



HCA Tech Note 613

Habitat Integration (updated 26-Mar-20)

Habitat is an integrator of various protocols that uses a hub that provides a protocol independent interface to devices of different types – a bit like HCA. The Habitat app allows you to add, view, and modify devices, and then to perform simple actions on them. The advantage of having HCA support Habitat is that all the device types they support are then available to HCA.

For HCA support, a user must have a Habitat hub installed. Any devices that will be controlled by Habitat need to be added to their system so the hub can control and receive signals from them.

NOTE: Habitat works in two ways: either local control or control via the Habitat Cloud. The HCA implementation uses Habitat only in local mode. This requires that you determine the IP address of the Habitat hub and ensure it doesn't change. Investigate what facilities your network router has for maintaining the same IP address for a device over time. Some routers call this "IP Reservations" or provides some facility in the LAN setup section. Each router is different but most modern ones have this ability somewhere in the settings. To receive events from the Habitat hub into HCA, the IP address of the computer running the HCA Server must also not change.

Follow the instructions in the Habitat documentation to setup your hub. In the portal.habitat.com website there is a "Find Hubs" button. This locates your hub IP address.



HCA Tech Note 613

The screenshot shows a web browser window for the 'Hubitat Elevation Portal' at IP address 192.168.0.31. The title bar says 'Main Menu'. The left sidebar has a dark background with white icons and text:

- Devices
- Apps
- Settings
- Advanced
- Apps Code
- Drivers Code
- System Events
- Logs

The main content area is titled 'Welcome to your Elevated Home.' It features several cards:

- Register Your Hub on Portal (with 'Register Hub' button)
- View your Location Settings and Modes (with 'Location and Modes' button)
- View all your connected devices (with 'Devices' button)
- Add new Apps and Show Installed Apps (with 'Apps' button)
- Access hub settings (with 'Settings' button)
- Discover new LAN, Zigbee and Z-Wave Devices (with 'Discover Devices' button)
- Documentation (with a question mark icon)

At the bottom, it says 'Location: Country Home' and includes links for Terms of Service, Documentation, Community, Support, and Copyright 2019 Hubitat, Inc.

Take note of the IP address shown. Next, click on “Apps” in the left menu.

TechSup@HCATech.com

www.hcatech.com



HCA Tech Note 613

The screenshot shows a web browser window for the Hubitat Elevation Portal. The URL is 192.168.0.31/installedapp/list. The left sidebar has a navigation menu with items: Devices, Apps (selected), Settings, Advanced, Apps Code, Drivers Code, System Events, and Logs. The main content area is titled 'Apps' and displays a message: 'No Apps are currently installed.' Below this, it says: 'Below are some of the Apps that are built in. To Install an app, click the "Add Built-In App" button above.' A large button labeled '+ Add Built-In App' is visible. On the right, there are four app cards:

- Rule Machine®**
Full rules engine to create simple to complex automation.
- Amazon Echo App**
Control devices and change modes with your voice
- Button Controller**
Use multi-button controllers to control any of your devices
- Hubitat® Safety Monitor**
Monitor for intrusion, smoke and water; alerts for safety
- Groups and Scenes**
A Group of lights can operate together, and a Scene sets lights to preset levels and colors.
- Hubitat® Simple Lighting**
Create simple lighting automations

At the bottom of the page, the location is set to 'Country Home' and there are links for Terms of Service, Documentation, Community, Support, and Copyright 2019 Hubitat, Inc.

Press the “Add Built-in App” button.

TechSup@HCATech.com

www.hcatech.com



HCA Tech Note 613

The screenshot shows the Habitat Elevation Portal interface. The left sidebar includes options like Devices, Apps (selected), Settings, Apps Code, Drivers Code, System Events, and Logs. The main content area displays a list of built-in apps:

- Maker API**: Allows for simple GET based requests to endpoints to get selected devices information.
- Mirror**: Slave Devices from Master.
- Mode Manager**: Manage automatic mode settings.
- Motion Lighting Apps**: Motion Lighting for motion activated lighting, and Mode Lighting for switch/button activated lighting.
- Notifications**

A modal dialog titled "Install New Built-In App" is open, containing the same list of apps. At the bottom of the modal are "Close" and "Install" buttons. Below the modal, there are two sections: "Habitat™ Simple Lighting" (Create simple lighting automations) and "Groups and Scenes" (A Group of lights can operate together, and a Scene sets lights to preset levels and colors). The status bar at the bottom shows "Location: Country Home" and the URL "192.168.0.31/installapp/create/28".

Select the “Maker API”

TechSup@HCATech.com

www.hcatech.com



HCA Tech Note 613

The screenshot shows a web browser window titled "Hubitat Elevation Portal - Find M..." with the URL "Not secure | 192.168.0.31/installedapp/configure/1/mainPage". The main content is the "Maker API" configuration page. On the left, there's a sidebar with icons for Devices, Apps, Settings, Advanced, Apps Code, Drivers Code, System Events, and Logs. The "Apps" icon is highlighted. The main area has a title "App List" with "«" and "»" buttons, a gear icon, and a question mark icon. Below this, it says "Maker API Label" and "Maker API". Under "Security", there are two toggle switches: "Allow Access via Remote / Cloud" (off) and "Allow Access via Local IP Address" (on). A section for "Allowed Hosts (for CORS)" includes a text input field with placeholder text "Enable these hosts (comma separated) for CORS support, ex. 'http://localhost:8080'" and a "Click to set" link. Under "Enable Logging", there's a toggle switch for "Enable Logging for debugging" (off). At the bottom, it says "Allow Endpoint to Control These Devices." The footer contains links for Terms of Service, Documentation, Community, Support, and Copyright 2019 Hubitat, Inc.

Slide the option “Allow Access via Local IP Address” to the right to enable it.



HCA Tech Note 613

The screenshot shows a web browser window titled "Maker API" with the URL "Not secure | 192.168.0.31/installedapp/configure/1/mainPage". The left sidebar has a "Devices" icon, followed by "Apps" (selected), "Settings", "Advanced", "Apps Code", "Drivers Code", "System Events", and "Logs". The main content area is titled "Maker API". It contains a text input for "Enable these hosts (comma separated) for CORS support, ex. 'http://localhost:8080'" with the placeholder "Click to set". Below it is a section for "Enable Logging" with a toggle switch labeled "Enable Logging for debugging". A "Select Devices" dropdown menu is open, showing two checked checkboxes: "Toggle All On/Off" and "Office Lights". An "Update" button is next to the dropdown. A warning message at the bottom states: "WARNING: Any device selected here will have full access to its attributes and events as well as have the ability to send any command to the device". At the bottom of the page, there are links for "Location: Country Home", "Terms of Service", "Documentation", "Community", "Support", and "Copyright 2019 Hubitat, Inc."

Scroll down and tick the checkbox next to the device you want to allow HCA to control.

TechSup@HCATech.com

www.hcatech.com



HCA Tech Note 613

The screenshot shows a web browser window with the following details:

- Title Bar:** Habitat Elevation Portal - Find M... (tab 1), Maker API (tab 2), 192.168.0.31/apps/api/1/devices/1/mainPage (active tab).
- Address Bar:** 192.168.0.31/installledapp/configure/1/mainPage
- Toolbar:** Apps, Bookmarks, Home Control Assis..., HCAAdmin Main M..., Anti-Spam Filtering..., JSON Parser Online, Other bookmarks.
- Left Sidebar (Habitat Elevation Portal):**
 - Devices
 - Apps
 - Settings
 - Advanced
 - Apps Code
 - Drivers Code
 - System Events
 - Logs
- Middle Content Area (Maker API):**
 - Office Lights**
 - WARNING:** Any device selected here will have full access to its attributes and events as well as have the ability to send any command to the device
 - Endpoints**
 - Local URLs**
 - URLS:**
 - Get All Devices**
 - Get All Devices with Full Details**
 - Get Device Info (replace [Device ID] with actual subscribed device id)**
http://192.168.0.31/apps/api/1/devices/[Device ID]?access_token=09d46f42-7cf3-4bd2-ac8b-0fef46606c51
 - Get Device Event History (replace [Device ID] with actual subscribed device id)**
http://192.168.0.31/apps/api/1/devices/[Device ID]/events?access_token=09d46f42-7cf3-4bd2-ac8b-0fef46606c51
 - Get Device Commands (replace [Device ID] with actual subscribed device id)**
http://192.168.0.31/apps/api/1/devices/[Device ID]/commands?access_token=09d46f42-7cf3-4bd2-ac8b-0fef46606c51
- Bottom Navigation:** Location: Country Home, Terms of Service, Documentation, Community, Support, Copyright 2019 Habitat, Inc.

Again, scroll down and take note of two key pieces of info:

1. Look at any of the URLs it displays and extract the number between the “/api/” and the “/devices”. In the above example that is “1” but yours may be different.
2. Extract the text from the end of the URL following the “access_token=”. In the above example, it is: 09d46f42-7cf3-4bd2-ac8b-0fef46606c51 but yours will be different.

You may want to cut and paste both items and save in a text file for now.

TechSup@HCATech.com

www.hcatech.com



HCA Tech Note 613

The screenshot shows a web browser window with the following details:

- Title Bar:** Habitat Elevation Portal - Find My Home, Maker API, 192.168.0.31/apps/api/1/devices/[Device ID]
- Address Bar:** Not secure | 192.168.0.31/installedapp/configure/1/mainPage
- Bookmarks Bar:** Apps, Bookmarks, Home Control Assis..., HCAAdmin Main M..., Anti-Spam Filtering..., Json Parser Online
- Content Area:**
 - Left Sidebar (Devices):** Devices, Apps, Settings (Advanced: Apps Code, Drivers Code), System Events, Logs.
 - Middle Content:** **Maker API** section with API endpoints:
 - Get Device Event History (replace [Device ID] with actual subscribed device id)
http://192.168.0.31/apps/api/1/devices/[Device ID]?access_token=09d46f42-7cf3-4bd2-ac8b-0fef46606c51
 - Get Device Commands (replace [Device ID] with actual subscribed device id)
http://192.168.0.31/apps/api/1/devices/[Device ID]/commands?
access_token=09d46f42-7cf3-4bd2-ac8b-0fef46606c51
 - Get Device Capabilities (replace [Device ID] with actual subscribed device id)
http://192.168.0.31/apps/api/1/devices/[Device ID]/capabilities?
access_token=09d46f42-7cf3-4bd2-ac8b-0fef46606c51
 - Send Device Command (replace [Device ID] with actual subscribed device id and [Command] with a supported command. Supports optional [Secondary value])
http://192.168.0.31/apps/api/1/devices/[Device ID]/[Command]?access_token=09d46f42-7cf3-4bd2-ac8b-0fef46606c51
 - Buttons:** Remove (red button), Done (grey button).

Scroll down to the bottom and press the “Done” button.

TechSup@HCATech.com

www.hcatech.com



HCA Tech Note 613

Info	Name	Type
	Maker API	Maker API

Showing 1 to 1 of 1 entries

Location: Country Home Terms of Service Documentation Community Support Copyright 2019 Hubitat, Inc.

The display updates to show that the Maker API has been added.

Adding the Hubitat Package

For HCA to discover and operate all the devices controlled by Hubitat, you next add to your design a package from the online library. Open the library browser (“Design” ribbon category, “Library Browser/Import” button) and locate the Hubitat package and add that to your design.

TechSup@HCATech.com

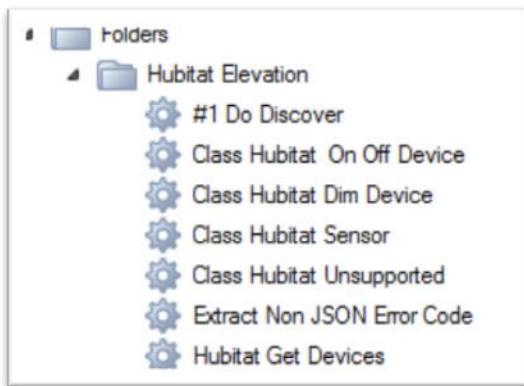
www.hcatech.com



HCA Tech Note 613

Name	Description	Version	Author	Added
Garage Door Monitor	Monitor garage doors and send an SMS message w...	1	Mark Stayton	2018-10-17 23:10:14
Home Mode for External Assistants	Make the HCA Home Mode available to external as...	1	Mark Stayton	2018-10-17 22:30:47
Home Mode Restore	Saves the home mode across HCA restarts	6	HCA Central	2019-06-25 13:13:14
Hubitat	Classes in support of the Hubitat integrating hub	1 (BETA)	HCA Central	2019-09-02 19:30:48
LEGACY Auto-Off	A set of programs that allows users of one-way X10 ...	1	Don M	2018-10-18 02:18:18
LIFX	Implements LIFX color changing light bulbs	1	HCA Central	2018-09-17 13:52:44
List Enumerator Template	Template for enumerating HCA objects using _Desig...	1	Mark Stayton	2018-10-17 22:31:48

The import creates a “Hubitat” folder containing these programs:



NOTE: there may be more or fewer of these programs as the package evolves.

Before doing anything else, open the Global variables inventory - “Control” ribbon category, “Tools” panel, “Variables” button.

There are three variables to update:

- HE_AccessToken
- HE_APICode
- HE_HubAddress

When adding the Hubitat package these variables are created, but all have empty values. Start with HE_HubAddress, and in the right section of the dialog change the values like this:



HCA Tech Note 613

The screenshot shows the 'Variables' section of the HCA software. A variable named 'HE_HubAddress' is being configured. The 'Name:' field contains 'HE_HubAddress'. The 'Current Value:' field contains '192.168.0.31', which is highlighted with a red arrow. Below it, the 'Variable usage or description:' field contains the text 'Hubitat hub IP address. Find from your router and configure the router to not change the IP ad'. Under 'When HCA loads the design file', the radio button 'Set it to this value' is selected (indicated by a red circle) and its value '192.168.0.31' is also highlighted with a red arrow. At the bottom are 'Save changes' and 'Delete' buttons, with a red arrow pointing to the 'Save changes' button.

Enter the IP address in the “Current Value”, choose the “Set it to this value” option and then enter the same address in the edit below that option. Then press the “Save Changes” button.

Set the “HE_AccessToken” and “HE_APICode” in the same way. Both the values came from what you cut from the example URLs when enabling the “Maker API” as described above. Make sure that you enter the values correctly. When all done, close the variables inventory.

Next, start the program “Hubitat Elevation - #1 Do Discover” to locate your Hubitat managed devices and to add them to your HCA design.



HCA Tech Note 613

This program locates all the devices managed by Hubitat Elevation and adds them to your HCA design. Before using this program, you must install the Hubitat hub and add your devices with the Hubitat browser interface.

Unfortunatly Hubitat doesn't have a concept of rooms like HCA does, so all your devices are added to the folder "Hubitat Devices". You can move them into other rooms or change their names and the connection to the device will not be lost.

Continue **Stop**

Press *Continue* and in a few seconds the devices are found.

These devices and the class they are managed by are now available in your design:

Hubitat Devices - Office Lights [Hubitat Dim Device]

I understand

The Hubitat has no concept of rooms, so all the newly discovered devices are added to a room called "Hubitat Devices". **You can move them to new rooms or change their names and the connection to the Hubitat device will not be lost.** The connection between the actual device and HCA is shown on the "User Class Object" tab of the device's properties.

"Hubitat Devices - Office Lights" Properties

Name	Notes	Room	Type	User Class Object	Poll	Transmission	Triggers	Restart	Icon	Display	Log
------	-------	------	------	-------------------	------	--------------	----------	---------	------	---------	-----

This device is controlled by a User Implemented Class. Information on this class and what the ID should contain is available in the User Object Class Manager accessible though the Protocols ribbon category.

ID of this device: 1

TechSup@HCATech.com

www.hcatech.com



HCA Tech Note 613

That ID is the way that HCA identifies each device when talking to the Hubitat hub. You can't change the ID.

How to control the initial icon choice

Every device that is imported is assigned an icon. That icon choice is controlled by the kind of device it is. For example, a lock gets a lock icon, a motion sensor gets a motion sensor icon. For other types a similar choice is made. If you want to override these choices for your import, on the “Tags” tab of the “Hubitat Get Devices” program you can make that mapping from type to icon. The tag name is the device type, and the tag value is the icon name. If you don’t use the tags to make changes, here are the icon selected for each type.

lock	Lock
presence	Room occupied
motion	Motion Sensor
contact	Door
water	Water Heater
switch / dim	Track Lights
nonDim	Relay A
button	CGrey
unknown	CRed

Using Hubitat devices

Once the devices are added you can control them like any other HCA device – ON, OFF, set to a percentage if dimmable, schedule it, use in programs, etc. Each action is handled by one of the class programs in the package.

For devices of the “Hubitat Sensor” class you can get their status.

Any devices assigned to the “Hubitat Unsupported” class will be non-functional as this class is a placeholder. See the next section.

NOTE: If a device goes offline, unfortunately HCA has no way of knowing this. When controlled or when requesting state, the Hubitat hub returns no error.



HCA Tech Note 613

Hubitat Device Types

The problem (and feature) of Hubitat is that it supports so much. Each device is represented by info that the server provides showing the “components” and “capabilities” of a device. Hubitat doesn’t provide a simple “type” – from some pre-determined list of available types – and isn’t immediately obvious what you can do with a device unless the capability list is examined. The HCA program that discovers your Hubitat devices attempts to understand the capabilities of each device and assigns that device to a class that handles devices with those capabilities.

At this point these classes were created, and each Hubitat device is assigned to one of those classes. The classes are:

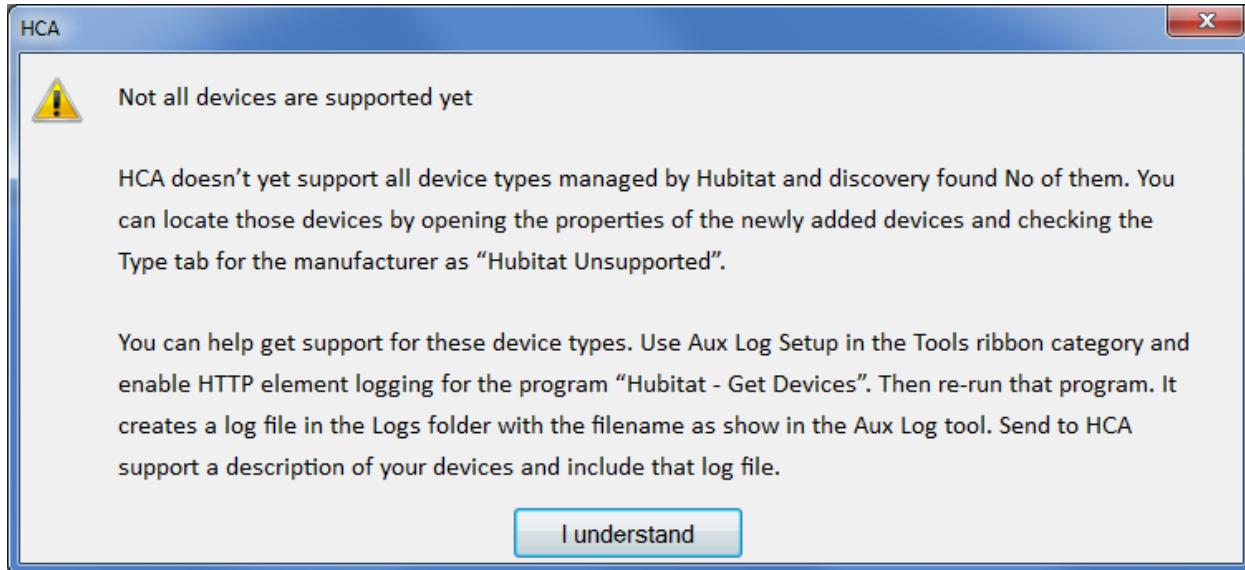
- Hubitat Dim Device
- Hubitat On Off Device
- Hubitat Sensor
- Hubitat Lock Device
- Hubitat Unsupported

The first two are for dimmable and non-dimmable switches, modules, etc. The third class is for sensors that have some sort of state - contact, motion, button, or water sensor. The Lock class is for locks, obviously. The final class is for everything else and is a placeholder so it doesn’t allow any control or status reading.

If during the Discover program’s execution any of the devices are managed by class “Hubitat Unsupported”, a second popup suggests what you should do.



HCA Tech Note 613



In the Hubitat package in the library is a program called “Check Batteries”. When battery powered devices report their full status to the Hubitat hub they also report other properties, one of which is the battery level. Some devices also report temperature. If the Hubitat programs see these kinds of settings it saves them as tags on the device. For example:

Tag Name	Current Value
HE_type	Generic Zigbee Motion Sensor
HE_devType	Sensor
HE_devSubType	motion
HE_Battery	12
HE_Temperature	61.45

These are shown on the “Tags” tab of a device properties. In this case the motion sensor is reporting a temperature of 61 (better put on a sweater!) and a battery level of 12 percent (better change it soon!). You can schedule the “Check Batteries” program to run once a week or how often you want to, and it sends you email for any device with a battery level getting low. Of course, for this to happen you will have to configure HCA to be able to send email. Feel free to modify the program to change how it reports, or the battery level you want to report about.



HCA Tech Note 613

Because of these tags you can write your own programs to make decisions based upon their values – temperature for example – using the Tag functions in the Compute and Compute-Test programmer element expressions.

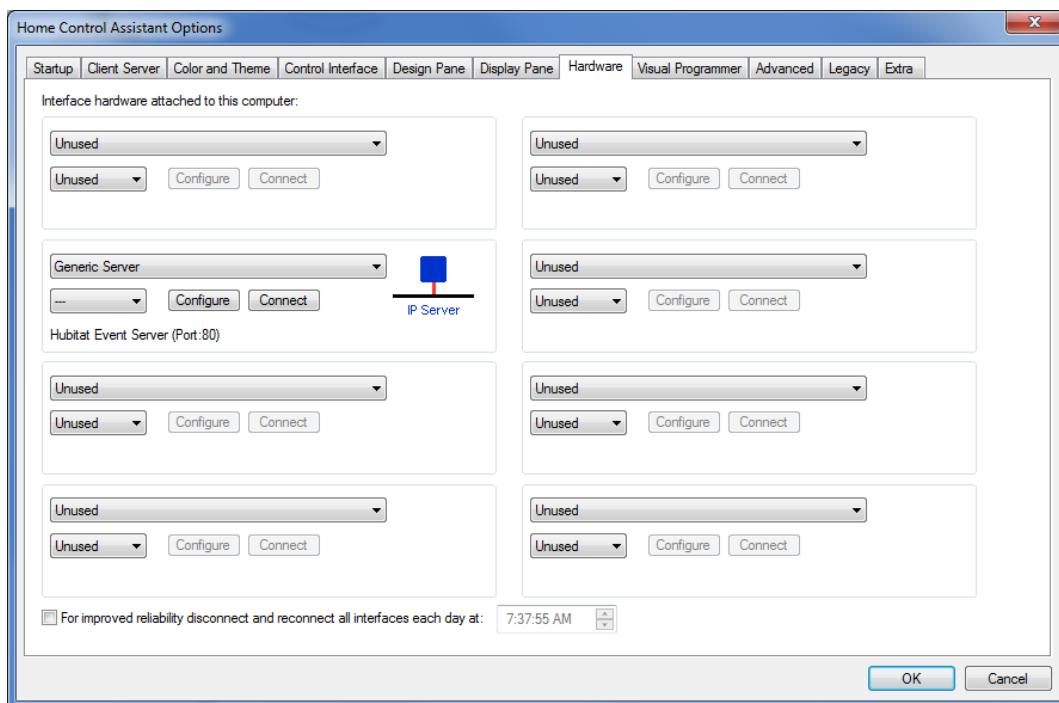


HCA Tech Note 613

Inbound Notifications – applies only to HCA 16.0.30 or later!

If you are running the latest version of the Habitat “Maker API” then you can configure HCA to handle events that are generated by devices that the Habitat Hub manages. Events like motion sensors becoming active, doors opening or closing, or switches being controlled on or off.

The first step is to configure an HCA Interface as a “Generic Server”. Can be any of the eight interfaces, it doesn’t matter which.

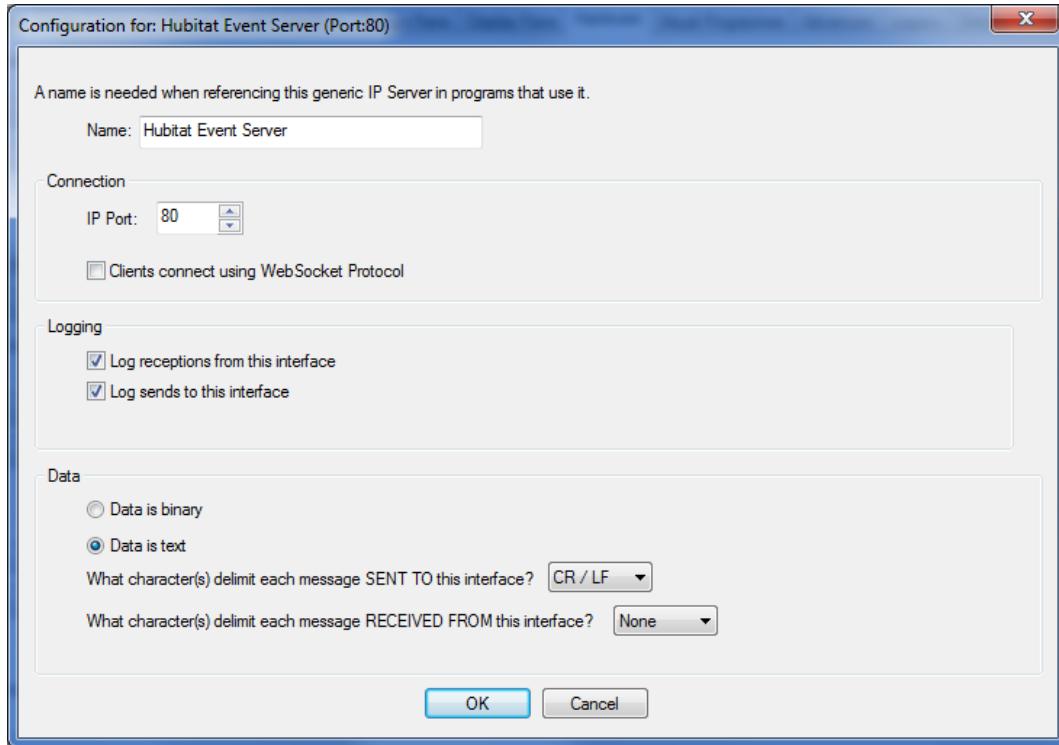


Press the Configure button to adjust the settings. The settings to use are: Name is “Habitat Event Server”, Port 80, Data is text, receive delimiter is None.

(Note: you can use another port number other than 80 if you want. In the section below where the Marker API settings are discussed you can see how you can choose a different port number.)



HCA Tech Note 613



You can tick the *logging* checkboxes or not. That's up to you.

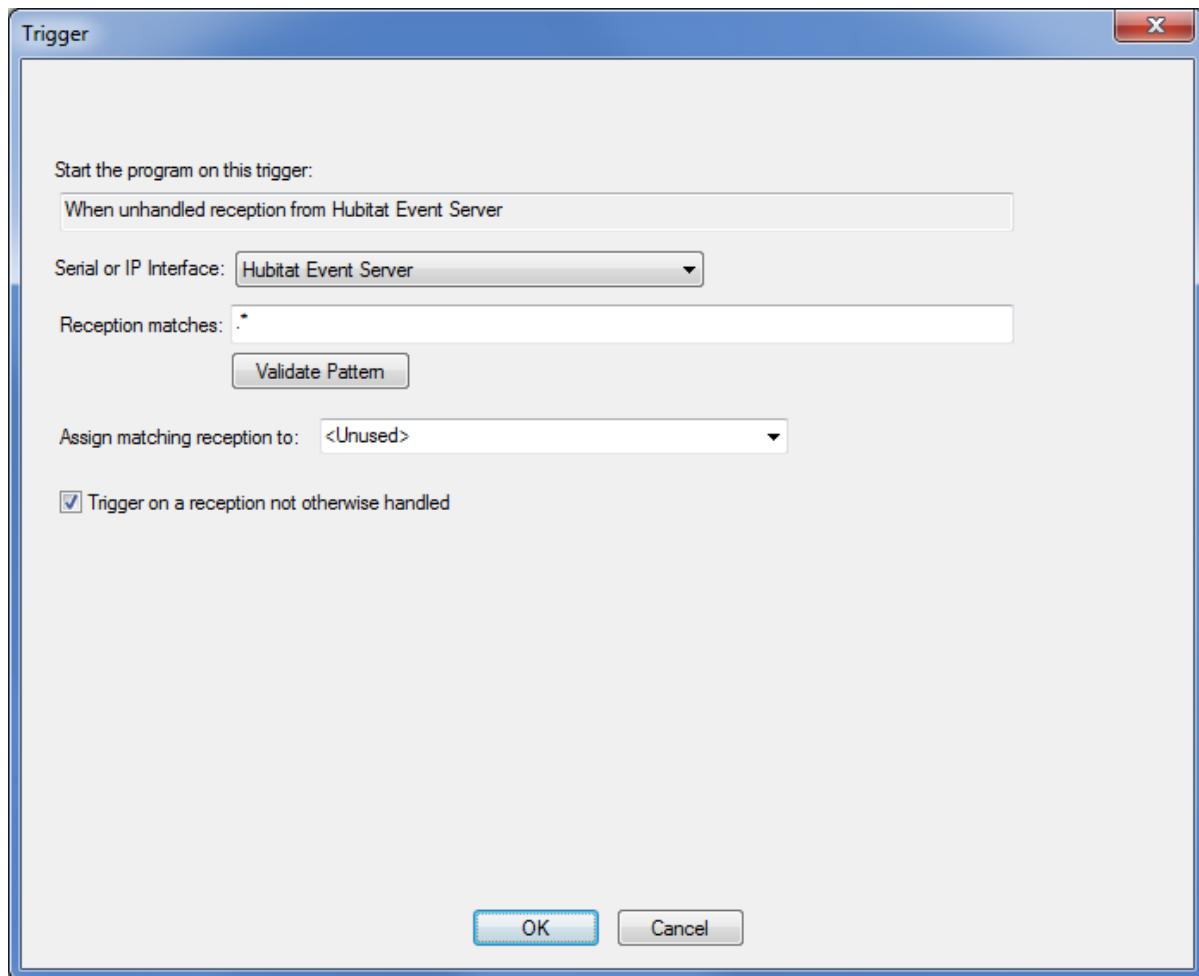
After importing the Hubitat package, you must next add a trigger on to the “Hubitat Event Handler” program.





HCA Tech Note 613

Choose “Port Reception” for the Trigger type then select the interface as “Hubitat Event Server” which is what you named the interface in the interface configuration. Tick the box that says: “Trigger on a reception not otherwise handled”. All other settings can remain unchanged



In the main *triggers* tab, at the bottom, configure as:



HCA Tech Note 613

If the trigger is received while the program is running:

- Ignore it
- Restart the program from the Start Here element
- Start a copy of the program and run it concurrently
- Add it to a received trigger queue and start the program with it when it stops after processing the current trigger

The next change is in the Maker API settings. As described at the start of this note, use a browser and connect to the Hubitat Hub, choose “Apps” from the left-hand menu, then “Marker API” and scroll down until you see this section:

URL to send device events to by POST

<http://192.168.0.88/Habitat>

[Include location events to be sent by POST?](#)

The image above shows an IP address, yours will be blank. Enter the IP address of your HCA Server in this format:

<http://<your HCA Server computer IP address>/Habitat>

If you want to use a port other than 80, here is an example for that same IP address but using port 8080. Note that the port number is at the end of the 4-part IP address and is preceded by a “:”.

URL to send device events to by POST

<http://192.168.0.88:8080/Habitat>

Just remember that if you specify a port in the Maker API you must use the same port number in the HCA Interface setup. With no port specified, 80 is assumed.

After entering the “URL to send device events”, then scroll down to the bottom of the Maker API configuration and press the *Done* button. This is VERY important!

To check if HCA is now receiving events use the HCA Interface Viewer. On the “Interfaces” ribbon category is an *Interface Viewer* button. Press that and a viewer window opens.



HCA Tech Note 613

When something happens to a device that Hubitat controls you should see something recorded in the viewer if that interface is enabled in the viewer. In this example, Hubitat is the 2nd interface (in programmer world that is shown as [1])

If nothing shows in the viewer, make sure you have all the ports and IP addresses correct and that you have entered in the URL correctly into the maker API. Also check that the Hubitat hub and the HCA Server computer are on the same network.

The screenshot shows the 'Interface Viewer' application window. In the top-left corner, there's a 'View Interfaces' section with checkboxes for 'Unused 1' through 'Unused 8'. The checkbox for 'Hubitat Event Server' is checked. Below this is a 'Show' section with several checkboxes: 'Initialize' (checked), 'Read' (checked), 'Write' (checked), 'Decode' (checked), 'Exceptions' (checked), 'Log Entries' (checked), 'Internal' (unchecked), 'Misc' (unchecked), 'Queue' (unchecked), 'Packets' (unchecked), 'S/R Queue' (unchecked), 'UPB Idle' (unchecked), 'MM' (unchecked), and 'X10' (unchecked). The main area displays a log of events:

```
06:41:32 Program "Hubitat Elevation - Hubitat User Event Handler" execution task ending
06:41:47 [1]: Event Socket 1bc8: 00000008 00000000
06:41:47 [1]: RD POST /Hubitat HTTP/1.1<cr><lf>Accept: */*<cr><lf>Content-Length: 164<cr><lf>Content-Type: application/json<cr><lf>Host: 192.168.0.88Connection: keep-alive
06:41:47 [1]: Extracted Message = "POST /Hubitat HTTP/1.1Accept: */*Content-Length: 164Content-Type: application/jsonHost: 192.168.0.88Connection: keep-alive"
06:41:47 Hubitat Elevation - Hubitat User Event Handler Exit code = 0
06:41:47 Hubitat Elevation - Hubitat Event Handler Exit code = 0
06:41:47 Starting task for program Hubitat Event Handler
06:41:47 Begin Task program ""Hubitat Elevation - Hubitat Event Handler"" 1404
06:41:48 Begin Task program ""Hubitat Elevation - Hubitat User Event Handler"" 1AB8
06:41:48 Program "Hubitat Elevation - Hubitat Event Handler" execution task ending
06:41:48 Program "Hubitat Elevation - Hubitat User Event Handler" execution task ending
```

At the bottom of the window, there are buttons for 'Clear', 'Save', 'Viewer Is ON' (which is highlighted in green), and 'Close'.



HCA Tech Note 613

Processing events yourself

When an event arrives from the Habitat hub, the program named “Hubitat Event Handler” starts (that’s the program you added the trigger to). It processes the event to update HCA state. But you may want to process the event yourself. There are two ways to do that. Here are two methods.

Method 1: Responding to a device that changes state based upon an event

When a motion sensor sees something move, it sends an event that is processed by the “Hubitat Event Handler” program. That looks at the event data and sees that it applies to a motion sensor, so it changes the state HCA has for that sensor to be “ON”. You can create a program that triggers on a change of state for that device from OFF (inactive) to ON (active). Create a trigger on your program like this:

Trigger

Trigger Type: State Change

Start the program on this trigger:
When Habitat Devices - Motion Office goes ON

When Habitat Devices - Motion Office Goes Off
 Goes On

If you want to respond to the motion sensor going inactive, add a second “ON to OFF” trigger and then in the program, test for the trigger that stated the program.

In HCA everything is either ON or OFF. With sensors their state is ON/OFF but the meaning depends upon the sensor.

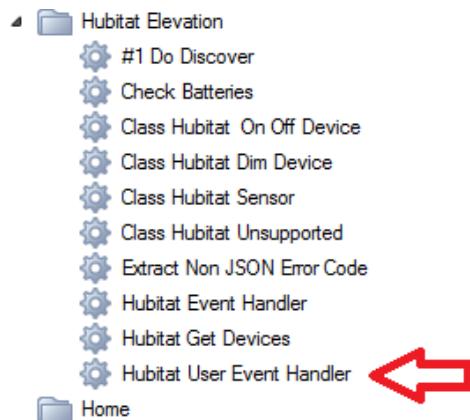
Device Type	OFF means	ON Means
Motion Sensor	Inactive (no motion)`	Active (Motion)
Contact Sensor	Closed	Open
Water Sensor	Dry	Wet
Presence Sensor	Not present	Present
Button	(other)	Pushed/double/held



HCA Tech Note 613

Method 2: Responding to an event directly by a program

When the state change trigger works fine for events that change the state of the device, some events don't do that, and you must process the event yourself. **To do that find the program "Hubitat User Event Handler" in the package, copy it, and paste it into one of your folders.**



This is a parameterized program and is supplied with the elements of the event message.

Parameters			
Count:	6	Value	Default for value parameter if not supplied by calling program
Parameter 1:	device	Type	HCA device name that reported the event
Parameter 2:	value	Type	Value from event report
Parameter 3:	displayName	Type	Display name from event report
Parameter 4:	descriptionText	Type	Description text from event report
Parameter 5:	unit	Type	Unit from event report
Parameter 6:	data	Type	Data from event report

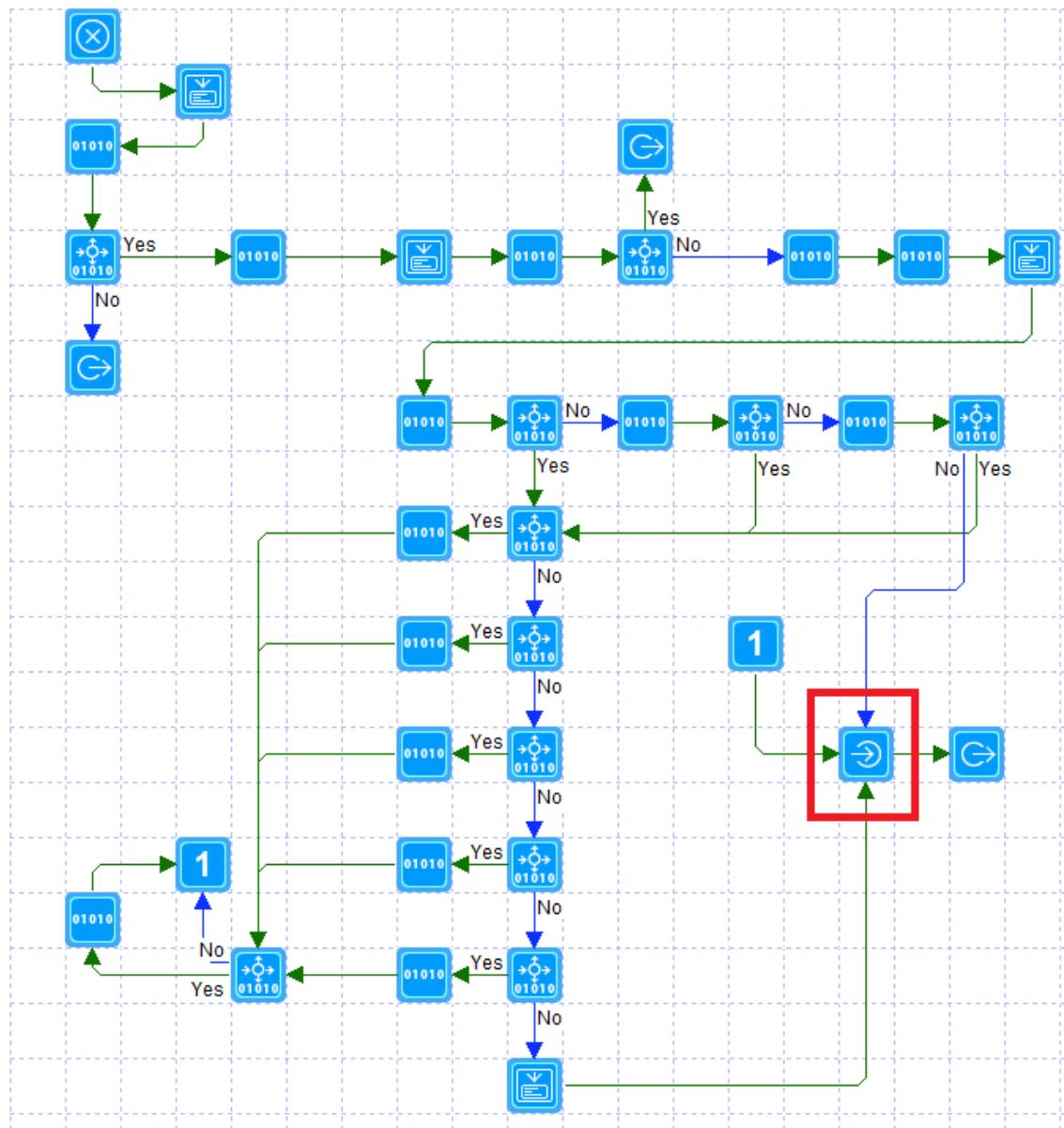
The Event processing program in the package ("Hubitat Event Handler") has already extracted out all the elements of the event JSON, determined which HCA device it applies to, and then can start one of your programs so you also can process the event if you need to once the package handler is done with it.



HCA Tech Note 613

There is a placeholder program in the package called “Hubitat User Event Handler”. Don’t modify that program but copy it and paste it into one of your own folders with any name you want.

Next, open the properties of the package event handler program “Hubitat Elevation – Hubitat Event Handler” and find this element:



TechSup@HCATech.com

www.hcatech.com



HCA Tech Note 613

Change the properties of this Start-Program element to use your program and not the placeholder program in the package:

Change this:

This element starts another program and waits until it finishes before continuing the current program unless the "don't wait" option enabled.

Start this program:

Don't wait for this program to complete before continuing to the next element

To this:

This element starts another program and waits until it finishes before continuing the current program unless the "don't wait" option enabled.

Start this program:

Don't wait for this program to complete before continuing to the next element

In this example, the “Hubitat User Event Handler” was copied and pasted into the folder named “My Home” and renamed to “My Hubitat Event Handler”.

What your program does is up to you. You can test the various parameter values using the Compute and Compute-Test elements.



HCA Tech Note 613

Using Habitat Elevation with Z-Wave devices

The v5 Habitat Hub has less than stellar Z-Wave sensitivity so it is a good idea to add at least 2 range extenders to your Z-Wave mesh before trying to add any devices. We have had good experience with the Aeotec Range Extender 6 repeaters and they are also cited often on the Habitat Community as the best overall. Generally speaking, powered Z-Wave devices with repeater capabilities are insufficient to add reliability to your Z-Wave mesh when they are used with the HE hub, so adding these extenders is a good idea especially if you want to control Z-Wave locks. You will need to place one range extender within two feet of your hub and the other(s) within 10-15 feet of the devices you want to control.

When you add these devices to your Z-Wave, they will be discovered as generic devices. Be sure to change the Type from “Device” to “Generic Z-Wave Repeater”. Make the Device label anything you want.

The screenshot shows the Habitat Elevation web interface. On the left, a sidebar menu includes: Devices (selected), Apps, Settings, Advanced (disabled), Apps Code, Drivers Code, System Events, and Logs. The main content area shows the following sections:

- State Variables:** A blank input field.
- Preferences:** A message stating "No preferences defined for this device."
- Device Information:**
 - Device Name ***: Device (with an [edit](#) link)
 - Device Network Id ***: 04
 - Device Label**: Aeotec Repeater 1
 - Type ***: Generic Z-Wave Repeater (with a dropdown arrow)
- Zigbee Id**: An empty input field.

A large "Save Device" button is located at the bottom of the form.



HCA Tech Note 613

Next, repair your Z-Wave network from the settings/Z-Wave Details page on your HE hub:

The screenshot shows the Habitat HE hub interface. On the left is a sidebar with icons for Dashboards, Devices, Apps, Settings, Advanced (with sub-options Apps Code and Drivers Code), and a gear icon. The main area is titled "Z-Wave Details". Below it is a sub-section titled "Z-Wave Radio Devices". There are two entries in the table:

Node	Device ID	Device Type	Clusters	Device
004 (04)	117	260	in: 0x5E, 0x26, 0x33, 0x70, 0x85, 0x59, 0x72, 0x86, 0x7A, 0x73, 0x5A, out:	Aeotec Repeater 1
005 (05)	117	260	in: 0x5E, 0x26, 0x33, 0x70, 0x85, 0x59, 0x72, 0x86, 0x7A, 0x73, 0x5A, out:	Aeotec Repeater 2

After the repair finishes, it is best to wait 24 hours before trying to add devices to allow the mesh to fully re-establish itself.

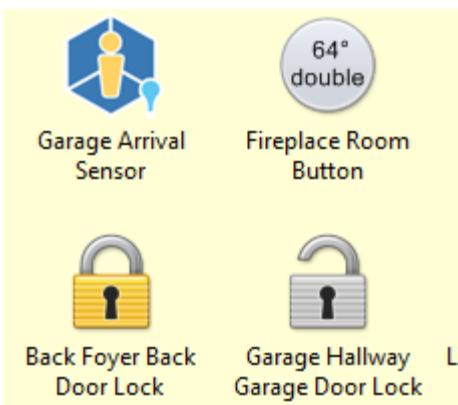
It is also a good idea to put fresh batteries in the device you want to add if it is battery powered. We have had situations occur with lock devices where their full capabilities were not recognized if the batteries weren't fresh. In one case it led to the lock being successfully added but subsequent commands not being recognized, and in another case, the lock was added but its auto-refresh capabilities weren't discovered. Put in fresh batteries to be sure.



HCA Tech Note 613

Using Buttons with Hubitat Elevation

Zigbee buttons such as the inexpensive Samsung SmartThings Button are great little devices to have in Hubitat because not only can they be used to initiate programs in HCA with various tap and hold options, but they can also report their temperature. HCA recognizes these buttons as sensors and creates a special icon for them that shows the last press event as well as the button's current temperature:



However, in order to make these buttons useful, some programming needs to be done using the Hubitat Rule machine. If you don't already have this installed, it can be found in the built-in apps. From the Hubitat browser interface, select "Apps" in the left menu, then press the "Add Built-in App" button and find "Rule Machine" and select it to add.

The problem is this: buttons have three "state variables" associated with the push actions they recognize: "pressed", "doubletapped", and "held".

Double Tap doubleTap NUMBER: <input type="text"/>	Hold hold NUMBER: <input type="text"/>	Push push NUMBER: <input type="text"/>	Current States <ul style="list-style-type: none">battery : 92doubleTapped : 0held : 0numberOfButtons : 1pushed : nulltemperature : 64.21
---	--	--	---

The problem is that when a button is manipulated, the associated State gets set to a 1 and it stays that way forever. This makes it impossible for HCA to request a status that returns the last press action or for you to know that the button got



HCA Tech Note 613

tapped in any manner. So how to solve this? One way is to manually reset the state by performing the action from the device tab. Entering a "0" in the NUMBER field above for the desired state to reset, then performing the action is one way to do this but obviously not ideal.

Enter the Rules engine which is really the primary programming feature of Hubitat. Using the engine we can trigger on a press, hold or doubletap event, then create actions that reset the state via custom command after a 3 second delay.

Creating these rules is very tedious but not hard. Here are the steps.

- In the “Rule machine” app click “Create new rule”
- “Name this rule” (click to set). [Click] Enter the name “Reset button state” [Enter]
- “Select Trigger Events” (click to set). [Click]
- “Select capability for new Event Trigger” (click to set). [Click]. Choose “Button Device” (not “Button”)
- “Select Button Device” (click to set). [Click]. Tick box next to your button [Update]
- “Prefill all button actions for” (click to set). [Click]. Tick all three. Click elsewhere and the display updates.
- “Select Actions for button 1 pushed” (click to set) [Click]
- “Select Action Type to add” dropdown, select “Set mode or variables. Run custom action”
- “Select Which Action” dropdown. Select “Run custom action”.
- “Select capability of action device” (click to set) [click] Choose “button”
- “Select Button devices to act on” (click to set) [click]
- Tick box next to your button [Update]
- “Select custom command” dropdown. Choose “push”
- “parameter type” dropdown. Choose number
- “integer value” Enter 0 [Enter]
- Tick “delay”
- Enter 3 under seconds. [Enter]
- “Done with this action” [click]



HCA Tech Note 613

- [Done with Actions for button 1 push] [Click]

Repeat this section starting with “Select actions for button 1 held” and then repeat again for “Select actions for button 1 doubleTapped” doing the same thing but in the “Select custom command” step, choose not “push” but the appropriate command.

When all three are done, this is what you should see:

The screenshot shows the "Define Buttons and Actions" app interface. At the top, there are navigation buttons: "« App List", a gear icon, and a question mark icon. Below the title, it says "Define Buttons and Actions". Under "Capability Button Device", it lists "Select Button Device * Office Button". In the main area, there are three sections: "Define Actions for [Office Button]" with three items: "Select Actions for button 1 pushed push(0) on Office Button --> delayed: 0:00:03", "Select Actions for button 1 held hold(0) on Office Button --> delayed: 0:00:03", and "Select Actions for button 1 doubleTapped doubleTap(0, 0) on Office Button --> delayed: 0:00:03". Each item has a right-pointing arrow icon at the end.

"Done" [click]

"Done" [Click] This should get you back to the Apps list and you should see your new rule listed below the Rule Machine.



HCA Tech Note 613

If you click on that rule you should see this:

« App List

Define a Rule Using a Button Device

Name this Rule*
Reset Office Button

Select Button Device *
Office Button

Define Actions for [Office Button]

Select Actions for button 1 pushed
push(0) on Office Button --> delayed: 0:00:03 ►

Select Actions for button 1 held
hold(0) on Office Button --> delayed: 0:00:03 ►

Select Actions for button 1 doubleTapped
doubleTap(0) on Office Button --> delayed: 0:00:03 ►

OK, that was tedious and I bet you appreciate the HCA visual programmer even more.

When the button is pressed, double pressed, or held, HCA is notified and then 3 seconds later the button state clears and HCA is notified again.



HCA Tech Note 613

So how does the automation map into HCA? Here's how:

- The button is pressed, double tapped or held
- HCA gets an event from Habitat and the built-in event handler stores the action in the button's "HE_LastAction" tag, It then changes the button's state to "On", changes the button's icon to its "On" depiction, and updates the text displayed on the button.
- **KEY POINT:** The "On" state change causes any programs with a state change trigger for that button "off" to "on". They program can query the HE_LastAction tag to find out what event occurred and act accordingly.
- Three seconds later, the Habitat rule kicks in and resets the state.
- HCA gets another event from this, changes the button's state to "Off" and changes the button's icon to its "Off" depiction.
- The "Off" state change causes any programs with a state change trigger for that button "on" to "off". They program can query the HE_LastAction tag to find out what event occurred and act accordingly.

TIP: In the package there is an example program "Habitat respond to push". Don't modify the program but copy it and paste it into one of your own folders. Then add state change triggers for your button. You can modify the program to do whatever you want when the button is pressed, double tapped, or held.

TIP: If you want to change the size of the text on the button, add a tag to the "Habitat Event Handler" and "Class Habitat Sensor" programs. The tag name is "ButtonTextPointSize" and the tag value is the point size. For example, 10 point text.

Tag Name	Current Value
ButtonTextPointSize	10

Enjoy using these versatile Habitat devices!

##end##