



# HALO Smart Sensor BASIC API Developer's Guide v2.3

## Forward

This document describes the group of facilities of the Halo Smart Sensor known collectively as the **BASIC** API, or Application Programming Interface. This discussion is intended for use by programmers or integrators who are interested in integrating one or more HALO Smart Sensors (HALOs) with 3<sup>rd</sup> party (non-IPVideo) software components or systems.

In general, the HALO API is intended to transfer information efficiently from HALO over a conventional Ethernet network to the external program. To accomplish this goal, the API is divided into three sections: Event Driven Socket Connection, Heartbeat Socket Connection, and Event Data URL. BACnet interface is also present and covered in a separate document.

## API Design

The API is designed using industry standard formats such as TCP/IP, HTTP, HTTPS, and JSON. The design does not require any special or proprietary techniques or libraries to be used in the development of the external program or application.

The API is flexible and can be configured and programmed to deliver exactly the data required and in the most efficient manner. The details of the operation of each of the above sections are covered in the following sections of this guide.

## External Messaging

This facility is used to deliver alerts or alarms and Event data to an external program, VMS system, server, etc. when an Event is triggered (is set). Optional messages can also be enabled to signal when an Event clears (is reset). This delivery can be made to a TCP/IP socket or HTTP/S server in real time. There is a range of configurable protocols with customizable contents. Authentication and encryption are available.

## Heartbeat

Heartbeat messages are sent at a configurable interval (instead of when Events are triggered) to provide proof of live/availability. They have a similar range of capabilities as External Messaging but would typically be configured to contain general state information rather than details about a particular event.

## Event Data URL

This facility is only available under an NDA and should only be used when the external program requires access to any and all the Event values, thresholds, and state flags. This data is generally retrieved on demand by the external program but not at a very high frequency. This method generally incurs some latency when a modest polling rate is used. Typical polling rates range from once per minute to once per 5 seconds with an absolute maximum rate of once per second. This method can also be used to retrieve additional supporting data when an Event (alert) is received.



# HALO Smart Sensor BASIC API Developer's Guide v2.3

## External Messaging details

A section of the HALO web interface Integration popup provides for configuration of a single 3<sup>rd</sup> party connection where various values can be sent to a remote TCP socket or an HTTP/HTTPS server. Place holders (tokens) are used to insert live values into the transmitted text. Although labeled "External Messaging," this channel can be used for almost any purpose requiring real time Event triggers, actively delivered by HALO. This arrangement is quite flexible because selections on the "Actions" determine which HALO Events transmit through this channel.

### Actions

Event Identifier	Email Set	Email Reset	Msg Set	Msg Reset	Relay
Aggression	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---
AQI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	---
Gunshot	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---

In HTTP mode, the Set and Reset Strings are the URLs that must be entered and formatted as required by the desired destination server. A User and Password field can be used for authentication. See HTTP mode below.

### Integration

#### External Messaging

Protocol  TCP  HTTP      Repeat Holdoff 5 sec

Set String   On  Off

Reset String   On  Off

**Above you can use:**

%NAME% - device name	%THR% - event threshold	%FWVER% - firmware version
%IP% - ip address	%VAL% - sensor value	\\ - 1 back slash
%MAC% - mac address	%DATE% - local date of event	\n - new line
%EID% - event id	%TIME% - local time of event	\r - carriage return
%SOURCE% - data source	%PSWD% - password	\u### - hex char code
	%USER% - user	etc.

User  Password



## HALO Smart Sensor BASIC API Developer's Guide v2.3

In TCP mode, the Set and Reset Strings are only the data of a single message that is sent to the receiving TCP socket. They can be formatted as needed by the destination. The destination is specified in the Address and Port fields. See TCP Mode below.

### Integration

#### External Messaging

Protocol  TCP  HTTP Repeat Holdoff 5 sec

Set String `location=%NAME%_event=%EID%_detected=true`  On  Off

Reset String `{"location":"%NAME%" "event":"%EID%" "detected":false}`  On  Off

**Above you can use:**

%NAME% - device name	%THR% - event threshold	%FWVER% - firmware version
%IP% - ip address	%VAL% - sensor value	\\ - 1 back slash
%MAC% - mac address	%DATE% - local date of event	\n - new line
%EID% - event id	%TIME% - local time of event	\r - carriage return
%SOURCE% - data source	%PSWD% - password	\u#### - hex char code
	%USER% - user	etc.

Address 0.0.0.0 Port 1234

For either mode, status from the most recent message is displayed which can aid in fixing connection or other issues. You could use the Event TEST buttons on the Actions popup to force a message:

Status: Connection refused

Global On/Off for Set or Reset must be On to enable those types of messages. Reset is often not used because only the onset of an Event is of interest, but that can vary. Each Event can independently specify if it will use either the Set or the Reset message on the Actions popup.

The eyeball buttons will display a rough representation of what is sent after keyword substitutions and formatting.

Repeat Holdoff can be used to throttle frequent messages by delaying before another one can be sent. This is done independently per Event. HALO has a built-in hold time for events of 15 seconds to prevent rapid retriggering of Events. If you wanted to ensure not more than 1 Event of a type is sent per minute, you could set Repeat Holdoff to 60 (seconds).



## HALO Smart Sensor BASIC API Developer's Guide v2.3

### Heartbeat details

The Heartbeat transmissions function in a similar way to the above except that there is no interaction with the Actions page. Instead, the Heartbeat transmission occurs on a regular basis as configured with the Interval field,

In HTTP mode, the Set and Reset Strings are the URLs that must be entered and formatted as required by the desired destination server. A User and Password field can be used for authentication. See HTTP mode below.

### Heartbeat

Protocol  TCP  HTTP

Message `https://server.com/heartbeat?  
location=%NAME%,Temp=%SENSOR:F%,RH=%SENSOR:RH%,Lux=%SENSOR:Lux%,TVOC=%SENSOR:TVOC%,CO2eq=%SENSOR:CO2eq%,PM2.5=%SENSOR:PM2.5%,PM10=%SENSOR:PM10%,NH3=%SENSOR:NH3%,NO2=%SENSOR:NO2%,CO=%SENSOR:CO%,Noise=%SENSOR:Noise%,Move=%SENSOR:Move%,AQI=%SENSOR:AQI%,Triggered=%ACTIVE%`  On  Off

**Above you can use:**

%NAME% - device name	%EVENTS% - list of event states	%TIME% - local time
%IP% - ip address	%EVENTVALS% - list of event values	\\ - 1 back slash
%MAC% - mac address	%ACTIVE% - list of active events	\n - new line
%FWVER% - firmware version	%SENSOR:id% - value of selected sensor	\r - carriage return
	%DATE% - local date	\u#### - hex char code

Interval 60 sec User Password

Save

While the primary purpose of the Heartbeat is to provide *proof of life* of a HALO Smart Sensor to a remote application, this message can also be used to transmit selected sensors or current Event state information. The example above sends a long string parameter with the URL that include the Halo name, a majority of sensor values, and lastly Triggered=%ACTIVE% which could be empty or a list of currently triggered Events.

### HTTP (and HTTPS) Mode (for both External Messaging and Heartbeat)

External Messaging and Heartbeat strings can be http: or https: URLs as needed. Path and parameters can be entered as needed by the destination server. Keywords like %NAME% (HALO device name) or %EID% (Event id) can be inserted as needed and will be replaced with respective data when the message is sent. A list of commonly used keywords are shown for quick reference.



## HALO Smart Sensor BASIC API Developer's Guide v2.3

The URL path can contain keywords as well as the parameters to the URL. The parameters might be NAME=VALUE pairs or a JSON object, or a custom format depending on the destination server.

Examples for External Messaging would include %EID% to indicate the Event that triggered:

`https://server.com/event/%NAME%/%EID%`

`https://server.com/event?location=%NAME%&event=%EID%`

`https://server.com/event>{"location": "%NAME%", "event": "%EID%"}`

Examples for Heartbeat might add %ACTIVE% (currently triggered Events) or a sensor value:

`https://server.com/alive?location=%NAME%&Triggered=%ACTIVE%`

`https://server.com/event>{"location": "%NAME%", "NH3": %SENSOR:NH3%}`

The %SENSOR:...% values use the names found in the right-hand sensor column headings in evtYYYYMMDD.csv log files. They are typically:

%SENSOR:C%	%SENSOR:NO2%	%SENSOR:ACCY%
%SENSOR:F%	%SENSOR:CO%	%SENSOR:ACCZ%
%SENSOR:RH%	%SENSOR:Noise%	%SENSOR:Move%
%SENSOR:Lux%	%SENSOR:HGMic%	%SENSOR:P-Hg%
%SENSOR:TVOC%	%SENSOR:LGMic%	%SENSOR:P-mb%
%SENSOR:CO2eq%	%SENSOR:KW1%	%SENSOR:AQI%
%SENSOR:PM1%	%SENSOR:KW2%	%SENSOR:NO2AQI%
%SENSOR:PM2.5%	%SENSOR:KW3%	%SENSOR:COAQI%
%SENSOR:PM10%	%SENSOR:Gun%	%SENSOR:PM10AQI%
%SENSOR:NH3%	%SENSOR:AccX%	%SENSOR:PM25AQI%

If the destination server prefers HTTP PUT or POST instead of GET requests, you can prefix the URL with PUT: or POST:. Independently, you can add a JSON payload which is popular with many servers by adding the [JSONBODY] keyword followed by a JSON formatted object. Example:

`PUT:https://server.com/event[JSONBODY>{"location": "%NAME%", "event": "%EID%"}`

The URL supports typical IP address (and IPv6) and port and user-password options, or you can use the User and Password fields if needed by the destination server for authentication methods like Basic or Digest:

`https://username:password@123.321.123.321:9876/event...`

### TCP Mode (for both External Messaging and Heartbeat)

External Messaging and Heartbeat strings are just for data since the Address and Port fields specify the destination. The Address supports names, IPv4 and IPv6.

The string can be formatted like the data portions of HTTP messages described above, or as required by the destination server.

Examples for External Messaging would include %EID% to indicate the Event that triggered:

`location=%NAME%,event=%EID%`

`{"location": "%NAME%", "event": "%EID%"}`

Examples for Heartbeat might add %ACTIVE% (currently triggered Events) or a sensor value:

`location=%NAME%&Triggered=%ACTIVE%`



# HALO Smart Sensor BASIC API Developer's Guide v2.3

{ "location": "%NAME%", "NH3": "%SENSOR:NH3%" }

## Actions

Event Identifier	Email Set	Email Reset	Msg Set	Msg Reset	Relay 1	Relay 2	LED Color	LED Pattern	LED Priority	Sound	Actions
Aggression	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---	---	1 Sec Blink	Hig	---	Test
AQI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---	---	Steady	Hig	---	Test
Gunshot	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---	Blue	1 Sec Blink	Hig	Emergency2C2N.wav	Test
Help	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---	Viole	1 Sec Blink	Hig	Emergency2C2N.wav	Test
Masking	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---	---	Steady	Hig	---	Test
Tamper	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---	Red	1 Sec Blink	Hig	Siren-Euro.wav	Test
THC	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---	---	Steady	Hig	---	Test
Vape	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	---	---	---	Steady	Hig	---	Test

Speaker Volume: 86%

Upload sound file:

Delete sound file:

Checkboxes in the "Msg Set" and "Msg Reset" columns determine which Events trigger sending. More on the setup of Events and Actions is available in the HALO Administrator's Guide.

### Delivery of JSON Event Messages

Some developers prefer to receive Event data formatted as industry standard self-labeled JSON rather than plain ASCII text as the former is more reliably and more easily parsed.

On the HALO web page "Messaging" tab, you can supply JSON messages in "External Messaging" settings "Set String" and "Reset String" and in the "Heartbeat" "Message."

Examples:

#### "External Messaging" Settings Set String:

```
{ "device": "%NAME%", "event": "%EID%", "alarm": "yes" }
```

This will send a single TCP or UDP JSON message to the specified server reporting the friendly device name, event name and that it just started.

#### "External Messaging" Settings Reset String:

```
{ "device": "%NAME%", "event": "%EID%", "alarm": "no" }
```

This will send a single TCP or UDP JSON message to the specified server reporting the friendly device name, event name and that the condition has now stopped.



## HALO Smart Sensor BASIC API Developer's Guide v2.3

### **"Heartbeat" Message:**

```
{ "device": "%NAME%", "alive": "%DATE% %TIME%" }
```

This will periodically send a TCP or UDP JSON message to the specified server reporting that the HALO is alive at the indicated time.