Type:

Structure	
dlap_version	Data log access protocol version.. Data log access protocol version number. Currently must be 1.
	unsigned short Minimum: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
requested_log	Requested log number.. The log number of the data log to be accessed. Logs are numbered from 1 to number_of_logs.
	unsigned long Minimum: *1* Maximum: *65535* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
last_time	Timestamp of last record fetched. The first record with a timestamp after this timestamp is returned. If this field is invalid, the first record in the data log is returned. . Timestamp of last record fetched. The first record with a timestamp after this timestamp is returned. If this field is invalid, the first record in the data log is returned.
	SNVT_time_stamp_p

Formats:

SCPTlogRequest: text("%d %d %d %f", dlap_version, requested_log, last_time.second, last_time.hundredths)

SCPTlogResponse

Overview:

Data log access response.. Data log access response message format. Not used as a CP. This response has a variable size--unused fields are not included.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index:	*341*
Obsolete:	*no*
Size:	*16*
Programmatic Name:	*SCPTlogResponse*
Default:	*0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0*

Neuron C Type:

Structure	
response	.
	Union
	log_record: .
	unsigned short [16] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	closest_version: Closest data log access protocol version.. Closest data log access protocol version number supported by this device. Returned for LRC_VER_MISMATCH responses.
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	number_of_logs: Number of logs.. Number of data logs in the device. Returned for LRC_BAD_LOG_INDEX responses.
	unsigned long Minimum: *1* Maximum: *65535* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTlogResponse: text("%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d", response.log_record[0], response.log_record[1], response.log_record[2], response.log_record[3], response.log_record[4],

response.log_record[5], response.log_record[6], response.log_record[7], response.log_record[8], response.log_record[9], response.log_record[10], response.log_record[11], response.log_record[12], response.log_record[13], response.log_record[14], response.log_record[15])

SCPTlogSize

Overview:

Log size. (bytes) . Capacity of a data log.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*326*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTlogSize*
Default:	*0*
Neuron C Type:	s32_type Minimum: *-2147483648* Maximum: *2147483647* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTlogSize: text("%d")*
Used by:	*SFPTdataLogger*

SCPTlogTimestampEnable

Overview:

Data log enable timestamp.. Enables time stamping of each data value. When True, the data logger includes a timestamp of the receipt time for each value received by the data logger.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*329*
Obsolete:	*no*

Size:	*1*
Programmatic Name:	*SCPTlogTimestampEnable*
Default:	*1*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *1* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTlogTimestampEnable: text("%d")*
Used by:	*SFPTdataLogger*

SCPTlogType

Overview:

Data log type.. Specifies the method used to store data in a data log.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*327*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTlogType*
Minimum:	*LT_CIRCULAR*
Maximum:	*3*
Default:	*LT_CIRCULAR*
Neuron C Type:	*log_type_t*
Formats:	*SCPTlogType: text("%m")*
Used by:	*SFPTdataLogger*

SCPTlowLimDly

Overview:

Low limit delay. The time limit during normal operation before the alarm air temp low alarm is recognized

This configuration property sets the minimum time that a value must be below the low limit before an alarm is generated.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*129*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTlowLimDly*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTlowLimDly: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTlowLimTemp

Overview:

Low limit temperature. The low alarm set point for the alarm air temp network variable

This configuration property sets the low alarm temperature set point.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*128*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTlowLimTemp*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTlowLimTemp#SI: text("%f", *1+0(0:854))* *SCPTlowLimTemp#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTlowLimit1

Overview:

.

This configuration property sets the first alarm low limit against which the value primary network output network variable value is tested for alarm conditions. The data type is the same as the value field of the output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*18*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTlowLimit1: text("%d")*
Used by:	*SFPTchannelMonitor SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor SFPTunitHeater SFPTwallUnit*

SCPTlowLimit1Enable

Overview:

Low limit 1 Enable. Controls whether low limit 1 is in effect

Controls whether low limit 1 is in effect.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*298*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTlowLimit1Enable*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*
Formats:	*SCPTlowLimit1Enable: text("%m")*
Used by:	*SFPTchannelMonitor*

SCPTlowLimit2

Overview:

.

This configuration property sets the second alarm low limit against which the primary network output network variable value is tested for alarm conditions. The data type is the same as the value field of the output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*19*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTlowLimit2: text("%d")*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTlowLimit2Enable

Overview:

Low limit 2 Enable. Controls whether low limit 2 is in effect

Controls whether low limit 2 is in effect.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*299*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTlowLimit2Enable*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*
Formats:	*SCPTlowLimit2Enable: text("%m")*

SCPTluxSetpoint

Overview:

Setpoint, illumination level. The illumination level setpoint for the controller

This configuration property sets the illumination-level setpoint. The setpoint value may also be changed temporarily by an input network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*82*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTluxSetpoint*
Default:	*0*
Neuron C Type:	*SNVT_lux*
Formats:	*SCPTluxSetpoint: text("%d")*
Used by:	*SFPTconstantLightController*

SCPTmaintDate

Overview:

Maintenance date. The date of last maintenance for the device

This configuration property sets the date of last maintenance (cleaning, inspection, test, etc.) for a device.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*147*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTmaintDate*
Default:	*0/0/0 0:0:0*
Neuron C Type:	*SNVT_time_stamp*
Formats:	*SCPTmaintDate: text("%d/%d/%d %d:%d:%d", year, month, day,*

hour, minute, second)

SCPTmaintDate#LO: text(date(year, month, day), (" "), time(hour, minute, second))

Used by:

SFPTaudibleFireIndicator SFPTfireSmokeDamperActuator SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator SFPTuniversalFireIndicator SFPTuniversalFireInitiator SFPTvisibleFireIndicator

SCPTmanOvrTime

Overview:

Manual override time. The maximum time that the controller will stay in a manual mode following the last request by a network variable input. Zero disables the timer.

This configuration property sets the maximum time that a controller will stay in a manual mode when the manual mode was requested by a network variable input, without receiving an update on that network variable. For example, if an override request is received via the network and an update is not received within the Manual Override Time, the controller will go back to the default value. Updates to the network variable input will restart the timer. The specific network variable inputs for which this timer is used are manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*35*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmanOvrTime*
Default:	*0*
Neuron C Type:	*SNVT_time_min*
Formats:	*SCPTmanOvrTime: text("%f")*
Used by:	*SFPTchilledCeilingController SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTunitVentilatorController*

SCPTmanfDate

Overview:

Manufacture date. The date of manufacture for the device

This configuration property sets the date of manufacture for a device. The value is set by the device manufacturer.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*148*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTmanfDate*
Default:	*0/0/0 0:0:0*
Neuron C Type:	*SNVT_time_stamp*
Formats:	*SCPTmanfDate: text("%d/%d/%d %d:%d:%d", year, month, day, hour, minute, second)* *SCPTmanfDate#LO: text(date(year, month, day), (" "), time(hour, minute, second))*
Used by:	*SFPTaudibleFireIndicator SFPTfireSmokeDamperActuator SFPThvacValvePositioner SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator SFPTuniversalFireIndicator SFPTuniversalFireInitiator SFPTvisibleFireIndicator*

SCPTmanualAllowed

Overview:

Manual allowed (boolean) . Provides a clock, with a manual time input, the possibility to permit manual time updating

This configuration property enables manual updating of a clock with a manual time input. A True (1) value indicates that the manual time input will be used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*101*
Obsolete:	*no*

Size:	*1*
Programmatic Name:	*SCPTmanualAllowed*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTmanualAllowed: text("%d")*
Used by:	*SFPTlightingPanelController SFPTrealTimeKeeper*

SCPTmasterSlave

Overview:

Master-slave operation (boolean) . Used to select master clock or slave clock, non-zero indicates this is master clock

This configuration property provides sets the associated functional block as a master clock or slave clock. It is used when there are multiple clocks in a network. A True value indicates that this is the master clock for the network.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*97*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTmasterSlave*
Default:	*1*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTmasterSlave: text("%d")*
Used by:	*SFPTrealTimeKeeper*

SCPTmaxCameraPrepositions

Overview:

Maximum pre-positions (units) . The maximum number of pre-positions supported by a device

This configuration property sets the maximum number of prepositions supported in a camera telemetry receiver. It is manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*174*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTmaxCameraPrepositions*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTmaxCameraPrepositions: text("%d")*

SCPTmaxDefrostTime

Overview:

Maximum defrost time. Maximum time for defrost to run if terminated on temperature

This configuration property sets the maximum defrost time for defrost functional blocks configured to terminate on temperature. If terminate on time is selected, it indicates the defrost time. If terminate on temperature is selected and this timer expires, an alarm will be generated by the defrost functional block. This configuration property is identical to the SCPTmaxDefrstTime configuration property (without an “o”), except that it has units of minutes rather than seconds. This provides a higher range of time but lower resolution of time (65534 minutes compared to 109 minutes).

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*

Index:	*221*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxDefrostTime*
Default:	*0*
Neuron C Type:	*SNVT_time_min*
Formats:	*SCPTmaxDefrostTime: text("%d")*

SCPTmaxDefrstTemp

Overview:

Defrost stop temperature. The temperature at which to terminate defrost for objects set to terminate on temperature

This configuration property set the temperature at which to terminate defrost for functional blocks configured to terminate on temperature. If the functional block is set to terminate on time and this temperature is exceeded then an alarm is generated.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*110*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxDefrstTemp*
Minimum:	*-100,00*
Maximum:	*150,00*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTmaxDefrstTemp#SI: text("%f", *1+0(0:854))* *SCPTmaxDefrstTemp#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerDefrost*

SCPTmaxDefrstTime

Overview:

Maximum defrost time. The maximum defrost time for defrost objects set to terminate on temperature

This configuration property sets the maximum defrost time for defrost functional blocks configured to terminate on temperature. If terminate on time is selected, it indicates the defrost time. If terminate on temperature is selected and this timer expires, an alarm will be generated by the defrost functional block. This configuration property is identical to the SCPTmaxDefrostTime configuration property (with an “o”), except that it has units of seconds rather than minutes. This provides a higher resolution of time but lower range of time (109 minutes compared to 65534 minutes).

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*107*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxDefrstTime*
Default:	*3600,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTmaxDefrstTime: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerDefrost*

SCPTmaxDischargeAirCoolingSetpoint

Overview:

Maximum discharge air cooling. Setpoint for the maximum discharge air cooling

This configuration property sets the maximum discharge-air cooling setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*205*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxDischargeAirCoolingSetpoint*
Default:	*20,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTmaxDischargeAirCoolingSetpoint#SI: text("%f", *1+0(0:854))* *SCPTmaxDischargeAirCoolingSetpoint#US: text("%f",*

	**1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTmaxDischargeAirHeatingSetpoint

Overview:

Maximum discharge air heating. Setpoint for the maximum discharge air heating

This configuration property sets the maximum discharge-air heating setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*207*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxDischargeAirHeatingSetpoint*
Default:	*50,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTmaxDischargeAirHeatingSetpoint#SI: text("%f", *1+0(0:854))* *SCPTmaxDischargeAirHeatingSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTmaxDuctStaticPressureSetpoint

Overview:

Maximum duct static pressure. Setpoint for maximum duct static pressure

This configuration property sets the maximum duct static-pressure setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*190*
Obsolete:	*no*
Size:	*2*

Programmatic Name:	*SCPTmaxDuctStaticPressureSetpoint*
Default:	*1000*
Neuron C Type:	*SNVT_press_p*
Formats:	*SCPTmaxDuctStaticPressureSetpoint#SI: text("%f")* *SCPTmaxDuctStaticPressureSetpoint#US: text("%f", *0.0040217+0(0:954))* *SCPTmaxDuctStaticPressureSetpoint#US_psi: text("%f", *1.4504e-4+0(0:875))*
Used by:	*SFPTdischargeAirController*

SCPTmaxFanIn

Overview:

Maximum fan-in. (data sources) . Specifies the maximum number of data sources that may be connected to a network variable. The functional block determines data sources by examining the source address of each update.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*332*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxFanIn*
Default:	*0*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *65534* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTmaxFanIn: text("%d")*
Used by:	*SFPTdataLogger*

SCPTmaxFlow

Overview:

Maximum flow. The maximum flow

This configuration property sets the maximum flow through a device such as a VAV box.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*51*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxFlow*
Default:	*65535*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTmaxFlow#SI: text("%d")* *SCPTmaxFlow#US: text("%d", *0.26418+0(0:837))* *SCPTmaxFlow#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTvariableAirVolume*

SCPTmaxFlowHeat

Overview:

Maximum heating airflow. The maximum airflow setpoint of a VAV terminal while heating

This configuration property sets the maximum airflow setpoint of a device such as a VAV terminal while heating. The value of the heating maximum-flow setpoint must be validated against the value of the heating minimum-flow setpoint as follows:

0<= heating minimum flow setpoint <=heating maximum flow setpoint

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*37*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxFlowHeat*

Default:	*65535*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTmaxFlowHeat#SI: text("%d")* *SCPTmaxFlowHeat#US: text("%d", *0.26418+0(0:837))* *SCPTmaxFlowHeat#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTmaxFlowSetpoint

Overview:

Maximum flow. Setpoint for the operational high flow limit

This configuration property sets the high-flow limit for the working area of a pump.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*237*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxFlowSetpoint*
Default:	*10,00*
Neuron C Type:	*SNVT_flow_p*
Formats:	*SCPTmaxFlowSetpoint#SI: text("%f", *1+0(0:1408))* *SCPTmaxFlowSetpoint#US: text("%f", *0.588578+0(0:1405))*
Used by:	*SFPTpumpController*

SCPTmaxFlowUnit

Overview:

Unit maximum air flow. Unit maximum airflow for dual duct VAV Terminal units

Unit maximum airflow for dual-duct VAV Terminal units.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*262*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxFlowUnit*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTmaxFlowUnit#SI: text("%d")* *SCPTmaxFlowUnit#US: text("%d", *0.26418+0(0:837))* *SCPTmaxFlowUnit#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTmaxLevelVolt

Overview:

Maximum Dim Voltage.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*349*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxLevelVolt*
Default:	*0,0*
Neuron C Type:	*SNVT_volt*
Formats:	*SCPTmaxLevelVolt: text("%f")*
Used by:	*SFPToutdoorLuminairController*

SCPTmaxNVLength

Overview:

Maximum network variable length (bytes) . Maximum length of a type that may be assigned to the network variable

This configuration property specifies the maximum length of a type that may be assigned to a network variable that supports changeable types. It is used with SCPTnvType.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*255*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTmaxNVLength*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *1* Maximum: *31* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTmaxNVLength: text("%d")*
Used by:	*SFPTdataLogger SFPTscheduler*

SCPTmaxOut

Overview:

Maximum output value. The maximum value limit of the associated output network variable

This configuration property sets maximum value limit of the associated output network variable. A value of 0% indicates there is no maximum value.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*93*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTmaxOut*

Default:	*100,0*
Neuron C Type:	*SNVT_lev_cont*
Formats:	*SCPTmaxOut: text("%f")*
Used by:	*SFPTswitch*

SCPTmaxPower

Overview:

Maximum power.. Power level at which the sunblind actuator detects a blocked motor and switches off automatically.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*317*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxPower*
Default:	*0,0*
Neuron C Type:	*SNVT_power*
Formats:	*SCPTmaxPower: text("%f")*
Used by:	*SFPTisiSunblindActuator*

SCPTmaxPressureSetpoint

Overview:

Maximum pressure. Setpoint for the operational high pressure limit

This configuration property sets the high-pressure limit for the working area of a pump.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*235*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxPressureSetpoint*
Default:	*0,0*

Neuron C Type:	*SNVT_press*
Formats:	*SCPTmaxPressureSetpoint#SI: text("%f")* *SCPTmaxPressureSetpoint#US: text("%f", *0.2953+0(0:1402))*
Used by:	*SFPTpumpController*

SCPTmaxPrivacyZones

Overview:

Maximum privacy zones (units) . The maximum number of privacy zones supported by a device

This sets the maximum number of privacy zones supported by a camera telemetry receiver. The value is manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*173*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTmaxPrivacyZones*
Default:	*1*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTmaxPrivacyZones: text("%d")*

SCPTmaxRcvT

Overview:

Maximum receive time. The maximum time elapsed after the last update before the actuator adopts the default output

This configuration property sets the maximum time that elapses after an update to an input network variable of an actuator object before the actuator adopts the default output.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*21*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTmaxRcvT*
Maximum:	*0 17:59:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTmaxRcvT: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTmaxRcvT#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopActuator SFPTopenLoopActuator*

SCPTmaxRcvTime

Overview:

Maximum receive time. The maximum period of time that may expire with no updates on the associated input network variables before the object goes into heartbeat failure mode. A zero value disables

This configuration property sets the maximum time that elapses after an update to a bound network input. A value of zero (0) disable the receive failure detect mechanism.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*48*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxRcvTime*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTmaxRcvTime: text("%f")*
Used by:	*SFPTaudibleFireIndicator SFPTboilerController SFPTchilledCeilingController SFPTchiller SFPTdamperActuator*

SFPTdischargeAirController SFPTelevatorArrivalGong SFPTelevatorDirectionLantern SFPTelevatorFireSystemsPort SFPTelevatorHallLantern SFPTelevatorPositionIndicator SFPTentryExit SFPTfanCoilUnit SFPTfireSmokeDamperActuator SFPThardwiredFireAlarmShutdown SFPThardwiredFullVentilation SFPThardwiredGasDetectionShutdown SFPThardwiredRecirculation SFPThardwiredSafetyInstrumentedSystemSIS SFPTheatPump SFPThvacValvePositioner SFPTpumpController SFPTroofTopUnit SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTschedulerSimple SFPTspaceComfortController SFPTsunblindController SFPTthermostat SFPTunitHeater SFPTunitVentilatorController SFPTuniversalFireIndicator SFPTvariableAirVolume SFPTvariableSpeedMotorDrive SFPTvisibleFireIndicator SFPTwallUnit

SCPTmaxRemoteFlowSetpoint

Overview:

Maximum remote flow. Setpoint for the operational high flow limit (remote sensor)

This configuration property sets the high-flow limit for a remote sensor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*242*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxRemoteFlowSetpoint*
Default:	*0,00*
Neuron C Type:	*SNVT_flow_p*
Formats:	*SCPTmaxRemoteFlowSetpoint#SI: text("%f", *1+0(0:1408))* *SCPTmaxRemoteFlowSetpoint#US: text("%f", *0.588578+0(0:1405))*
Used by:	*SFPTpumpController*

SCPTmaxRemotePressureSetpoint

Overview:

Maximum remote pressure. Setpoint for the operational high pressure limit (remote sensor)

This configuration property sets the high-pressure limit for a remote sensor. This value replaces the manufacturer-defined setpoint limit when the remote sensor is used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*240*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxRemotePressureSetpoint*
Default:	*0,0*
Neuron C Type:	*SNVT_press*
Formats:	*SCPTmaxRemotePressureSetpoint#SI: text("%f")* *SCPTmaxRemotePressureSetpoint#US: text("%f", *0.2953+0(0:1402))*
Used by:	*SFPTpumpController*

SCPTmaxRemoteTempSetpoint

Overview:

Maximum remote temperature. Setpoint for the operational high temperature limit (remote sensor)

This configuration property sets the high-temperature limit for a remote sensor. This value replaces the Manufacturer-defined setpoint limit when the remote sensor is used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*244*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxRemoteTempSetpoint*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*

Formats:	*SCPTmaxRemoteTempSetpoint#SI: text("%f", *1+0(0:854))*
	*SCPTmaxRemoteTempSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTpumpController*

SCPTmaxReturnExhaustFanCapacity

Overview:

Maximum return/exhaust fan capacity. Setpoint for maximum return/exhaust fan capacity

This configuration property sets the maximum return/exhaust fan capacity setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*187*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxReturnExhaustFanCapacity*
Default:	*100,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTmaxReturnExhaustFanCapacity: text("%f")*
Used by:	*SFPTdischargeAirController*

SCPTmaxRnge

Overview:

.

This configuration sets the maximum value limit of a network variable. The data type is the same as the network variable. The value must be greater than any specified Minimum Range configuration property.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*20*
Obsolete:	*no*

Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTmaxRnge: text("%d")*
Used by:	*SFPTclosedLoopSensor SFPTdamperActuator SFPTopenLoopSensor*

SCPTmaxSendTime

Overview:

Maximum send time. The maximum period of time between consecutive transmissions of the current value

This configuration property sets the maximum period of time before the associated output network variables are automatically updated.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*49*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxSendTime*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTmaxSendTime: text("%f")*
Used by:	*SFPTairVelocitySensor SFPTaudibleFireIndicator SFPTautomaticTransferSwitch SFPTboilerController SFPTchannelContinuityMonitor SFPTchilledCeilingController SFPTchiller SFPTclothesWasherDomestic SFPTco2Sensor SFPTconstantLightController SFPTdamperActuator SFPTdataLogger SFPTdeviceMonitor SFPTdischargeAirController SFPTelevatorFireSystemsPort SFPTentryExit SFPTfanCoilUnit SFPTfireSmokeDamperActuator SFPTfrostSensor SFPTgeneratorSet SFPTglobalSolarRadiation SFPThardwiredFireAlarmShutdown SFPThardwiredFullVentilation SFPThardwiredGasDetectionShutdown SFPThardwiredRecirculation SFPThardwiredSafetyInstrumentedSystemSIS SFPTheatPump SFPThvacRelativeHumiditySensor SFPThvacTempSensor SFPThvacValvePositioner SFPTidentifierSensor SFPTisiKeypad*

SFPTisiLampActuator SFPTisiOccupancySensor
SFPTisiSunblindActuator SFPTlightSensor SFPToccupancySensor
SFPToutdoorLuminairController SFPTpressureSensor
SFPTpullStationFireInitiator SFPTpumpController
SFPTrailcarAudioController SFPTrainSensor
SFPTrefrigDisplayCaseControllerDefrost
SFPTrefrigDisplayCaseControllerEvaporator
SFPTrefrigDisplayCaseControllerThermostat SFPTroofTopUnit
SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule
SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator
SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator
SFPTsccVAV SFPTscheduler SFPTschedulerSimple
SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli
SFPTspaceComfortController SFPTsunblindActuator
SFPTsunblindController SFPTswitch SFPTthermalFireInitiator
SFPTthermostat SFPTunitHeater SFPTunitVentilatorController
SFPTuniversalFireInitiator SFPTvariableAirVolume
SFPTvariableSpeedMotorDrive SFPTvisibleFireIndicator
SFPTwallUnit

SCPTmaxSetpoint

Overview:

Maximum setpoint. Either the maximum angle of rotation for an actuator or the maximum airflow for an actuator depending on actuator category

This configuration property sets the maximum angle of rotation or the maximum fluid flow for an actuator. Its value has to be greater than or equal to the SCPTminSetpoint value, if any.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*50*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxSetpoint*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTmaxSetpoint: text("%f")*
Used by:	*SFPThvacValvePositioner SFPTvariableSpeedMotorDrive*

SCPTmaxSndT

Overview:

Maximum send time. The maximum period of time between consecutive transmissions of the current value

This configuration property sets the maximum period of time that expires before the functional block automatically transmits the current value of the associated output network variable. This provides a heartbeat output that can be used by destination objects to ensure that the device is still healthy.

When used with the node object, the maximum send time is used for the nvoStatus output network variable, and the status of each object on the device (including the node object) is returned sequentially in round-robin fashion, one object status per expiration of the timer.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*22*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTmaxSndT*
Maximum:	*0 17:59:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTmaxSndT: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTmaxSndT#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopSensor SFPTnodeObject SFPTopenLoopSensor*

SCPTmaxStroke

Overview:

Maximum stroke. The maximum stroke limit

This configuration property sets the maximum stroke.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*253*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxStroke*
Default:	*0,0*
Neuron C Type:	*SNVT_length_mil*
Formats:	*SCPTmaxStroke#SI: text("%f")* *SCPTmaxStroke#US: text("%f", *0.03937+0(0:844))*
Used by:	*SFPThvacValvePositioner*

SCPTmaxSupplyFanCapacity

Overview:

Maximum supply fan capacity. Setpoint for maximum supply fan capacity

This configuration property sets the maximum supply fan capacity setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*185*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmaxSupplyFanCapacity*
Default:	*100,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTmaxSupplyFanCapacity: text("%f")*
Used by:	*SFPTdischargeAirController*

SCPTmeasurementInterval

Overview:

Measurement interval (seconds) . Time period used for a measurement

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*360*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmeasurementInterval*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTmeasurementInterval: text("%f")*
Used by:	*SFPTisiLampActuator*

SCPTminDefrostTime

Overview:

Minimum defrost time. Minimum time for defrost to run if terminated on temperature

This configuration property sets the minimum defrost time for defrosts. If the defrost terminates on temperature before this time expires, then the defrost will cycle on and off about the termination temperature until the timer expires.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*220*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDefrostTime*
Default:	*0*
Neuron C Type:	*SNVT_time_min*
Formats:	*SCPTminDefrostTime: text("%d")*

SCPTminDeltaAngl

Overview:

Damper angle send on delta. The minimum change in damper actuator angle required to be treated as significant

This configuration property sets the minimum delta on a damper actuator angle required to send an update of the damper angle.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*43*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDeltaAngl*
Minimum:	*0,00*
Default:	*0,00*
Neuron C Type:	*SNVT_angle_deg*
Formats:	*SCPTminDeltaAngl: text("%f")*
Used by:	*SFPTdamperActuator*

SCPTminDeltaCO2

Overview:

Minimum delta CO2 level. The minimum change in CO2 level required to be treated as significant

This configuration property sets the minimum carbon dioxide (CO_2) level change required to update the CO_2 level output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*63*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDeltaCO2*
Default:	*10*
Neuron C Type:	*SNVT_ppm*
Formats:	*SCPTminDeltaCO2: text("%d")*
Used by:	*SFPTco2Sensor SFPTsccCommandModule*

SCPTminDeltaFlow

Overview:

Flow send on delta. The minimum change in airflow required to be treated as significant

This configuration property sets the minimum delta on the air flow to update the output variable for air flow.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*47*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDeltaFlow*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminDeltaFlow#SI: text("%d")* *SCPTminDeltaFlow#US: text("%d", *0.26418+0(0:837))* *SCPTminDeltaFlow#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTdamperActuator*

SCPTminDeltaLevel

Overview:

Send on delta. The minimum change required to force transmission of the output value

This configuration property sets the amount by which a value must change before the associated output network variable is updated.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*88*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTminDeltaLevel*
Default:	*0,0*
Neuron C Type:	*SNVT_lev_cont*

Formats:	*SCPTminDeltaLevel: text("%f")*
Used by:	*SFPTconstantLightController SFPTlightSensor*

SCPTminDeltaRH

Overview:

Minimum delta relative humidity. The minimum change in RH level required to be treated as significant

This configuration property sets the minimum relative humidity change required before the associated output network variable is updated.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*62*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDeltaRH*
Minimum:	*0,000*
Default:	*1,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTminDeltaRH: text("%f")*
Used by:	*SFPThvacRelativeHumiditySensor SFPTsccCommandModule*

SCPTminDeltaTemp

Overview:

Minimum delta temperature. The minimum change in temperature required to be treated as significant

This configuration property sets the minimum temperature change required before the associated output network variable is updated.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*64*

Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDeltaTemp*
Minimum:	*0,00*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTminDeltaTemp#SI: text("%f", *1+0(0:854))* *SCPTminDeltaTemp#US: text("%f", *1.8+0(0:855))*
Used by:	*SFPTdamperActuator SFPThvacTempSensor SFPTsccCommandModule SFPTthermostat*

SCPTminDischargeAirCoolingSetpoint

Overview:

Minimum discharge air cooling. Setpoint for the minimum discharge air cooling

This configuration property sets the minimum discharge-air cooling setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*206*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDischargeAirCoolingSetpoint*
Default:	*10,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTminDischargeAirCoolingSetpoint#SI: text("%f", *1+0(0:854))* *SCPTminDischargeAirCoolingSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTminDischargeAirHeatingSetpoint

Overview:

Minimum discharge air heating. Setpoint for the minimum discharge air heating

This configuration property sets the minimum discharge-air heating setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*208*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDischargeAirHeatingSetpoint*
Default:	*30,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTminDischargeAirHeatingSetpoint#SI: text("%f", *1+0(0:854))* *SCPTminDischargeAirHeatingSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTminDuctStaticPressureSetpoint

Overview:

Minimum duct static pressure. Setpoint for minimum duct static pressure

This configuration property sets the minimum duct static-pressure setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*191*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminDuctStaticPressureSetpoint*
Default:	*100*
Neuron C Type:	*SNVT_press_p*
Formats:	*SCPTminDuctStaticPressureSetpoint#SI: text("%f")* *SCPTminDuctStaticPressureSetpoint#US: text("%f", *0.0040217+0(0:954))* *SCPTminDuctStaticPressureSetpoint#US_psi: text("%f", *1.4504e-*

	4+0(0:875))
Used by:	*SFPTdischargeAirController*

SCPTminFlow

Overview:

Minimum flow. The minimum flow

This configuration property sets the minimum flow through a device such as a VAV box.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*54*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminFlow*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminFlow#SI: text("%d")*
	*SCPTminFlow#US: text("%d", *0.26418+0(0:837))*
	*SCPTminFlow#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTvariableAirVolume*

SCPTminFlowHeat

Overview:

Minimum heating airflow. The minimum airflow setpoint of a VAV terminal while heating

This configuration property sets the minimum flow a VAV controller will control to when reheat is utilized.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*55*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminFlowHeat*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminFlowHeat#SI: text("%d")* *SCPTminFlowHeat#US: text("%d", *0.26418+0(0:837))* *SCPTminFlowHeat#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTvariableAirVolume*

SCPTminFlowHeatStby

Overview:

Standby heating minimum air flow. Heating or ventilated deck minimum flow of a dual duct VAV Terminal unit during occupied standby mode

Heating or ventilated deck minimum flow of a dual-duct VAV Terminal unit during occupied standby mode.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*263*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminFlowHeatStby*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminFlowHeatStby#SI: text("%d")* *SCPTminFlowHeatStby#US: text("%d", *0.26418+0(0:837))* *SCPTminFlowHeatStby#US_cfm: text("%d", *2.1189+0(0:966))*

Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTminFlowSetpoint

Overview:

Minimum flow. Setpoint for the operational low flow limit

This configuration property sets the low-flow limit for the working area of a pump.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*236*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminFlowSetpoint*
Default:	*0,00*
Neuron C Type:	*SNVT_flow_p*
Formats:	*SCPTminFlowSetpoint#SI: text("%f", *1+0(0:1408))* *SCPTminFlowSetpoint#US: text("%f", *0.588578+0(0:1405))*
Used by:	*SFPTpumpController*

SCPTminFlowStby

Overview:

Minimum flow for standby. The minimum flow through the VAV box in standby mode

This configuration property sets the minimum flow through a device, such as VAV box, in standby mode.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*56*
Obsolete:	*no*

Size:	*2*
Programmatic Name:	*SCPTminFlowStby*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminFlowStby#SI: text("%d")* *SCPTminFlowStby#US: text("%d", *0.26418+0(0:837))* *SCPTminFlowStby#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTvariableAirVolume*

SCPTminFlowUnit

Overview:

Unit minimum air flow. Unit minimum air flow for dual duct VAV Terminal units

Unit minimum airflow for dual-duct VAV Terminal units.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*261*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminFlowUnit*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminFlowUnit#SI: text("%d")* *SCPTminFlowUnit#US: text("%d", *0.26418+0(0:837))* *SCPTminFlowUnit#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTminFlowUnitHeat

Overview:

Unit Heating Minimum Flow. Minimum airflow setpoint of a single duct, or the unit minimum airflow setpoint of a dual duct VAV terminal when using a unit (local) heating source

Minimum airflow setpoint of a single duct, or the unit minimum-airflow setpoint of a dual-duct VAV terminal when using a unit (local) heating source.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*270*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminFlowUnitHeat*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminFlowUnitHeat#SI: text("%d")* *SCPTminFlowUnitHeat#US: text("%d", *0.26418+0(0:837))* *SCPTminFlowUnitHeat#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTminFlowUnitStby

Overview:

Standby unit minimum air flow. Total unit minimum airflow for dual duct units during occupied standby mode

Total unit minimum airflow for dual-duct units during occupied standby mode.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*264*

Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminFlowUnitStby*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminFlowUnitStby#SI: text("%d")* *SCPTminFlowUnitStby#US: text("%d", *0.26418+0(0:837))* *SCPTminFlowUnitStby#US_cfm: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTminOutdoorAirFlowSetpoint

Overview:

Minimum outdoor air flow. Setpoint for the default minimum outdoor air flow

This configuration property sets the default minimum outdoor-airflow setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*198*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminOutdoorAirFlowSetpoint*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTminOutdoorAirFlowSetpoint#SI: text("%d")* *SCPTminOutdoorAirFlowSetpoint#US: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTdischargeAirController*

SCPTminPressureSetpoint

Overview:

Minimum pressure. Setpoint for the operational low pressure limit

This configuration property sets the low-pressure limit for the working area of a pump.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*234*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminPressureSetpoint*
Default:	*0,0*
Neuron C Type:	*SNVT_press*
Formats:	*SCPTminPressureSetpoint#SI: text("%f")* *SCPTminPressureSetpoint#US: text("%f", *0.2953+0(0:1402))*
Used by:	*SFPTpumpController*

SCPTminRemoteFlowSetpoint

Overview:

Minimum remote flow. Setpoint for the operational low flow limit (remote sensor)

This configuration property sets the low-flow limit for a remote sensor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*241*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminRemoteFlowSetpoint*
Default:	*0,00*
Neuron C Type:	*SNVT_flow_p*
Formats:	*SCPTminRemoteFlowSetpoint#SI: text("%f", *1+0(0:1408))* *SCPTminRemoteFlowSetpoint#US: text("%f", *0.588578+0(0:1405))*
Used by:	*SFPTpumpController*

SCPTminRemotePressureSetpoint

Overview:

Minimum remote pressure. Setpoint for the operational low pressure limit (remote sensor)

This configuration property sets the low-pressure limit for a remote sensor. This value replaces the manufacturer-defined setpoint limit when the remote sensor is used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*239*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminRemotePressureSetpoint*
Default:	*0,0*
Neuron C Type:	*SNVT_press*
Formats:	*SCPTminRemotePressureSetpoint#SI: text("%f")* *SCPTminRemotePressureSetpoint#US: text("%f", *0.2953+0(0:1402))*
Used by:	*SFPTpumpController*

SCPTminRemoteTempSetpoint

Overview:

Minimum remote temperature. Setpoint for the operational low temperature limit (remote sensor)

This configuration property sets the low-temperature limit for a remote sensor. This value replaces the manufacturer-defined setpoint limit when the remote sensor is used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*243*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminRemoteTempSetpoint*
Default:	*0,00*

Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTminRemoteTempSetpoint#SI: text("%f", *1+0(0:854))* *SCPTminRemoteTempSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTpumpController*

SCPTminReturnExhaustFanCapacity

Overview:

Minimum return/exhaust fan capacity. Setpoint for minimum return/exhaust fan capacity

This configuration property sets the minimum return/exhaust fan capacity setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*188*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminReturnExhaustFanCapacity*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTminReturnExhaustFanCapacity: text("%f")*
Used by:	*SFPTdischargeAirController*

SCPTminRnge

Overview:

.

This configuration property sets the minimum value limit of an output network variable. The data type is the same as the value field of the output network variable. The value must be less than any specified Maximum Range configuration property.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*23*

Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTminRnge: text("%d")*
Used by:	*SFPTclosedLoopSensor SFPTdamperActuator SFPTdischargeAirController SFPTopenLoopSensor SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTunitVentilatorController*

SCPTminSendTime

Overview:

Minimum send time. The minimum period of time between consecutive transmissions of the current value

This configuration property sets the minimum period of time between output network variable transitions.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*52*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminSendTime*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTminSendTime: text("%f")*
Used by:	*SFPTairVelocitySensor SFPTboilerController SFPTcalendar SFPTchannelContinuityMonitor SFPTchilledCeilingController SFPTchiller SFPTco2Sensor SFPTconstantLightController SFPTdamperActuator SFPTdischargeAirController SFPTentryExit SFPTfanCoilUnit SFPTfrostSensor SFPTglobalSolarRadiation SFPThvacRelativeHumiditySensor SFPThvacTempSensor SFPThvacValvePositioner SFPTisiKeypad SFPTisiLampActuator SFPTisiSunblindActuator SFPTlightSensor SFPToutdoorLuminairController SFPTpressureSensor SFPTpumpController SFPTrailcarAudioController*

SFPTrainSensor SFPTrefrigDisplayCaseControllerDefrost SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTsceneController SFPTscenePanel SFPTschedulerSimple SFPTspaceComfortController SFPTswitch SFPTunitHeater SFPTunitVentilatorController SFPTvariableAirVolume SFPTvariableSpeedMotorDrive SFPTwallUnit

SCPTminSetpoint

Overview:

Minimum setpoint. The minimum setpoint, such as minimum angle of rotation or minimum air flow

This configuration property sets the minimum angle of rotation or the minimum fluid flow for an actuator. This value must be less than or equal to the SCPTmaxSetpoint value.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*53*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminSetpoint*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTminSetpoint: text("%f")*
Used by:	*SFPThvacValvePositioner SFPToutdoorLuminairController SFPTvariableSpeedMotorDrive*

SCPTminSndT

Overview:

Minimum send time. The minimum period of time between consecutive transmissions of the current value

This configuration property sets the minimum period between output network variable updates (maximum update rate). It provides a way to tailor the output network variable update rate to

available bandwidth.

Update rate limiting may be disabled by setting all fields to zero.

Default Value

If this configuration property is present the default value is set according to the bit rate of the transmission medium as follows:

Bit Rate	Default Minimum Time
2kbps	60 seconds
4kbps	60 seconds
10kbps	30 seconds
39kbps	15 seconds
78kbps	15 seconds
1.25Mbps	1 seconds

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*24*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTminSndT*
Maximum:	*0 17:59:59:999*
Default:	*0 0:0:15:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTminSndT: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTminSndT#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopSensor SFPTopenLoopSensor*

SCPTminStroke

Overview:

Minimum stroke. The minimum stroke limit

This configuration property sets the minimum stroke.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*252*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminStroke*
Default:	*0,0*
Neuron C Type:	*SNVT_length_mil*
Formats:	*SCPTminStroke#SI: text("%f")* *SCPTminStroke#US: text("%f", *0.03937+0(0:844))*
Used by:	*SFPThvacValvePositioner*

SCPTminSupplyFanCapacity

Overview:

Minimum supply fan capacity. Setpoint for minimum supply fan capacity

This configuration property sets the minimum supply fan capacity setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*186*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTminSupplyFanCapacity*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTminSupplyFanCapacity: text("%f")*
Used by:	*SFPTdischargeAirController*

SCPTmixedAirLowLimitSetpoint

Overview:

Mixed air low limit. Setpoint for the mixed air low temperature limit

This configuration property sets the mixed air low-limit setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*196*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmixedAirLowLimitSetpoint*
Default:	*7,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTmixedAirLowLimitSetpoint#SI: text("%f", *1+0(0:854))* *SCPTmixedAirLowLimitSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTmixedAirTempSetpoint

Overview:

Mixed air temperature. Setpoint for the default mixed air temperature

This configuration property sets the default mixed air temperature setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*197*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmixedAirTempSetpoint*
Default:	*11,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTmixedAirTempSetpoint#SI: text("%f", *1+0(0:854))* *SCPTmixedAirTempSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTmodeHrtBt

Overview:

Heartbeat, mode output. The time that must pass without an update for mode definitions to be automatically retransmitted, zero disables

This configuration property sets the time that must pass without an update for mode definitions to be automatically retransmitted. A value of 0 indicates that there is no heartbeat.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*105*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTmodeHrtBt*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTmodeHrtBt: text("%f")*
Used by:	*SFPTrealTimeBasedScheduler*

SCPTmonInterval

Overview:

Monitor Interval.. This configuration property defines the interval over which statistics are collected and averages are calculated

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*319*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTmonInterval*
Minimum:	*0 0:0:1:0*
Maximum:	*0 17:59:59:999*
Default:	*0 0:0:5:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTmonInterval: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)*

SCPTmonInterval#LO: text(("%d ", day), time(hour, minute, second, millisecond))

Used by: *SFPTchannelMonitor*

SCPTname1

Overview:

Name part 1. Part 1 of the name of the functional block to be used by optional user interface applications. May optionally used with SCPTname2 and SCPTname3. Must be implemented as a configuration network variable.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *306*

Obsolete: *no*

Size: *13*

Programmatic Name: *SCPTname1*

Default: *CE_UTF_8*

Neuron C Type:

Structure	
encoding	Character encoding.. Character encoding method
	char_encoding_t
name	.
	unsigned char [12] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTname1: text(("%m ", encoding),((encoding == 0) ? ("%s", name) :("%x %x %x %x %x %x %x %x %x %x %x %x", name[0], name[1], name[2], name[3], name[4], name[5], name[6], name[7], name[8], name[9], name[10], name[11])))

Used by: *SFPTisiKeypad SFPTisiLampActuator SFPTisiOccupancySensor SFPTisiSunblindActuator*

SCPTname2

Overview:

Name part 2. Part 2 of the name of the functional block to be used by optional user interface applications. Must be used with SCPTname1 and may optionally be used with SCPTname3. This part is concatenated after part 1, and may optionally be followed by part 3. Must be implemented as a configuration network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*309*
Obsolete:	*no*
Size:	*12*
Programmatic Name:	*SCPTname2*
Default:	*00*12*
Neuron C Type:	Structure

name	.
	unsigned char [12] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTname2: text("%c %c %c %c %c %c %c %c %c %c %c %c", name[0], name[1], name[2], name[3], name[4], name[5], name[6], name[7], name[8], name[9], name[10], name[11])

Used by:

SFPTisiKeypad SFPTisiLampActuator SFPTisiOccupancySensor SFPTisiSunblindActuator

SCPTname3

Overview:

Name part 3.. Part 3 of the name of the functional block to be used by optional user interface applications. Must be used with SCPTname1 and SCPTname2. This part, if present, is concatenated with parts 1 and 2. Must be implemented as a configuration network variable.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *310*

Obsolete: *no*

Size: *12*

Programmatic Name: *SCPTname3*

Default: *00*12*

Neuron C Type:

Structure	
name	.
	unsigned char [12] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTname3: text("%c %c %c %c %c %c %c %c %c %c %c %c", name[0], name[1], name[2], name[3], name[4], name[5], name[6], name[7], name[8], name[9], name[10], name[11])

Used by:

SFPTisiKeypad SFPTisiLampActuator SFPTisiOccupancySensor SFPTisiSunblindActuator

SCPTneuronId

Overview:

Neuron Identifier. A unique 6-byte identifier for a CEA-709.1 / EN14908 device

A unique 6-byte identifier for a LON device (numbers administered through the US Consumer Electronic Association (CEA).

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *301*

Obsolete: *no*

Size: *6*

Programmatic Name: *SCPTneuronId*

Default: *0-0-0-0-0-0*

Neuron C Type:

Structure	
id	.
	unsigned char [6] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTneuronId: text("%x-%x-%x-%x-%x-%x", id[0], id[1], id[2], id[3], id[4], id[5])

SCPTnightPurgePosition

Overview:

Night purge valve position. Valve position in percent open for night purge

This configuration property defines the valve position for the Night Purge HVAC mode.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*246*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnightPurgePosition*
Minimum:	*0,000*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTnightPurgePosition: text("%f")*
Used by:	*SFPThvacValvePositioner*

SCPTnomAirFlow

Overview:

Nominal air flow. Value used in calculating the air flow in an airflow control actuator

This configuration property sets the nominal airflow through a duct. It is used to calculate the airflow through the duct and is typically used by airflow control actuators. Because improper usage may cause a non-functional device, the device manufacturer can disable write access.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*57*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnomAirFlow*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTnomAirFlow#SI: text("%d")* *SCPTnomAirFlow#US: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTdamperActuator SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTvariableAirVolume*

SCPTnomAirFlowHeat

Overview:

Heating nominal flow. Nominal airflow volume of a hot or ventilation deck of a dual duct VAV terminal

Nominal airflow volume of a hot or ventilation deck of a dual duct VAV terminal.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*267*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnomAirFlowHeat*
Default:	*0*
Neuron C Type:	*SNVT_flow*
Formats:	*SCPTnomAirFlowHeat#SI: text("%d")*

	*SCPTnomAirFlowHeat#US: text("%d", *2.1189+0(0:966))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTnomAngle

Overview:

Nominal angle. The nominal angle for an actuator

This configuration property sets the nominal angle for an actuator. Because improper usage may cause a non-functional device, the device manufacturer can disable write access. Write access may also cause an automatic self-test to get the mechanical working range of the actuator.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*58*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnomAngle*
Minimum:	*0,00*
Maximum:	*95,00*
Default:	*90,00*
Neuron C Type:	*SNVT_angle_deg*
Formats:	*SCPTnomAngle: text("%f")*
Used by:	*SFPTdamperActuator*

SCPTnomFreq

Overview:

Nominal motor frequency. The nominal frequency of a motor

This configuration property sets the nominal frequency of a motor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*159*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnomFreq*
Minimum:	*0,0*
Maximum:	*100,0*
Default:	*60,0*
Neuron C Type:	*SNVT_freq_hz*
Formats:	*SCPTnomFreq: text("%f")*
Used by:	*SFPTvariableSpeedMotorDrive*

SCPTnomRPM

Overview:

Nominal motor speed. The nominal speed of a motor in RPM

This configuration property sets the nominal speed of a motor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*158*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnomRPM*
Default:	*1800*
Neuron C Type:	*SNVT_rpm*
Formats:	*SCPTnomRPM: text("%f")*
Used by:	*SFPTvariableSpeedMotorDrive*

SCPTnormalRotationalSpeed

Overview:

Normal rotational speed. The normal rotational speed in Hz

This configuration property sets the normal rotational speed of a mechanical part.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*180*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnormalRotationalSpeed*
Default:	*0,0*
Neuron C Type:	*SNVT_freq_hz*
Formats:	*SCPTnormalRotationalSpeed: text("%f")*

SCPTnsdsFbIndex

Overview:

Index of Functional Block.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*358*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnsdsFbIndex*
Default:	*0*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTnsdsFbIndex: text("%d")*
Used by:	*SFPTstaticProgrammable*

SCPTnumDampers

Overview:

Number of dampers. Used to inform the controller if it is in a single or dual duct system

Calibration constant used to calculate airflow.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*269*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnumDampers*
Minimum:	*1*
Maximum:	*2*
Default:	*1*
Neuron C Type:	*SNVT_count*
Formats:	*SCPTnumDampers: text("%d")*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTnumDigits

Overview:

Number of Digits on the Meter. This configuration property is used for setting the total number of digits on the meter

This configuration property is used for setting the total number of digits on the meter.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*293*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnumDigits*
Default:	*0*
Neuron C Type:	*SNVT_count*

Formats:	*SCPTnumDigits: text("%d")*
Used by:	*SFPTutilityMeter*

SCPTnumValves

Overview:

Number of output valves. Used to inform the controller whether it is in a one-valve or two-valve system

This configuration property sets the number of pipes or valves in a device. It is manufacturer-defined. The value 1 implies one output valve (two-pipe system), and the value 2 implies two output valves (four-pipe system).

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*59*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTnumValves*
Minimum:	*1*
Maximum:	*2*
Default:	*2*
Neuron C Type:	*SNVT_count*
Formats:	*SCPTnumValves: text("%d")*
Used by:	*SFPTfanCoilUnit SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTunitVentilatorController*

SCPTnvDynamicAssignment

Overview:

Network variable dynamic assignment. Assigns a dynamic network variable to a functional block member

This configuration property specifies an assignment of a dynamic network variable to a

functional block on the device containing the dynamic network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*256*
Obsolete:	*no*
Size:	*25*
Programmatic Name:	*SCPTnvDynamicAssignment*
Default:	*NV 65535 = FBlock 65535, Member 65535 PID 0:0:0:0:0:0:0:0, Scope 0, Index 1, NVT_CAT_INITIAL, 1 bytes, A=0, B=0, C=0*

Neuron C Type:

Structure	
nv_index	Network variable index (nv index) . Network variable index within the device
	unsigned long Minimum: *0* Maximum: *4095* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
fblock_index	Functional block index (fblock index) . Index of the functional block to which the network variable is assigned
	unsigned long Minimum: *0* Maximum: *4095* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
member_number	Member number (nv member number) . Member number of the functional block network variable member to which the network variable is assigned
	unsigned long Minimum: *1* Maximum: *4095* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
nv_type	.
	SNVT_nv_type

Formats:

SCPTnvDynamicAssignment: text("NV %d = FBlock %d, Member %d PID %x:%x:%x:%x:%x:%x:%x:%x, Scope %d, Index %d, %m, %d bytes, A=%d, B=%d, C=%d", nv_index, fblock_index, member_number, nv_type.type_program_ID[0], nv_type.type_program_ID[1], nv_type.type_program_ID[2], nv_type.type_program_ID[3], nv_type.type_program_ID[4], nv_type.type_program_ID[5], nv_type.type_program_ID[6],

nv_type.type_program_ID[7], nv_type.type_scope, nv_type.type_index, nv_type.type_category, nv_type.type_length, nv_type.scaling_factor_a, nv_type.scaling_factor_b, nv_type.scaling_factor_c)

SCPTnvPriority

Overview:

UNVT Flag. Defines whether the NV is of SNVT (=0) or UNVT (=1)

This configuration property is used to set the priorities of the inputs and defines whether the NV is of SNVT (=0) or UNVT (=1).

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*296*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTnvPriority*
Default:	*0 0*

Neuron C Type:

Structure	
user_flag	UNVT Flag. Defines whether the NV is of SNVT (=0) or UNVT (=1)
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
nv	NV Declaration Index. The member index from the profile
	bitfield Signed: *no* Width: *7* Offset: *1* Minimum: *0* Maximum: *127* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTnvPriority: text("%d %d", user_flag,nv)

Used by:

SFPTsunblindController

SCPTnvType

Overview:

Network variable type. Network variable type for network variables that support changeable types

This configuration property specifies the type of a network variable. A SCPTnvType configuration property specifies the network variable type for the network variable that it applies to. When a new value is specified for a SCPTnvType configuration property, the device application must validate that it supports the new setting, and report an error via the Node Object functional block if the new setting is not supported, or change the application processing required for the network variable based on the new type if the new type is supported. A SCPTmaxNVLength configuration property may be specified for a network

variable that supports changeable types to specify the maximum type length supported by the network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*254*
Obsolete:	*no*
Size:	*19*
Programmatic Name:	*SCPTnvType*
Default:	*PID 0:0:0:0:0:0:0:0, Scope 0, Index 1, NVT_CAT_INITIAL, 1 bytes, A=0, B=0, C=0*
Neuron C Type:	*SNVT_nv_type*
Formats:	*SCPTnvType: text("PID %x:%x:%x:%x:%x:%x:%x:%x, Scope %d, Index %d, %m, %d bytes, A=%d, B=%d, C=%d", type_program_ID[0], type_program_ID[1], type_program_ID[2], type_program_ID[3], type_program_ID[4], type_program_ID[5], type_program_ID[6], type_program_ID[7], type_scope, type_index, type_category, type_length, scaling_factor_a, scaling_factor_b, scaling_factor_c)*
Used by:	*SFPTdataLogger SFPTscheduler*

SCPTnvUsage

Overview:

NV usage. The SCPTnvUsage CPs shall be used to indicate whether the NVs are in use by the loaded program

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*364*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTnvUsage*
Default:	*in_use:0, mfg:0*

Neuron C Type:

Structure	
in_use	.
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
mfg	.
	bitfield Signed: *no* Width: *7* Offset: *1* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SCPTnvUsage: text("in_use:%d, mfg:%d", in_use,mfg)*

Used by: *SFPTstaticProgrammable*

SCPTnwrkCnfg

Overview:

Network configuration source. The value of this field determines the source of the node's network configuration.

The configuration property set the source for network configuration for a device. The source may be the device itself, called *self-installation*, or an external network tool. All devices that support self-installation must provide this configuration property to allow a network tool to take control of the device's network configuration.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index:	*25*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTnwrkCnfg*
Default:	*CFG_EXTERNAL*
Neuron C Type:	*SNVT_config_src*
Formats:	*SCPTnwrkCnfg: text("%m")*
Used by:	*SFPTelevatorArrivalGong SFPTelevatorDirectionLantern SFPTelevatorHallLantern SFPTelevatorPositionIndicator SFPTelevatorVoiceAnnouncer SFPTentryExit SFPTidentifierSensor SFPTnodeObject*

SCPTobjMajVer

Overview:

Object major version number. The major version number for the object

This configuration property sets the major version number for a functional block. It is used with the SCPTobjMinVer configuration property. See the *LONMARK Application-Layer Interoperability Guidelines* for details on usage.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*167*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTobjMajVer*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTobjMajVer: text("%d")*
Used by:	*SFPTcalendar SFPTchannelContinuityMonitor SFPTchannelMonitor SFPTclothesWasherDomestic SFPTdataLogger SFPTdeviceMonitor SFPTelevatorArrivalGong*

SFPTelevatorDirectionLantern SFPTelevatorFireSystemsPort
SFPTelevatorHallLantern SFPTelevatorPositionIndicator
SFPTelevatorVoiceAnnouncer SFPTentryExit
SFPThvacValvePositioner SFPTidentifierSensor SFPTisiKeypad
SFPTisiLampActuator SFPTisiOccupancySensor
SFPTisiSunblindActuator SFPTpumpController
SFPTrailcarAudioController SFPTrailcarAudioSensor
SFPTscheduler SFPTschedulerSimple SFPTstaticProgrammable
SFPTsunblindActuator SFPTsunblindController SFPTutilityMeter

SCPTobjMinVer

Overview:

Object minor version number. The minor version number for the object

This configuration property sets the minor version number for a functional block. It is used with the SCPTobjMajVer configuration property. See the *LONMARK Application-Layer Interoperability Guidelines* for details on usage.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*168*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTobjMinVer*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTobjMinVer: text("%d")*
Used by:	*SFPTcalendar SFPTchannelContinuityMonitor* *SFPTchannelMonitor SFPTclothesWasherDomestic* *SFPTdataLogger SFPTdeviceMonitor SFPTelevatorArrivalGong* *SFPTelevatorDirectionLantern SFPTelevatorFireSystemsPort* *SFPTelevatorHallLantern SFPTelevatorPositionIndicator* *SFPTelevatorVoiceAnnouncer SFPTentryExit* *SFPThvacValvePositioner SFPTidentifierSensor SFPTisiKeypad*

SFPTisiLampActuator SFPTisiOccupancySensor
SFPTisiSunblindActuator SFPTpumpController
SFPTrailcarAudioController SFPTrailcarAudioSensor
SFPTscheduler SFPTschedulerSimple SFPTstaticProgrammable
SFPTsunblindActuator SFPTsunblindController SFPTutilityMeter

SCPToccupancyBehavior

Overview:

Occupancy behavior. Specifies mapping of scheduled occupancy values to primary occupancy states based on local occupancy inputs

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*372*
Obsolete:	*no*
Size:	*5*
Programmatic Name:	*SCPToccupancyBehavior*
Default:	*OC_OCCUPIED OC_OCCUPIED OC_OCCUPIED OC_OCCUPIED OC_OCCUPIED*

Neuron C Type:

Structure	
ob_nul_value	Invalid output value. Primary occupancy value when the scheduled value is invalid (ob_nul) and a local occupancy condition is detected
	occup_t
ob_occupied_value	Occupied output value. Primary occupancy value when the scheduled value is occupied (ob_occupied) and a local occupancy condition is detected
	occup_t
ob_unoccupied_value	Unoccupied output value. Primary occupancy value when the scheduled value is unoccupied (ob_unoccupied) and a local occupancy condition is detected
	occup_t
ob_bypass_value	Bypass output value. Primary occupancy value when the scheduled value is bypass (ob_bypass) and a local occupancy condition is detected
	occup_t
ob_standby_value	Standby output value. Primary occupancy value when the scheduled value is standby (ob_standby) and a local occupancy condition is detected
	occup_t

Formats:

SCPToccupancyBehavior: text("%m %m %m %m %m", ob_nul_value, ob_occupied_value, ob_unoccupied_value, ob_bypass_value, ob_standby_value)

Used by:

SFPTschedulerSimple

SCPToccupancyThresholds

Overview:

Occupancy thresholds. Specifies the minimum number of occupancy sensors that must report the same value to override a scheduled output value

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *380*

Obsolete: *no*

Size: *3*

Programmatic Name: *SCPToccupancyThresholds*

Default: *0 0 0*

Neuron C Type:

Structure	
occupied	Occupied threshold. Number of occupancy sensors that must be reporting occupied to report an occupied output when the current schedule specifies an occupied state; if the number of occupancy sensors reporting occupancy is below this value, the occupancy output is set to standby
	unsigned short Minimum: *0* Maximum: *50* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
standby_to_occupied	Standby to occupied override threshold. Number of occupancy sensors that must be reporting occupied to report an occupied output when the current schedule specifies a standby state; if the number of occupancy sensors reporting occupancy is below this value, the occupancy output is set to standby
	unsigned short Minimum: *0* Maximum: *50* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
unoccupied_to_occupied	Unoccupied to occupied override threshold. Number of occupancy sensors that must be reporting occupied to report an occupied output when the current schedule specifies an unoccupied state; if the number of occupancy sensors reporting occupancy is below this value, the occupancy output is set to standby

	unsigned short Minimum: *0* Maximum: *50* Invalid: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SCPToccupancyThresholds: text("%d %d %d", occupied, standby_to_occupied, unoccupied_to_occupied)*

Used by: *SFPTschedulerSimple*

SCPToemType

Overview:

OEM label. The label, programmed by the OEM, to identify the unit name

This configuration property sets the manufacturer's unit name. It is manufacturer-defined.

The ascii field contains a nul-terminated string of up to 16 characters. All unused character should be zeros (0s). The default value is manufacturer-defined.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *61*

Obsolete: *no*

Size: *31*

Programmatic Name: *SCPToemType*

Minimum: *00*31*

Maximum: *FF*16-00*15*

Default: *00*31*

Neuron C Type: *SNVT_str_asc*

Formats: *SCPToemType: text("%s", ascii)*

Used by: *SFPTaudibleFireIndicator SFPTdamperActuator SFPTfireSmokeDamperActuator SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator SFPTuniversalFireIndicator*

SFPTuniversalFireInitiator SFPTvisibleFireIndicator

SCPToffDely

Overview:

Turn-off delay. The length of time that the load remains energized after a change from ON to OFF has been received

This configuration property sets the turn-off delay for a load. It is used to determine the length of time that the load remains energized after a request for a change from ON to OFF has been received. If this property is set to a non-zero value, the load will be kept on for the specified time after the request is set to OFF. The turn-off delay is disabled by setting all fields to zero.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*30*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPToffDely*
Maximum:	*0 17:59:59:999*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPToffDely: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPToffDely#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTclosedLoopActuator SFPTfireSmokeDamperActuator SFPTopenLoopActuator*

SCPToffset

Overview:

.

This configuration property sets the offset to be applied to a hardware sensor to generate a network variable output. It is typically used to calibrate the external hardware by specifying the level that the network variable should take based on the current data from the hardware at the time of the calibration. This offset applies after the use of any translation table or gain

factor. The data type must be the same as that of the network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*26*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPToffset: text("%d")*
Used by:	*SFPTairVelocitySensor SFPTclosedLoopSensor SFPTglobalSolarRadiation SFPTopenLoopSensor SFPTpressureSensor*

SCPToffsetCO2

Overview:

CO2 level offset. Used to calibrate external hardware with additive offset after transformation

This configuration property sets the offset to be applied to a carbon dioxide (CO_2) sensor to generate a network variable output. It is typically used to calibrate the external hardware by specifying the ppm-level that the network variable should take based on the current data from the hardware at the time of the calibration. This offset applies after the use of any translation table or gain factor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*68*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPToffsetCO2*
Default:	*0*
Neuron C Type:	*SNVT_ppm*
Formats:	*SCPToffsetCO2: text("%d")*
Used by:	*SFPTco2Sensor SFPTsccCommandModule*

SCPToffsetFlow

Overview:

Air flow offset. Offset value used to calculate the active airflow setpoint by adding nciFlowOffset to nviAirFlowSetpt

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*265*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPToffsetFlow*
Default:	*0*
Neuron C Type:	*SNVT_flow_f*
Formats:	*SCPToffsetFlow#SI: text("%f")* *SCPToffsetFlow#US: text("%f", *2.1189+0(0:966))* *SCPToffsetFlow#US_cfm: text("%f", *2.1189+0(0:966))* *SCPToffsetFlow#US_liq: text("%f", *0.26418+0(0:837))*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPToffsetRH

Overview:

Relative humidity offset. Used to calibrate external hardware with additive offset after transformation

This configuration property sets the offset to be applied to a humidity sensor to generate a network variable output. It is typically used to calibrate the external hardware by specifying the RH-level that the network variable should take based on the current data from the hardware at the time of the calibration. This offset applies after the use of any translation table or gain factor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*69*
Obsolete:	*No*

Size:	*2*
Programmatic Name:	*SCPToffsetRH*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPToffsetRH: text("%f")*
Used by:	*SFPThvacRelativeHumiditySensor SFPTsccCommandModule*

SCPToffsetTemp

Overview:

Temperature offset. Used to calibrate external hardware with additive offset after transformation

This configuration property sets the offset to be applied to a temperature sensor to generate a network variable output. It is typically used to calibrate the external hardware by specifying the temperature value that the network variable should take based on the current data from the hardware at the time of the calibration. This offset applies after the use of any translation table or gain factor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*70*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPToffsetTemp*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPToffsetTemp#SI: text("%f", *1+0(0:854))* *SCPToffsetTemp#US: text("%f", *1.8+0(0:855))*
Used by:	*SFPThvacTempSensor SFPTsccCommandModule*

SCPTOLCLimits

Overview:

OLC Limits Setpoints. MIN/MAX values for the status report are set here

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*345*
Obsolete:	*No*
Size:	*22*
Programmatic Name:	*SCPTOLCLimits*
Default:	*0 0 0 0 0 0 0 0 0,00000 0,0 0,0*

Neuron C Type:

Structure	
lamp_current_high	Limit high current.
	SNVT_amp_ac_mil
lamp_current_low	Limit low current.
	SNVT_amp_ac_mil
main_current_high	Limit main current high.
	SNVT_amp_ac_mil
main_current_low	Limit main current low.
	SNVT_amp_ac_mil
lamp_voltage_high	Limit lamp voltage high.
	SNVT_volt_ac
lamp_voltage_low	Limit lamp voltage low.
	SNVT_volt_ac
main_voltage_high	Limit main voltage high.
	SNVT_volt_ac
main_voltage_low	Limit main voltage low.
	SNVT_volt_ac
power_factor_low	Limit power factor low.
	SNVT_pwr_fact
power_high	Limit power high.
	SNVT_power
power_low	Limit power low.
	SNVT_power

Formats:

SCPTOLCLimits: text("%d %d %d %d %d %d %d %d %f %f %f", lamp_current_high, lamp_current_low, main_current_high, main_current_low, lamp_voltage_high, lamp_voltage_low, main_voltage_high, main_voltage_low, power_factor_low, power_high, power_low)

Used by:

SFPToutdoorLuminairController

SCPTorientation

Overview:

Orientation. The orientation angle of the display image (0 = landscape, 90 = portrait)

This configuration property sets the angle of a displayed or printed image. It is also used to set portrait or landscape mode.

Landscape mode is 0 degrees and Portrait mode is +90 degrees.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*231*
Obsolete:	*No*
Size:	*2*
Programmatic Name:	*SCPTorientation*
Default:	*0,00*
Neuron C Type:	*SNVT_angle_deg*
Formats:	*SCPTorientation: text("%f")*
Used by:	*SFPTelevatorPositionIndicator*

SCPToutdoorAirEnthalpySetpoint

Overview:

Outdoor air enthalpy. Setpoint for the outdoor air enthalpy economizer enable

This configuration property sets the default air-side economizer outdoor air enthalpy enable setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*200*
Obsolete:	*No*
Size:	*2*
Programmatic Name:	*SCPToutdoorAirEnthalpySetpoint*
Default:	*40,00*
Neuron C Type:	*SNVT_enthalpy*

Formats:	*SCPToutdoorAirEnthalpySetpoint#SI: text("%f")*
	*SCPToutdoorAirEnthalpySetpoint#US: text("%f", *0.429923+7.67(0:1400))*
	*SCPToutdoorAirEnthalpySetpoint#US_diff: text("%f", *0.429923+0(0:1400))*
Used by:	*SFPTdischargeAirController*

SCPToutdoorAirTempSetpoint

Overview:

Outdoor air temperature. Setpoint for the outdoor air temperature economizer enable

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*199*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPToutdoorAirTempSetpoint*
Default:	*10,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPToutdoorAirTempSetpoint#SI: text("%f", *1+0(0:854))*
	*SCPToutdoorAirTempSetpoint#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTdischargeAirController*

SCPTovrBehave

Overview:

Override behavior. This parameter is used to define the behavior when an override request is received.

This configuration property sets the behavior of a sensor or actuator when an override request is received for the associated functional block. The sensor or actuator can retain its last setting, go to a specified value, or go to the default output value.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*32*
Obsolete:	*No*
Size:	*1*
Programmatic Name:	*SCPTovrBehave*
Default:	*OV_RETAIN*
Neuron C Type:	*SNVT_override*
Formats:	*SCPTovrBehave: text("%m")*
Used by:	*SFPTcalendar SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTisiLampActuator SFPTopenLoopActuator SFPTopenLoopSensor SFPTscheduler*

SCPTovrValue

Overview:

.

This configuration property sets the value that a sensor or actuator should adopt when a functional block is overridden and the value of the SCPTovrBehave Override Behavior configuration property is OV_SPECIFIED.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*33*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTovrValue: text("%d")*
Used by:	*SFPTcalendar SFPTclosedLoopActuator SFPTclosedLoopSensor SFPTisiLampActuator SFPTopenLoopActuator SFPTopenLoopSensor SFPTscheduler*

SCPTpartNumber

Overview:

Part number. Manufacturer-defined part number string for the device

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*182*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTpartNumber*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTpartNumber: text("%s",ascii)*

SCPTpollRate

Overview:

Poll rate. (seconds) . Specifies the poll rate for each data source. When this value is greater than zero, the functional block polls each of the data sources identified in the source address array at the rate specified by this value.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*335*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTpollRate*
Default:	*0,0*
Neuron C Type:	s32_type Minimum: *0* Maximum: *2147483646* Invalid: *2147483647* Scaling (A,B,C): *1, -1, 0* Scaled value: *1 *10 $^{-1}$ *(Raw+0)* Resolution: *0.10000000149012*
Formats:	*SCPTpollRate: text("%f")*
Used by:	*SFPTdataLogger*

SCPTpowerProfile

Overview:

LC Power Profile.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*381*
Obsolete:	*no*
Size:	*10*
Programmatic Name:	*SCPTpowerProfile*
Default:	*0,0 0,0 0,0 0,0 0,0*
Neuron C Type:	*SNVT_power_profile*
Formats:	*SCPTpowerProfile: text("%f %f %f %f %f", typicalPower[0], typicalPower[1], typicalPower[2], typicalPower[3], typicalPower[4])*
Used by:	*SFPTsmartLuminaireController*

SCPTpowerupState

Overview:

Power-up state. The state of a light controller object after power-up or reset

This configuration property sets the state (mode) of a controller object after power-up or reset. The state can either be Auto or Off.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*87*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTpowerupState*
Default:	*SET_OFF 0,0 0,00*
Neuron C Type:	*SNVT_setting*
Formats:	*SCPTpowerupState: text("%m %f %f", function, setting, rotation)*
Used by:	*SFPTconstantLightController*

SCPTprimeVal

Overview:

Primary default value. The default output value when an area is occupied

This configuration property sets the default output value to be used when an area is occupied.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*155*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTprimeVal*
Default:	*100,0 1*
Neuron C Type:	*SNVT_switch*
Formats:	*SCPTprimeVal: text("%f %d", value, state)*
Used by:	*SFPToccupancyController*

SCPTprogCmdHistory

Overview:

Command History. Log of recent commands, with time stamp

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*356*
Obsolete:	*no*
Size:	*88*
Programmatic Name:	*SCPTprogCmdHistory*
Default:	*0/0/0 0:0:0 - RQ_NORMAL:*

Neuron C Type:

Structure	
timestamp	.
	SNVT_time_stamp
command	.
	object_request_t
description	.
	unsigned char [80] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTprogCmdHistory: text("%d/%d/%d %d:%d:%d - %m: %s", timestamp.year, timestamp.month, timestamp.day, timestamp.hour, timestamp.minute, timestamp.second, command, description)

SCPTprogCmdHistory#LO: text(date(timestamp.year, timestamp.month, timestamp.day), (" "), time(timestamp.hour, timestamp.minute, timestamp.second),(" - %m: %s", command, description))

Used by:

SFPTstaticProgrammable

SCPTprogErrorHistory

Overview:

State History. Log of recent status values, with time stamp

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *363*

Obsolete: *no*

Size: *8*

Programmatic Name: *SCPTprogErrorHistory*

Default: *0/0/0 0:0:0 PSE_NO_ERROR*

Neuron C Type:

Structure	
time_of_error	.
	SNVT_time_stamp
error	.
	program_status_error_t

Formats:

SCPTprogErrorHistory: text("%d/%d/%d %d:%d:%d %m", time_of_error.year, time_of_error.month, time_of_error.day, time_of_error.hour, time_of_error.minute, time_of_error.second, error)

SCPTprogErrorHistory#LO: text(date(time_of_error.year, time_of_error.month, time_of_error.day), (" "), time(time_of_error.hour, time_of_error.minute, time_of_error.second),(" - %m", error))

Used by:

SFPTstaticProgrammable

SCPTprogFileIndexes

Overview:

File Indexes. Indexes of first and last LonMark files where programs may be stored

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *355*

Obsolete: *no*

Size: *2*

Programmatic Name: *SCPTprogFileIndexes*

Default: *3 - 3*

Neuron C Type:

Structure	
first_file_index	.
	unsigned short Minimum: *3* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
last_file_index	.
	unsigned short Minimum: *3* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SCPTprogFileIndexes: text("%d - %d", first_file_index, last_file_index)*

Used by: *SFPTstaticProgrammable*

SCPTprogName

Overview:

Program Name. Name of currently loaded program

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *351*

Obsolete: *no*

Size: *31*

Programmatic Name: *SCPTprogName*

Default: *00*31*

Neuron C Type: *SNVT_str_asc*

Formats: *SCPTprogName: text("%s", ascii)*

Used by: *SFPTstaticProgrammable*

SCPTprogRevision

Overview:

Program Revision. Revision number and date of currently loaded program

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*352*
Obsolete:	*no*
Size:	*11*
Programmatic Name:	*SCPTprogRevision*
Default:	*v0.0.0 (0/0/0 0:0:0)*

Neuron C Type:

Structure	
major_version	.
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
minor_version	.
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
build_number	.
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
build_date	.
	SNVT_time_stamp

Formats:

SCPTprogRevision: text("v%d.%d.%d (%d/%d/%d %d:%d:%d)", major_version, minor_version, build_number, build_date.year, build_date.month, build_date.day, build_date.hour, build_date.minute, build_date.second)

SCPTprogRevision#LO: text(("v%d.%d.%d (", major_version,minor_version,build_number),date(build_date.year, build_date.month, build_date.day), (" "), time(build_date.hour, build_date.minute, build_date.second), (") "))

Used by:

SFPTstaticProgrammable

SCPTprogSelect

Overview:

Program Select. Buffer Id where the currently loaded program is stored

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*353*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTprogSelect*
Default:	*0*
Neuron C Type:	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTprogSelect: text("%d")*
Used by:	*SFPTstaticProgrammable*

SCPTprogSourceLocation

Overview:

Source Location. Location from where the current program was downloaded

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*354*
Obsolete:	*no*
Size:	*255*
Programmatic Name:	*SCPTprogSourceLocation*
Default:	*00*255*

Neuron C Type:

Structure	
location	.
	unsigned char [255] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTprogSourceLocation: text("%s", location)

Used by:

SFPTstaticProgrammable

SCPTprogStateHistory

Overview:

State History. Log of recent status values, with time stamp

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *357*

Obsolete: *no*

Size: *8*

Programmatic Name: *SCPTprogStateHistory*

Default: *0/0/0 0:0:0 PRS_NO_PROGRAM*

Neuron C Type:

Structure	
time_of_state_change	.
	SNVT_time_stamp
state	.
	program_state_t

Formats:

SCPTprogStateHistory: text("%d/%d/%d %d:%d:%d %m", time_of_state_change.year, time_of_state_change.month, time_of_state_change.day, time_of_state_change.hour, time_of_state_change.minute, time_of_state_change.second, state)

SCPTprogStateHistory#LO: text(date(time_of_state_change.year, time_of_state_change.month, time_of_state_change.day), (" "),

	time(time_of_state_change.hour, time_of_state_change.minute, time_of_state_change.second),(" - %m", state))
Used by:	*SFPTstaticProgrammable*

SCPTpulseValue

Overview:

Pulse and Transformer Constant. This configuration property is used to scale the raw pulse value to an energy-meter value

This configuration property is used to scale the raw pulse value to an energy-meter value.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*292*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTpulseValue*
Default:	*0 0*
Neuron C Type:	*SNVT_muldiv*
Formats:	*SCPTpulseValue: text("%d %d", multiplier,divisor)*
Used by:	*SFPTutilityMeter*

SCPTpumpCharacteristic

Overview:

Pump characteristic (speedmax, pressmax, flowmax) . The basic characteristic data for a pump

This configuration property sets the basic characteristic data for a pump. For further technical information refer to the manufacturer documentation.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*233*
Obsolete:	*no*
Size:	*6*

Programmatic Name: *SCPTpumpCharacteristic*

Default: *0 0,0 0,00*

Neuron C Type:

Structure	
speedMax	Maximum speed.
	SNVT_rpm
pressMax	Maximum pressure. Maximum pressure at zero flow
	SNVT_press
flowMax	Maximum flow. maximum flow at zero pressure
	SNVT_flow_p

Formats:

*SCPTpumpCharacteristic#SI: text("%d %f %f", speedMax, pressMax, flowMax*1+0(0:1408))*

*SCPTpumpCharacteristic#US: text("%d %f %f", speedMax, pressMax*0.2953+0(0:1402), flowMax*0.588578+0(0:1405))*

Used by:

SFPTpumpController

SCPTpumpDownDelay

Overview:

Pump down delay. The delay to use before starting the defrost

This configuration property sets the delay to use before defrost starts.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *113*

Obsolete: *no*

Size: *2*

Programmatic Name: *SCPTpumpDownDelay*

Default: *0,0*

Neuron C Type: *SNVT_time_sec*

Formats:

SCPTpumpDownDelay: text("%f")

Used by:

SFPTboilerController SFPTrefrigDisplayCaseControllerDefrost

SCPTpwmPeriod

Overview:

Pulse-width modulation period. The time period to be used in pulse-width modulation control strategy

This configuration property sets the time period to be used in a pulse-width modulation-control strategy.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*216*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTpwmPeriod*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTpwmPeriod: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTpwmPeriod#LO: text(("%d ", day), time(hour, minute, second, millisecond))*

SCPTpwrSendOnDelta

Overview:

Power send on delta.. The minimum change required to propagate the output value

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*315*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTpwrSendOnDelta*
Default:	*0,5*
Neuron C Type:	*SNVT_power*
Formats:	*SCPTpwrSendOnDelta: text("%f")*

SCPTpwrUpDelay

Overview:

Power-up delay. The minimum period of time after power-up or re-establishment of communications before a control action takes place

This configuration property controls the minimum period of time that expires before outputs are retransmitted. It also is the minimum amount of elapsed time after a power-up or re-establishment of communications before a control action takes place. This can be used to account for the settle-down time of a network.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*72*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTpwrUpDelay*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTpwrUpDelay: text("%f")*
Used by:	*SFPTchiller*

SCPTpwrUpState

Overview:

Chiller enable. The default power-up and restart modes of the device when the default behavior selector is set to zero

This configuration property sets the default power-up and restart modes of various devices (such as chillers, boilers, and similar devices) unless the SCPTdefltBehave configuration property is set equal to one. In the latter case, use the manufacturer-specified values, instead.

The value field is not used. A state value of 0 requests device activity to be Off; a state value of 1 requests device activity to be Auto (run). A state value of –1 (0xFF) indicates an invalid value. The default value is 0 (request device activity Off).

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*

Index:	*73*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTpwrUpState*
Default:	*0,0 0*
Neuron C Type:	*SNVT_switch*
Formats:	*SCPTpwrUpState: text("%f %d", value, state)*
Used by:	*SFPTboilerController SFPTchiller SFPToutdoorLuminairController*

SCPTrampDownTm

Overview:

Minimum ramp-down time. The ramp-down time of the device

This configuration property sets the ramp-down time of a device such as a motor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*161*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTrampDownTm*
Default:	*10,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTrampDownTm: text("%f")*
Used by:	*SFPTvariableSpeedMotorDrive*

SCPTrampUpTm

Overview:

Minimum ramp-up time. The ramp-up time of the device

This configuration property sets the ramp-up time of a device such as a motor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*160*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTrampUpTm*
Default:	*10,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTrampUpTm: text("%f")*
Used by:	*SFPToutdoorLuminairController SFPTvariableSpeedMotorDrive*

SCPTrandomizationInterval

Overview:

Randomization interval. Specifies an interval around a scheduled time that is used by a scheduler to calculate a random event time. Used to reduce simultaneous startup and shutdown of many devices by multiple schedulers.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*376*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTrandomizationInterval*
Default:	*0,0 0,0*
Neuron C Type:	

Structure	
earliest_time_offset	Earliest time offset.
	SNVT_time_sec
latest_time_offset	Latest time offset.
	SNVT_time_sec

Formats:	*SCPTrandomizationInterval: text("%f %f", earliest_time_offset, latest_time_offset)*
Used by:	*SFPTschedulerSimple*

SCPTreflection

Overview:

Reflection factor. The internal gain factor for the measured illumination level

This configuration property sets the internal gain factor for a measured illumination level. Adjusting is needed because the amount of the light reflected back to the sensor element from a light surface differs.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*89*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTreflection*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTreflection: text("%f")*
Used by:	*SFPTlightSensor*

SCPTrefrigGlide

Overview:

Refrigerant glide. Used to characterize the glide of the refrigerant used

This configuration property sets the glide of the refrigerant in a system.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*117*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTrefrigGlide*
Default:	*-274,0*
Neuron C Type:	*SNVT_temp*
Formats:	*SCPTrefrigGlide#SI: text("%f", *1+0(0:854))*

	*SCPTrefrigGlide#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerEvaporator*

SCPTrefrigType

Overview:

Refrigerant type (Refrigerant name, A, B, C) .

This configuration property sets the refrigerant type used in a system. Its primary use is for temperature-pressure conversion.

The formula used is:

$$t = \frac{B}{\ln(p) - A} - C$$

Where:

- *t* is temperature in °C
- *p* is the pressure in Bar absolute
- *A*, *B*, *C* are constants defined for a particular type of refrigerant.

Refrigerant details and numbers are published by the US-based ASHRAE. The default values are manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*119*
Obsolete:	*no*
Size:	*18*
Programmatic Name:	*SCPTrefrigType*
Default:	*0,0,0*

Neuron C Type:

Structure	
refrigerant	Refrigerant name (array of 6 characters) .
	unsigned char [6] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
A	Constant A.
	float Minimum: *-3.4028234663853E+038* Maximum: *3.4028234663853E+038*
B	Constant B.
	float Minimum: *-3.4028234663853E+038* Maximum: *3.4028234663853E+038*
C	Constant C.
	float Minimum: *-3.4028234663853E+038* Maximum: *3.4028234663853E+038*

Formats:

SCPTrefrigType: text("%s %f,%f,%f", refrigerant, A, B, C)

SCPTrefrigType#LO: text("%s %f|%f|%f", refrigerant, A, B, C)

Used by:

SFPTrefrigDisplayCaseControllerEvaporator

SCPTregName

Overview:

Register name. The name of a utility data logger register device

This configuration property sets the name of a utility data logger register device.

The ascii field contains a nul-terminated string of up to 30 characters (making a total of 31

characters). The default value is nul string (all zeroes).

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*163*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTregName*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTregName: text("%s", ascii)*
Used by:	*SFPTutilityDataLoggerRegister*

SCPTreturnFanStaticPressureSetpoint

Overview:

Return fan pressure. Setpoint for the return fan static pressure

This configuration property sets the return fan static pressure setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*194*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTreturnFanStaticPressureSetpoint*
Default:	*10*
Neuron C Type:	*SNVT_press_p*
Formats:	*SCPTreturnFanStaticPressureSetpoint#SI: text("%f")* *SCPTreturnFanStaticPressureSetpoint#US: text("%f", *0.0040217+0(0:954))* *SCPTreturnFanStaticPressureSetpoint#US_psi: text("%f", *1.4504e-4+0(0:875))*
Used by:	*SFPTdischargeAirController*

SCPTrunHrAlarm

Overview:

Running hours alarm threshold level. The alarm threshold for the running hours counter

This configuration property sets the alarm threshold for a running hours counter output network variable. When the threshold level is exceeded, an alarm is sent via the node object. The days, minutes, seconds, and milliseconds of the SNVT_elapsed_tm structure are not used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*136*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTrunHrAlarm*
Maximum:	*65535 23:0:0:0*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTrunHrAlarm: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTrunHrAlarm#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTisiKeypad SFPTisiLampActuator SFPTlampActuator*

SCPTrunHrInit

Overview:

Running hours counter initialization. The initial value of the running hours counter network variable

This configuration property sets the initial value for a running hours counter. The days, minutes, seconds, and milliseconds of the SNVT_elapsed_tm structure are not used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*135*
Obsolete:	*no*

Size:	*7*
Programmatic Name:	*SCPTrunHrInit*
Maximum:	*65535 23:0:0:0*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTrunHrInit: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTrunHrInit#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTisiKeypad SFPTisiLampActuator SFPTlampActuator*

SCPTrunTimeAlarm

Overview:

Runtime Alarm. This configuration property can be used to provide the alarm threshold for the run time counter output network variable

This configuration property can be used to provide the alarm threshold for the run time counter output network variable.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*290*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTrunTimeAlarm*
Default:	*0 0 0 0 0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTrunTimeAlarm: text("%d %d %d %d %d", day, hour, minute, second, millisecond)*
Used by:	*SFPThvacValvePositioner*

SCPTsafExtCnfg

Overview:

.

Mode that a device has to be brought to when a safety external request state is pending.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*257*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTsafExtCnfg: text("%d")*
Used by:	*SFPTentryExit*

SCPTsaturationDelay

Overview:

Saturation delay.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*271*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsaturationDelay*
Default:	*0*
Neuron C Type:	*SNVT_time_min*
Formats:	*SCPTsaturationDelay: text("%d")*
Used by:	*SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTscanTime

Overview:

Scan Time.. Duration in which all devices are being queried

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*321*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTscanTime*
Default:	*0 0:0:0:0*
Neuron C Type:	*SNVT_elapsed_tm*
Formats:	*SCPTscanTime: text("%d %d:%d:%d:%d", day, hour, minute, second, millisecond)* *SCPTscanTime#LO: text(("%d ", day), time(hour, minute, second, millisecond))*
Used by:	*SFPTdeviceMonitor*

SCPTscene

Overview:

Scene configuration. Scene definition used to create a scene table. This SCPT defines the minimum entries required by the ISI profiles. May be used in combination with SCPTsceneTiming.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*307*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTscene*
Default:	*0 0,0 0 0*

Neuron C Type:

Structure	
scene_number	Scene number.. Scene number used to uniquely identify a scene. A scene number to be recalled or learned is typically received from a SNVT_switch_2 or SNVT_scene input NV.

	unsigned short Minimum: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
setting	Scene setting level (% of full level) . Setting value that is applied when the scene specified by scene_number is recalled. Also used to set or specify a factor to be multiplied with the setting.
	unsigned short Minimum: *0* Maximum: *200* Invalid: *255* Scaling (A,B,C): *5, -1, 0* Scaled value: *5 *10 $^{-1}$ *(Raw+0)* Resolution: *0.5*
rotation	Scene rotation angle (degrees) . Rotation angle that is applied when the scene specified by scene_number is recalled. Only applies to devices that support a rotation setting such as blinds.
	signed short Minimum: *-90* Maximum: *90* Invalid: *-128* Scaling (A,B,C): *2, 0, 0* Scaled value: *2 *10 ^{0} *(Raw+0)* Resolution: *2*
unoccupied_scene	Unoccupied scene number.. Scene to be activated when the scene specified by scene_number is active, and an unoccupied input is received.

	unsigned short Minimum: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats: *SCPTscene: text("%d %f %d %d", scene_number, setting, rotation, unoccupied_scene)*

Used by: *SFPTisiLampActuator SFPTisiSunblindActuator*

SCPTsceneColor

Overview:

Scene color configuration. Scene color definition used to supplement a scene table created with a SCPTscene array. This SCPT defines optional color scene table entries for the ISI profiles. When used, it must be used in combination with a SCPTscene array.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*343*
Obsolete:	*no*
Size:	*5*
Programmatic Name:	*SCPTsceneColor*
Default:	*EC=COLOR_CIE31_LUMEN X=0,000 Y=0,000 ABSOLUT_Y=0*
Neuron C Type:	*SNVT_color_2*

Formats: *SCPTsceneColor: text(("EC=%m ",encoding),((encoding == 0)?("X=%f Y=%f ABSOLUT_Y=%d", color_value.CIE1931_lumen.x,color_value.CIE1931_lumen.y, color_value.CIE1931_lumen.absolute_Y) :((encoding == 1) ?("X=%f Y=%f PERCENT_Y=%f", color_value.CIE1931_percent.x,color_value.CIE1931_percent.y, color_value.CIE1931_percent.percent_Y):((encoding == 2) ?("RED=%d GREEN=%d BLUE= %d",color_value.RGB.red,color_value.RGB.green,color_value.RGB.blue):*

((encoding == 3) ?("COL_TEMP= %d",color_value.color_temperature):("UNKNOWN"))))))

SCPTsceneDef

Overview:

LC Scene Definition.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*384*
Obsolete:	*no*
Size:	*3*
Programmatic Name:	*SCPTsceneDef*
Default:	*0 0 0*
Neuron C Type:	*SNVT_scene_def*
Formats:	*SCPTsceneDef: text("%d %d %d", scene_number, setting, unoccupied_scene)*
Used by:	*SFPTsmartLuminaireController*

SCPTsceneName

Overview:

Scene name.. Name for a scene to be used by optional user interface applications. Used to create an array that supplements a scene table created with a SCPTscene array.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*316*
Obsolete:	*no*
Size:	*13*
Programmatic Name:	*SCPTsceneName*
Default:	*CE_UTF_8*

Neuron C Type:

Structure	
encoding	Character encoding.. Character encoding method
	char_encoding_t
name	.
	unsigned short [12] Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTsceneName: text(("%m ", encoding),((encoding == 0) ? ("%s", name) :("%x %x %x %x %x %x %x %x %x %x %x %x", name[0], name[1], name[2], name[3], name[4], name[5], name[6], name[7], name[8], name[9], name[10], name[11])))

Used by:

SFPTisiKeypad SFPTisiLampActuator SFPTisiSunblindActuator

SCPTsceneNmbr

Overview:

Scene number (Numeric selector) . The number of the first scene for the panel, other numbers are subsequent

This configuration property sets the number of the first scene for a scene panel. Other numbers are subsequent. The total number of scenes is hardware dependent.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*94*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTsceneNmbr*
Default:	*1*

Neuron C Type:	unsigned short	
	Minimum:	*1*
	Maximum:	*255*
	Invalid:	*0*
	Scaling (A,B,C):	*1, 0, 0*
	Scaled value:	*1 *10 ^{0} *(Raw+0)*
	Resolution:	*1*
Formats:	*SCPTsceneNmbr: text("%d")*	
Used by:	*SFPTscenePanel*	

SCPTsceneOffset

Overview:

Scene offset. The offset for the scene number when data is forwarded from primary to secondary

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*	
Index:	*157*	
Obsolete:	*no*	
Size:	*1*	
Programmatic Name:	*SCPTsceneOffset*	
Default:	*0*	
Neuron C Type:	unsigned short	
	Minimum:	*0*
	Maximum:	*255*
	Scaling (A,B,C):	*1, 0, 0*
	Scaled value:	*1 *10 ^{0} *(Raw+0)*
	Resolution:	*1*
Formats:	*SCPTsceneOffset: text("%d")*	
Used by:	*SFPTpartitionWallController*	

SCPTsceneTiming

Overview:

Scene timing configuration. Scene timing definition used to supplement a scene table created with a SCPTscene array. This SCPT defines the optional scene table entries for the ISI profiles. When used, it must be used in combination with a SCPTscene array.

This configuration property sets an offset for a scene number. The offset is added to the scene number when data is forwarded from a primary side input to a secondary side output. The offset is subtracted from the scene number when data is forwarded from the secondary side input to the primary side output. The application should take care of possible overflows and underflows of the 8-bit scene number.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *308*

Obsolete: *no*

Size: *4*

Programmatic Name: *SCPTsceneTiming*

Default: *0,0 0,0*

Neuron C Type:

Structure	
fade_time	Scene fade time (seconds) . Time to ramp to a new setting. Fading starts after any delay specified by the delay_time field, so the total time to move to a new value is delay_time plus fade_time seconds.
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, -1, 0* Scaled value: *1 *10 $^{-1}$ *(Raw+0)* Resolution: *0.10000000149012*
delay_time	Scene delay time (seconds) . Delay time from the time a new scene is selected until any change is made to the light intensity or appliance state.
	unsigned long Minimum: *0* Maximum: *65535* Scaling (A,B,C): *1, -1, 0* Scaled value: *1 *10 $^{-1}$ *(Raw+0)* Resolution: *0.10000000149012*

Formats: *SCPTsceneTiming: text("%f %f", fade_time,delay_time)*

Used by: *SFPTisiLampActuator SFPTisiSunblindActuator*

SCPTschedule

Overview:

Schedule. Describes the attributes of a daily schedule definition

This configuration property is used to create an array of schedule descriptions, each including a temporary schedule flag, priority, and an index into the time-value list defined by a SCPTscheduleTimeValue array. Optional schedule names may be provided using a SCPTscheduleName array of the same size as the SCPTschedule array. When multiple schedules are active, the first schedule in the SCPTschedule array with a valid value and the highest priority (lowest priority value) is used.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*274*
Obsolete:	*no*
Size:	*3*
Programmatic Name:	*SCPTschedule*
Default:	*0 127 65535*

Neuron C Type:

Structure	
temporary	Temporary flag. Identifies a temporary schedule. Temporary schedules are deleted at the end of the day that they are active
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
schedule_priority	Schedule priority. Specifies the priority for this schedule. Low priority values specify high priority, and high priority values specify low priority. Zero (0) is the highest priority and 255 is the lowest
	bitfield Signed: *no* Width: *7* Offset: *1* Minimum: *0* Maximum: *127* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
time_value_index	Time-value array index. Identifies the starting entry of a list of time-value events in a SCPTscheduleTimeValue array. The end of the list is identified by the terminator field in the SCPTscheduleTime entry
	unsigned long Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTschedule: text("%d %d %d", temporary, schedule_priority, time_value_index)

Used by: *SFPTscheduler*

SCPTscheduleDates

Overview:

Schedule dates. A range of dates with an optional qualifier that specifies when a schedule is active

This configuration property is used to create an array of schedule dates. Each entry consists of a starting date, ending date, and date qualifier for recurring events.

Starting date and/or ending date may contain wildcards (see below). The following table contains all possible combinations of wildcards and their interpretation.

Start End	**-**-**	**-**-DE	**-ME-**	**-ME-DE	YE-**-**	YE-**-DE	YE-ME-**	YE-ME-DE
--**	always	each month from 1st to DEth day of month	each year from 01-01 to last day of month ME	each year from 01-01 to DEth day of month ME	until YE-12-31	until YE-12-DE	until YE-ME-31	until YE-ME-DE
--DS	each month from DSth day to last day of month	each month from DSth to DEth day of month	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
-MS-	each year from first day of month MS to 12-31	Invalid	each year from 1st day of month MS to last day of month ME	Invalid	Invalid	Invalid	Invalid	Invalid
**-MS-DS	each year from DSth day of month MS to 12-31	Invalid	Invalid	each year from MS-DS to ME-DE	Invalid	Invalid	Invalid	Invalid
YS-**-**	from YS-01-01	Invalid	Invalid	Invalid	from YS-01-01 to YE-12-31	Invalid	Invalid	Invalid
YS-**-DS	from YS-01-DS	Invalid	Invalid	Invalid	Invalid	from YS-01-DS to YE-12-DE	Invalid	Invalid
YS-MS-**	from YS-MS-01	Invalid	Invalid	Invalid	Invalid	Invalid	from YS-MS-01 to YE-ME-31	Invalid
YS-MS-	from YS-MS-DS	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	YS-MS-DS to

DS								YE-ME-DE

** ... wildcard (unspecified)

YS/YE ... Start/End Year

MS/ME ... Start/End Month

DS/DE ... Start/End Day

All other combinations are not specified and thus their behavior is a local matter. A possible behavior would be to disable the schedule date entry, that is, treat it as "never".

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*273*
Obsolete:	*no*
Size:	*12*
Programmatic Name:	*SCPTscheduleDates*
Default:	*-1 0 DM_EVERY_DAY -1 0 0 DM_EVERY_DAY MN_EVERY_MONTH DM_EVERY_DAY 65535*
Neuron C Type:	Structure
	start — Schedule start.

	Structure	
	year	Starting year. Starting year for schedule Zero (0) means the starting year is not specified ("wildcard").
		signed long Minimum: *-1* Maximum: *3000* Invalid: *-1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	month	Starting month. Starting month for schedule Zero (0) means the starting month is not specified ("wildcard").
		unsigned short Minimum: *0* Maximum: *12* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	day	Starting day. Starting day for schedule A value of DM_EVERY_DAY means the starting day is not specified ("wildcard"). If a value that corresponds to multiple days such as DM_EVERY_MON is specified for start.day, the first day of the month that matches the value is used. In this case the start month must not be unspecified ("a wildcard").
		days_of_month_t
end	Schedule end.	
	Structure	

	year	Ending year. Ending year for schedule Zero (0) means the starting ending year is not specified ("wildcard").
		signed long Minimum: *-1* Maximum: *3000* Invalid: *-1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	temporary	Temporary flag. Identifies a temporary schedule. Temporary schedules are deleted at the end of the day that they are active
		bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	month	Ending month. Ending month for schedule Zero (0) means the ending month is not specified ("wildcard").

		bitfield Signed: *no* Width: *7* Offset: *1* Minimum: *0* Maximum: *12* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	day	Ending day. Ending day for schedule A value of DM_EVERY_DAY means the ending day is not specified ("wildcard"). If a value that corresponds to multiple days such as DM_EVERY_MON is specified for end.day, the last day of the month that matches the value is used. In this case the end month must not be unspecified ("a wildcard").
		days_of_month_t
qualifier	.	
	Structure	
	months	Month qualifier. Months within the dates specified by the start and end dates
		months_t
	days	Days qualifier. Days within the dates specified by the start and end dates
		days_of_month_t
schedule_index	Schedule number. Index into a schedule or schedule name array	

	unsigned long Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTscheduleDates: text("%d %d %m %d %d %d %m %m %m %d", start.year, start.month, start.day, end.year, end.temporary, end.month, end.day, qualifier.months, qualifier.days, schedule_index)

Used by:

SFPTcalendar

SCPTscheduleException

Overview:

Exception schedule. An event that overrides a daily schedule; typically used for special events

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *374*

Obsolete: *no*

Size: *8*

Programmatic Name: *SCPTscheduleException*

Default: *MN_NUL DM_EVERY_DAY 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0*

Neuron C Type: *SNVT_sched_exc*

Formats:

SCPTscheduleException: text("%m %m %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d", months, days, start_time_value.sunrise_relative_flag, start_time_value.sunset_relative_flag, start_time_value.negative_time_offset_flag, start_time_value.hour, start_time_value.occ_value_ignored_flag, start_time_value.gp_value_ignored_flag, start_time_value.minutes, start_time_value.occupancy, start_time_value.gp_value, end_time.sunrise_relative_flag, end_time.sunset_relative_flag, end_time.negative_time_offset_flag, end_time.hour, end_time.start_offset_enable_flag, end_time.stop_offset_enable_flag, end_time.minutes)

Used by: *SFPTschedulerSimple*

SCPTscheduleFriday

Overview:

Friday schedule. A structure containing an array of seven time-value pairs that specify the daily schedule for Friday; unused time-value pairs have an invalid value (31) for the hour; if two time-value pairs specify the same time, the first with a valid output value is used

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *370*

Obsolete: *no*

Size: *28*

Programmatic Name: *SCPTscheduleFriday*

Default: *0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0*

Neuron C Type:

Structure	
time_value [7]	.
	SNVT_time_val_2

Formats:

SCPTscheduleFriday: text("%d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d", time_value[0].sunrise_relative_flag, time_value[0].sunset_relative_flag, time_value[0].negative_time_offset_flag, time_value[0].hour, time_value[0].occ_value_ignored_flag, time_value[0].gp_value_ignored_flag, time_value[0].minutes, time_value[0].occupancy, time_value[0].gp_value, time_value[1].sunrise_relative_flag, time_value[1].sunset_relative_flag, time_value[1].negative_time_offset_flag, time_value[1].hour, time_value[1].occ_value_ignored_flag, time_value[1].gp_value_ignored_flag, time_value[1].minutes, time_value[1].occupancy, time_value[1].gp_value, time_value[2].sunrise_relative_flag, time_value[2].sunset_relative_flag, time_value[2].negative_time_offset_flag, time_value[2].hour,

time_value[2].occ_value_ignored_flag, time_value[2].gp_value_ignored_flag, time_value[2].minutes, time_value[2].occupancy, time_value[2].gp_value, time_value[3].sunrise_relative_flag, time_value[3].sunset_relative_flag, time_value[3].negative_time_offset_flag, time_value[3].hour, time_value[3].occ_value_ignored_flag, time_value[3].gp_value_ignored_flag, time_value[3].minutes, time_value[3].occupancy, time_value[3].gp_value, time_value[4].sunrise_relative_flag, time_value[4].sunset_relative_flag, time_value[4].negative_time_offset_flag, time_value[4].hour, time_value[4].occ_value_ignored_flag, time_value[4].gp_value_ignored_flag, time_value[4].minutes, time_value[4].occupancy, time_value[4].gp_value, time_value[5].sunrise_relative_flag, time_value[5].sunset_relative_flag, time_value[5].negative_time_offset_flag, time_value[5].hour, time_value[5].occ_value_ignored_flag, time_value[5].gp_value_ignored_flag, time_value[5].minutes, time_value[5].occupancy, time_value[5].gp_value, time_value[6].sunrise_relative_flag, time_value[6].sunset_relative_flag, time_value[6].negative_time_offset_flag, time_value[6].hour, time_value[6].occ_value_ignored_flag, time_value[6].gp_value_ignored_flag, time_value[6].minutes, time_value[6].occupancy, time_value[6].gp_value)

Used by: *SFPTschedulerSimple*

SCPTscheduleHoliday

Overview:

Holiday or vacation schedule. An event that overrides a daily schedule; typically used for holiday or vacation event

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*375*
Obsolete:	*no*
Size:	*8*
Programmatic Name:	*SCPTscheduleHoliday*
Default:	*MN_EVERY_MONTH DM_EVERY_DAY 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0*

Neuron C Type:	*SNVT_sched_exc*
Formats:	*SCPTscheduleHoliday: text("%m %m %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d", months, days, start_time_value.sunrise_relative_flag, start_time_value.sunset_relative_flag, start_time_value.negative_time_offset_flag, start_time_value.hour, start_time_value.occ_value_ignored_flag, start_time_value.gp_value_ignored_flag, start_time_value.minutes, start_time_value.occupancy, start_time_value.gp_value, end_time.sunrise_relative_flag, end_time.sunset_relative_flag, end_time.negative_time_offset_flag, end_time.hour, end_time.start_offset_enable_flag, end_time.stop_offset_enable_flag, end_time.minutes)*
Used by:	*SFPTschedulerSimple*

SCPTscheduleInternal

Overview:

Internal schedule (boolean) . Enable internal scheduling

This configuration property determines whether to internally schedule light switching according to a predefined schedule. The SCPTlightOn and SCPTlightOff Lights Timers configuration properties can be used to define the schedule.

A True value enables internal scheduling; a False value disables any schedules. Any non-zero values are assumed to be True.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*226*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTscheduleInternal*
Default:	*BOOL_FALSE*
Neuron C Type:	*boolean_t*
Formats:	*SCPTscheduleInternal: text("%m")*

SCPTscheduleMonday

Overview:

Monday schedule. A structure containing an array of seven time-value pairs that specify the daily schedule for Monday; unused time-value pairs have an invalid value (31) for the hour; if two time-value pairs specify the same time, the first with a valid output value is used

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *366*

Obsolete: *no*

Size: *28*

Programmatic Name: *SCPTscheduleMonday*

Default: *0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0*

Neuron C Type:

Structure	
time_value [7]	.
	SNVT_time_val_2

Formats:

SCPTscheduleMonday: text("%d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d", time_value[0].sunrise_relative_flag, time_value[0].sunset_relative_flag, time_value[0].negative_time_offset_flag, time_value[0].hour, time_value[0].occ_value_ignored_flag, time_value[0].gp_value_ignored_flag, time_value[0].minutes, time_value[0].occupancy, time_value[0].gp_value, time_value[1].sunrise_relative_flag, time_value[1].sunset_relative_flag, time_value[1].negative_time_offset_flag, time_value[1].hour, time_value[1].occ_value_ignored_flag, time_value[1].gp_value_ignored_flag, time_value[1].minutes, time_value[1].occupancy, time_value[1].gp_value, time_value[2].sunrise_relative_flag, time_value[2].sunset_relative_flag, time_value[2].negative_time_offset_flag, time_value[2].hour, time_value[2].occ_value_ignored_flag, time_value[2].gp_value_ignored_flag, time_value[2].minutes, time_value[2].occupancy, time_value[2].gp_value, time_value[3].sunrise_relative_flag, time_value[3].sunset_relative_flag, time_value[3].negative_time_offset_flag, time_value[3].hour,

time_value[3].occ_value_ignored_flag, time_value[3].gp_value_ignored_flag, time_value[3].minutes, time_value[3].occupancy, time_value[3].gp_value, time_value[4].sunrise_relative_flag, time_value[4].sunset_relative_flag, time_value[4].negative_time_offset_flag, time_value[4].hour, time_value[4].occ_value_ignored_flag, time_value[4].gp_value_ignored_flag, time_value[4].minutes, time_value[4].occupancy, time_value[4].gp_value, time_value[5].sunrise_relative_flag, time_value[5].sunset_relative_flag, time_value[5].negative_time_offset_flag, time_value[5].hour, time_value[5].occ_value_ignored_flag, time_value[5].gp_value_ignored_flag, time_value[5].minutes, time_value[5].occupancy, time_value[5].gp_value, time_value[6].sunrise_relative_flag, time_value[6].sunset_relative_flag, time_value[6].negative_time_offset_flag, time_value[6].hour, time_value[6].occ_value_ignored_flag, time_value[6].gp_value_ignored_flag, time_value[6].minutes, time_value[6].occupancy, time_value[6].gp_value)

Used by: *SFPTschedulerSimple*

SCPTscheduleName

Overview:

Schedule name. Used to create an array of names for schedules defined by a SCPTschedule array or a SCPTscheduleDates array.

This configuration property is used to create an array of schedule names. Each schedule name is associated with a schedule in a SCPTschedule array in a Scheduler, or with schedules referenced by a SCPTscheduleDates array in a Calendar. When used with a SCPTschedule array in a Scheduler, the schedule with the same index in the SCPTschedule array is active when the SCPTscheduleName name matches a name identified as active by a Calendar functional block or other functional block that has a SNVT_date_event output.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*279*
Obsolete:	*no*
Size:	*22*
Programmatic Name:	*SCPTscheduleName*

Maximum: ~~~~~~~~~~~~~~~~~~~~~~

Default: *00*22*

Neuron C Type:

Structure	
name	.
	unsigned char [22] Minimum: *0* Maximum: *126* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTscheduleName: text("%s", name)

Used by:

SFPTcalendar SFPTscheduler

SCPTscheduleSaturday

Overview:

Saturday schedule. A structure containing an array of seven time-value pairs that specify the daily schedule for Saturday; unused time-value pairs have an invalid value (31) for the hour; if two time-value pairs specify the same time, the first with a valid output value is used

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *371*

Obsolete: *no*

Size: *28*

Programmatic Name: *SCPTscheduleSaturday*

Default: *0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0*

Neuron C Type:

Structure	
time_value [7]	.
	SNVT_time_val_2

Formats:

SCPTscheduleSaturday: text("%d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d

%m %d", time_value[0].sunrise_relative_flag,
time_value[0].sunset_relative_flag,
time_value[0].negative_time_offset_flag, time_value[0].hour,
time_value[0].occ_value_ignored_flag,
time_value[0].gp_value_ignored_flag, time_value[0].minutes,
time_value[0].occupancy, time_value[0].gp_value,
time_value[1].sunrise_relative_flag,
time_value[1].sunset_relative_flag,
time_value[1].negative_time_offset_flag, time_value[1].hour,
time_value[1].occ_value_ignored_flag,
time_value[1].gp_value_ignored_flag, time_value[1].minutes,
time_value[1].occupancy, time_value[1].gp_value,
time_value[2].sunrise_relative_flag,
time_value[2].sunset_relative_flag,
time_value[2].negative_time_offset_flag, time_value[2].hour,
time_value[2].occ_value_ignored_flag,
time_value[2].gp_value_ignored_flag, time_value[2].minutes,
time_value[2].occupancy, time_value[2].gp_value,
time_value[3].sunrise_relative_flag,
time_value[3].sunset_relative_flag,
time_value[3].negative_time_offset_flag, time_value[3].hour,
time_value[3].occ_value_ignored_flag,
time_value[3].gp_value_ignored_flag, time_value[3].minutes,
time_value[3].occupancy, time_value[3].gp_value,
time_value[4].sunrise_relative_flag,
time_value[4].sunset_relative_flag,
time_value[4].negative_time_offset_flag, time_value[4].hour,
time_value[4].occ_value_ignored_flag,
time_value[4].gp_value_ignored_flag, time_value[4].minutes,
time_value[4].occupancy, time_value[4].gp_value,
time_value[5].sunrise_relative_flag,
time_value[5].sunset_relative_flag,
time_value[5].negative_time_offset_flag, time_value[5].hour,
time_value[5].occ_value_ignored_flag,
time_value[5].gp_value_ignored_flag, time_value[5].minutes,
time_value[5].occupancy, time_value[5].gp_value,
time_value[6].sunrise_relative_flag,
time_value[6].sunset_relative_flag,
time_value[6].negative_time_offset_flag, time_value[6].hour,
time_value[6].occ_value_ignored_flag,
time_value[6].gp_value_ignored_flag, time_value[6].minutes,
time_value[6].occupancy, time_value[6].gp_value)

Used by:

SFPTschedulerSimple

SCPTscheduleSunday

Overview:

Sunday schedule. A structure containing an array of seven time-value pairs that specify the daily schedule for Sunday; unused time-value pairs have an invalid value (31) for the hour; if two time-value pairs specify the same time, the first with a valid output value is used

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *365*

Obsolete: *no*

Size: *28*

Programmatic Name: *SCPTscheduleSunday*

Default: *0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0*

Neuron C Type:

Structure	
time_value [7]	.
	SNVT_time_val_2

Formats:

SCPTscheduleSunday: text("%d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d", time_value[0].sunrise_relative_flag, time_value[0].sunset_relative_flag, time_value[0].negative_time_offset_flag, time_value[0].hour, time_value[0].occ_value_ignored_flag, time_value[0].gp_value_ignored_flag, time_value[0].minutes, time_value[0].occupancy, time_value[0].gp_value, time_value[1].sunrise_relative_flag, time_value[1].sunset_relative_flag, time_value[1].negative_time_offset_flag, time_value[1].hour, time_value[1].occ_value_ignored_flag, time_value[1].gp_value_ignored_flag, time_value[1].minutes, time_value[1].occupancy, time_value[1].gp_value, time_value[2].sunrise_relative_flag, time_value[2].sunset_relative_flag, time_value[2].negative_time_offset_flag, time_value[2].hour, time_value[2].occ_value_ignored_flag, time_value[2].gp_value_ignored_flag, time_value[2].minutes,

time_value[2].occupancy, time_value[2].gp_value,
time_value[3].sunrise_relative_flag,
time_value[3].sunset_relative_flag,
time_value[3].negative_time_offset_flag, time_value[3].hour,
time_value[3].occ_value_ignored_flag,
time_value[3].gp_value_ignored_flag, time_value[3].minutes,
time_value[3].occupancy, time_value[3].gp_value,
time_value[4].sunrise_relative_flag,
time_value[4].sunset_relative_flag,
time_value[4].negative_time_offset_flag, time_value[4].hour,
time_value[4].occ_value_ignored_flag,
time_value[4].gp_value_ignored_flag, time_value[4].minutes,
time_value[4].occupancy, time_value[4].gp_value,
time_value[5].sunrise_relative_flag,
time_value[5].sunset_relative_flag,
time_value[5].negative_time_offset_flag, time_value[5].hour,
time_value[5].occ_value_ignored_flag,
time_value[5].gp_value_ignored_flag, time_value[5].minutes,
time_value[5].occupancy, time_value[5].gp_value,
time_value[6].sunrise_relative_flag,
time_value[6].sunset_relative_flag,
time_value[6].negative_time_offset_flag, time_value[6].hour,
time_value[6].occ_value_ignored_flag,
time_value[6].gp_value_ignored_flag, time_value[6].minutes,
time_value[6].occupancy, time_value[6].gp_value)

Used by: *SFPTschedulerSimple*

SCPTscheduleThursday

Overview:

Thursday schedule. A structure containing an array of seven time-value pairs that specify the daily schedule for Thursday; unused time-value pairs have an invalid value (31) for the hour; if two time-value pairs specify the same time, the first with a valid output value is used

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*369*
Obsolete:	*no*
Size:	*28*
Programmatic Name:	*SCPTscheduleThursday*
Default:	*0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0*

OC_OCCUPIED 0

Neuron C Type:

Structure	
time_value [7]	.
	SNVT_time_val_2

Formats:

SCPTscheduleThursday: text("%d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d", time_value[0].sunrise_relative_flag, time_value[0].sunset_relative_flag, time_value[0].negative_time_offset_flag, time_value[0].hour, time_value[0].occ_value_ignored_flag, time_value[0].gp_value_ignored_flag, time_value[0].minutes, time_value[0].occupancy, time_value[0].gp_value, time_value[1].sunrise_relative_flag, time_value[1].sunset_relative_flag, time_value[1].negative_time_offset_flag, time_value[1].hour, time_value[1].occ_value_ignored_flag, time_value[1].gp_value_ignored_flag, time_value[1].minutes, time_value[1].occupancy, time_value[1].gp_value, time_value[2].sunrise_relative_flag, time_value[2].sunset_relative_flag, time_value[2].negative_time_offset_flag, time_value[2].hour, time_value[2].occ_value_ignored_flag, time_value[2].gp_value_ignored_flag, time_value[2].minutes, time_value[2].occupancy, time_value[2].gp_value, time_value[3].sunrise_relative_flag, time_value[3].sunset_relative_flag, time_value[3].negative_time_offset_flag, time_value[3].hour, time_value[3].occ_value_ignored_flag, time_value[3].gp_value_ignored_flag, time_value[3].minutes, time_value[3].occupancy, time_value[3].gp_value, time_value[4].sunrise_relative_flag, time_value[4].sunset_relative_flag, time_value[4].negative_time_offset_flag, time_value[4].hour, time_value[4].occ_value_ignored_flag, time_value[4].gp_value_ignored_flag, time_value[4].minutes, time_value[4].occupancy, time_value[4].gp_value, time_value[5].sunrise_relative_flag, time_value[5].sunset_relative_flag, time_value[5].negative_time_offset_flag, time_value[5].hour, time_value[5].occ_value_ignored_flag, time_value[5].gp_value_ignored_flag, time_value[5].minutes, time_value[5].occupancy, time_value[5].gp_value,

time_value[6].sunrise_relative_flag, time_value[6].sunset_relative_flag, time_value[6].negative_time_offset_flag, time_value[6].hour, time_value[6].occ_value_ignored_flag, time_value[6].gp_value_ignored_flag, time_value[6].minutes, time_value[6].occupancy, time_value[6].gp_value)

Used by: *SFPTschedulerSimple*

SCPTscheduleTimeValue

Overview:

Schedule time-value pair. Specifies the time and value for a scheduled event

This configuration property is used to create an array of time-value pairs, organized as terminated lists.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *275*

Obsolete: *no*

Size: *3*

Programmatic Name: *SCPTscheduleTimeValue*

Default: *1 0 0 0 255*

Neuron C Type:

Structure	
invalid	Invalid flag. Identifies an undefined schedule entry
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
terminator	Terminator flag. Identifies the last entry in a time-value list

	bitfield Signed: *no* Width: *1* Offset: *1* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
hour	Hour. Hours since midnight for a scheduled event
	bitfield Signed: *no* Width: *6* Offset: *2* Minimum: *0* Maximum: *47* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
minute	Minute. Minute within the hour for a scheduled event
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
value	Scheduler value. Specifies the value to output when a time-value pair is active. The value must be mapped to a value that matches the type of the output network variable
	SNVT_sched_val

Formats:

SCPTscheduleTimeValue: text("%d %d %d %d %d", invalid, terminator, hour, minute, value)

Used by:

SFPTscheduler

SCPTscheduleTuesday

Overview:

Tuesday schedule. A structure containing an array of seven time-value pairs that specify the daily schedule for Tuesday; unused time-value pairs have an invalid value (31) for the hour; if two time-value pairs specify the same time, the first with a valid output value is used

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *367*

Obsolete: *no*

Size: *28*

Programmatic Name: *SCPTscheduleTuesday*

Default: *0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0*

Neuron C Type:

Structure	
time_value [7]	.
	SNVT_time_val_2

Formats:

SCPTscheduleTuesday: text("%d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d", time_value[0].sunrise_relative_flag, time_value[0].sunset_relative_flag, time_value[0].negative_time_offset_flag, time_value[0].hour, time_value[0].occ_value_ignored_flag, time_value[0].gp_value_ignored_flag, time_value[0].minutes, time_value[0].occupancy, time_value[0].gp_value, time_value[1].sunrise_relative_flag, time_value[1].sunset_relative_flag, time_value[1].negative_time_offset_flag, time_value[1].hour, time_value[1].occ_value_ignored_flag, time_value[1].gp_value_ignored_flag, time_value[1].minutes, time_value[1].occupancy, time_value[1].gp_value, time_value[2].sunrise_relative_flag, time_value[2].sunset_relative_flag, time_value[2].negative_time_offset_flag, time_value[2].hour, time_value[2].occ_value_ignored_flag, time_value[2].gp_value_ignored_flag, time_value[2].minutes, time_value[2].occupancy, time_value[2].gp_value, time_value[3].sunrise_relative_flag, time_value[3].sunset_relative_flag,

time_value[3].negative_time_offset_flag, time_value[3].hour, time_value[3].occ_value_ignored_flag, time_value[3].gp_value_ignored_flag, time_value[3].minutes, time_value[3].occupancy, time_value[3].gp_value, time_value[4].sunrise_relative_flag, time_value[4].sunset_relative_flag, time_value[4].negative_time_offset_flag, time_value[4].hour, time_value[4].occ_value_ignored_flag, time_value[4].gp_value_ignored_flag, time_value[4].minutes, time_value[4].occupancy, time_value[4].gp_value, time_value[5].sunrise_relative_flag, time_value[5].sunset_relative_flag, time_value[5].negative_time_offset_flag, time_value[5].hour, time_value[5].occ_value_ignored_flag, time_value[5].gp_value_ignored_flag, time_value[5].minutes, time_value[5].occupancy, time_value[5].gp_value, time_value[6].sunrise_relative_flag, time_value[6].sunset_relative_flag, time_value[6].negative_time_offset_flag, time_value[6].hour, time_value[6].occ_value_ignored_flag, time_value[6].gp_value_ignored_flag, time_value[6].minutes, time_value[6].occupancy, time_value[6].gp_value)

Used by: *SFPTschedulerSimple*

SCPTscheduleWednesday

Overview:

Wednesday schedule. A structure containing an array of seven time-value pairs that specify the daily schedule for Wednesday; unused time-value pairs have an invalid value (31) for the hour; if two time-value pairs specify the same time, the first with a valid output value is used

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*368*
Obsolete:	*no*
Size:	*28*
Programmatic Name:	*SCPTscheduleWednesday*
Default:	*0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0 0 0 0 0 0 0 0 OC_OCCUPIED 0*

Neuron C Type:

Structure	
time_value [7]	.
	SNVT_time_val_2

Formats:

SCPTscheduleWednesday: text("%d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d %d %d %d %d %d %d %d %m %d", time_value[0].sunrise_relative_flag, time_value[0].sunset_relative_flag, time_value[0].negative_time_offset_flag, time_value[0].hour, time_value[0].occ_value_ignored_flag, time_value[0].gp_value_ignored_flag, time_value[0].minutes, time_value[0].occupancy, time_value[0].gp_value, time_value[1].sunrise_relative_flag, time_value[1].sunset_relative_flag, time_value[1].negative_time_offset_flag, time_value[1].hour, time_value[1].occ_value_ignored_flag, time_value[1].gp_value_ignored_flag, time_value[1].minutes, time_value[1].occupancy, time_value[1].gp_value, time_value[2].sunrise_relative_flag, time_value[2].sunset_relative_flag, time_value[2].negative_time_offset_flag, time_value[2].hour, time_value[2].occ_value_ignored_flag, time_value[2].gp_value_ignored_flag, time_value[2].minutes, time_value[2].occupancy, time_value[2].gp_value, time_value[3].sunrise_relative_flag, time_value[3].sunset_relative_flag, time_value[3].negative_time_offset_flag, time_value[3].hour, time_value[3].occ_value_ignored_flag, time_value[3].gp_value_ignored_flag, time_value[3].minutes, time_value[3].occupancy, time_value[3].gp_value, time_value[4].sunrise_relative_flag, time_value[4].sunset_relative_flag, time_value[4].negative_time_offset_flag, time_value[4].hour, time_value[4].occ_value_ignored_flag, time_value[4].gp_value_ignored_flag, time_value[4].minutes, time_value[4].occupancy, time_value[4].gp_value, time_value[5].sunrise_relative_flag, time_value[5].sunset_relative_flag, time_value[5].negative_time_offset_flag, time_value[5].hour, time_value[5].occ_value_ignored_flag, time_value[5].gp_value_ignored_flag, time_value[5].minutes, time_value[5].occupancy, time_value[5].gp_value, time_value[6].sunrise_relative_flag,

time_value[6].sunset_relative_flag, time_value[6].negative_time_offset_flag, time_value[6].hour, time_value[6].occ_value_ignored_flag, time_value[6].gp_value_ignored_flag, time_value[6].minutes, time_value[6].occupancy, time_value[6].gp_value)

Used by: *SFPTschedulerSimple*

SCPTschedulerOptions

Overview:

.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *379*

Obsolete: *no*

Size: *1*

Programmatic Name: *SCPTschedulerOptions*

Default: *0 0 0 0*

Neuron C Type:

Structure	
reserved	.
	bitfield Signed: *no* Width: *5* Offset: *0* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
alternate_time_source	Alternate time source option flag. Set to one if the device supports an alternate time source such as an interface to an NTP or SNTP server, GPS clock, or radio atomic-clock source

	bitfield Signed: *no* Width: *1* Offset: *5* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
general_purpose_output	General-purpose option flag. Set to one if the general purpose output is supported; if zero, only the occupancy output is used
	bitfield Signed: *no* Width: *1* Offset: *6* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
sunrise_sunset_relative	Sunrise and sunset relative scheduling option flag. Set to one if the scheduler supports sunrise and sunset relative scheduling; set to zero if sunrise and sunset relative scheduling is not supported
	bitfield Signed: *no* Width: *1* Offset: *7* Minimum: *0* Maximum: *1* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTschedulerOptions: text("%d %d %d %d", reserved, alternate_time_source, general_purpose_output,

	sunrise_sunset_relative)
Used by:	*SFPTschedulerSimple*

SCPTscrollSpeed

Overview:

Scroll speed. The scroll speed of the display image

The configuration property sets the rate at which a display image scrolls across the display screen or visible area.

The value field sets the rate from zero to the maximum allowed by the device. The state field enables scrolling. When the state field is set to one, the scroll speed is enabled at the rate set by the value field. When the state field is set to zero, the scroll speed is zero regardless of the value setting.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*229*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTscrollSpeed*
Default:	*0,0 0*
Neuron C Type:	*SNVT_switch*
Formats:	*SCPTscrollSpeed: text("%f %d", value, state)*
Used by:	*SFPTelevatorDirectionLantern SFPTelevatorHallLantern SFPTelevatorPositionIndicator*

SCPTsecondVal

Overview:

Secondary default value. The default output value when the neighboring area is occupied

This configuration property sets the default occupancy value when a neighboring area is occupied.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*156*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsecondVal*
Default:	*0,0 0*
Neuron C Type:	*SNVT_switch*
Formats:	*SCPTsecondVal: text("%f %d", value, state)*
Used by:	*SFPToccupancyController*

SCPTsensConstTmp

Overview:

Temperature sensor constant. Calibration value for a duct temperature sensor

This configuration property sets the calibration multiplier for a temperature sensor.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*65*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsensConstTmp*
Default:	*1,0000*
Neuron C Type:	*SNVT_multiplier*
Formats:	*SCPTsensConstTmp: text("%f")*
Used by:	*SFPTdamperActuator*

SCPTsensConstVAV

Overview:

VAV sensor constant. Calibration constant used to calculate airflow

This configuration property sets the airflow sensor calibration multiplier for a VAV sensor. It is typically used to calculate the air flow by an airflow control actuator.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*67*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsensConstVAV*
Default:	*1,0000*
Neuron C Type:	*SNVT_multiplier*
Formats:	*SCPTsensConstVAV: text("%f")*
Used by:	*SFPTdamperActuator SFPTdischargeAirController SFPTsccAHU SFPTsccChilledCeiling SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController*

SCPTserialNumber

Overview:

Serial number. Manufacturer-defined serial number string for the device

This configuration property sets a manufacturer-defined serial number. The serial number is specified by a string with up to 30 characters followed by a terminating null character (ASCII value zero) totaling 31 characters.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*179*
Obsolete:	*no*
Size:	*31*
Programmatic Name:	*SCPTserialNumber*
Default:	*00*31*
Neuron C Type:	*SNVT_str_asc*
Formats:	*SCPTserialNumber: text("%s", ascii)*

SCPTsetPnts

Overview:

This configuration property sets the occupancy temperature setpoints for heat and cool mode.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*60*
Obsolete:	*no*
Size:	*12*
Programmatic Name:	*SCPTsetPnts*
Minimum:	*-10,00,-10,00,-10,00,-10,00,-10,00,-10,00*
Maximum:	*35,00,35,00,35,00,35,00,35,00,35,00*
Default:	*23,00,25,00,28,00,21,00,19,00,16,00*
Neuron C Type:	*SNVT_temp_setpt*
Formats:	*SCPTsetPnts#SI: text("%f,%f,%f,%f,%f,%f", occupied_cool, standby_cool, unoccupied_cool, occupied_heat, standby_heat, unoccupied_heat)* *SCPTsetPnts#SI_LO: text("%f|%f|%f|%f|%f|%f", occupied_cool, standby_cool, unoccupied_cool, occupied_heat, standby_heat, unoccupied_heat)* *SCPTsetPnts#US: text("%f,%f,%f,%f,%f,%f", occupied_cool*1.8+32(0:855), standby_cool*1.8+32(0:855), unoccupied_cool*1.8+32(0:855), occupied_heat*1.8+32(0:855), standby_heat*1.8+32(0:855), unoccupied_heat*1.8+32(0:855))* *SCPTsetPnts#US_LO: text("%f|%f|%f|%f|%f|%f", occupied_cool*1.8+32(0:855), standby_cool*1.8+32(0:855), unoccupied_cool*1.8+32(0:855), occupied_heat*1.8+32(0:855), standby_heat*1.8+32(0:855), unoccupied_heat*1.8+32(0:855))*
Used by:	*SFPTchilledCeilingController SFPTdischargeAirController SFPTfanCoilUnit SFPTheatPump SFPTroofTopUnit SFPTsccAHU SFPTsccChilledCeiling SFPTsccCommandModule SFPTsccFanCoil SFPTsccHeatPump SFPTsccRadiator SFPTsccRooftop SFPTsccSelfContained SFPTsccUnitVentilator SFPTsccVAV SFPTspaceComfortController SFPTthermostat SFPTunitHeater SFPTunitVentilatorController SFPTvariableAirVolume SFPTwallUnit*

SCPTsetpoint

Overview:

.

This configuration property sets a setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*213*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTsetpoint: text("%d")*

SCPTsluiceCnfg

Overview:

Sluice-lock master/slave control. Role of a device in a sluice-lock connection. A sluice-lock is an interlock mechanism between two entry/exit devices, or a sluice manager and several entry/exit devices, to ensure that only one single entry/exit device is opened at any point in time.

Role of a device in a sluice-lock connection. A sluice-lock is an interlock mechanism between two entry/exit devices, or a sluice manager and several entry/exit devices, to ensure that only one single entry/exit device is opened at once.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*259*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTsluiceCnfg*
Default:	*MSC_UNKNOWN*
Neuron C Type:	*master_slave_t*
Formats:	*SCPTsluiceCnfg: text("%m")*
Used by:	*SFPTentryExit*

SCPTsmokeDayAlrmLim

Overview:

Daytime alarm limit. The daytime alarm limit sensitivity value for the fire initiator in percentage obscuration by smoke

This configuration property sets the daytime alarm limit sensitivity value for an initiator.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*40*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsmokeDayAlrmLim*
Default:	*0,000*
Neuron C Type:	*SNVT_smo_obscur*
Formats:	*SCPTsmokeDayAlrmLim: text("%f")*
Used by:	*SFPTsmokeFireInitiatorIntelli*

SCPTsmokeDayPreAlrmLim

Overview:

Daytime pre-alarm limit. The daytime pre-alarm limit sensitivity value for the fire initiator in percentage obscuration by smoke

This configuration property sets the daytime pre-alarm limit sensitivity value for an initiator.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*138*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsmokeDayPreAlrmLim*
Default:	*0,000*
Neuron C Type:	*SNVT_smo_obscur*
Formats:	*SCPTsmokeDayPreAlrmLim: text("%f")*
Used by:	*SFPTsmokeFireInitiatorIntelli*

SCPTsmokeNightAlrmLim

Overview:

Nighttime alarm limit. The nighttime alarm limit sensitivity value for the fire initiator in percentage obscuration by smoke

This configuration property sets the nighttime alarm limit sensitivity value for an initiator.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*127*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsmokeNightAlrmLim*
Default:	*0,000*
Neuron C Type:	*SNVT_smo_obscur*
Formats:	*SCPTsmokeNightAlrmLim: text("%f")*
Used by:	*SFPTsmokeFireInitiatorIntelli*

SCPTsmokeNightPreAlrmLim

Overview:

Nighttime pre-alarm limit. The nighttime pre-alarm limit sensitivity value for the fire initiator in percentage obscuration by smoke

This configuration property sets the nighttime pre-alarm limit sensitivity value for an initiator.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*140*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsmokeNightPreAlrmLim*
Default:	*0,000*
Neuron C Type:	*SNVT_smo_obscur*
Formats:	*SCPTsmokeNightPreAlrmLim: text("%f")*

Used by:	*SFPTsmokeFireInitiatorIntelli*

SCPTsmokeNomSens

Overview:

Nominal sensitivity. The nominal sensitivity value for the fire initiator in percentage obscuration by smoke

This configuration property sets the nominal sensitivity value for an initiator.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*39*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsmokeNomSens*
Default:	*0,000*
Neuron C Type:	*SNVT_smo_obscur*
Formats:	*SCPTsmokeNomSens: text("%f")*
Used by:	*SFPTsmokeFireInitiatorIntelli*

SCPTsndDelta

Overview:

.

This configuration property sets the amount by which the value obtained by a data acquisition application must change before an output network variable is updated. The data type is the same as the output network variable. For network variable types that are not zero-based (for example SNVT_temp) this value is represented as a difference in two values of the SNVT concerned. For SNVT_switch the continuous (level) part follows this convention. Thus to represent a delta value of 0.5 deg C, the SCPTsndDelta value should be set to 5. Some network monitoring and control tools may display this value as -273.5°C.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*27*

Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTsndDelta: text("%d")*
Used by:	*SFPTairVelocitySensor SFPTclosedLoopSensor SFPTglobalSolarRadiation SFPTisiLampActuator SFPTopenLoopSensor SFPTpressureSensor SFPTunitHeater SFPTutilityDataLoggerRegister SFPTutilityMeter SFPTwallUnit*

SCPTsourceAddress

Overview:

Source address.. Specifies a source address or element of an array of source addresses for and input to a functional block.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*336*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsourceAddress*
Default:	*0 0*

Neuron C Type:

Structure	
subnet	Subnet ID.. ANSI/CEA-709.1 subnet ID.
	unsigned short Minimum: *1* Maximum: *255* Invalid: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
reserved	Reserved.. Set to 0. Applications must ignore this bit.
	bitfield Signed: *no* Width: *1* Offset: *0* Minimum: *0* Maximum: *0* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
node	Node ID.. ANSI/CEA-709.1 node ID. The value 0 is invalid.
	bitfield Signed: *no* Width: *7* Offset: *1* Minimum: *0* Maximum: *127* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTsourceAddress: text("%d %d", subnet, node)

Used by:

SFPTdataLogger

SCPTspaceHumSetpoint

Overview:

Space humidification. Setpoint for the default space humidification

This configuration property sets the default space humidification setpoint.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*203*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTspaceHumSetpoint*
Default:	*30,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTspaceHumSetpoint: text("%f")*
Used by:	*SFPTdischargeAirController*

SCPTstandbyRotationalSpeed

Overview:

Standby rotational speed. The standby rotational speed in Hz

This configuration property sets the rotational speed of a mechanical part when in standby mode.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*181*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTstandbyRotationalSpeed*
Default:	*0,0*
Neuron C Type:	*SNVT_freq_hz*
Formats:	*SCPTstandbyRotationalSpeed: text("%f")*

SCPTstep

Overview:

Maximum step. The maximum step that the associated controller may take to approach the target level

This configuration property sets the maximum step that the associated controller is allowed to take to approach the target level. The bigger the steps are, the shorter the response time is for level changes.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*83*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTstep*
Default:	*0,0*
Neuron C Type:	*SNVT_lev_cont*
Formats:	*SCPTstep: text("%f")*
Used by:	*SFPTconstantLightController*

SCPTstepValue

Overview:

Step value, ramp or master fade. The step value for up/down ramps or fade control

This configuration property sets the step value for up/down ramps or fade controllers. When up/down push buttons are used, this configuration property can be used to adjust the total ramp time from 0 to 100%. The SCPTminSendTime configuration property sets the time between subsequent updates.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*92*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTstepValue*
Default:	*2,5*
Neuron C Type:	*SNVT_lev_cont*

Formats:	*SCPTstepValue: text("%f")*
Used by:	*SFPTscenePanel SFPTswitch*

SCPTstrtupDelay

Overview:

Startup delay. The time to delay after power-up, defrost, or pack fail

This configuration property sets the time delay before unrestricted control begins after power up, defrost, or pack fail.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*111*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTstrtupDelay*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTstrtupDelay: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerDefrost* *SFPTrefrigDisplayCaseControllerEvaporator*

SCPTstrtupOpen

Overview:

Startup valve opening. Maximum valve opening to use after power-up, defrost, or pack fail

This configuration property sets the maximum valve opening to use after power up, pack fail, or defrost.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*115*
Obsolete:	*no*
Size:	*2*

Programmatic Name:	*SCPTstrtupOpen*
Default:	*0,000*
Neuron C Type:	*SNVT_lev_percent*
Formats:	*SCPTstrtupOpen: text("%f")*
Used by:	*SFPTrefrigDisplayCaseControllerEvaporator*

SCPTsummerTime

Overview:

Summer time, start date and time. The start of summer time for purposes of daylight-savings time, all zeros disables

This configuration property sets the start of summer time for purposes of daylight-savings time. At the defined summer date the clock will increase its time by one hour. Setting this table to all zeroes disables daylight savings time. Year, minutes, and seconds are ignored.

The valid range for this configuration property is 1 January 0 hours, to 31 December 23 hours. Year, minutes, and seconds should be set to zero. The default value is a manufacturer-specific.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*99*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTsummerTime*
Minimum:	*0-0-0T0:0:0*
Maximum:	*0-12-31T23:0:0*
Default:	*0-1-1T0:0:0*
Neuron C Type:	*SNVT_time_stamp*
Formats:	*SCPTsummerTime: text("%d/%d/%d %d:%d:%d", year, month, day, hour, minute, second)* *SCPTsummerTime#LO: text(date(year, month, day), (" "), time(hour, minute, second))* *SCPTsummerTime#SI: text("%d-%d-%dT%d:%d:%d", year, month, day, hour, minute, second)*
Used by:	*SFPTrealTimeKeeper*

SCPTsunriseTime

Overview:

Sunrise time. Time used for sunrise-relative scheduling; must be implemented as a configuration network variable; only the time fields are used for scheduling--the date fields indicate the date used for the configured time

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*377*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTsunriseTime*
Default:	*0 0 0 0 0 0*
Neuron C Type:	*SNVT_time_stamp*
Formats:	*SCPTsunriseTime: text("%d %d %d %d %d %d", year, month, day, hour, minute, second)*
Used by:	*SFPTschedulerSimple*

SCPTsunsetTime

Overview:

Sunset time. Time used for sunset-relative scheduling; must be implemented as a configuration network variable; only the time fields are used for scheduling--the date fields indicate the date used for the configured time

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*378*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTsunsetTime*
Default:	*0 0 0 0 0 0*
Neuron C Type:	*SNVT_time_stamp*
Formats:	*SCPTsunsetTime: text("%d %d %d %d %d %d", year, month, day, hour, minute, second)*

Used by:	*SFPTschedulerSimple*

SCPTsuperHtRefInit

Overview:

Super heat reference initialization. Default value for the super heat target network variable

This configuration property sets the default value of the target super heat reference.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*114*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsuperHtRefInit*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTsuperHtRefInit#SI: text("%f", *1+0(0:854))* *SCPTsuperHtRefInit#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerEvaporator*

SCPTsuperHtRefMax

Overview:

Super heat reference maximum. Maximum value for the target super heat network variable

This configuration property sets the maximum value of the target super heat reference.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*118*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsuperHtRefMax*
Default:	*0,00*

Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTsuperHtRefMax#SI: text("%f", *1+0(0:854))*
	*SCPTsuperHtRefMax#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerEvaporator*

SCPTsuperHtRefMin

Overview:

Super heat reference minimum. Minimum value for the target super heat network variable

This configuration property sets the minimum value of the target super heat reference.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*116*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTsuperHtRefMin*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_p*
Formats:	*SCPTsuperHtRefMin#SI: text("%f", *1+0(0:854))*
	*SCPTsuperHtRefMin#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTrefrigDisplayCaseControllerEvaporator*

SCPTtempOffset

Overview:

Temperature offset. Temperature offset for calibration

This configuration property sets an offset for calibrating temperature probes. The specified offset (positive or negative) is added to the corresponding temperature measurement before it is used in any calculations or as a network output variable. This configuration property replaces the SCPToffsetTemp (70) configuration property.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*227*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTtempOffset*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_diff_p*
Formats:	*SCPTtempOffset#SI: text("%f", *1+0(0:854))* *SCPTtempOffset#US: text("%f", *1.8+0(0:855))*

SCPTtemperatureHysteresis

Overview:

Temperature hysteresis. General-purpose temperature hysteresis differential between on-point and off-point

This configuration property sets a temperature hysteresis. It provides for some number of degrees between a turn-on point and a turn-off point to avoid rapid cycling of a heating or cooling system. It is not based on any differential of time, but only a differential of temperature.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*214*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTtemperatureHysteresis*
Default:	*0,00*
Neuron C Type:	*SNVT_temp_diff_p*
Formats:	*SCPTtemperatureHysteresis#SI: text("%f", *1+0(0:854))* *SCPTtemperatureHysteresis#US: text("%f", *1.8+0(0:855))*

SCPTtermTimeTemp

Overview:

Defrost termination setting. The defrost termination condition

This configuration property determines whether defrost should terminate on time, temperature, or a combination. If the defrost was started by a scheduler, the duration is indicated in the start/stop times. If a termination condition occurs which is not the selected termination condition the functional block will generate an alarm.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*112*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTtermTimeTemp*
Default:	*DFT_TERM_TEMP*
Neuron C Type:	*SNVT_defr_term*
Formats:	*SCPTtermTimeTemp: text("%m")*
Used by:	*SFPTrefrigDisplayCaseControllerDefrost*

SCPTthermAlrmROR

Overview:

Thermal rate of change/rise trip value. The thermal alarm trip rate of rise

This configuration property sets the thermal alarm trip rate of rise for the initiator. This configuration property is manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*142*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTthermAlrmROR*
Default:	*0,0*
Neuron C Type:	*SNVT_temp_ror*
Formats:	*SCPTthermAlrmROR#SI: text("%f", *1+0(0:1012))* *SCPTthermAlrmROR#US: text("%f", *1.8+0(0:1011))*
Used by:	*SFPTthermalFireInitiator*

SCPTthermMode

Overview:

Thermostat mode. The thermostat control strategy

This configuration property sets the control strategy.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*120*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTthermMode*
Default:	*THERM_NO_CONTROL*
Neuron C Type:	*SNVT_therm_mode*
Formats:	*SCPTthermMode: text("%m")*
Used by:	*SFPTrefrigDisplayCaseControllerThermostat*

SCPTthermThreshold

Overview:

Thermal alarm trip threshold. Thermal alarm trip threshold for the initiator

This configuration property sets the thermal alarm trip threshold for an initiator. This configuration property is manufacturer-defined.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*152*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTthermThreshold*
Default:	*-274,0*
Neuron C Type:	*SNVT_temp*
Formats:	*SCPTthermThreshold#SI: text("%f", *1+0(0:854))*

	*SCPTthermThreshold#US: text("%f", *1.8+32(0:855))*
Used by:	*SFPTthermalFireInitiator*

SCPTtimeEvent

Overview:

Time event entry (record type, hour, minute, event mode) . Event or mode definitions to be transmitted if the time in the record is reached

This configuration property sets a daily schedule with an event-mode for a given time. It's used to create a configuration property array that defines a schedule. The configuration property includes record_type, hour, minute, and event_mode fields. If the record_type field is set to 0, it indicates that this element is the end of the list. A value of 1 indicates a scene, and 2 indicates a mode.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*104*
Obsolete:	*no*
Size:	*4*
Programmatic Name:	*SCPTtimeEvent*
Default:	*EMT_END_OF_LIST,0:0,0*

Neuron C Type:

Structure	
record_type	Type of time event record (time event record type names) .
	event_mode_type_t
hour	Hour (hours) .
	unsigned short Minimum: *0* Maximum: *23* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
minute	Minute (minutes) .
	unsigned short Minimum: *0* Maximum: *59* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
event_mode	Event mode information (event mode number) .
	unsigned short Minimum: *0* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTtimeEvent: text("%m,%d:%d,%d", record_type, hour, minute, event_mode)

SCPTtimeEvent#LO: text(("%m\|", record_type), time(hour, minute), ("\|%d", event_mode))

Used by:

SFPTrealTimeBasedScheduler

SCPTtimePeriod

Overview:

Historical Period. This input configuration network variable defines the period of time between transfer of a values to the historical register

This input configuration network variable defines the period of time between transfer of a values to the historical register.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*291*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTtimePeriod*
Default:	*IOM_MINUTE 0*
Neuron C Type:	Structure

units	.
	interval_of_month_t
value	.

	Union	
	minutes_interval	.
		unsigned short Minimum: *1* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	date_of_month	.
		unsigned short Minimum: *1* Maximum: *31* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	hour_of_day	.
		unsigned short Minimum: *0* Maximum: *23* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
	day_of_week	.
		days_of_week_t
	hours_interval	.
		unsigned short Minimum: *1* Maximum: *255* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:	*SCPTtimePeriod: text(("%m ", units),((units == 0) ? ("%d ", value.minutes_interval) :((units == 1) ? ("%d ", value.hours_interval) :((units == 2) ? ("%d ", value.hour_of_day) :((units == 3) ? ("%d ", value.day_of_week) :((units == 4) ? ("%d ", value.date_of_month) : ("UNKNOWN")))))))*
Used by:	*SFPTutilityMeter*

SCPTtimeSource

Overview:

Time source. Specifies the source of time

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*373*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTtimeSource*
Default:	*TMS_SCHEDULER_NV*
Neuron C Type:	*time_source_t*
Formats:	*SCPTtimeSource: text("%m")*

SCPTtimeZone

Overview:

Time zone descriptor. Time zone of node (offset from GMT, start and end of DST)

Default Value

```
SNVT_time_zone TimeZone;
TimeZone.second_time_offset = 0;
TimeZone.type_of_description = CAL_MEU;
TimeZone.hour_of_start_DST = 2;
TimeZone.minute_of_start_DST = 0;
TimeZone.second_of_start_DST = 0;
TimeZone.start_DST.M_start_DST.month_of_start_DST = 4;
TimeZone.start_DST.M_start_DST.week_of_start_DST = 1;
TimeZone.start_DST.M_start_DST.dateday_of_start_DST = 0;
TimeZone.hour_of_end_DST = 2;
TimeZone.minute_of_end_DST = 0;
```

TimeZone.second_of_end_DST = 0;
TimeZone.end_DST.M_end_DST.month_of_end_DST = 10;
TimeZone.end_DST.M_end_DST.week_of_end_DST = 5;
TimeZone.end_DST.M_end_DST.dateday_of_end_DST = 0;

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*154*
Obsolete:	*no*
Size:	*15*
Programmatic Name:	*SCPTtimeZone*
Default:	*0 CAL_MEU 2:0:0 4 1 DAY_SUN 2:0:0 10 5 DAY_SUN*
Neuron C Type:	*SNVT_time_zone*
Formats:	*SCPTtimeZone: text(("%d %m %d:%d:%d ", second_time_offset, type_of_description,hour_of_start_DST, minute_of_start_DST,second_of_start_DST), ((type_of_description == 0) ?("%d %d:%d:%d %d" , start_DST.G_day_of_start_DST, hour_of_end_DST,minute_of_end_DST, second_of_end_DST, end_DST.G_day_of_end_DST):((type_of_description == 1) ?("%d %d:%d:%d %d", start_DST.J_day_of_start_DST,hour_of_end_DST, minute_of_end_DST, second_of_end_DST, end_DST.J_day_of_end_DST): ((type_of_description == 2) ?("%d %d %m %d:%d:%d %d %d %m", start_DST.M_start_DST.month_of_start_DST, start_DST.M_start_DST.week_of_start_DST, start_DST.M_start_DST.dateday_of_start_DST,hour_of_end_DST, minute_of_end_DST, second_of_end_DST, end_DST.M_end_DST.month_of_end_DST, end_DST.M_end_DST.week_of_end_DST, end_DST.M_end_DST.dateday_of_end_DST):("UNKNOWN"))))) *SCPTtimeZone#LO: text(("%d %m ", second_time_offset, type_of_description), time(hour_of_start_DST, minute_of_start_DST, second_of_start_DST), ((type_of_description == 0) ?(" %d " ,start_DST.G_day_of_start_DST), time(hour_of_end_DST, minute_of_end_DST, second_of_end_DST), (" %d", end_DST.G_day_of_end_DST)) :((type_of_description == 1) ?(" %d ", start_DST.J_day_of_start_DST), time(hour_of_end_DST, minute_of_end_DST, second_of_end_DST), (" %d", end_DST.J_day_of_end_DST)) :((type_of_description == 2) ?(" %d %d %m ", start_DST.M_start_DST.month_of_start_DST, start_DST.M_start_DST.week_of_start_DST, start_DST.M_start_DST.dateday_of_start_DST), time(hour_of_end_DST,minute_of_end_DST,second_of_end_DST), (" %d*

%d %m", end_DST.M_end_DST.month_of_end_DST,
end_DST.M_end_DST.week_of_end_DST,
end_DST.M_end_DST.dateday_of_end_DST)) :("UNKNOWN")))))

SCPTtimeout

Overview:

Response timeout. The timeout for a controlling device to respond, during control permission request

This configuration property sets a response timeout value for a controlling device, within which a controllable device must respond, during a control permission request.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*170*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTtimeout*
Default:	*0,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTtimeout: text("%f")*

SCPTtrnsTblX

Overview:

Translation table X. Used in conjunction with Translation table Y to scale and linearize a value

This configuration property is used in conjunction with the Translation Table Y configuration property to create a translation table that determines how to scale and linearize the raw input signal as received from a hardware device such as a sensor, or how to scale or linearize actuator movement with respect to an input network variable.

There may be an unlimited number of table entries (up to the maximum number of configuration properties). A Translation Table Y value must be specified for each Translation Table X value. The default is manufacturer-defined. See SNVT_trans_table for invalid value (3) in each bitfield.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*

Index:	*28*
Obsolete:	*no*
Size:	*30*
Programmatic Name:	*SCPTtrnsTblX*
Default:	*0 0 0 0 0 0 0 0 0 0 0 0 0 0*
Neuron C Type:	*SNVT_trans_table*
Formats:	*SCPTtrnsTblX: text("%f %f %f %f %f %f %f %d %d %d %d %d %d %d", point[0], point[1], point[2], point[3], point[4], point[5], point[6], interp_pts_0_to_1, interp_pts_1_to_2, interp_pts_2_to_3, interp_pts_3_to_4, interp_pts_4_to_5, interp_pts_5_to_6, interp_pts_6_to_0)*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPThvacValvePositioner SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTtrnsTblX2

Overview:

Valve-Plug Characteristic Table X. This configuration property will be used in conjunction with the translation-table Y configuration property to create a translation table that dictates how to scale the flow with respect to the mechanical stroke

This configuration property will be used in conjunction with the translation-table Y2 configuration property to create a translation table that dictates how to scale the flow with respect to the mechanical stroke.

There may be an unlimited number of table entries (up to the maximum number of configuration properties). A Translation Table Y2 value must be specified for each Translation Table X2 value. The default is manufacturer-defined. See SNVT_trans_table for the invalid value definition (3) for each bitfield.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*285*
Obsolete:	*no*
Size:	*30*
Programmatic Name:	*SCPTtrnsTblX2*
Default:	*0 0 0 0 0 0 0 0 0 0 0 0 0 0*
Neuron C Type:	*SNVT_trans_table*

Formats:	*SCPTtrnsTblX2: text("%f %f %f %f %f %f %f %d %d %d %d %d %d %d", point[0], point[1], point[2], point[3], point[4], point[5], point[6], interp_pts_0_to_1, interp_pts_1_to_2, interp_pts_2_to_3, interp_pts_3_to_4, interp_pts_4_to_5, interp_pts_5_to_6, interp_pts_6_to_0)*
Used by:	*SFPThvacValvePositioner*

SCPTtrnsTblX3

Overview:

Combination-Flow Characteristic Table X. This configuration property will be used in conjunction with the translation-table Y configuration property to create a translation table that dictates how to scale the flow with respect to the valve capacity

This configuration property will be used in conjunction with the translation-table Y3 configuration property to create a translation table that dictates how to scale the flow with respect to the valve capacity.

There may be an unlimited number of table entries (up to the maximum number of configuration properties). A Translation Table Y3 value must be specified for each Translation Table X3 value. The default is manufacturer-defined. See SNVT_trans_table for the invalid value definition (3) for each bitfield.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*288*
Obsolete:	*no*
Size:	*30*
Programmatic Name:	*SCPTtrnsTblX3*
Default:	*0 0 0 0 0 0 0 0 0 0 0 0 0 0*
Neuron C Type:	*SNVT_trans_table*
Formats:	*SCPTtrnsTblX3: text("%f %f %f %f %f %f %f %d %d %d %d %d %d %d", point[0], point[1], point[2], point[3], point[4], point[5], point[6], interp_pts_0_to_1, interp_pts_1_to_2, interp_pts_2_to_3, interp_pts_3_to_4, interp_pts_4_to_5, interp_pts_5_to_6, interp_pts_6_to_0)*
Used by:	*SFPThvacValvePositioner*

SCPTtrnsTblY

Overview:

Translation table Y. Used in conjunction with Translation table X to scale and linearize a value

This configuration property is used in conjunction with the Translation Table X configuration property to create a translation table that determines how to scale or linearize the raw input signal as received from a hardware device such as a sensor, or how to scale or linearize actuator movement with respect to an input network variable.

There may be an unlimited number of table entries (up to the maximum number of configuration properties). A Translation Table X value must be specified for each Translation Table Y value. The default is manufacturer-defined. See SNVT_trans_table for invalid value (3) in each bitfield.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*29*
Obsolete:	*no*
Size:	*30*
Programmatic Name:	*SCPTtrnsTblY*
Default:	*0 0 0 0 0 0 0 0 0 0 0 0 0 0*
Neuron C Type:	*SNVT_trans_table*
Formats:	*SCPTtrnsTblY: text("%f %f %f %f %f %f %f %d %d %d %d %d %d %d", point[0], point[1], point[2], point[3], point[4], point[5], point[6], interp_pts_0_to_1, interp_pts_1_to_2, interp_pts_2_to_3, interp_pts_3_to_4, interp_pts_4_to_5, interp_pts_5_to_6, interp_pts_6_to_0)*
Used by:	*SFPTclosedLoopActuator SFPTclosedLoopSensor SFPThvacValvePositioner SFPTopenLoopActuator SFPTopenLoopSensor*

SCPTtrnsTblY2

Overview:

Valve-Plug Characteristic Table Y. This configuration property will be used in conjunction with the translation-table X configuration property to create a translation table that dictates how to scale the flow with respect to the mechanical stroke

This configuration property will be used in conjunction with the translation-table X2 configuration property to create a translation table that dictates how to scale the flow with

respect to the mechanical stroke.

There may be an unlimited number of table entries (up to the maximum number of configuration properties). A Translation Table X2 value must be specified for each Translation Table Y2 value. The default is manufacturer-defined. See SNVT_trans_table for the invalid value definition (3) for each bitfield.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*286*
Obsolete:	*no*
Size:	*30*
Programmatic Name:	*SCPTtrnsTblY2*
Default:	*0 0 0 0 0 0 0 0 0 0 0 0 0 0*
Neuron C Type:	*SNVT_trans_table*
Formats:	*SCPTtrnsTblY2: text("%f %f %f %f %f %f %f %d %d %d %d %d %d %d", point[0], point[1], point[2], point[3], point[4], point[5], point[6], interp_pts_0_to_1, interp_pts_1_to_2, interp_pts_2_to_3, interp_pts_3_to_4, interp_pts_4_to_5, interp_pts_5_to_6, interp_pts_6_to_0)*
Used by:	*SFPThvacValvePositioner*

SCPTtrnsTblY3

Overview:

Combination-Flow Characteristic Table Y. This configuration property will be used in conjunction with the translation-table X configuration property to create a translation table that dictates how to scale the flow with respect to the valve capacity

This configuration property will be used in conjunction with the translation-table X3 configuration property to create a translation table that dictates how to scale the flow with respect to the valve capacity.

There may be an unlimited number of table entries (up to the maximum number of configuration properties). A Translation Table X3 value must be specified for each Translation Table Y3 value. The default is manufacturer-defined. See SNVT_trans_table for the invalid value definition (3) for each bitfield.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*

Index:	*289*
Obsolete:	*no*
Size:	*30*
Programmatic Name:	*SCPTtrnsTblY3*
Default:	*0 0 0 0 0 0 0 0 0 0 0 0 0 0*
Neuron C Type:	*SNVT_trans_table*
Formats:	*SCPTtrnsTblY3: text("%f %f %f %f %f %f %f %d %d %d %d %d %d %d", point[0], point[1], point[2], point[3], point[4], point[5], point[6], interp_pts_0_to_1, interp_pts_1_to_2, interp_pts_2_to_3, interp_pts_3_to_4, interp_pts_4_to_5, interp_pts_5_to_6, interp_pts_6_to_0)*
Used by:	*SFPThvacValvePositioner*

SCPTupdateRate

Overview:

Update rate, time stamp. The update rate of the master clock to the associated network variable

This configuration property sets the update rate of a master clock output.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*98*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTupdateRate*
Default:	*60,0*
Neuron C Type:	*SNVT_time_sec*
Formats:	*SCPTupdateRate: text("%f")*
Used by:	*SFPTrealTimeKeeper*

SCPTvalueDefinition

Overview:

.

This configuration property is used to create an array of output values to be used for a schedule. A schedule time-value event specifies a value as an index into a SCPTvalueDefinition array. For example, a time-value event may specify a value of 1 a 0800 and a value of 0 at 1700. If a SCPTvalueDefinition array is declared that is called cpValueDef, the value to be produced by the scheduler at 0800 is specified by cpValueDef[1] and the value to be produced by the scheduler at 1700 is specified by cpValueDef[0].

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*276*
Obsolete:	*no*
Default:	*00*4*
Neuron C Type:	Inheriting from network variable
Formats:	*SCPTvalueDefinition: text("%d")*
Used by:	*SFPTscheduler*

SCPTvalueName

Overview:

Value name. Used to create an array of value names for each of the values defined in a SCPTvalueDefinition array

This configuration property is used to create an array of value names for each of the values defined in a SCPTvalueDefinition array. For example, a SNVT_switch value of “100.0 1” can be given a name of “On” and a value of “0.0 0” can be given a name of “Off.”

The name field contains a nul-terminated string of up to 22 characters. The default value is a nul string (all zeroes). The NUL terminator is not required if the string is 22 characters. The name may include a reference to a string defined in a language file. The 0x80 value is reserved to delimit a language string reference. See *Language File String Reference* in the SNVT_alarm_2 definition for more details.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*277*
Obsolete:	*no*
Size:	*22*
Programmatic Name:	*SCPTvalueName*
Default:	*00*22*

Neuron C Type:

Structure	
name	Name (array of 22 characters) . Nul-terminated name string of up to 22 characters. The nul terminator is not required if the string is 22 characters.
	unsigned char [22] Minimum: *32* Maximum: *126* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTvalueName: text("%c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c", name[0], name[1], name[2], name[3], name[4], name[5], name[6], name[7], name[8], name[9], name[10], name[11], name[12], name[13], name[14], name[15], name[16], name[17], name[18], name[19], name[20], name[21])

Used by:

SFPTscheduler

SCPTvalveFlowCharacteristic

Overview:

Valve flow characteristic. Actual flow characteristic of the valve

This configuration property sets the actual flow characteristic of the valve. This characteristic will build through the characteristic of the plug or ball and in the firmware implemented conversation table.

Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00-0*

Index: *248*

Obsolete: *no*

Size: *1*

Programmatic Name: *SCPTvalveFlowCharacteristic*

Minimum: *DCM_QUICK_OPEN*

Maximum: *DCM_FREE_DEFINED*

Default: *DCM_QUICK_OPEN*

Neuron C Type: *SNVT_dev_c_mode*

Formats:

SCPTvalveFlowCharacteristic: text("%m")

Used by:	*SFPThvacValvePositioner*

SCPTvalveKvs

Overview:

Valve Flow. This configuration property can be used to provide the flow through an open valve at 1 bar differential pressure

This configuration property can be used to provide the flow through an open valve at 1 bar differential pressure.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*282*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTvalveKvs*
Default:	*0*
Neuron C Type:	*SNVT_flow_p*
Formats:	*SCPTvalveKvs: text("%d")*
Used by:	*SFPThvacValvePositioner*

SCPTvalveNominalSize

Overview:

Nominal Valve Size. This configuration property can be used to provide the nominal size of the valve body

This configuration property can be used to provide the nominal size of the valve body.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*281*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTvalveNominalSize*
Default:	*0*

Neuron C Type:	*SNVT_length_mil*
Formats:	*SCPTvalveNominalSize: text("%d")*
Used by:	*SFPThvacValvePositioner*

SCPTvalveOperatingMode

Overview:

Valve operating mode. The normal operating mode of the valve

This configuration property sets the normal working mode of a valve.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*249*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTvalveOperatingMode*
Default:	*VALVE_NORMAL*
Neuron C Type:	*SNVT_valve_mode*
Formats:	*SCPTvalveOperatingMode: text("%m")*
Used by:	*SFPThvacValvePositioner*

SCPTvalveStroke

Overview:

Valve Stroke. This configuration property can be used to provide the stroke to fully open the valve

This configuration property can be used to provide the stroke to fully open the valve.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*280*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTvalveStroke*

Default:	*0*
Neuron C Type:	*SNVT_length_mil*
Formats:	*SCPTvalveStroke: text("%d")*
Used by:	*SFPThvacValvePositioner*

SCPTvalveType

Overview:

Valve Type. This configuration property can be used to provide the valve type

This configuration property can be used to provide the valve type.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*283*
Obsolete:	*no*
Size:	*1*
Programmatic Name:	*SCPTvalveType*
Minimum:	*DCM_INVFNC_QCK_OPN*
Maximum:	*DCM_INVFNC_QUAD*
Default:	*0*
Neuron C Type:	*SNVT_dev_c_mode*
Formats:	*SCPTvalveType: text("%m")*
Used by:	*SFPThvacValvePositioner*

SCPTvisOutput

Overview:

Visible light output intensity (candela) . Visible light output intensity specification of the device at 0 deg viewing angle

This configuration property sets the visible output (strobe intensity) in candelas (Cd). It contains the fire strobe output specification in candelas (Cd) at a 0 Deg viewing angle. It is manufacturer-defined.

The valid type range and raw range may be further restricted based on target market and code

jurisdiction.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*143*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTvisOutput*
Default:	*0,0*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *10000* Invalid: *65535* Scaling (A,B,C): *1, -1, 0* Scaled value: *1 *10 $^{-1}$ *(Raw+0)* Resolution: *0.10000000149012*
Formats:	*SCPTvisOutput: text("%f")*
Used by:	*SFPTvisibleFireIndicator*

SCPTweeklySchedule

Overview:

Weekly schedule. Identifies a schedule to be active for each day of the week

This configuration property is a structure containing an active schedule for each day of the week.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*278*
Obsolete:	*no*
Size:	*14*
Programmatic Name:	*SCPTweeklySchedule*
Default:	*65535 65535 65535 65535 65535 65535 65535*

Neuron C Type:

Structure	
schedule_index	Time-value index array. Identifies the starting entry of a list of time-value events in a SCPTscheduleTimeValue array. The end of the list is identified by the terminator field in the SCPTscheduleTime entry
	unsigned long [7] Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*

Formats:

SCPTweeklySchedule: text("%d %d %d %d %d %d %d", schedule_index[0], schedule_index[1], schedule_index[2], schedule_index[3], schedule_index[4], schedule_index[5], schedule_index[6])

Used by:

SFPTscheduler

SCPTwinterTime

Overview:

Winter time, start date and time. The start of winter time for purposes of daylight-savings time, all zeros disables

This configuration property sets the start of wintertime for purposes of daylight-savings time. At the defined winter date the clock will decrease its time by one hour. Setting this table to all zeroes disables daylight savings time. Year, minutes, and seconds are ignored.

The valid range for this configuration property is 1 January 0 hours, to 31 December 23 hours. Year, minutes, and seconds should be set to zero. The default value is a manufacturer-specific.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*100*
Obsolete:	*no*
Size:	*7*
Programmatic Name:	*SCPTwinterTime*
Minimum:	*0-0-0T0:0:0*
Maximum:	*0-12-31T23:0:0*

Default:	*0-1-1T0:0:0*
Neuron C Type:	*SNVT_time_stamp*
Formats:	*SCPTwinterTime: text("%d/%d/%d %d:%d:%d", year, month, day, hour, minute, second)* *SCPTwinterTime#LO: text(date(year, month, day), (" "), time(hour, minute, second))* *SCPTwinterTime#SI: text("%d-%d-%dT%d:%d:%d", year, month, day, hour, minute, second)*
Used by:	*SFPTrealTimeKeeper*

SCPTzoneNum

Overview:

Zone number (zone number) . The zone number for the device

This configuration property sets the zone number for a device.

Details:

Resource Set:	*Standard 00:00:00:00:00:00:00:00-0*
Index:	*141*
Obsolete:	*no*
Size:	*2*
Programmatic Name:	*SCPTzoneNum*
Default:	*0*
Neuron C Type:	unsigned long Minimum: *0* Maximum: *65535* Invalid: *65535* Scaling (A,B,C): *1, 0, 0* Scaled value: *1 *10 ^{0} *(Raw+0)* Resolution: *1*
Formats:	*SCPTzoneNum: text("%d")*
Used by:	*SFPTaudibleFireIndicator SFPTelevatorFireSystemsPort SFPTfireSmokeDamperActuator SFPTpullStationFireInitiator SFPTsmokeFireInitiatorConvent SFPTsmokeFireInitiatorIntelli SFPTthermalFireInitiator SFPTuniversalFireIndicator*