Calypso Action Control Language (CACL)

Version: **15.0** 20 Oct 2016

Command		Description
ccc..@{macroname}..ccc		macro replacement

Macro rules:

Macros can be used to substitute common text within an action string. There are some limitations as to where macros can be used:
  -They can be used for an entire command;
  -They can be used anywhere within the command brackets, e.g. #CMD[..in here..], except
  -They can contain a quoted string, but cannot cross the quote boundaries.
In short, a macro cannot contain text which crosses either a square bracket ([..]) boundary or a quoted string ("..", or '..') boundary.
In addition, macro substitution does not currently occur inside quoted strings.

A common use for macros would be to define a network device:
  NETDEV => I192.168.1.101,P7262
and then use the macro in all relevant actions containing NET commands:
  #NET[F1, @{NETDEV}, "power on"];
Such an arrangement would allow the target network device to be changed by simply changing the macro definition, rather than requiring all the various NET commands to be edited with the new IP info.

Command		Description
ccc..,*nm*=arg,..ccc..|..ccc		style string (character sequence)
Style specifications:		
no	reset to base style (must be first within a given style).	
bc=xxx	set background color; uses a 12-bit hexadecimal RGB argument much like that used in web-based CCS color speicifications (e.g. 00f is bright blue, fff is white, 000 is black).	
fc=xxx	set foreground (or font) color; uses same RGB argument as bc.	
ff=nam	set font family/face using 3-character name abbreviation nam: for example the generic families 'ser' (serif), 'san' (sans-serif), and 'mon' (monospaced); or specific families like 'tmr' (timesroman) or 'hlv' (helvetica) (which are highly device dependent).	
fs=c	set font size in a relative manner indicated by character c: '+' is next defined larger size, '-' is next defined smaller size, 's' is defined small font, 'n' is defined normal/base font, and 'l' is defined large font. The actual font size used (in pixels or points) depends on the display destination.	
fn	set font-style to normal (resets from fb, fi, and/or fu).	
fb	set font-style to bold (cumulative with fi or fu).	
fi	set font-style to italic (cumulative with fb or fu).	
fu	set font-style to underlined (cumulative with fb or fi).	
ta=c	set text alignment as indicated by character c: 'c' is centered, 'l' is left-aligned, and 'r' is right-aligned.	
A style string can contain multiple styles, referenced by their order in the string and separated by a pipe ('|') character. Each style can contain multiple style specifications, each separated by a comma (',') character. Styles are accumulative, but can be reset to the base if the first spec is no. For example `fc=0f0,fb|fc=f00` defines two styles (\s01 and \s02 because of their order) where \s01 uses a bold green font and \s02 uses a bold red font (still bold because it inherits the fb from the previous style). Any publication text using this style set would initially use \s01 unless the text style was set explicitly to something else. Not all devices support all (or sometimes any) style specs, but unsupported styles are simply ignored in such cases. There should be no spaces in a style string.		

Command		Description
ccc..\ esc..ccc		non-hex mode formatted string (character sequence)
Escape sequences:		
\a	insert character 0x07 (the bell character) into the string character sequence.	
\b	insert character 0x08 (the backspace character) into the string character sequence.	
\f	insert character 0x0C (the form-feed character) into the string character sequence.	
\n	insert character 0x0A (the newline character) into the string character sequence.	
\r	insert character 0x0D (the carriage return character) into the string character sequence.	
\snn	insert character 0x01 (SOH) followed by the character 0xnn into the string character sequence (for devices supporting the #PUB command only).	
\vn	insert a base-10 string representation of stored value n into the string character sequence; this escape sequence can result in multiple characters being inserted.	
\vxn	insert a base-16 string representation of stored value n into the string character sequence; this escape sequence results in 4 characters being inserted.	
\xnn	insert character 0xnn into the string character sequence (allows for a single hexadecimal-value character to be inserted).	
\xvn.0	insert the low-order byte of stored value n into the string character sequence.	
\xvn.1	insert the high-order byte of stored value n into the string character sequence.	
\c	insert the character c into the string character sequence; e.g. "\\"; this is the result if c is not one of the characters listed above.	

Command		Description
#AUDn[*tag*];		command context only
#AUDn[*I*v, *M*v, *V*v];		action or command context
n	a decimal value between 1 and maxAUD, which targets a specific audio device.	
tag	a sequence of printable characters delimited by double quotes; the sequence, if used, is returned as the prefix to the status data; it should be a fairly short string (~10 chars), anything too long will be truncated. (Query mode ONLY)	
Iv	input control; controls the current input (or channel); if v is either '+' or '-', then the input is incremented or decremented, otherwise if v is a decimal value between 1 and maxInput then that input/channel is made active.	
Mv	mute control; controls the current mute state; if v is either '+' or '-', then the mute state is toggled, otherwise if v is 0 muting is disabled and if it is 1 muting is enabled.	
Vv	volume control; controls the current volume level; if v is either '+' or '-', then the volume is incremented or decremented (by the next natural internal volume step), and if the +/- is followed by a decimal value (between 1 and 9) then the change is that number of steps (e.g. 'V+3' increments 3 steps); otherwise if v is a decimal value between 0 and 100 then the volume level is set proportionately (0 is minimum, 100 is maximum volume).	
In query mode, the data returned is in the format: `tagII M VVV` where the optional tag comes first, then the input value (0-99, with 0 = no input active), the mute state (0 = mute OFF (audible), 1 = mute ON (silent)), and the volume (0-100). Values which take up fewer than the specified digits are left-adjusted and space-filled. For example, "`CM:1 0 100`" or "`2 0 5 `" would be possible returns. The actual input values that can be returned is dependent on the specific device.		

Command		Description
#AINn[*Iv, tag***];**		command context only
#AINn[Iv, Lv];		action or command context
n	a decimal value between 1 and maxAUD, which targets a specific audio device.	
tag	a sequence of printable characters delimited by double quotes; the sequence, if used, is returned as the prefix to the status data; it should be a fairly short string (~10 chars), anything too long will be truncated. (Query mode ONLY)	
Iv	input selector; specifies the input (or channel) to act upon; v is a decimal value between 1 and maxInput.	
Lv	level control; controls the current input level; if v is either '+' or '-', then the level is incremented or decremented (by the next natural internal step), and if the +/- is followed by a decimal value (between 1 and 9) then the change is that number of steps (e.g. 'V+3' increments 3 steps); otherwise if v is a decimal value between 0 and 100 then the level is set proportionately (0 is minimum, 100 is maximum level).	

In query mode, the data returned is in the following formats:
#AINn[tag] (no input specified):

```
tagNN
```

where the optional tag comes first, then the maximum input value (0-99);
#AINn[In,tag] (input specified):

```
tagNN LLL
```

where the optional tag comes first, then the value of the specified input (0-99), and then the level (0-100). Values which take up fewer than the specified digits are left-adjusted and space-filled. For example, "`CM:1 100`" or "`2  5  `" would be possible returns. The actual input values that can be returned is dependent on the specific device.

Command		Description
#VIDn[*tag*];		command context only
#VIDn[*I*v, *B*v, *F*v];		action or command context
n	a decimal value between 1 and maxVID, which targets a specific video device.	
tag	a sequence of printable characters delimited by double quotes; the sequence, if used, is returned as the prefix to the status data; it should be a fairly short string (~10 chars), anything too long will be truncated. (Query mode ONLY)	
Iv	input control; controls the current input (or channel); if v is either '+' or '-', then the input is incremented or decremented, otherwise if v is a decimal value between 1 and maxInput then that input/channel is made active.	
Bv	blank control; controls the current blank-video state; if v is either '+' or '-', then the blank state is toggled, otherwise if v is 0 blank is disabled and if it is 1 blank is enabled.	
Fv	freeze control; controls the current freeze-video state; if v is either '+' or '-', then the freeze state is toggled, otherwise if v is 0 freeze is disabled and if it is 1 freeze is enabled.	

In query mode, the data returned is in the format:

```
tagII B F
```

where the optional tag comes first, then the input value (0-99, with 0 = no input active), the blank state (0 = blank OFF (visible), 1 = blank ON (blank)), and the freeze state (0 = freeze OFF (moving), 1 = freeze ON (frozen)). Values which take up fewer than the specified digits are left-adjusted and space-filled. For example, "`CM:1 0 1`" or "`2 0 0`" would be possible returns. The actual input values that can be returned is dependent on the specific device.

Command		Description
#GPIn**[***F*n**];**		command context only
#GPIn**[**string**];**		trigger context only
n	a decimal value between 1 and maxGPI, which targets a specific input connection.	
string	a sequence of printable characters delimited by double quotes; the sequence should be either "OPEN" or "CLOSE" (case insensitive).	
Fn	flag setting; n indicates the error return mode.	

Command		Description
#GPOn**[***F*n**];**		command context only
#GPOn**[**string**, *D*n**];**		action or command context
n	a decimal value between 1 and maxGPO, which targets a specific relay connection.	
string	a sequence of printable characters delimited by double quotes; the sequence should be either "OPEN" or "CLOSE" (case insensitive).	
Dn	duration of pulse; causes the relay to temporarily change to the given state for a period of n units of time, and then change to the opposite state. (1 unit = 4.4 ms, 2 <= n <= 65535)	
Fn	flag setting; n indicates the error return mode.	

Command		Description
#ROTn**[];**		command context only
#ROTn**[**string**, *L*n, *H*n**];**		trigger context only
n	a decimal value between 1 and maxROT, which targets a specific rotary connection.	
string	a sequence of printable characters delimited by double quotes; the sequence should be either "CC" or "CW" (case insensitive), meaning counter-clockwise and clockwise, respectively.	
Ln	low (minimum) tick count; will only trigger if rotary dial has been rotated n ticks or more. Default is 1.	
Hn	high (maximum) tick count; will only trigger if rotary dial has been rotated, but not more than n ticks. Maximum setting is 254, default is 0 (meaning no max).	

Command		Description
#SDT[Ttgt, *S*n, *I*n**];**		command context only
#SDT[Ttgt, *S*n, *I*n**,** string**];**		trigger context only
tgt	a 3-character designator which targets a specific detection type/device. The current legal targets are: AUD = audio signal detection VID = video sync detection STM = stream detection	
Sn	source selection; if there are multiple targets of the given type, this allows selection between them. A value of 0 indicates "any" of the matching targets; this is the default and is currently the only available source.	
In	input selection; if there are multiple multiple inputs on the given target, this allows selection between them. A value of 0 indicates "any" of the inputs; this is the default and is currently the only available input setting.	
string	a sequence of printable characters delimited by double quotes; the sequence should be either "ON" or "OFF" (case insensitive).	
The SDT command is a general state/signal detection trigger command. Its use is highly dependent upon the implementation in the device, including which targets are available and the configuration of those targets.		

Command		Description
#COMn[*Cn, Tn, Wn***];**		command context only
#COMn[*flags*, string**];**		action or command context
n	a decimal value between 1 and maxCOM, which targets a specific serial port connection.	
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence may include special characters by using certain escape (backslash) sequences; or the entire string may be encoded as hexadecimal character pairs (ref hex flag below).	
Flags:		
Cn	clear buffer; if not specified, then the receive buffer is cleared prior to any transmit; otherwise: 0 = don't clear the receive buffer 1 = clear the buffer prior to any transmit and/or after data is received	
Dn	delay period; specifies a delay after transmission for a period of n units of time. (0 <= n <= 255; for older devices 1 unit = 4.4 ms, on newer devices 1 unit = 0.1 sec)	
Fn	flag setting; n indicates the error return mode [IGNORED].	
M1	master mode; causes ALL serial ports to transmit the string.	
Tn	type of data string; indicates the type of data returned and/or expected: 1 = standard string data, the default 2 = hexadecimal encoded data	
Wn	wait period; specifies a wait before grabbing the receive buffer for a period of n seconds. Note that this will cause the command to return "received" data, even when also transmitting data, and therefore should NOT be used in action context.	

Command		Description
#CMPn[*flags*, string**];**		trigger context only
n	a decimal value which targets a specific serial port connection; currently only serial port 1 may be used for this command.	
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence may include special characters by using certain escape (backslash) sequences; or the entire string may be encoded as hexadecimal character pairs (ref hex flag below).	
Flags:		
Cn	type of match string; indicates the type of data expected: 1 = standard string data, the default 2 = hexadecimal encoded data	
Ln	match length expected; a non-zero value means the number of bytes remaining in the receive buffer starting from the match position must be n; a value of zero means this test will not be performed (length doesn't matter). A non-zero value must always be greater than or equal to the length of the match string.	
Mn	match position; a non-zero value means start the match at 1-relative position n in the receive buffer; a value of zero means the match will succeed if the match string is found anywhere in the receive buffer.	

Command		Description
#XRIn[*Fn*]**;**		command context only
#XRIn[*F1*, string]**;**		trigger context only
n	a decimal value between 1 and maxIRin, which targets a specific IR input line (most products have only a single IR input line with multiple connection points).	
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence must be a valid CIRT code (two hexadecimal pairs, case insensitive).	
Fn	flag for type; n indicates the type of IR data returned or expected: 1 = CIRT data, which is the default and is the only setting allowed for triggers 2 = Universal data (grab first or only burst) 4 = Noise data 6 = Universal data, skip first burst (grab second burst) 7 = Universal data, skip first 2 bursts (grab third burst) Only the F1 setting may be used for triggers, and may be omitted entirely since it is the default. The other settings may be used in learn command context.	

Command		Description
#XROn[*flags*, string]**;**		action or command context
n	a decimal value between 1 and maxIRout, which targets a specific IR output line.	
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence may be either a valid CIRT code or a valid Universal code including the Calypso header data. (Must be pairs of hexadecimal characters, case insensitive.)	
Flags:		
Bn	break period; specifies an inter-command idle period of n units of time. (1 unit = 4.4 ms, 0 <= n <= 255)	
Fn	flag for type; n indicates the type of IR data returned or expected: 1 = CIRT data, which is the default on devices which support it 2 = Universal data No other flag values are valid for output.	
Rn	repeat count; specifies the number of times to repeat the command (with a break between each repeat, as specified above).	

Command		Description
#MSG[address**,** subject**,** message**];**		action or command context [SMTP only]
address	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence should form a valid Email address, e.g. name@somewhere.domain.	
subject	a short sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched.	
message	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence may include special characters by using certain escape (backslash) sequences.	

Command		Description
#EVTn**[***flags***];**		command context only
n	a decimal value between 0 and maxEvent, which targets a specific database event.	
Flags:		
Fx	flag setting; a pair of hexadecimal characters which specify a set of bit flag values, all OR-ed together: 01 = run action 1 02 = run action 2 04 = run action 3 08 = run action 4 10 = obey the enable state of the event (don't run it if it is disabled) The resulting flag value may NOT be zero. By default this command will ignore the enable state of the event. The flag value "F0F" is the default setting and means run all actions.	
Wn	wait period; specifies a wait of n seconds before invoking the event. (Max 255 sec.)	
The details of use for this command may vary across devices and time. The most recent implementation of the command is not available as part of an action. In addition, the 'run action' flags are ignored for devices using the ACA architecture (where actions are defined separately from events).		

Command		Description
#NET[*flags*, *Aaddr* or *Iaddr*, string**];**		action or command context
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence may include special characters by using certain escape (backslash) sequences; or the entire string may be encoded as hexadecimal character pairs (ref hex flag below). In the special ICMP "ping" mode, the string is optional but when used specifies the expected sequence number (in ASCII).	
Aaddr	address string (no quotes); specifies a DNS-style named address, e.g. "ion.calypsocontrol.com". One, and only one, of the Aaddr or Iaddr arguments must be given.	
Iaddr	IP address string (no quotes); specifies an IPv4-style numeric address, e.g. "192.168.1.101". One, and only one, of the Iaddr or Aaddr arguments must be given.	
Flags:		
Fn	flag setting; specifies the output mode: 1 = use TCP; the default 2 = use UDP 3 = use CDP (a UDP-based Calypso protocol) 4 = use ICMP (ping) [Note: port and string parameters have special meanings] 5 = use ARP (announce only) [Note: ignores all other parameters]	
Pn	port number; specifies a target port number to be used for the network communication (uses the currently configured Remote Port by default). In the special ICMP "ping" mode, the port is REQUIRED and specifies the return address "slot" to use (1 <= n <= maxPings (5)).	
Tn	type of data string; indicates the type of data expected: 1 = standard string data, the default 2 = hexadecimal encoded data	

Command		Description
#HTP[uri-spec, *flags***];**		action or command context
uri-spec	a sequence of printable characters forming a valid URI specification. The sequence must be a fully formed URI (Uniform Resource Identifier). Currenly only the 'http:' scheme is supported.	
Flags:		
Pstring	POST data; string must be a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The entire string must be URI-encoded. If this option is used an HTTP POST request will be sent, otherwise an HTTP GET request will be sent (where any data must be in the URI string).	
This command is essentially a shorthand for making HTTP network requests. The same results can be obtained by using the NET command, but the format is a little more cumbersome in that case.		

Command		Description
#NCM[*flags*, string**];**		trigger context only
#NCM[F4, *pingflags*, string**];**		trigger context only; special ICMP ("ping") mode
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence may include special characters by using certain escape (backslash) sequences; or the entire string may be encoded as hexadecimal character pairs (ref hex flag below). In the special "ping" mode, the string is optional but when used specifies the expected sequence number (in ASCII).	
Flags:		
C**n**	type of match string; indicates the type of data expected: 1 = standard string data, the default 2 = hexadecimal encoded data	
L**n**	match length expected; a non-zero value means the number of bytes remaining in the receive buffer starting from the match position must be **n**; a value of zero means this test will not be performed (length doesn't matter). A non-zero value must always be greater than or equal to the length of the match string but less than 255.	
M**n**	match position; a non-zero value means start the match at 1-relative position **n** in the receive buffer; a value of zero means the match will succeed if the match string is found anywhere in the receive buffer.	
Ping Flags:		
F4	Indicates use of the special "ping" mode for testing.	
Aaddr	address string (no quotes); specifies a DNS-style named address, e.g. "ion.calypsocontrol.com". Use only one of the Aaddr or Iaddr arguments; if neither is given, this will match ping responses from ANY remote host (on this network).	
Iaddr	IP address string (no quotes); specifies an IPv4-style numeric address, e.g. "192.168.1.101". Use only one of the Iaddr or Aaddr arguments should be given, or neither.	
Note that since the network receive buffer is filled from the same port as that on which CNAP commands are received, strings longer than 6 characters which also begin with '#' will not be seen as potential NCM strings. Instead, such strings will be treated as potential CNAP commands and processed accordingly.		

Command		Description
#TMR[Wweek**];**		trigger context only
#TMR[Dday**];**		trigger context only
#TMR[Hhour**];**		trigger context only
#TMR[Mmodulus, *O*offset**];**		trigger context only
week	a numeric time value; specifies a specific time of the week in day:hour:min form, where 1=Sunday and 7=Saturday, e.g. 6:13:45 means every Friday at 1:45 pm, while 1:0:0 means every Sunday at midnight.	
day	a numeric time value; specifies a specific time of the day in hour:min form, e.g. 18:30 means every day at 6:30 pm.	
hour	a numeric time value; specifies a specific time in the hour in minutes, e.g. 15 means every hour at a quarter after the hour.	
modulus	a numeric time value; specifies a cycle period in minutes, e.g. 60 means every hour, 1440 mean every day, 10 means every 10 minutes.	
offset	a numeric time value; specifies an offset in minutes from the top of the hour but within the modulus, e.g. offset = 10 with modulus = 60 means 10 minutes after every hour. By default the offset is 0.	
Note that the W, D, and H formats are easy to use shorthands for certain modulus + offset pairs. In particular, the character specifies the modulus and the value is converted into an offset in minutes. So W3:0:10 (every Tues at 10 after midnight) is the same as M10080, O2890.		

Command		Description
#VALn**[***F2***];**		(GET mode) command context only
#VALn**[***F1*, *func*value**];**		(SET mode) action or command context
n	a decimal value between 1 and maxVAL, which targets a specific stored value. (The values maxVAL+1 .. maxVAL+5 are also available, but only for the GET mode. See discussion under #AQS command.)	
value	a numeric value; if the value string begins with "0x" the rest of the string is interpreted as a hexadecimal number. The maximum value a stored value can hold is 0xFFFF (a 16-bit unsigned value).	
Fn	flag setting; specifies various control options: 1 = allow function to cause a rollover, e.g. 0xFFFF + 1 = 0; default is no rollover, e.g. 0xFFFF + 1 = 0xFFFF 2 = return current value in hexadecimal format; default is to return the value in decimal format	
Functions		
+	add the value to the current stored value.	
-	subtract the value from the current stored value.	
&	bitwise AND the value with the current stored value.	
|	bitwise OR the value with the current stored value.	
^	bitwise exclusive-OR (XOR) the value with the current stored value.	
none	if no function is specified, set the stored value to the given value.	

Command		Description
#TSTn**[***func*value**];**		trigger context only
#TSTn**[**value**-**value**];**		trigger context only
n	a decimal value between 1 and maxVAL, which targets a specific stored value. (The values greater than maxVAL are NOT available, use #AQT instead.)	
value	a numeric value; if the value string begins with "0x" the rest of the string is interpreted as a hexadecimal number. The maximum value a stored value can hold is 0xFFFF (a 16-bit unsigned value).	
Functions		
=	true if the current stored value is EQUAL to the given value.	
!	true if the current stored value is NOT EQUAL to the given value.	
<	true if the current stored value is LESS THAN the given value.	
>	true if the current stored value is GREATER THAN the given value.	
-	true if the current stored value is BETWEEN the 2 given values, inclusive.	

Command		Description
#ALM[];		command context only
#ALM[string, *D*n, *L*n, *S*n**];**		action or command context
string	a sequence of printable characters delimited by quotes; the sequence should be one of "ON", "BEEP" or "OFF" (case insensitive); some devices may also support "WARN", and possibly "RING" and "BUSY".	
Dn	duration of alarm; causes the alarm to temporarily change to the given state for a period of n units of time, and then turn OFF. (1 unit = 4.4 ms, 2 <= n <= 65535)	
Ln	light mode; specifies the mode to be used: 0 = no lights (allow sound only), the default 1 = lights on 2 = lights blink This setting is ignored for "OFF".	
Sn	sound mode; specifies the mode to be used: 0 = no sound (allow lights only) 1 = higher frequency tone, the default 2 = lower frequency tone This setting is ignored for "OFF".	

Command		Description
#PAS[];		command context only
#PAS[string**];**		action or command context
#PAS["WFR", *W*n**];**		action or command context
string	a sequence of printable characters delimited by quotes; the sequence should be one of "WFR", "CAS", "ADS", "AAS", "HPS" or "OFF" (case insensitive).	
Wn	wait period for wait-for-reply (WFR); specifies the period for which WFR will remain active, in n minutes, and then turn become inactive. When not specified, the default period is 5 minutes. (1 <= n <= 1000)	
The wait-for-reply (WFR) option allows dynamic control over whether the "ADS" and "AAS" messages will be recognized (or ignored). When those messages are configured to recognize the WFR state, then they will be ignored unless * a #PAS["WFR"] command has previously been received AND * the WFR wait period has not expired AND * no previous similar message has been received within that period.		

Command		Description
#AQTn[*flags*, *func*value**];**		trigger context only
#AQTn[*flags*, value-value**];**		trigger context only
#AQTn[*flags***];**		trigger context only
n	a decimal reference value between 1 and 4 (maxAQ), which targets the specific auto-query return value. The auto-query functionality is designed with projector devices in mind, so queries are always translated into the following reference locations: 1 = Power 2 = Input Source 3 = Lamp Hours 4 = Filter Hours The special value 0 may also be used, but ONLY to test global connection status and/or whether any of the return values have changed (i.e. global change status). Note that not all references will hold valid data in all cases (e.g. some projectors may not have input source information available). See the discussion under the AQS command for more on interpreting translated values.	
value	a numeric value; if the value string begins with "0x" the rest of the string is interpreted as a hexadecimal number. The maximum value a stored value can hold is 0xFFFF (a 16-bit unsigned value).	
Functions		
=	true if the current stored value is EQUAL to the given value.	
!	true if the current stored value is NOT EQUAL to the given value.	
<	true if the current stored value is LESS THAN the given value.	
>	true if the current stored value is GREATER THAN the given value.	
-	true if the current stored value is BETWEEN the 2 given values, inclusive.	
Flags:		
Cn	test the connection state: 1 = connected / connection established 2 = no connection / connection lost If specified, this test is next in priority after the "new/change" test.	
Nn	test whether the value is new (i.e. has changed) since the last trigger cycle: 1 = new / changed 2 = not new / no change If specified, this test takes first priority.	
Note that when flags are specified, the tests are effectively ANDed together; so for example #AQT3[C1,N1,>1000] will trigger if the connection is established and the lamp hour value has changed and that value is greater than 1000. The flags are optional except in the case of #AQT0, where no function is allowed.		

Command		Description
#AQSn[*flags*];		command context only
n	a decimal reference value between 1 and 4 (maxAQ), which targets the specific auto-query return value. The auto-query functionality is designed with projector devices in mind, so queries are always translated into the following reference locations: 1 = Power 2 = Input Source 3 = Lamp Hours 4 = Filter Hours The special value 0 may also be used, but ONLY to test global connection status and/or whether any of the return values have changed (i.e. global change status). Note that when raw data is requested (i.e. the F0 flag), the above reference locations do not necessarity apply. In that case the actual untranslated data is returned, and may not have the same meaning as the defined translation location (e.g. a raw reference 2 may return the raw lamp hrs value, even though translation will put this in reference location 3). Translated data references are of consistent meaning--though not all references will have valid data for all configurations (e.g. some projectors do not return input source data). The meaning of raw data references depends on the current auto-query configuration.	
Flags:		
Fn	data format: 0 = state byte plus value (hexadecimal, raw, little-endian) 1 = value only (decimal, translated) - non-global only 2 = (default) value only (hexadecimal, translated, big-endian) If the 0 reference is used, the F1 setting may NOT be used, while F2 will return 8 bytes of data and F0 will return 12 bytes (see discussion below).	
Pn	get the data from the specified polling level: 0 = no polling -- cause the auto-queries to fire and return the most current data 1 = return the most recent data from the hardware polling subsystem 2 = (default) return the most recent data from the event trigger subsystem	

The translated auto-query values are stored at the event trigger level and so are also available through the #VALn command (GET mode only) using values of n starting at maxVAL + 1:
51 = Power (0 = unknown, 1 = OFF, 2 = ON, 3 = STDBY, 0xFFFF = disconnect/error)
52 = Input Source (1 = first source, 2 = second source, etc; 4 is the usual max)
53 = Lamp Hours
54 = Filter Hours
55 = Flag states for the above 4 queries
The flag states are reported in 2-bits each--with values as described below for the global raw "state" values--where
bits 0,1 -> query 1
bits 2,3 -> query 2
bits 4.5 -> query 3
bits 6,7 -> query 4
Hexadecimal data, raw or translated, will be returned using 2 hex digits per byte, e.g. "a4" representing decimal value 164.
When global translated status values are requested, e.g. #AQS0[F2,P0], the returned query-data will include all 4 values in 8 bytes (big-endian format):
byte 1 = Power value, most-significant byte
byte 2 = Power value, least-significant byte		

byte 3 = Input Source value, most-significant byte
byte 4 = Input Source value, least-significant byte
byte 5 = Lamp Hours value, most-significant byte
byte 6 = Lamp Hours value, least-significant byte
byte 7 = Filter Hours value, most-significant byte
byte 8 = Filter Hours value, least-significant byte

When global raw status values are requested, e.g. #AQS0[F0,P0], the returned query-data will contain 12 hexadecimal bytes with the following meanings:

byte 1 = Query 1 state (1 = valid, 2 = invalid, 3 = disconnected)
byte 2 = Query 1 raw value, least-significant byte
byte 3 = Query 1 raw value, most-significant byte
byte 4 = Query 2 state (1 = valid, 2 = invalid)
byte 5 = Query 2 raw value, least-significant byte
byte 6 = Query 2 raw value, most-significant byte
byte 7 = Query 3 state (1 = valid, 2 = invalid)
byte 8 = Query 3 raw value, least-significant byte
byte 9 = Query 3 raw value, most-significant byte
byte 10 = Query 4 state (1 = valid, 2 = invalid)
byte 11 = Query 4 raw value, least-significant byte
byte 12 = Query 4 raw value, most-significant byte

Command		Description
#STMn**[];**		command context only
#STMn**[**state, *flags*]**;**		action or command context
n	a decimal value between 1 and maxSTM, which targets a specific stream.	
state	a sequence of printable characters delimited by double quotes; the sequence should be either "ON" or "OFF" (case insensitive). Must come first, and if "OFF", then no other options should follow.	
Flags:		
Bn	byte-order of packets; specified by n: 1 = use least-significant-first order (little-endian) 2 = use most-significant-first order (big-endian) Not valid for the RTP protocol since this value is predetermined in that case.	
Cn	channel count; specifies the number of channels to be used for the stream. Not valid for the RTP protocol since this value is predetermined in that case.	
Fn	flag setting; specifies the output protocol: 1 = use Raw PCM 2 = use RTP (L16 custom)	
Hhexstring	custom RTP header; hexstring must be a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The entire string must be encoded as hexadecimal character pairs, case insensitive. Only valid for the RTP protocol. The entire custom part of the header must be included (the bytes following the extended length field).	
Iaddr	IP address string (no quotes); specifies an IPv4-style numeric address, e.g. "192.168.1.101". Note that the network part of the address must be the same as for the device itself, but the address must not be a self-reference. By default (if this option is not given) the network broadcast address is used, e.g. "192.168.1.255".	
Pn	port number; specifies a target port number to be used for the stream.	
Rn	sample rate; specifies the sample rate in kHz to be used for the stream. Which sample rates are supported is device dependent.	
Sn	size of packets; specifies the maximum number of audio data bytes in each output packet.	
Any flag options which are missing will result in the use of configured default values for those options. In query mode, the returned string will be either "ON" or "OFF" (witout the quotes).		

Command		Description
#PUBn**[***F0, flags, Sstylespec*, **string];**		action or command context
#PUBn**[F1, L**n*, flags, Sstylespec*, **string];**		action or command context
#PUBn**[F2, L**n*, flags*];**		action or command context
n	a decimal reference value between 1 and maxPUB, which targets a specific display or display method; maxPUB varies by device: on the CB-6000, for example, 1 implies use the current screen (don't change screens), while 2 will force a switch to the "big" screen.	
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. The sequence may include special characters by using certain escape (backslash) sequences. Beyond the usual escape sequences, the sequence '\sxx' indicates a switch to style 'xx', where 'xx' is 2 hexidecimal characters which reference a style (see below).	
Fn	function setting; specifies the reference mode: 0 = temporary, n reference ignored; the default 1 = reference a text publication; "show", "set", or "clear" (see below) 2 = reference a graphic publication; available for "show" only	
Sstylespec	style spec, which is a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. If a style spec is included, it MUST come before the text string. This parameter is illegal when the F2 graphic reference is used. Note: style handling is dependent on the device; if not supported this param is silently ignored.	
Flags:		
Ln	local location; specifies a decimal reference value between 1 and maxREF, which targets a specific locally stored publication; maxREF varies depending on the device and F flag; illegal in the case of F0.	
Pn	priority number; specifies a priority to be used for the specified publication. This can affect how the publication is displayed. A priority in the range 0 <= n <= 5 is considered informational, while a priority in the range 6 <= n <= 10 is considered an alert. Priorities in the 11 <= n <= 15 range are special case alerts. If not given, a default priority of 1 is used. A publication request with a lower priority than the current publication will be ignored.	
Tn	timeout period; specifies the number of seconds to publish the specified information. A value of 0 (the default) implies no timeout. (0 <= n <= 65535 sec)	
A combination of the F parameter and the string control how this command will operate. Since graphic publications cannot take a string, the F2 setting can only be used to show a publication (no set or clear). When F1 is used, then the operation depends on the string: if the string is not given, then it will be a show operation; if the string is empty (""), then it will be a clear-location operation; and if the string is not empty, it will be a set-location operation. For F2, the set and clear operations work on the stored data in the background; only the show operation causes an immediate visual change. When F0 is used, or no F parameter is specified, then only the clear and show operations are possible (there is no set for temporary publications). For this function, both show and clear cause an immediate visual change if the priority is sufficient (for show) or the current string is temporary (for clear).		

Command		Description
#PWR[string **or** bit-flag**];**		action or command context
string	a sequence of printable characters delimited by double quotes; the sequence should be one of "ON" or "OFF" (case insensitive); "OFF" puts the device into its minimum power usage state, while "ON" powers up all components.	
bit-flag	a hexidecimal string, e.g. 0x03, which specifies a bit-flag controlling which components should be powered up/down; for example, if there are two components whose power can be controlled, a flag of 0x01 would power down component 2 and power up component 1. If a bit is "on" for a non-existent component, the command will mis-parse.	

Command		Description
#SLP[Tn**];**		action or command context
Tn	sleep time; n specifies the amount of time to sleep, in 0.1 second increments, 1-65535.	
The sleep command is generally useful only in events with multiple actions which are defined to run in order. It will NOT enforce any delay between separate events.		

Command		Description
#TRG[state**,** list**];**		action or command context
state	a sequence of printable characters delimited by double quotes; the sequence should be either "ON" or "OFF" (case insensitive). Must come first.	
list	a comma-separated list of event ranges; each range can be either a single event number, or a dash-separated range. For example, "3, 5, 20-25, 31, 36-42".	
Temporarily changes the "enable" state of the specified events to the given state. On reset or power-cycle, the state of the events will return to that saved in the database.		

Command		Description
#OPTn**[P**n**];**		command context only
#OPTn**[P**n**, I**n**];**		action or command context
#OPTn**[P**n**,** string**];**		action or command context
n	a decimal value between 1 and maxOPT, which targets a specific option.	
Pn	parameter specifier (indicates which option parameter is being set); this is an ID (1 <= n <= 65535) which is specific to the device.	
string	a sequence of printable characters delimited by quotes and denoting a string quotes may be used, as long as they are matched. The sequence may include special characters by using certain escape (backslash) sequences.	
In	an integer value (0 <= n <= 65535) for the specified paramter	

Command		Description
#RECn[];		command context only
#RECn[string];		action or command context
n	a decimal value between 1 and maxREC, which targets a specific recording device.	
string	a sequence of printable characters delimited by quotes; the sequence should be one of "OFF", "ON", or "HOLD" (case insensitive).	
The "HOLD" string puts the specified recording in a paused state. Depending on the underlying device, this may fail if the recording is not already active. Similarly, the READ format of the command may not work on all devices.		

Command		Description
#MEMn**[R**s**, A**addr**];**		(READ mode) command context only
#MEMn**[W**s**, A**addr**,** value**];**		(WRITE mode) action or command context
n	a decimal value between 1 and maxMEM, which targets a specific memory component.	
value	a numeric value; if the value string begins with "0x" the rest of the string is interpreted as a hexadecimal number.	
Aaddr	memory address specified in hex; the number of bytes specified should reflect the address size, e.g. A2F would be address 2F in an 8-bit memory space, while A0000002F would be the same address in a 32-bit space.	
Rs	read mode; n specifies the read size in bytes, 1-8 depending on device.	
Ws	write mode; n specifies the read size in bytes, 1-8 depending on device.	
This is a special purpose command and should be used with care.		

Command		Description
#ICCn**[***flags*, string**];**		non-URI command context only
n	a decimal value between 1 and maxICC, which targets a specific ICC component.	
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. By default, the entire string must be encoded as hexadecimal character pairs, case insensitive. If the type flag is set for a standard string (see below), then stanard ASCII characters can be used; in this case special characters may be included by using certain escape (backslash) sequences.	
Flags:		
Cn	clear receive buffer; note that, if there is data to transmit, then the receive buffer is always cleared prior to the transmit; otherwise: 0 = don't clear the buffer after data is received (default) 1 = clear after data is received	
Tn	type of data string; indicates the type of data returned and/or expected: 1 = standard string data 2 = hexadecimal encoded data, the default	
This is a special purpose command and should be used with care.		

Command		Description
#CTL[string**];**		non-URI command context only
string	a sequence of printable characters delimited by quotes; either single or double quotes may be used, as long as they are matched. Case sensitive. Must be one of: "CLR" = clear the command queue (often used to terminate an infinite loop); "COMSTAT:n" = show serial port info / status flags (outputs-no n: the port numbers available, e.g. '0 1', '1', '1 2', etc.); (output n: 'TX[(data-len)@(data-index) (status),(delay)]' 'RX[(data-len)/(max-len)@(data-position) -(lost-chars)] 0x(flags)' "ICCSTAT" = show the IntraComm processing status info (outputs '(icc-stat-flag) (active-cmd)-(active-subcmd) (timeout)((retry-cnt)) (..other dev-specific info..)'); "NETSTAT" = show the network processing status flags (outputs '(rmt-tcp-stat-flag) (hicp-tcp-stat-flag)'); "QSTAT" = show the command and web queue "allocation" flags (outputs '0x(pkt-use-flags) ((net-cnt)n, (rsrv-cnt)r) ((evt-used-cnt)ue, (evt-rsrvd-cnt)ur) 0x(web-use-flags) ah->(run-head) fh->(free-head)'); "QDATAn" = show the command queue data for entry n (a decmimal integer 0 - maxQ) (outputs 'cq: (n state qref) -> (next); pkt: (len typ dev src tag timeout stat)'; most values are in hex); "RESETCONFIG" = send a command to the ICC layer to reset all its EEPROM data back to factory default values. "SERIALPGMn" = send a command to the ICC layer to enable(1)/disable(0) the program mode avaliable through the serial port (n = 0 or 1). "SOFTRESET" = cause the primary processor (ZWorld/Microchip) code to reset itself. "SYSTEMRESET" = cause the entire system to reset itself.	
This is a special purpose command and should be used with care. Which subcommands are supported depends on the device.		

Command		Description
#WHO[*flags***];**		command context only
Flags:		
S<u>str</u>	specifies the separator string, <u>str</u>; the default is to use a single tab character (char(0x09)). The string cannot be more than 15 characters in length.	
I	include the IP Address.	
R	include the Remote Port preference (port 7262 will always respond to WHO).	
M	include the MAC Address.	
D	include the Device name.	
V	include the Version of the firmware.	
C	include the CACL version.	
N	include the Name assigned by the user.	
Note that when flags are specified, the order of the flags does not effect the field order in the output. Unless a field is dropped because it is not included, the order of the fields is: IPAddr RmtPort MACAddr DevName Version CACL Name and all fields are separated by the separator string (see above). If no flags are specified, all fields are returned. The maximum length of the returned string buffer is WHO_RESPONSE_LENGTH (128), so in some rare cases the user assigned name could be truncated.		