



HOME CONTROL ASSISTANT

V13 Change Notes 28-March-2016

Parameterized programs	3
Parameterized Triggers	9
HCA Dynamic DNS.....	12
HCA Alexa support	12
Generic Triggers	13
PING	15
Change to Start-Program element.....	18
Alerts system refresh	20
Log Viewer Changes.....	28
New Compute functions	30
Insteon changes	32
Insteon Multi-Unit devices.....	37
Additional Insteon Device Support	39
Changes to Mobile Clients	40
Geofence support in the iPad/iPhone HCA app.....	40
Changes to Windows client.....	44
HTML Folder Sync	45

Changes to the Tiled displays.....	47
New Icons.....	49
Thermostat Support.....	51
Status Tile.....	56
UPB Import.....	57
Date-Book out, Calendar In.....	61
Support for Insteon On/Off outlet.....	64
Generic Serial/Port I/O interface binary mode.....	64
HCA Keypads.....	67
Miscellaneous	74

Parameterized programs

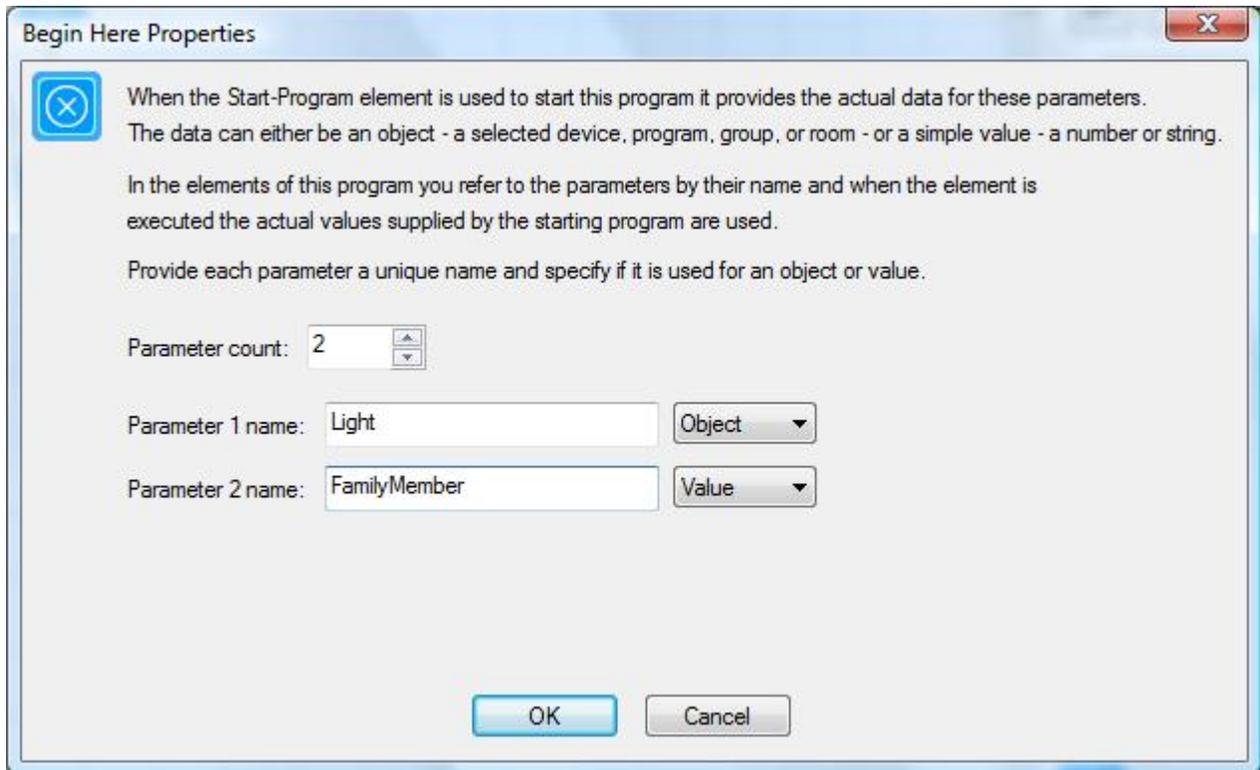
In previous versions of HCA one program could start another program but, unlike a more traditional programming language, there was no way to pass “parameters” to the started program. You could use flags (variables) for this but they are always global to the whole design.

This has changed in HCA 13 with the ability for a program to accept arguments – objects and values – from other programs.

Suppose Program A wants to start program B and tell B what device to work on and also pass another piece of data that is a string.

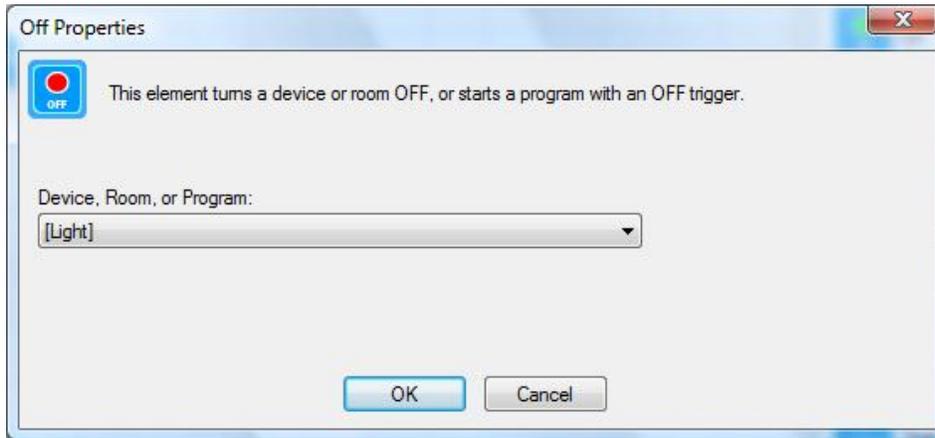
In traditional programming language, Program B is a “subroutine” of A.

In HCA 13, after program B is created, on the “Advanced tab” you must first tick the “This program supports parameters” box. One that is done that then the properties of the Begin-Here element can be opened.



In the Begin-Here properties you select the number of parameters and given them names and what they are used for. Parameters can either be an object – a device, program, group, room - or a value – a string numbers, date-time, etc.

Once the parameters are defined then they can be used. For example here is an ON element in program B:

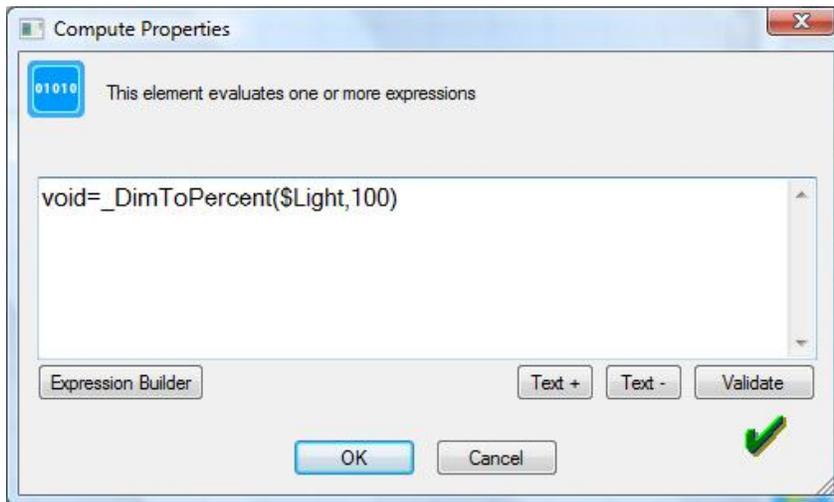
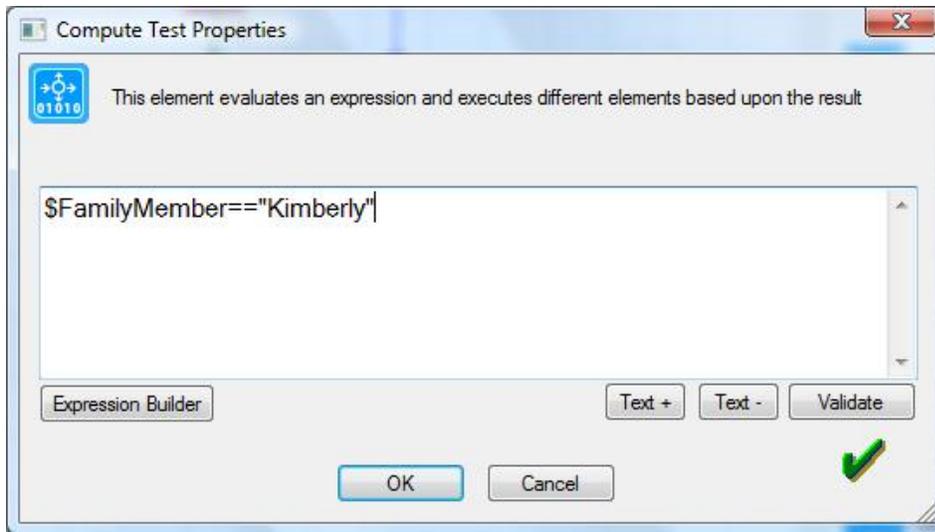


Note how the selection, in addition to all the usual “on-able” things are the parameter names. It added to the dropdown those parameters specified in the Begin-Here element that are used for objects. In this example, in the drop down you will not see “[FamilyMember]” since it is a parameter for a value.

The elements where you can select an object parameter are the ones you would expect

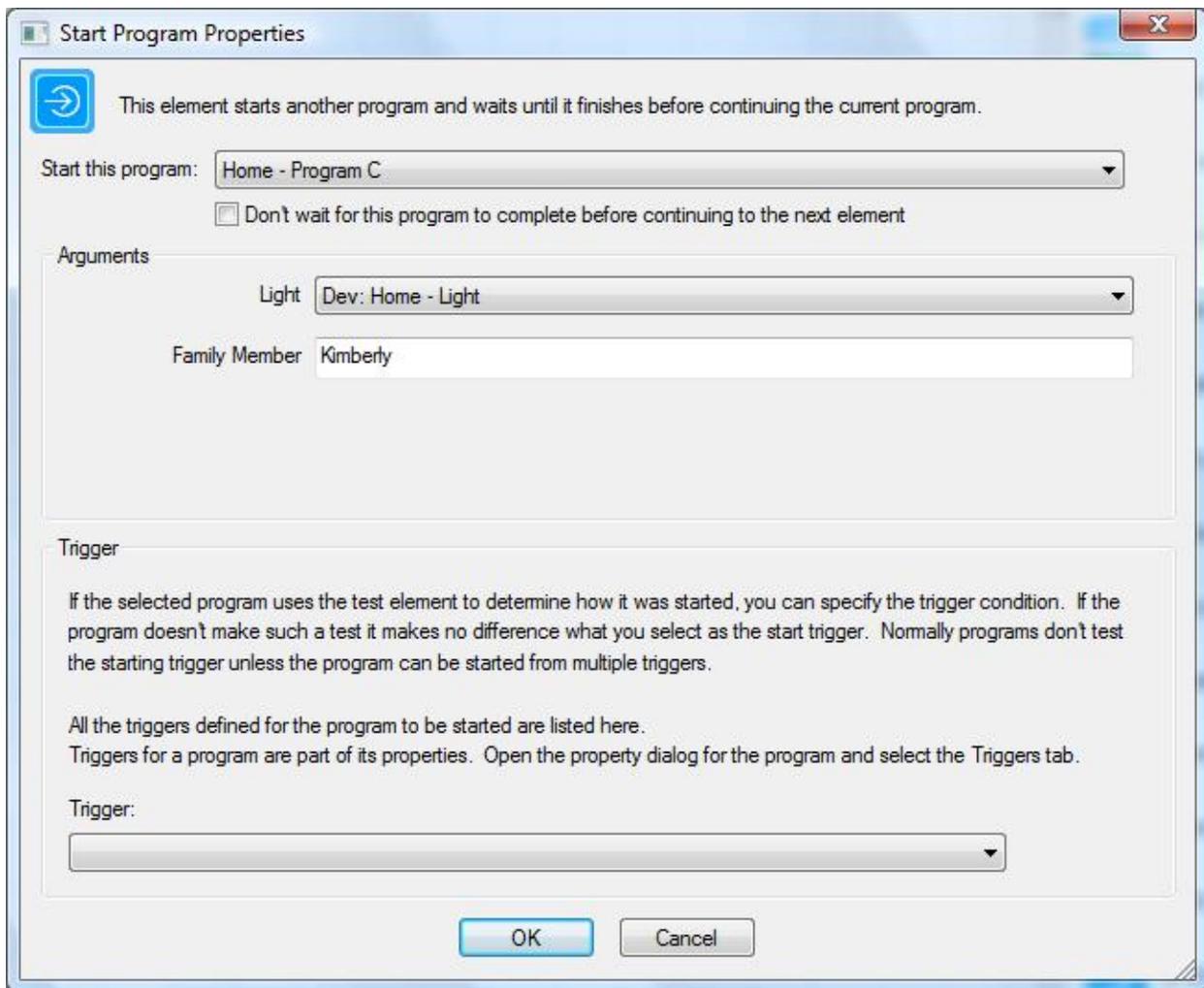
- On
- Off
- Dim
- Multi
- Thermostat
- Test (IsOn, IsOff, IsDim, IsSuspended)
- Thermostat-Test
- UPB-Blink
- Suspend
- Resume
- Stop
- ShowDisplay
- Auto-Off

In a compute or compute-test element you can also use the value parameters as well as the object parameters. Here are two elements from this program. The first is seeing which Family Member is being targeted and the second controls the light to a level.



Note that you refer to program parameters by their name prefixed with a \$. This lets the parser disambiguate flag names from parameter names.

Now let's turn to the calling program. Here is the Start-Program element in program A.



Since the Begin-Here element of the started program has defined what kind of argument it expects – object or value – it presents the UI you would expect. Object parameters get a dropdown listing all the devices, programs, groups, etc. Value parameters get a simple edit control.

Note: The text entered into the edit control for a value parameter may contain expressions. Those expressions are embedded in %'s like other HCA elements that take expressions embedded in text. For example if there was variable named "count" containing the value 23, then if this was entered as a value parameter argument:

Count is %count + 1%

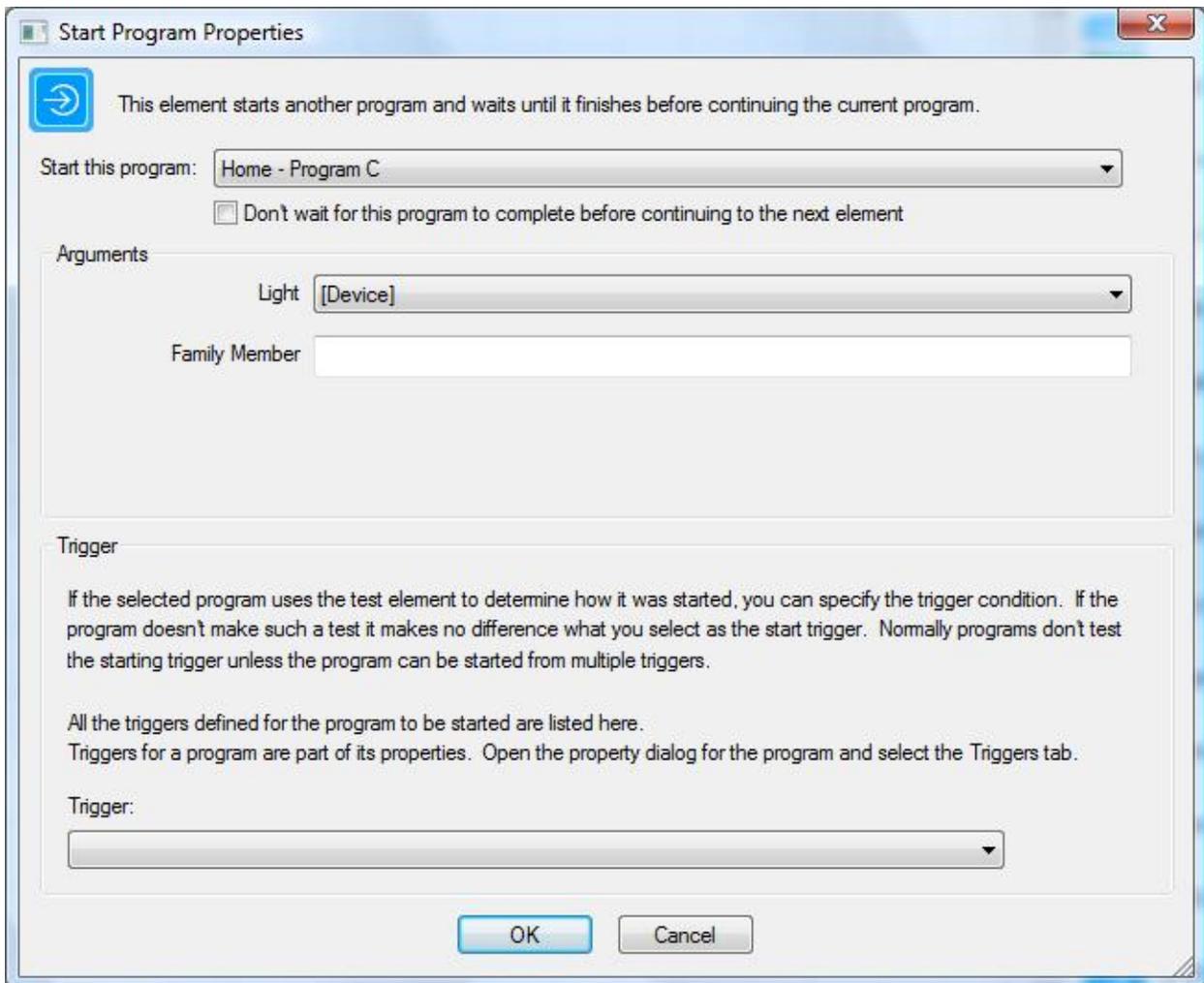
The started program would receive a parameter of text "Count is 24"

Here is some “Advanced Level” Info for the most sophisticated users

Suppose you want program A to Start-Program on program B and program B to Start-Program on program C. Also suppose that when program B starts program C you want to pass the value of one of its (that is, Program B) parameters to program C.

As an example, program B has a parameter called “Device”. Program C has a parameter called “Light”. In the start-program element in program A starting program B you select “Home-Lamp” for “Device”.

Now in program B in the Start-Program element starting program C you can do this:



Note that in the selection for the “Light” parameter to program C this example has selected the name of one of program B’s parameters. Suppose that when program B was started by program A, program A passed “Home-Light” to program B. Now program B passes “Home-Light” to program C.

Things get a bit more complicated for value parameters due to the UI. When you are filling in the Start-Program properties for a value parameter you want to be able to enter in text and also to be able to select one of the program parameters.

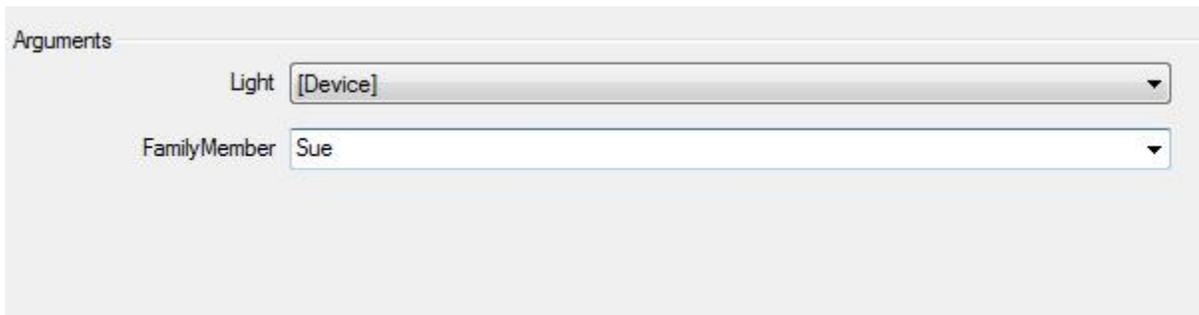
HCA uses one of the Windows UI controls HCA tries to avoid: The dropdown. This is a combination of an edit control and a droplist. The only other place it is used in HCA is in the Flag elements where you can either select an existing flag or type in a new one.

Here I am selecting one of the program parameters for “Family Member”



The screenshot shows a dialog box titled "Arguments". It contains two dropdown menus. The first is labeled "Light" and has "[Device]" selected. The second is labeled "FamilyMember" and has "[Name]" selected. The dropdown menu for "FamilyMember" is open, showing a list of options with "[Name]" highlighted.

Here I am just entering a value for “Family Member”



The screenshot shows the same "Arguments" dialog box. The "Light" dropdown menu still has "[Device]" selected. The "FamilyMember" dropdown menu now has "Sue" entered in the text field, and the dropdown arrow is visible on the right side.

Using the “dropdown” you can either select something from the list or type in the value you want.

This only is used in the case where one parameterized program uses start-program on another parameterized program.

As an example of the utility of this feature, in a sample home we have duplicated several programs each to work with devices and motion sensors in a room. They are all the same program except they work on different devices. Now we can make one common program and use it where needed. This can also be exported and given it to someone else and they could make use of it as is.

Why would you use this feature?

If you are comfortable with traditional programming languages the concept of "subroutines" will be familiar. Using this new feature you can create programs that perform an action without the need to duplicate the programs. For example a program that implements the interaction between a switch, a keypad, and a motion sensor in a room, can be generalized in such a way – by passing in the switch, motion sensor, and keypad objects – so it can be used for more than one room. Previously you would have to duplicate the program and change the elements that operate upon the specific keypad, motion sensor, and switch in the room.

Parameterized Triggers

In addition to the above changes, triggers can also be parameterized. First some background.

What we have now is that if program A starts program B using the Start-Program VP element then program A can pass parameters to program B.

But there is another way to start a program: a trigger also starts a program. So if program A has triggers and takes parameters then why not let the trigger pass parameters to the program?

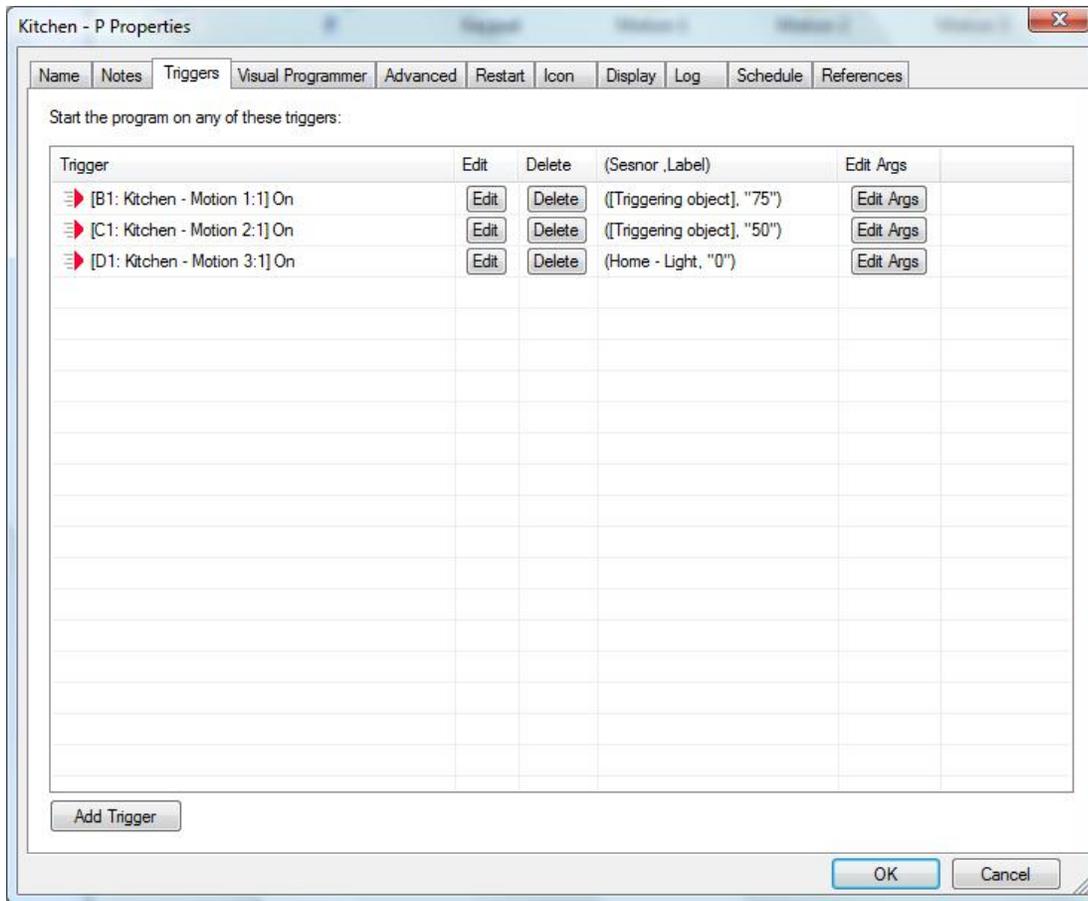
Simplest case: Suppose that program A takes a single object parameter.

Supposed program A is triggered based upon one or more devices. For example three different switches. The triggering device should be available to the program as an argument. That way the program could operate upon the triggering device without having to test for the starting trigger. And if additional triggers are added then it all just works without having to change the program other than adding the trigger.

This is of course only good for triggers that have an associated object. The X10, Insteon, and UPB triggers have an associated object. A weather trigger for example doesn't.

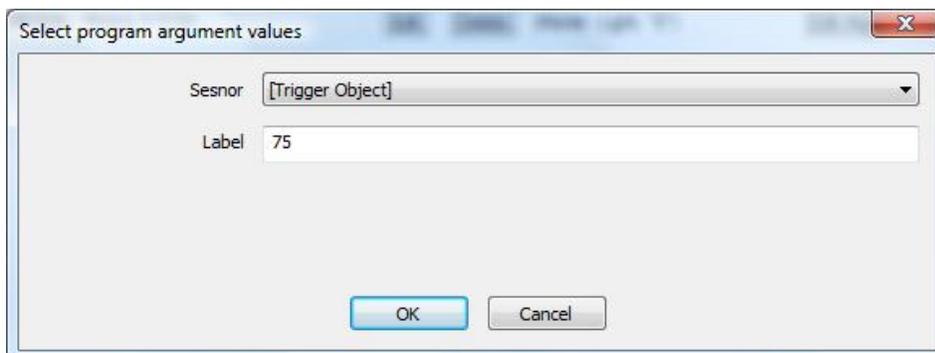
A more complex case: Suppose that associated with each trigger are the arguments to pass to the program for its parameters. It would be part of the trigger configuration. You would specify in the trigger what values and objects to associate with each of the program parameters.

To implement this, the trigger tab in an object's properties has been changed. Here is what the trigger tab looks like now for programs without parameters.



There is now a column for the parameters headed with the parameter names. For each trigger the row shows the arguments for each parameter. For an object parameter you can select any of the HCA objects – programs, devices, etc – or [nothing] or [triggering object]. This last case is where the object, if there is one, that is associated with the trigger is passed to the program. For example the keypad object that the user pressed a button on.

This is the dialog when you press the “Edit Args” button.



Why would you use this feature?

Same reasons as you would use parameterized programs in general. The ability to pair a trigger with an object or a value can make it simpler in creating the program. If for example you were to have a program that triggers on messages from many different devices you could use the triggering device in the program elements that operate upon that device.

HCA Dynamic DNS

A new feature has been added to HCA 13. This feature allows for simpler connection to the HCA Server from clients, services and web browsers when accessing HCA from outside the local network.

There is a separate technical note on this feature.

HCA Alexa support

Support has been added to HCA to allow it to be controlled by Amazon Alexa. While you may have heard of the Amazon Echo, Alexa support is in more than just that. Alexa support is available in:

- Amazon Echo
- Amazon FireTV
- Amazon Fire Tablet
- Amazon FireTV App....
- Anything in the future that Amazon bring out that they add Alexa to.

HCA is available as part of the “Connected Home” facilities. This means that after Alexa knows about your devices, programs, and groups – and you will configure in HCA which those are and how they should be named – you can then say for example:

- Alexa, turn on kitchen lights
- Alexa, turn on home away

There is a separate technical note describing how to configure Alexa and HCA to work together.

Generic Triggers

Triggers are what start programs running. The most used triggers come from when physical devices transmit messages. For example, pressing a keypad button. HCA has extensive configuration for defining these triggers.

But sometimes programs aren't connected to a real world device like a keypad. A program could be started by tapping an icon on a client screen. Or by opening the control page for the program on a client and pressing the "Start", "On" or "Off" buttons.

You may have wondered why when you right-click on a program in the development UI you sometimes get a "Start" menu pick and sometimes "On" and "Off" menu picks. It has to do with if the program has any triggers defined and if they are "On" or "Off" type triggers. If it doesn't have "On" or "Off" type triggers then the popup menu has "Start". Same reason for a client when in the control page you see "Start" or "On" and "Off".

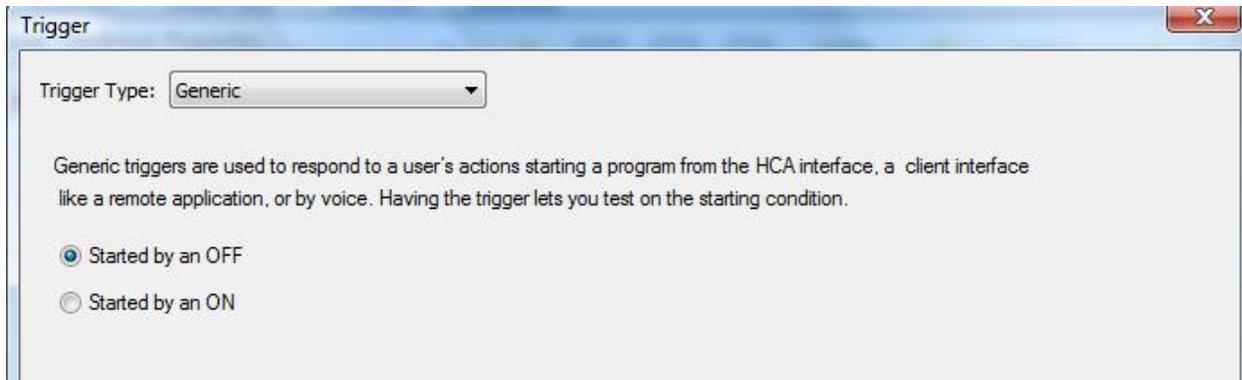
Also, the question comes up of what does an ON and OFF mean for a program. Both start the program – OFF doesn't mean stop. In the case of "On" it starts the program running with an "On" trigger and the same for "Off" – it starts the program with an OFF trigger. The key fact is that you can then test in the program for "Started by ON" or "Started by OFF" and do different things.

Testing for "Started by ON" and "Started by OFF" is very useful when creating programs that Alexa can control since everything in her world is ON and OFF.

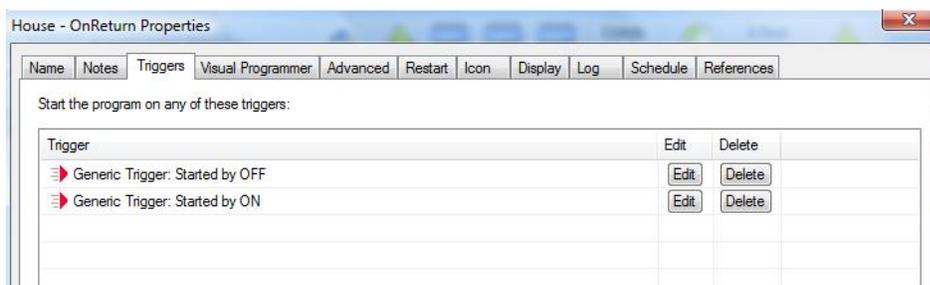
The problem was when you wanted to make the test for how the program was started, you couldn't if there were no triggers. The solution up until now was to define "phony" triggers. X10 was very useful for this. You defined X10 triggers like "P2 ON" and "P2 OFF" and then the Test element could be used to test for this. This worked even if there was nothing generating X10. The triggers were there just to allow you to configure the "Test" element

Starting in this version of HCA there is no longer a need to create these phony triggers. HCA now has a new trigger type: Generic triggers.

To create a trigger of this type, in the trigger tab of a program press the "Add Trigger" button then select "Generic" as the trigger type.

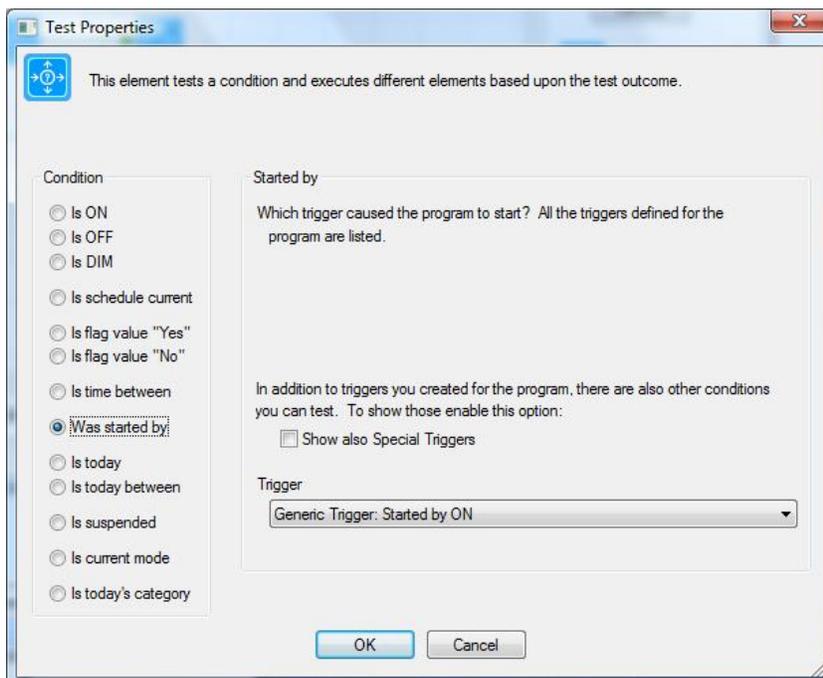


Here is how these triggers appear in the trigger list:



When a program has these triggers then the popup menu and the client control panel will have "On" and "Off" and not "Start".

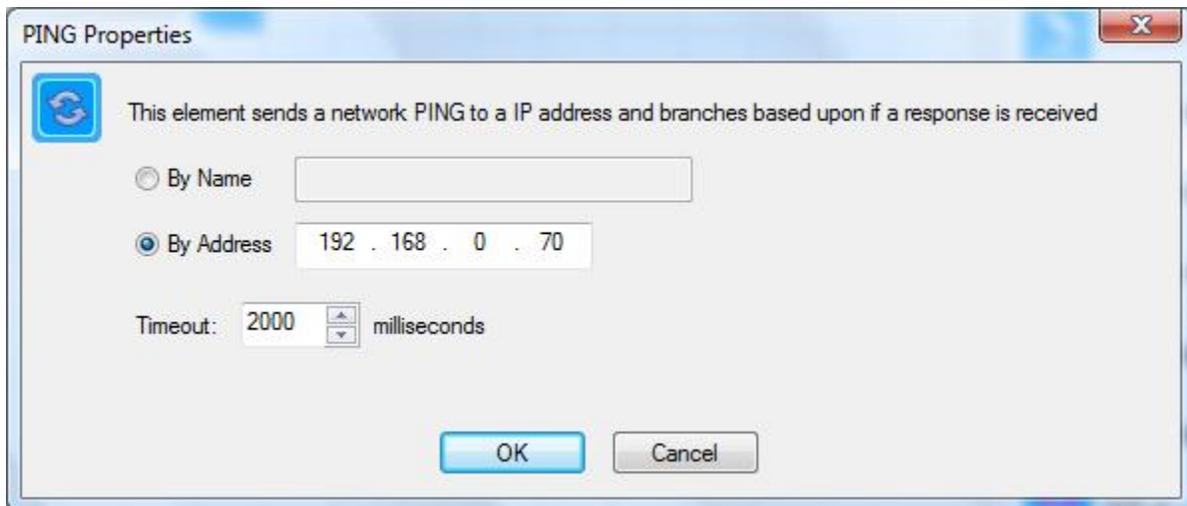
Once you have these triggers then you can test for that in the Test element:



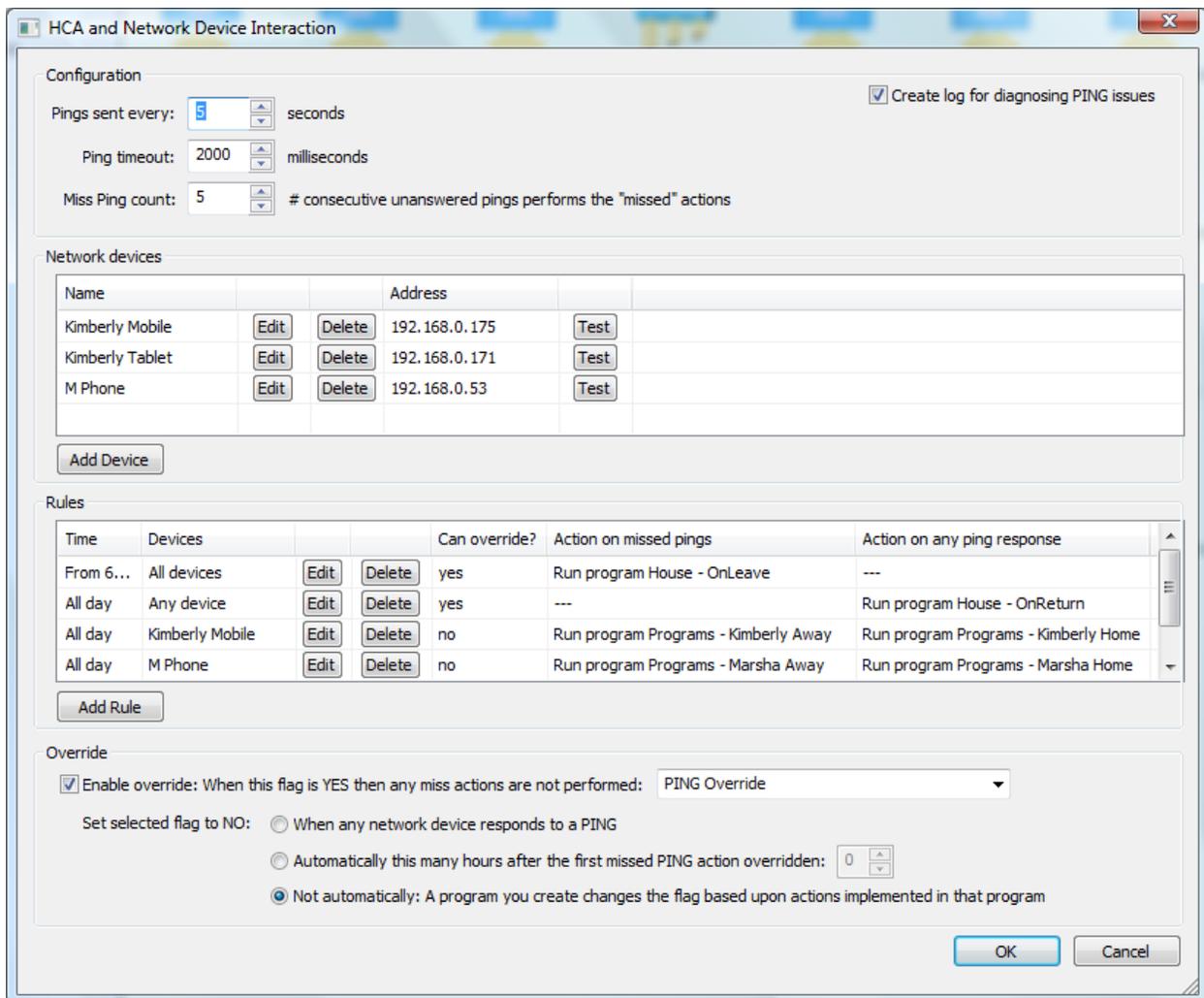
This type of trigger can be very useful when working with Alexa as you can now say "Alexa, Turn On <name>" and "Alexa, turn off <name>" and in the program test for what was wanted.

PING

There is now a VP PING Element that branches if the device responds or not.

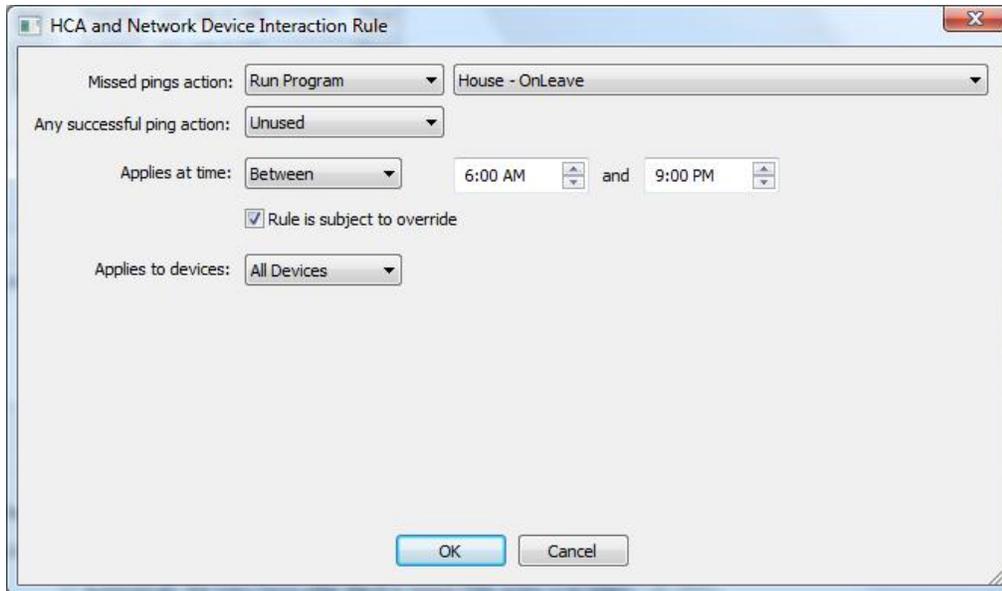


There is also a new tool that lets you determine how HCA interacts with network devices via PINGs. This is in the Design category in the Home Configuration panel called "Network Devices"



You can input up to 32 network devices and specify how often they are pinged. Also specified are a series of rules as to what happens when devices respond or don't respond to a PING. Each rule can apply to all devices, any device, or a selected set of devices.

You can create as many rules as you need. A rule specifies the time of day it is in effect and which devices it applies to. You can specify a specific device or devices, or choose "any" or "all". Each rule specifies what happens on a miss (no reply to the ping) or a response (the device responds to the ping). A rule can either change home modes or start a program.



A rule that applies to “Any” device will be useful for conditions where you want to see if anyone is at home.

A rule that applies to “All” devices will be useful for conditions when no one is at home.

The “All” and “Any” rules are most useful when you are changing the home mode.

Rules tied to specific devices will be helpful to track the comings and goings of individual devices.

It is important to remember that the “miss” action only happens if the device doesn’t respond after ‘n’ consecutive pings. ‘n’ being set in the configuration settings.

If you are starting a program on a response condition, the program should have a single value parameter. A string representing the name of the device that caused the response is passed to that program as that value parameter. The name is the name of the network device as displayed in the “Network Devices” list.

There is a logging facility that you can enable (creates “ping.log” in the HCA documents area Logs folder) that will let you track how it works and get a sense on how your devices respond.

There is also an override mechanism you can use to control the actions of the PING Tool. Once the override is enabled, if the selected flag has a value of YES then actions of any rules that are subject to the override are never executed. There are three options to clear the override flag.

- Clear the flag (set its value to No) when any device responds to a PING
- Automatically after a specified number of hours

- Never automatically cleared. In this case you are responsible for creating a program or some mechanism to clear the override flag.

Why would you use this feature?

Everyone now has some sort of a smart device that they often have with them. Using the feature you can tell if that person is within range of the computer or not – that is, are they home or are they not home?

Change to Start-Program element

One program can cause another program to start in one of two ways: using the Start-Program element or using the ON or OFF elements.

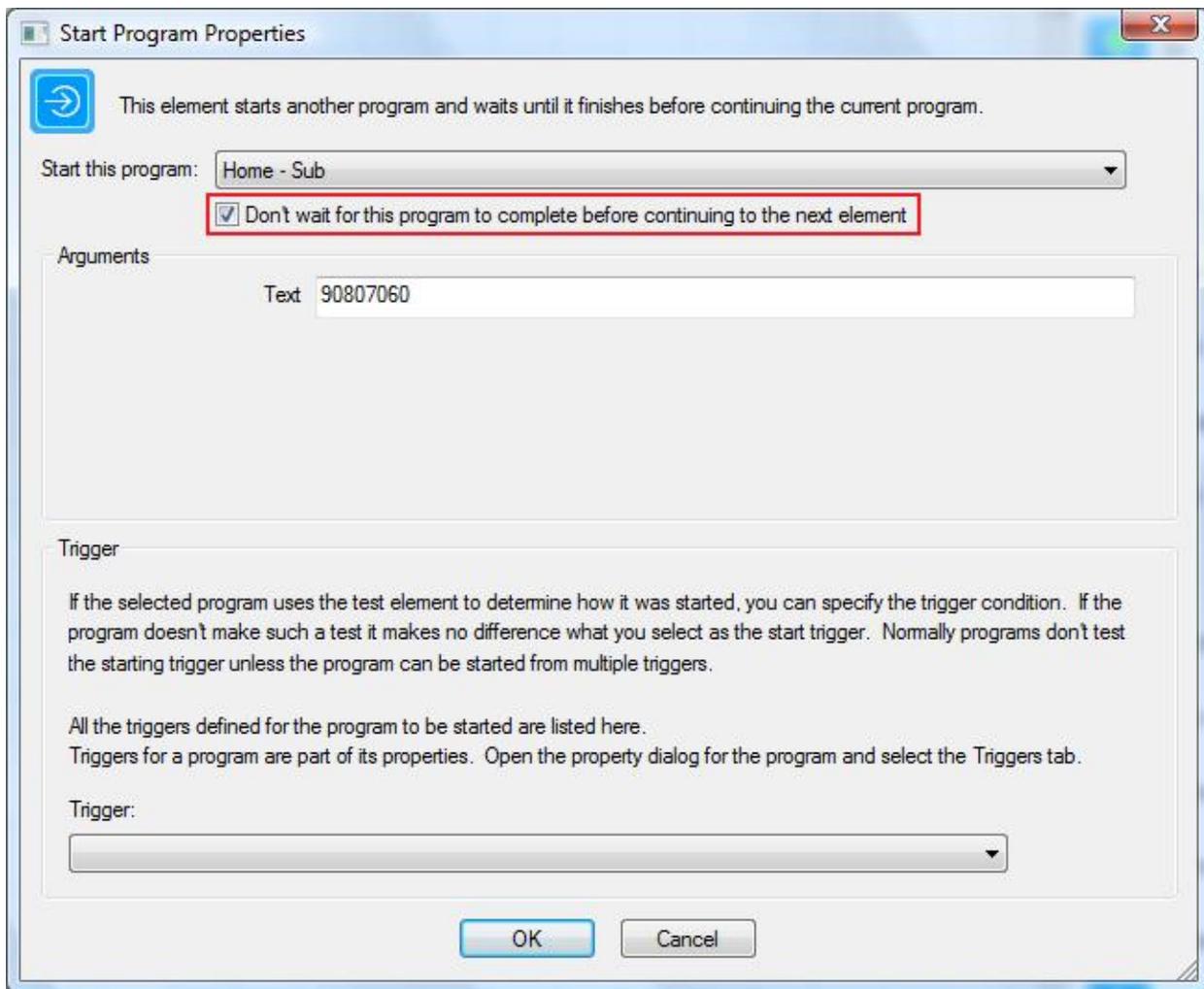
In the case of the Start-Program element, the named program acts as a "subroutine" of the calling program. For example if program A contained a Start-Program element naming program B, then when the Start-Program element is executed program B starts and program A stops until program B completes.

In the case of the ON and OFF elements, the named program runs as a co-routine to the starting program. That is, they both run concurrently. For example if program A contains an ON element naming program B, then when the ON element executes program B starts and program A immediately continues on at the next element after the ON element.

There is a major advantage of the Start-Program element instead of the ON and OFF elements: With Start-Program you get to specify a trigger to use when starting the named program and with the V13 changes for parameterized programs, you also got to supply arguments for those parameters. Neither could be done with the ON or OFF element. Also to choose a program as the target of the ON or OFF element the program had to have an ON or OFF trigger.

This made it impossible to have a program started as a co-routine and provide it with a trigger it can test within the program or supply parameters.

In V13 a change was made to the Start-Program element:



The one change is the option in the red box above. If this option is enabled then the program is started as a co-routine as described above.

The major advantage over the ON and OFF elements is that the started program now can be supplied with parameters and/or a trigger. And since the ON and OFF elements require that a program have an ON or OFF trigger, the started program no longer needs those triggers to execute as a co-routine when started by the start-program element.

Why would you use this feature?

This is feature that may make it simpler to build programs that implement your automation design.

Alerts system refresh

In V13 there are lots of changes in the Alert Manager. In Summary:

- The Alert Manager is now totally split off from the Inspector. They are now configured and viewed differently.
- The Inspector reports on potential problems that can be determined from a static analysis of a design file. The Alerts System looks for problems that occur over time while the system is running. It looks for dynamic problems.
- The Alerts system has decoupled counting problems from logging problems. You can count classes of problems, you can log problems, or you can count and log. It all depends upon how it is configured.

The new Alert viewer has two tabs: Details and summary.

1. The Details page is the log where all logged alerts show up. Columns are – alert name, time happened, and info. There is a checkbox and “Clear” button to remove an alert from the log.
2. The Summary page shows the count for each alert since last cleared and the date/time of last occurrence.

Alert viewer summary tab

The Alert system reports on problems that are dynamic - events that occur as HCA runs your design. For example status requests not answered, devices that don't acknowledge control commands, etc. Each alert type can be configured on how it reports.

Alert	State	Count	Last Occurred	Clear	Configure
Insteon: Unknown address	Green	0		Clear	Configure
Insteon: Missing command acknowledgement	Green	0		Clear	Configure
UPB: Unknown unit Id	Green	2	5/16/2015 8:29 AM	Clear	Configure
UPB: Missing sequence in transmission	Green	0		Clear	Configure
UPB: Missing command acknowledgement	Green	0		Clear	Configure
Wireless: Unknown unit	Green	0		Clear	Configure
X10: Unknown address	Green	0		Clear	Configure
Status request and no reply	Green	0		Clear	Configure
Confirm receipt needed retransmission	Green	0		Clear	Configure
Confirm receipt failure after all retransmissions	Green	0		Clear	Configure
Program error	Green	0		Clear	Configure
Weather observation failed	Green	0		Clear	Configure
Interface Connection	Green	0		Clear	Configure
Connected interface reported error	Green	0		Clear	Configure
USB Disconnection	Green	0		Clear	Configure
Power out	Disabled	0		Clear	Configure
Power restored	Disabled	0		Clear	Configure
Client disconnect	Green	0		Clear	Configure
Category 1 devices overdue	Green	0		Clear	Configure
Category 2 devices overdue	Green	0		Clear	Configure
Category 3 devices overdue	Green	0		Clear	Configure
Category 4 devices overdue	Green	0		Clear	Configure
User Alert 1	Green	0		Clear	Configure
User Alert 2	Green	0		Clear	Configure
User Alert 3	Green	0		Clear	Configure
User Alert 4	Green	0		Clear	Configure

Clear All

Close

The Summary tab lets you clear all the counts or just a single alert count. It also shows the alert level – green, yellow or red - for each alert determined from its count. You can also re-configure the alert from this view. Press the “Configure” button or double click the line to open the alert configuration dialog.

All possible alerts are listed. If you want you can disable an alert. When an alert is disabled, if the condition that alert tests for happens, nothing is done – it doesn't log, it doesn't count. It is just like it never happen.

Alert Type	Enabled?	Log Frequency	Yellow After	Red After	Run Program	Clear Option
Insteon: Unknown address	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
Insteon: Missing command acknowledgement	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
UPB: Unknown unit Id	<input type="button" value="Edit"/>	Yes	Max once a day	5	10	Daily
UPB: Missing sequence in transmission	<input type="button" value="Edit"/>	Yes	Max once a day	5	10	Daily
UPB: Missing command acknowledgement	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
Wireless: Unknown unit	<input type="button" value="Edit"/>	Yes	Never	5	10	Daily
X10: Unknown address	<input type="button" value="Edit"/>	Yes	Each time for occurrences 2 to 10	5	10	Daily
Status request and no reply	<input type="button" value="Edit"/>	Yes	Each Time	5	10	Daily
Confirm receipt needed retransmission	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
Confirm receipt failure after all retransmissions	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
Program error	<input type="button" value="Edit"/>	Yes	Max once a day	1	2	Daily
Weather observation failed	<input type="button" value="Edit"/>	Yes	Max once a day	5	10	Daily
Interface Connection	<input type="button" value="Edit"/>	Yes	Each Time	1	1	Daily
Connected interface reported error	<input type="button" value="Edit"/>	Yes	Each Time	1	10	Daily
USB Disconnection	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
Power out	<input type="button" value="Edit"/>	No				
Power restored	<input type="button" value="Edit"/>	No				
Client disconnect	<input type="button" value="Edit"/>	Yes	Each Time	1	10	Daily
Category 1 devices overdue	<input type="button" value="Edit"/>	Yes	Each Time	1	2	Weekly
Category 2 devices overdue	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
Category 3 devices overdue	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
Category 4 devices overdue	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
User Alert 1	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
User Alert 2	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
User Alert 3	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily
User Alert 4	<input type="button" value="Edit"/>	Yes	Never	1	2	Daily

Alert Frequency

Configuration for alert:

Disable this Alert

If this alert occurs:

... Create an alert log entry

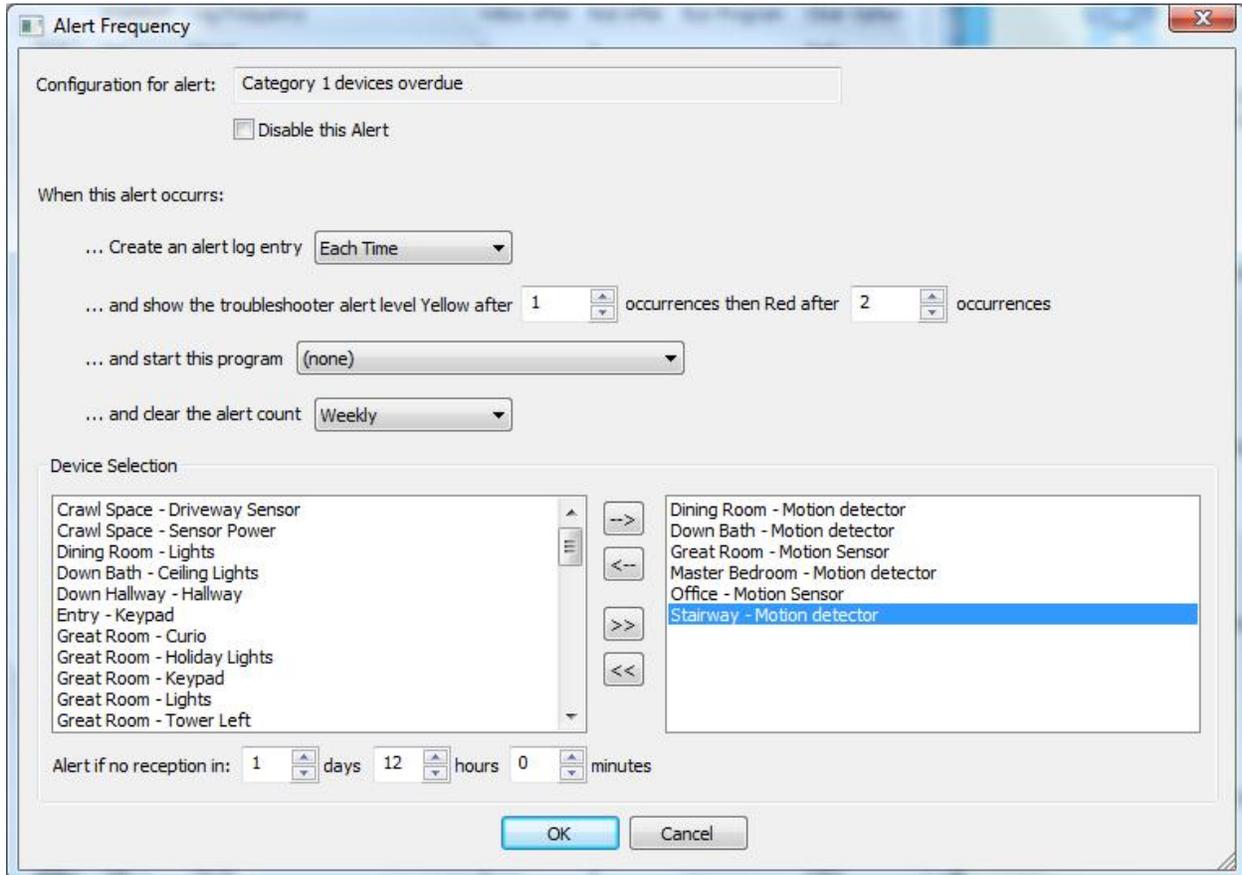
... and show the inspector alert level Yellow after occurrences then Red after occurrences

... and start this program

... and clear the alert count

In V13 four "user" alerts were added that can be triggered by programs using "_AlertAdd" function in a Compute element.

There is now a new Alert called "Category x devices overdue". This alert lets you specify a set of devices and a time. The time specified is the duration since that last reception from the device where it will be considered "overdue". That is, HCA should have seen a reception within that time.



If any device in the set doesn't have a reception within the specified time then the alert happens.

There are 4 sets of devices so you can create different "overdue" requirements if needed.

This alert is a bit different than the others. In that the alert can automatically clear when the device is heard from. Normally alerts only clear by user action. But in the overdue alerts, if a device is currently overdue and then is heard from, then any alert for it is cleared.

As with all alerts you can configure an action if the alert occurs to run a program. In the case of overdue alert the program is only run once even if the device continues to be overdue. The program doesn't get run again until after the device has checked in and then goes overdue again. For example suppose that device D is overdue if not heard from in 2 hours. Here is a timeline of what happens.

00:00: HCA Starts

00:50: Device D reception

02:50 Two hours since device D was heard from so the overdue action happens and that can include starting program P

05:00: Device D is still overdue but program P isn't started

05:04: Device D is received. This clears the overdue alert

07:04: Device D is again overdue. This time program P is run as part of raising the alert

The overdue mechanism only checks every 15 minutes for overdue devices so you can't rely on to-the-minute reporting.

Per device alerts

Alerts that apply to devices now keep track of the count per device. This means that you can say, for example, "log once" and that means log once per device. The alerts that this kind of counting applies to:

- UPB: Missing sequence in transmission
- UPB: Missing ACK
- Insteon: Message ACK
- Status request and no reply
- Confirm receipt needed retransmission
- Confirm receipt failure after all retransmissions
- Program error
- Overdue devices

While the logging for these alert conditions is driven by the alert setup and the per-device count, the red/yellow/green determination for the alert comes from the totals of all devices.

Suppose, for example if you have the "Program error" configured as "log once a day", yellow after 4, red after 8.

1. If program P makes an error it will log since it is the 1st for "P" today.
2. If program P makes another error it will not log since it is the 2nd today
3. If program P1 makes an error it will log since it is the first for "P1" today

In this alert (program error) each program treats the log options based upon its own counts. To continue this example, at this point there have been 3 total program errors and that's still a green condition.

4. If program C now makes an error, it logs – first for "C" today - and since the total count is now 4 a yellow condition exists.

The main ribbon for the troubleshooter now shows counts for “red”, “yellow”, and “alerts”.



Clicking the Inspector opens the Inspector viewer, clicking on the Alerts opens the Alerts viewer.

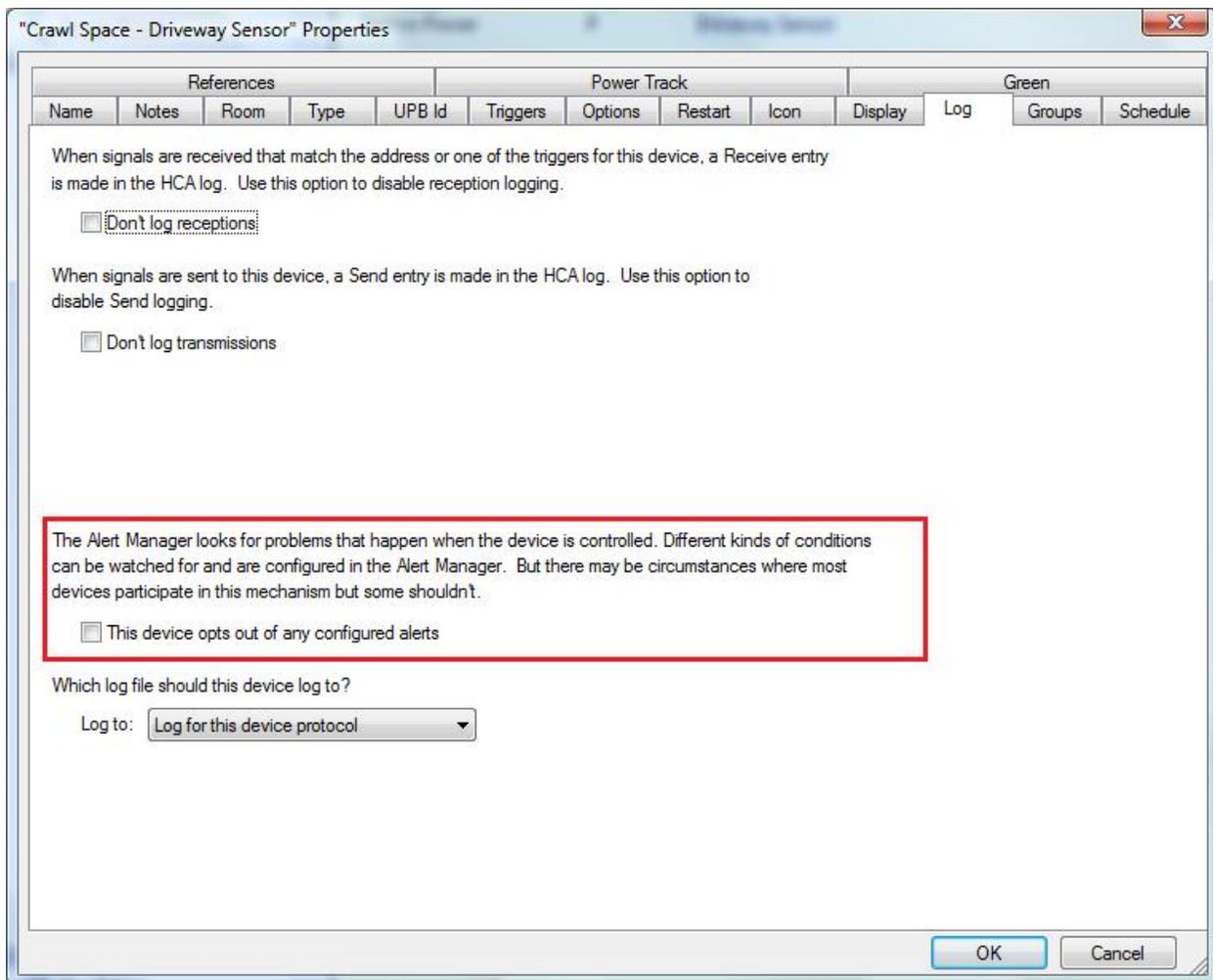
An alert that appears in the details log no longer has - in itself - a red/yellow/green level. This is different than the static inspector which classifies all things it finds as red or yellow so each item listed in the inspector report does have a level for it.

The alert mechanism is different and looks at all alert counts and makes a determination if overall there is a green/yellow/red level for alerts. So the determination of the current level is the higher level of the static inspector level and the alerts level.

Functional change: There are two issues related to the interfaces. In previous HCA versions, the inspector was reporting on both interface capabilities issues (you have an UPB device and no UPB interface configured) and interface problems (you have a UPB interface configured but it isn't connecting).

This was changed in HCA 13. Interface capabilities problems are still reported as an inspector issue. Issues with interfaces configured but not connected are now reported to as an Alert issue.

Finally, there is now a "disable" for devices to opt out of the alert general mechanism. Suppose for example you have a device that tends to have status request response problems and you have a program that polls it several times to get the result. You may want that device to opt out of the alert mechanism so it doesn't get reported upon.



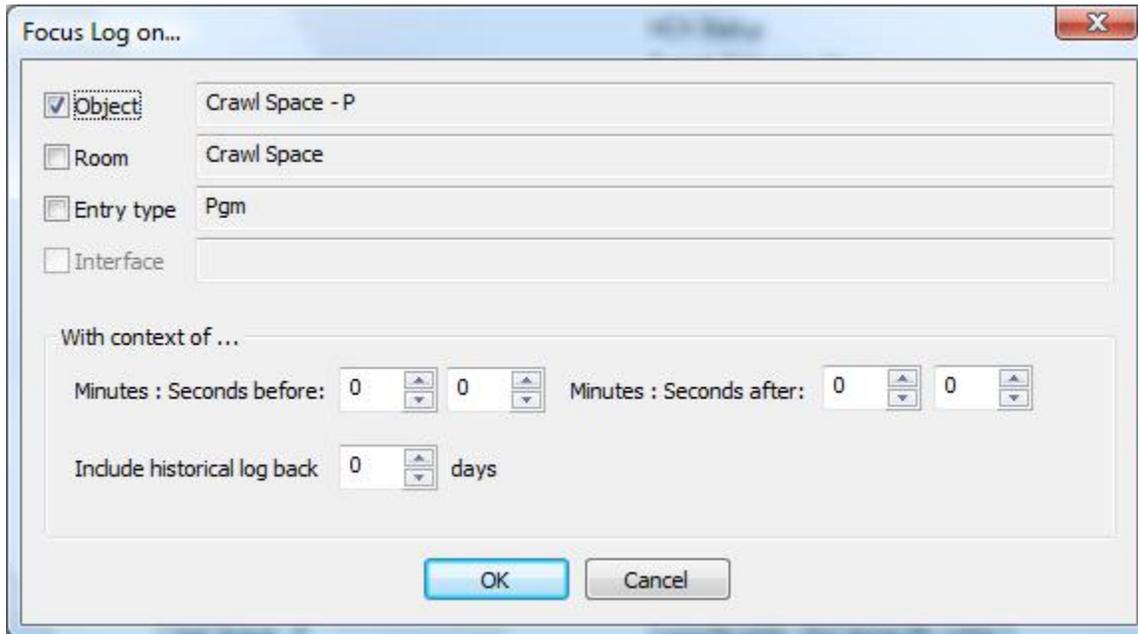
Why would you use this feature?

Alerts are the key feature to keep track of what happens as HCA executes the design. Do device respond to commands sent? Do weather update happen? Are device that should be heard from daily really heard from? Think of Alerts as a way to keep track of the "health" of your design.

Log Viewer Changes

The log viewer icons have been made more regular and easy to see.

A powerful new tool was added to the log viewer to enable you to better locate useful information in tracking down problems. If you double-click on a line in the log viewer a “Focus” dialog opens.



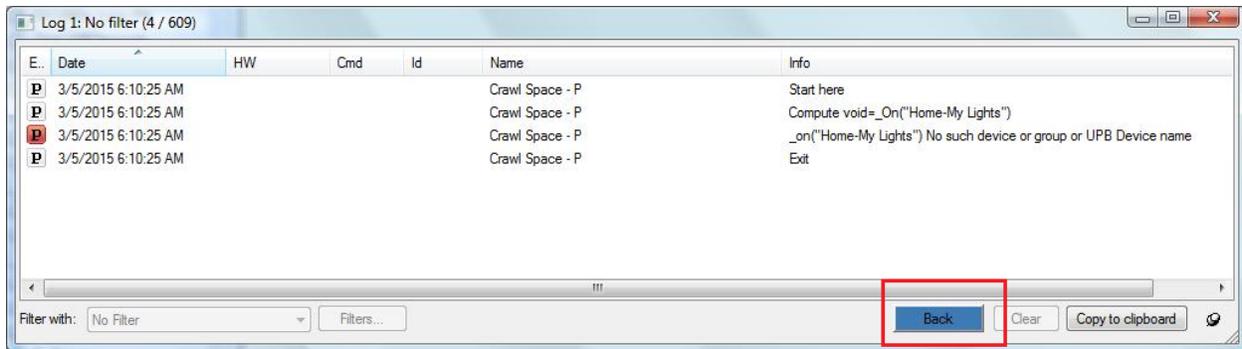
You can choose what you want to focus on. The log entry you double-clicked on pre-loads the details into this dialog and from the checkboxes you can specify what you want to focus on. Each box checked filters out only those log entries that have all of those properties.

But this is more than a simple filter. The “context” settings let you not only see the log entries that meet the filter requirements but also to show log entries before and/or after these to show the context of what you are looking at.

For example suppose that you focus in on a program but that program is started by some reception. Using the context settings you can focus on the program and the action that started the program.

Once you close the focus dialog with OK then the log viewer shows the focused entries. At the bottom of the log viewer is a “Back” button to remove the focus.

You can create a new focus on a view that was itself created by a focus. The back button moves you back a focus level.



The log focus has one additional property. You can have the focus go back into the historical log – if you have created one – a number of days. This is the first time that HCA itself supports viewing of log entries in the historical log.

Why would you use this feature?

The log is a great debugging device but it can be difficult to find what you are looking for and more importantly the "context" of what you found. This feature lets you answer the questions like, why did this device go on at this time and discover from the context the other things happening around that time.

New Compute functions

Many new functions have been added to the compute and compute test element. These are:

Void = _AlertAdd (alert #, text)

Raises an alert in the Alert Manager. The Alert manager lets you configure for user alerts. These four “user alerts” are there to tie into this compute function. What happens with the alert – does it log, does it change the alert level – is all part of the alert configuration. The _AddAlert function only causes the alert to be raised what happens is up to the alert configuration.

DateTime = _AutoOffTime (device-or-room-name)

Returns the time when a device or room will auto-off. If there is no auto off timer running returns a bool of zero. Use the _IsBool on the result before assuming it is a date-time.

DateTime = _LastReceptionTime (device-or-room-name)

Returns the last reception time for the named device. If a room name is supplied then it returns the most recent reception time for any device in the room.

Number = _RGB (red, green, blue)

Returns the encoded color value with the red, green, and blue color values

Void = _TileUpdate (tilename, code, x, y)

The tile name is the name of the tile to change

Codes are:

- 0: Change label
- 1: change tile colors
- 2: change image path for image tiles
- 3: change text for text tiles
- 4: refresh tile. Like the “refresh” checkbox in the TileUpdate element

The _TileUpdate automatically refreshes the tile after the changes are made. Code=4 only for those times where you want to refresh without making any other changes.

The “code” argument determines what the x and y arguments are used for:

- 0: string (label text)
- 1: number (tile color), number (tile text color)
- 2: String (image path)
- 3: string (tile text)
- 4: x and y can be omitted

For code 1 the new `_RGB` function will come in handy.

In addition to these new functions, several other functions were added in a point version of HCA 11 but not documented. They now are in the expression builder and documented in the expressions User Guide chapter.

- `_FileOpen`
- `_FileClose`
- `_FileWriteString`
- `_FileReadString`
- `_FileExists`
- `_Output`
- `_HomeMode`
- `_SetHomeMode`
- `_IconChangeEx`
- `_CurrentWattage`
- `_SetCurrentWattage`
- `_InsteonBeep`
- `_ChangeSchedule`

The `_IsOn`, `_IsOff`, and `_IsDim` function have changed. If the device supports status requests, and if the optional parameter doesn't say to not use a status request, then if the device doesn't respond, a numeric code is returned rather than a `BOOL`.

You can test for this:

```
X = _IsOn("Home – Lights");
```

In a Compute Test element you could decide if it was not responded to by:

```
_IsBool(x)
```

If that function returns "Yes" then the status request was answered and "x" contains yes or no. If `_IsBool` returns `FALSE` then the device didn't answer the status request.

As documented in another section, these thermostat functions should no longer be used:

- `_temperature`
- `_setpoint`
- `_mode`
- `_IsFanOn`
- `_IsEconomy`
- `_IsAuxheat`
- `_Humidity`

These have been replaced by `_GetThermostat` and `_SetThermostat`

Why would you use this feature?

This all are features requested by users and can be thought of as new tools in the toolbox when constructing your own programs.

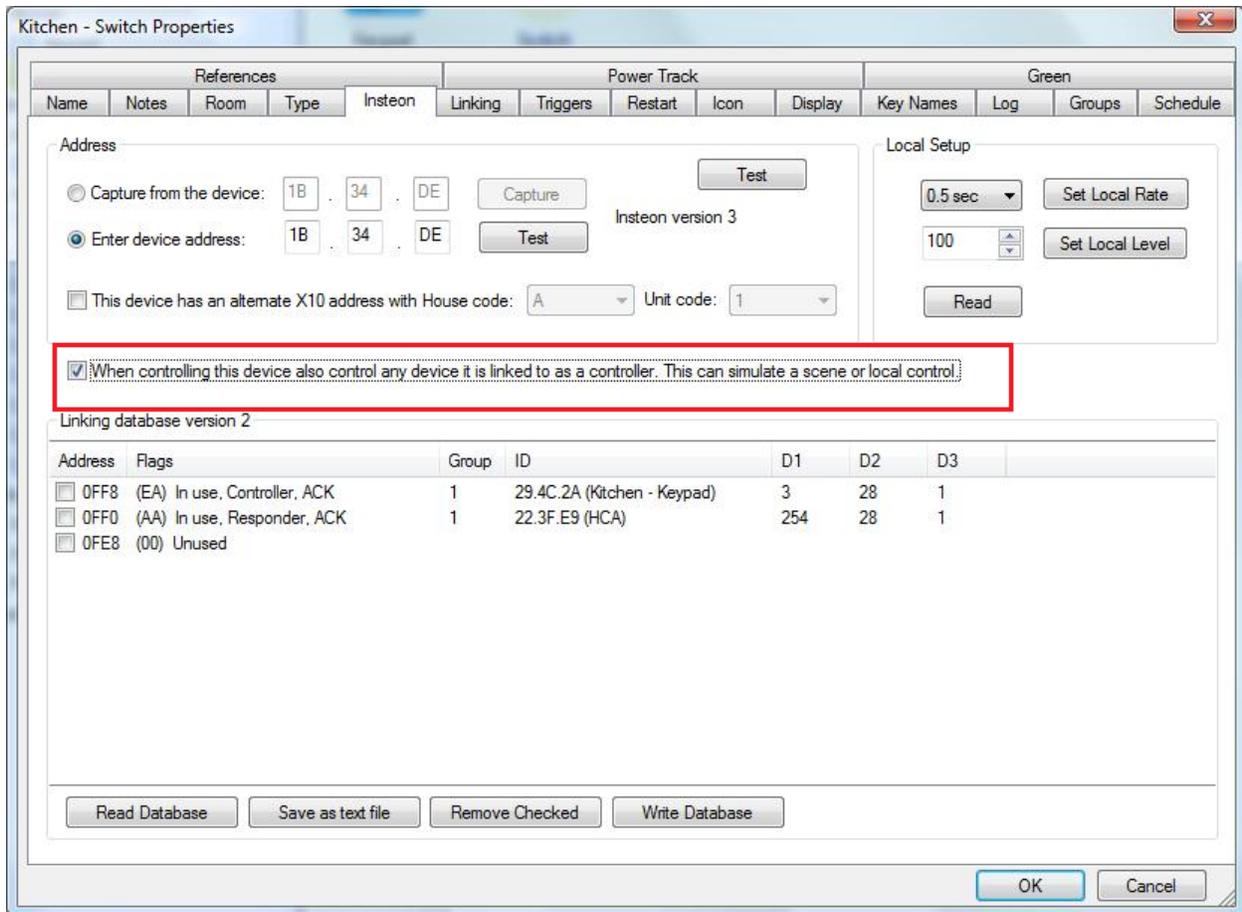
Insteon changes

One usual problem with Insteon devices because of its linking model is that no controller can pretend to be another controller. An example:

Suppose you have a switch that controls a load linked as a controller of a KeypadLinc button LED. You tap on the paddle, the load comes on, and the KeypadLinc button turns on as well.

What happens if you use HCA to turn on the load? The load comes on but the KeypadLinc button does not. This is because HCA can't "pretend" to be the switch and have the KeypadLinc button LED respond. Up until now you had to either create a scene where HCA was the controller and have the scene include the switch and the button LED. Or you had to create a program that controlled the switch and the button LED. In either case you had to remember to use the scene or program rather than control the switch directly.

A new option has been added to make this simpler.



With this option enabled – and if HCA has the current linking table of the device – then when the device is controlled, HCA also controls any device that is linked to the device as a responder.

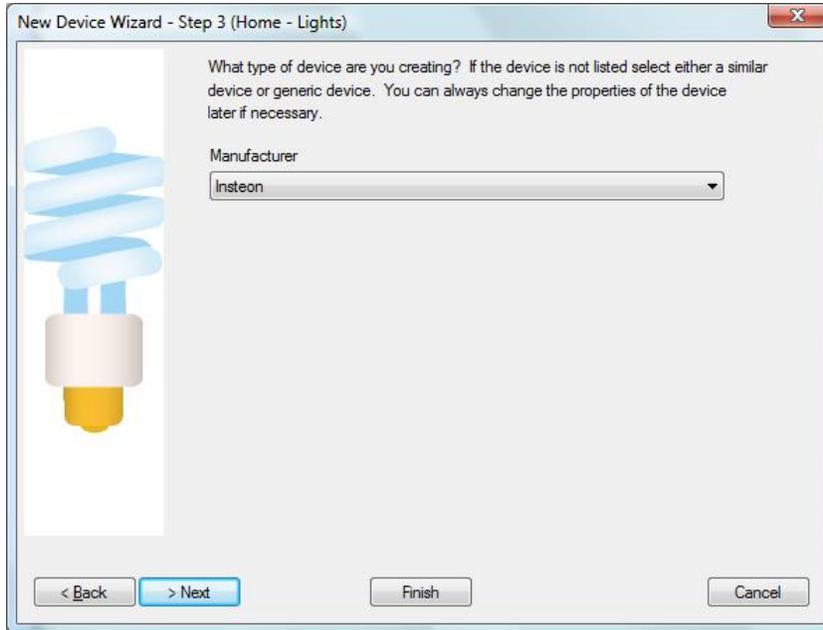
Why would you use this feature?

This can make it much simpler to, for example, control a device and also keep a keypad button LED correctly in sync with that device. Previously you had to create a scene and have HCA use that scene to control both the device and the keypad button.

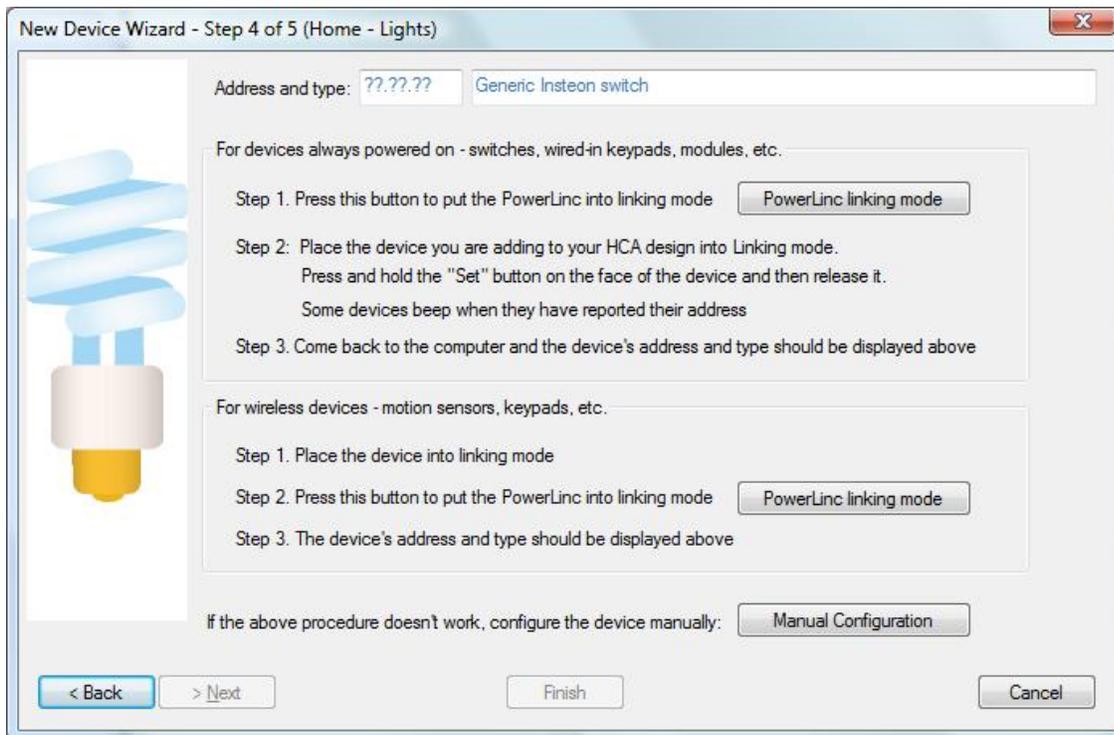
Support for the older model Insteon Hub has been moved to legacy status. This means that you can use it as an interface once you enable the Legacy option for it in the HCA Options "Legacy" tab. We do not at this time anticipate adding support for the new model Insteon hub.

Insteon Device Wizard and Properties

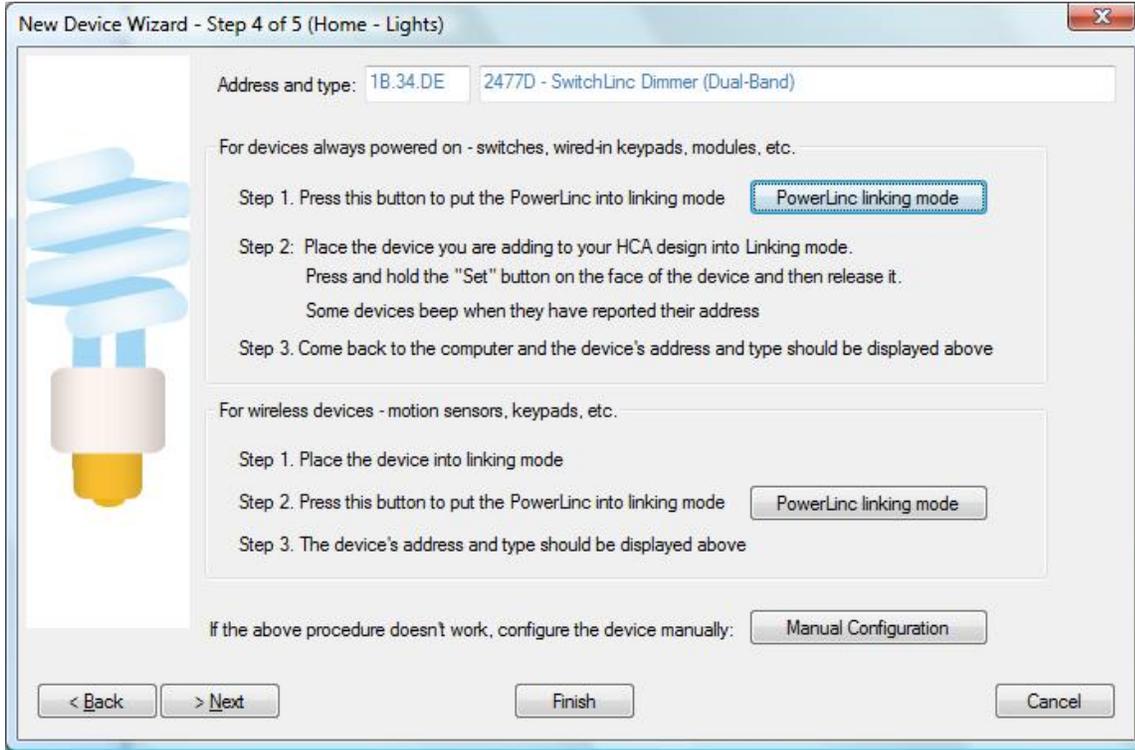
The device wizard and device properties for Insteon devices has changed. At the step where the type of device is selected, for Insteon you no longer get to choose the device model.



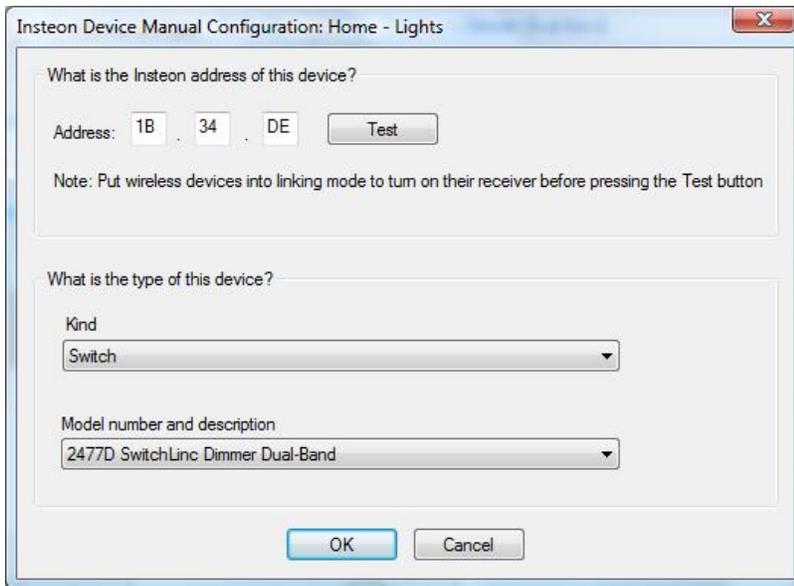
The wizard next step:



HCA now captures the address and the device type – like Insteon Multi-Add already does. When the “Linking mode” button is pressed and the device is then put into link mode, then the device address and type displays.

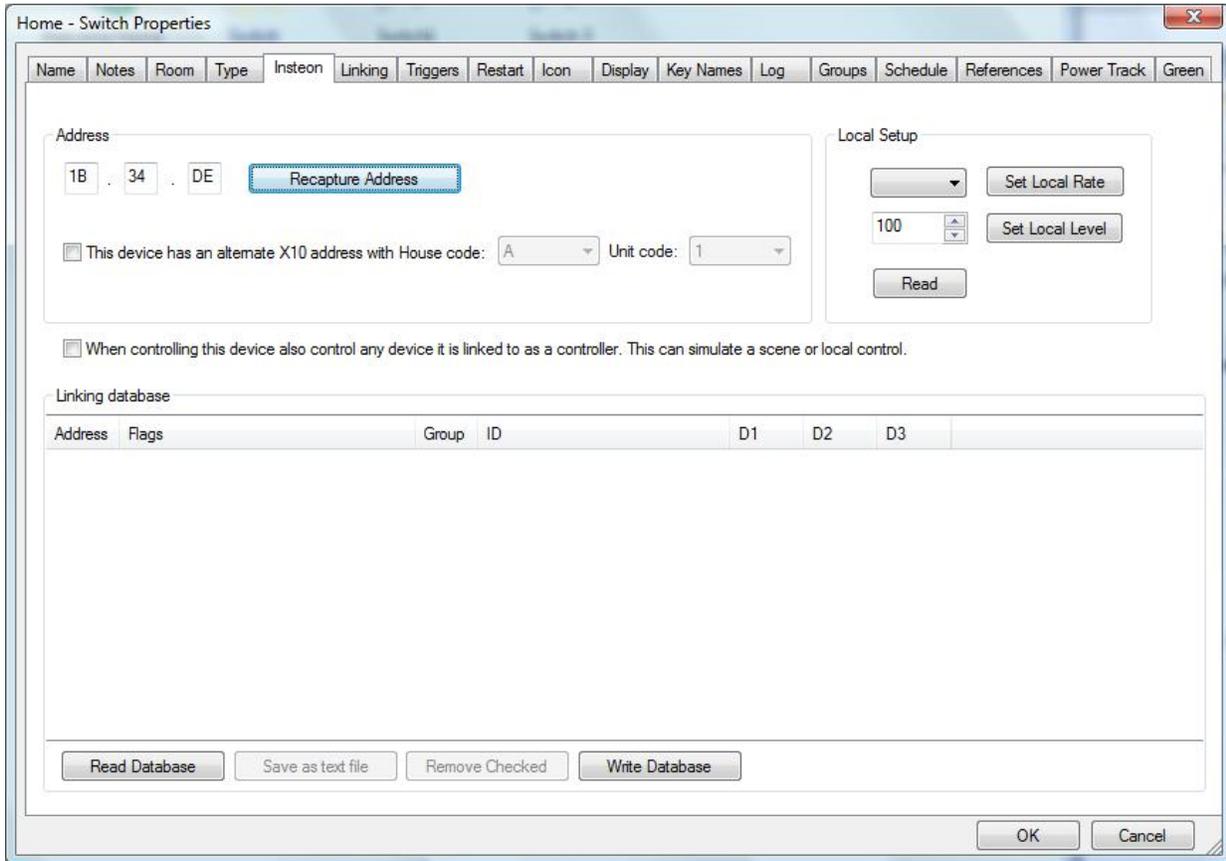


If you have a device that can't be handled in this way then you can use manual setup:



The “Test” button sends an Insteon get version command and if it isn’t answered a dialog appears giving you get a chance to put the device into linking mode so the responder link gets built as needed for the newest firmware devices.

The properties dialog has also changed.



The “Recapture address” button opens a very similar dialog to the Wizard step.

Why would you use this feature?

This makes it much easier to add Insteon device to you design and have HCA build the necessary links needed to control them.

Insteon Multi-Unit devices

When you add the EZIO4O, EZIO2X4, EZFlora, OutletLinc, FanLinc, IOLinc multiple HCA devices are now created - one for each sub-unit of the device. These can be used like any other Insteon device for control and display. Only the primary unit is used to access the address, linking table, and other Insteon parameters.

Device Kind	Primary Unit
EZIO4O	Output 1
EZIO2X4	Input 1
EZFlora	Zone 1
OutletLinc	Top outlet
FanLinc	Light
IOLinc	Input

The general changes are:

- As described above, for these device types, HCA devices are created for all units of an Insteon device
- The secondary devices are ignored when using the Insteon Map and Insteon Clean tools
- Insteon Multi-Add and the Designer when working with Insteon devices perform the same creation of sub-unit devices as the Wizard does.
- The "Add newly discovered devices" in the Insteon Capture Network feature has been removed.
- When using the "Recapture address" feature on a multi-unit device that new address is propagated to all secondary devices.
- Any existing files containing any of the device types are fixed to use this new model. That is, devices of these types with the same Insteon address are connected together in the primary and secondary device model they would have been in if newly added.
- Device replace now ignores secondary devices and updates any secondary devices with the new device address when complete.

A major change in the Visual Scene Editor (VSE) was completed. How it effects each device type is described below.

ON-OFF Outlet

Using the VSE you can now build scenes that are:

- Controlled by HCA to turn on or off the top and/or bottom outlets and other devices simultaneously
- Controlled by an Insteon controller (KeypadLinc, IO Module Input) that turns on or off the top and/or bottom outlets

FanLinc

Using the VSE you can now build scenes that are:

- Controlled by HCA to turn on or off the fan and/or light and other devices simultaneously
- Controlled by an Insteon controller (KeypadLinc, IO Module Input) that turns on or off the fan and/or light

Open-Close Sensor, Hidden Door Sensor, Leak Detector

Using the VSE you can build scenes that are controlled by the sensor and control Insteon receivers (switches, KeypadLinc LEDs, output devices). When using the VSE, the programming pauses when the scene contains a wireless device to give you time to place it into linking mode so its receiver is on and can then receive the programming commands.

EZFlora

Each zone can be independently controlled. HCA works the same way as the device does -only one zone is active at a given time - and updates the other zone devices to show their state as appropriate.

Messages from controllers that are linked to the outputs are received and cause HCA to keep its state up to date as long as the linking table of the controller and the EZFlora are up to date.

Using the VSE you can build scenes that are:

- Controlled by HCA to turn on or off a zone and other devices simultaneously
- Controlled by an Insteon controller (KeypadLinc, IO Module Input) that turns on or off a zone.

IOLinc

The one input can be linked to HCA and used as a trigger to programs

Messages from controllers that are linked to the output are received and cause HCA to keep its state up to date as long as the linking table of the controller and the IOLinc are up to date.

For the IOLinc output the VSE lets you build scenes that are:

- Controlled by HCA to turn on or off the output and other devices simultaneously
- Controlled by an Insteon controller (KeypadLinc, IO Module Input) that turns on or off the output.

EZIO2X4

The EZIO2X4 doesn't implement the Insteon command set as do other devices and so it is not possible to remotely link the inputs and to remotely write its linking table. In order to use the inputs as triggers it is necessary to manually link them to the PowerLinc. The tech note on this device describes the manual linking process.

Also, it isn't possible to use the Visual Scene Editor with this device for inputs or outputs.

Once the device is added to HCA and the inputs are manually linked then it can be used to trigger HCA programs. The outputs can also be controlled from schedules or programs.

If you keep HCA's copy of the device's linking table up to date – by reading it from the Insteon tab of the device or by Network Database Capture – then HCA can correctly track the outputs state if you have added it to other scenes by manual linking.

Additional Insteon Device Support

HCA 13 has improved support for these Insteon device types. [There are tech notes for each of them explaining the changes.](#)

- Hidden Door Sensor
- Leak Detector
- Open Close Sensor
- Thermostat – improved handling of auto reports from the thermostat
- On/Off Outlet – explained in its own section in this document
- Mini-Remote

Changes to Mobile Clients

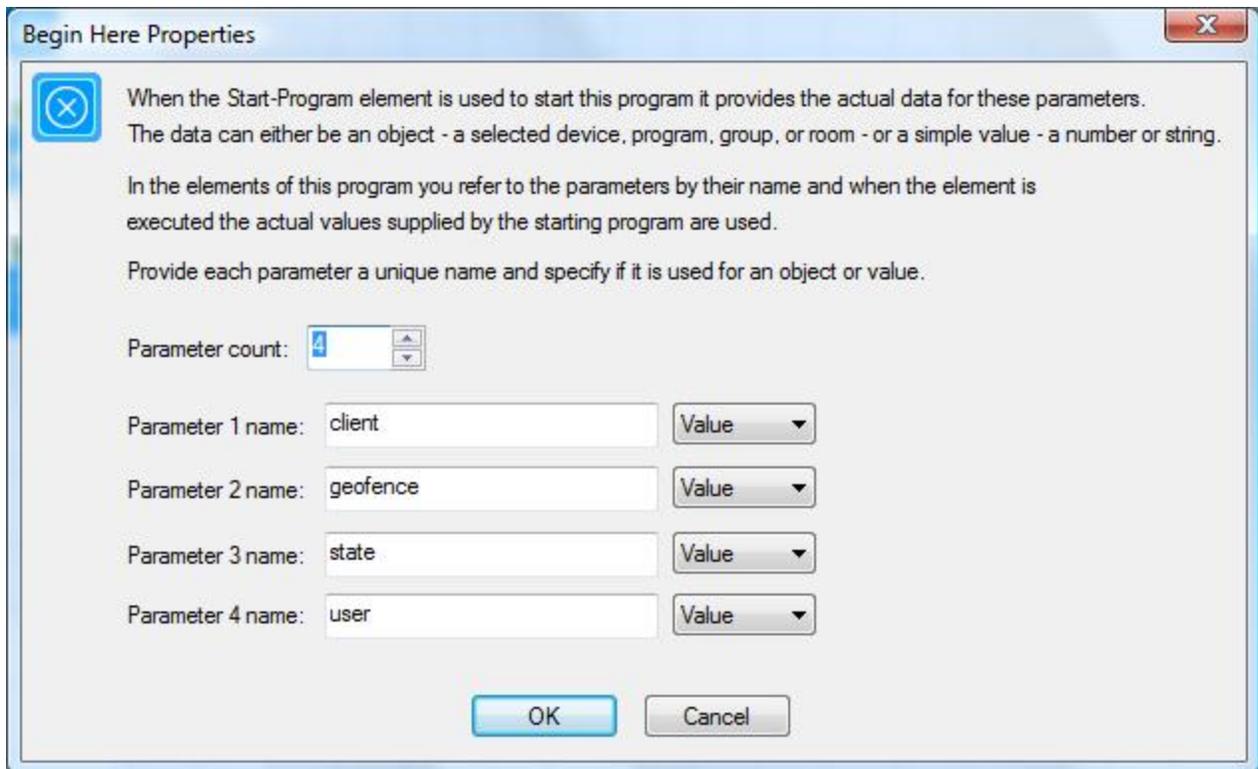
- iOS app implements “Geofences”. This is a new method for location aware programs. See description below.
- iOS app now has a Log viewer. Select “Settings” - “Activity Logs” – “HCA Log” in the app to view
- iOS app now loads new and/or changed icons from the server use by the application. Select “Settings” – “Custom Icons” – “Icons” to sync the app icons with the server.
- iOS app now shows both static and dynamic HTML displays. A static HTML display has a URL. A dynamic HTML display is created by HCA by reading a template file and replacing placeholders with values before display.

Geofence support in the iPad/iPhone HCA app

A GeoFence is a circle drawn around a physical point on a map with a given diameter. Crossing into and out of this area triggers a GeoFence alert. Once Geofences are configured in the iOS application, a program on the HCA server is started when crossing the fence.

On the HCA side, create a “Location Aware Programs” folder – must be named exactly that. In that folder place the programs that you want to select from started when configuring a fence. You can have many different programs or only a single program. You select what program to run during the configuration of the fence. You could have a single program used for all fences and have the program test a parameter to see which fence it is, or you could have different programs for different fences. That’s up to you.

When a program is started as the result of entering or leaving a fenced area, the programs receives as arguments several pieces of info. Each program that is started by entering or exiting a Geofenced area should have 4 parameters like this:

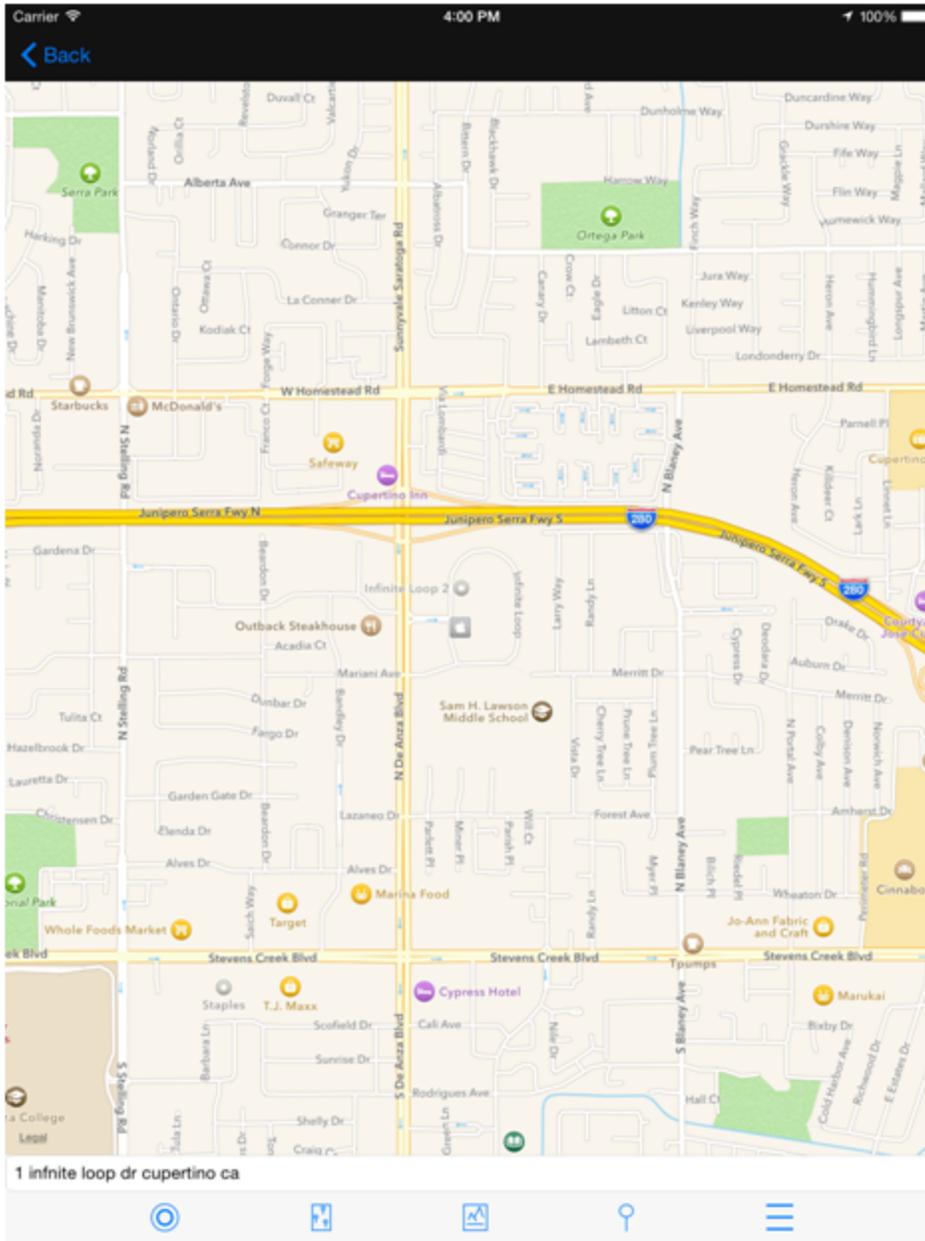


When the program is started:

- Parameter 1 has the name of the client from the iOS App HCA Settings “Client Name”.
- Parameter 2 is the name of the GeoFence that is happening. This is the name entered on the iOS device when the fence is created.
- Parameter 3 is the state. The string “**Left**” is used when the area is left and “**Arrived**” when it is entered.
- Parameter 4 is the description text, if any, given when creating the Geofence in the iOS app

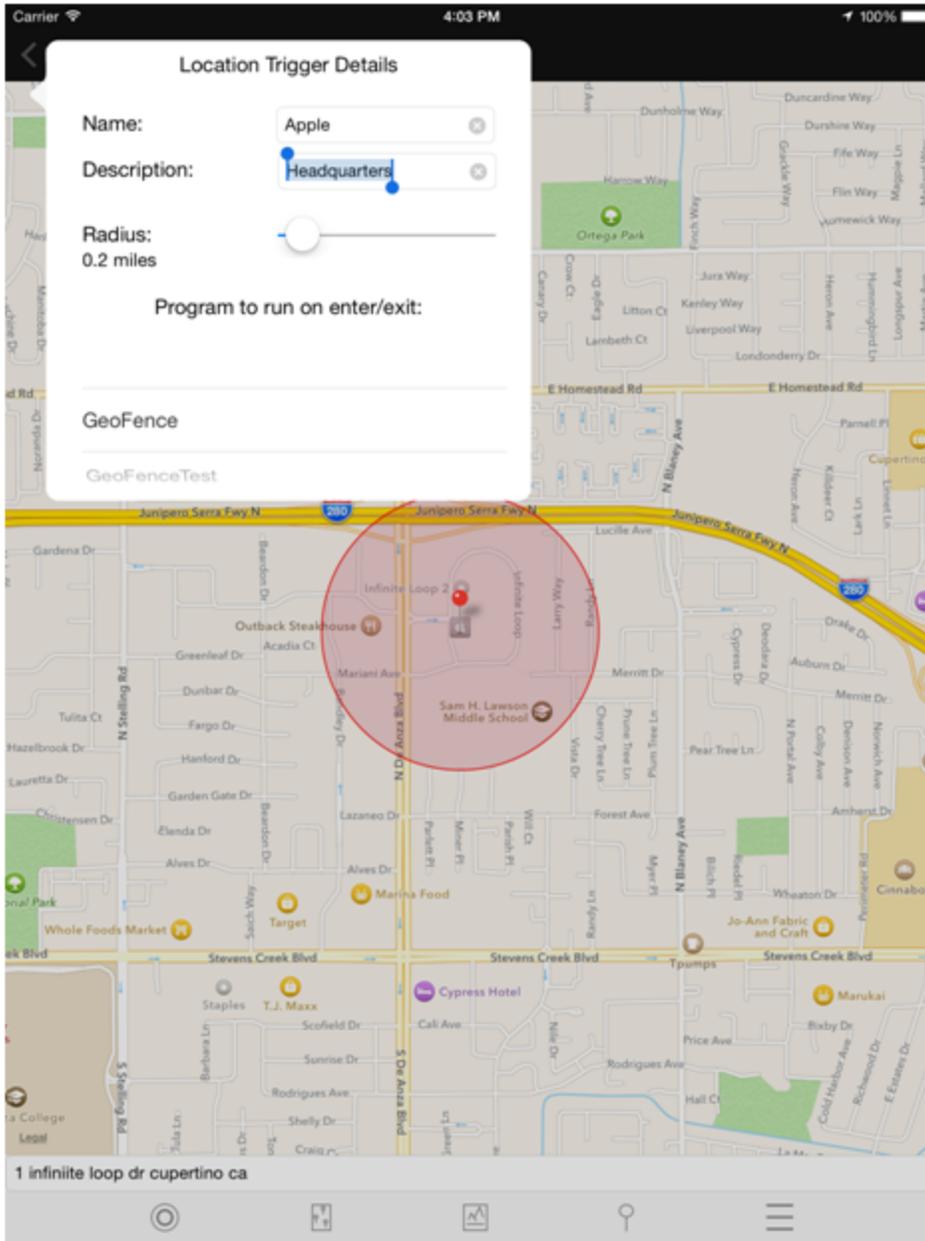
Configuring a Geofence On the iOS device

In the iOS app after connecting, click on “Settings” in the app menu (left side “slide out panel” on iPad and the base screen on the iPhone app) and then “Geofences” in the “Location Triggers” section. The map opens.



Zoom and move the map to center the desired location and then tap the 4th icon from the left at the bottom of the map - the “circle on a stick” icon. It creates a point on the map and a popup appears for specifying its configuration.

You can also long-press an area on the map to create a Geofence at that location. You can also center the map location by using the “Enter address to search” at the bottom of the map.



Enter a name for this Geofence – it is passed to the associated HCA Program when that program is started, a description – also passed as an argument to the program, the radius of the fenced area, and the HCA program to run when entering or exiting the area. Only the programs in the “Location Aware Programs” folder are available for selection.

The other icons at the bottom of the map are (left to right):

- Enable/Disable automatic map zooming to the current location. It is enabled when the icon has a solid blue center and disabled when the inner circle is not filled in.

- Set zoom level to show all fences.
- Show / Hide Terrain overlay on the map.
- Create a point on the map.
- Open a list of fences. Once on that list you can “tap” to open its properties or delete it. To delete, slide it left until the “Delete” button appears then press the “Delete” button.

Why would you use this feature?

These changes make the iOS client much more powerful – the log viewer and custom icon additions. Geofences can make it possible to have HCA react to the coming and goings of occupants of the home.

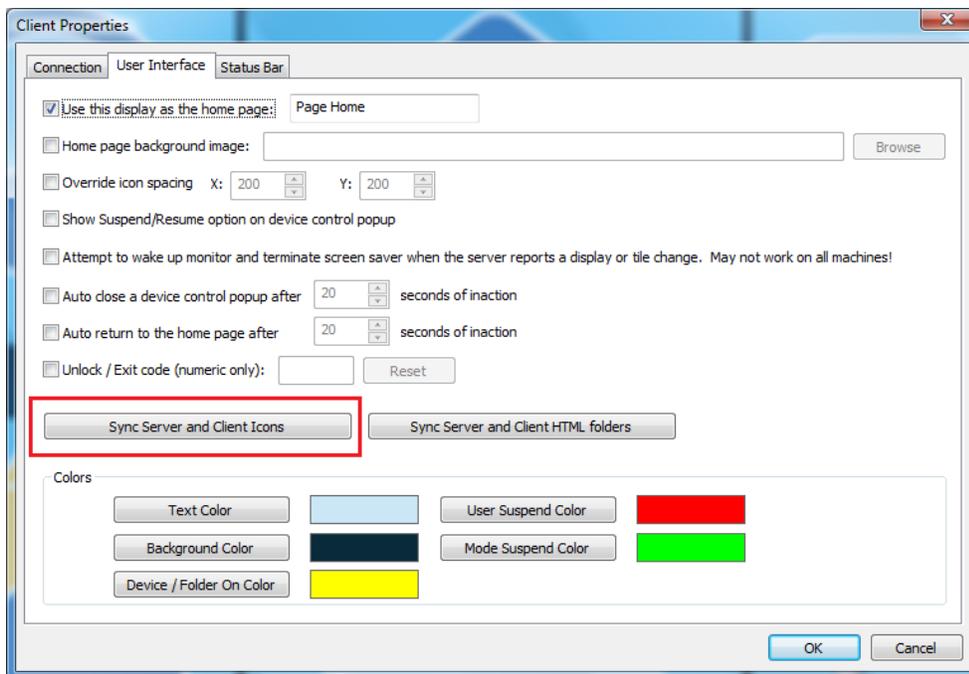
Changes to Windows client

The HCA Windows client can now access the three logs maintained by the HCA Server.

In previous versions of the Windows client, any custom icons created by users were sometimes moved by the client to the icon theme on the client machine. The rules for if an icon got moved or not were complex. This has been replaced by the “Sync Server and Client Icons” operation. This is started from the client control panel settings on the “User Interface” tab.

Any icons in the theme folders on the server machine that are not on the client machine or are newer than those on the client machine are copied from the server to the client.

All you need do now is to remember to place your new or changed icons in the server computer theme folders and then on each client use the “Sync” button.



The Windows client now uses the icon theme for a display chosen as the “Control UI Theme” in the display’s properties on the “Color and Theme” tab. Previously it always used the 100 x 100 theme. The client now installs the 36, 54, and 110 themes.

The client “Server status” now shows the Alert log rather than the Inspector log in previous versions.

The client also shows in the “Alerts tile” the same list of alerts as is shown in HCA in the Alerts details display. The color of the client tile bar reflects the Alert color – green, yellow, red – in the same way that HCA shows in the Alerts ribbon icon.

The Inspector list is no longer available in the Windows client. Only the alerts log is now available.

Why would you use this feature?

Using these new features you can craft a much improved user interface.

HTML Folder Sync

When the Windows client encounters a dynamic HTML display – a HTML display where HCA generates a HTML file from a template file by replacing placeholders with actual data - the client requests the server to generate the HTML and send it to the client. The client stores the HTML file in a sub-folder in its “Temp” folder. The folder is named with the name of the display (possibly made into a legal filename). It

then stores the HTML it receives from the server in a file named with the display name with an “html” file type.

For example, if the display was called “Current Status”, the client creates this folder.

C:\users\kimberly\HCAclient\Temp\Current Status

And the HTML it received from the server would be stored in that folder in a file named “Current Status.html”. The client then uses the Windows browser component to render the HTML in a window by passing to it the path to the HTML file.

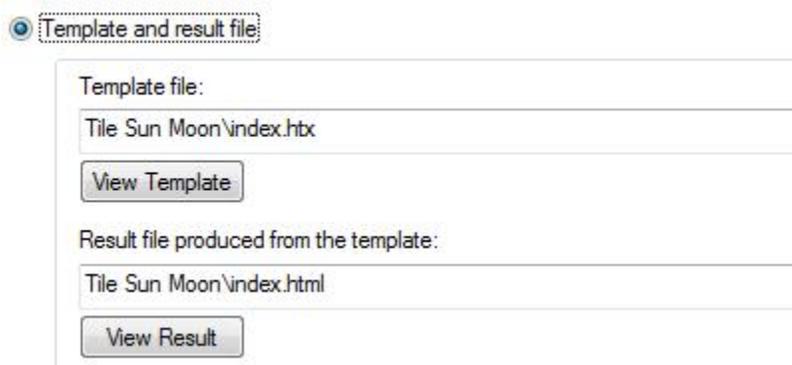
If the HTML file references any images or a style sheet then those auxiliary files must be in the same folder as the HTML file or in sub-folders referenced by relative paths. For example the HTML may reference an image as “apple.jpg” or “images\apple.jpg”.

It is a good practice to configure the display to locate the HTML template file and result file in its own folder. In this way any images or style sheets specific to this HTML file will not conflict with the files for other HTML displays

In this HCA design configuring the display properties using relative paths makes this all work. For example, on the server computer is a folder named “Tile Sun Moon” that contains the template, result, and all the images files along with the style sheet. That folder is a sub-folder of the folder that contains the design .HCA file.

For example, if the HCA file was in c:\users\kimberly\HCA, then the folder that contains the HTML template and auxiliary files would in c:\users\kimberly\HCA\Tile Sun Moon

The HTML display is configured as:



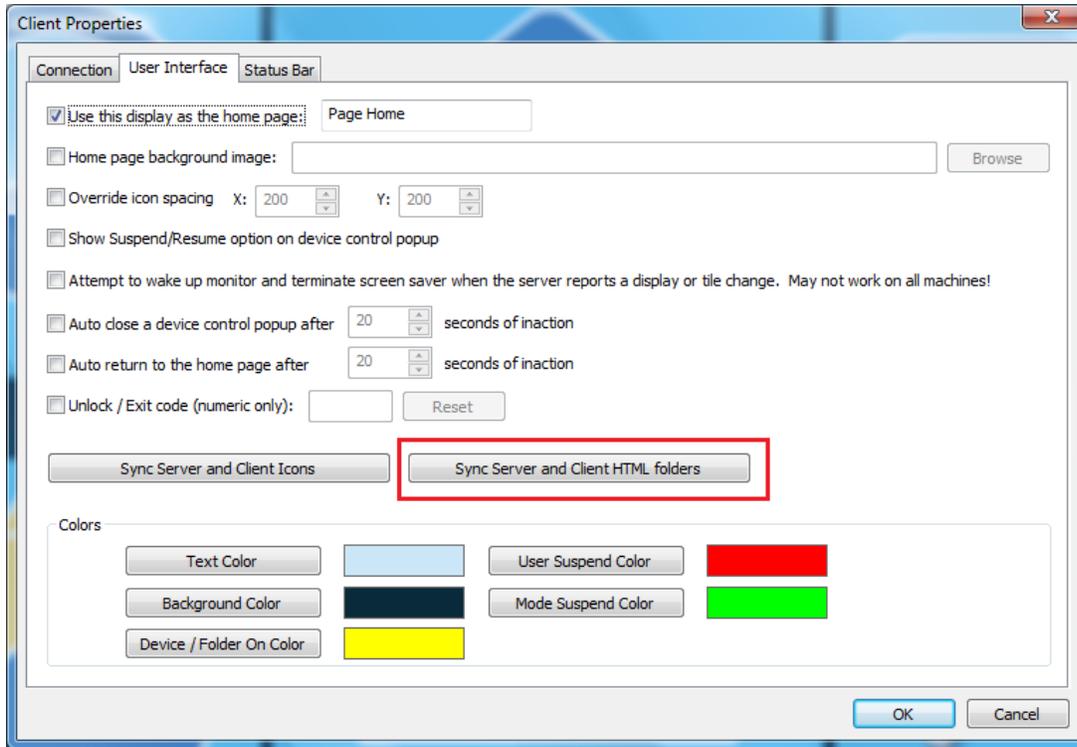
Template and result file

Template file:
Tile Sun Moon\index.htm
View Template

Result file produced from the template:
Tile Sun Moon\index.html
View Result

It is best practice for each dynamic HTML display to have its own folder that contains the HTML template, result, and whatever auxiliary files it needs.

The problem was that in HCA 12 there was no method built-in to the HCA client to move any auxiliary files from the server to the client. In HCA 13 and the client this is now possible. In the Client Properties dialog on the User Interface tab is a button for this.



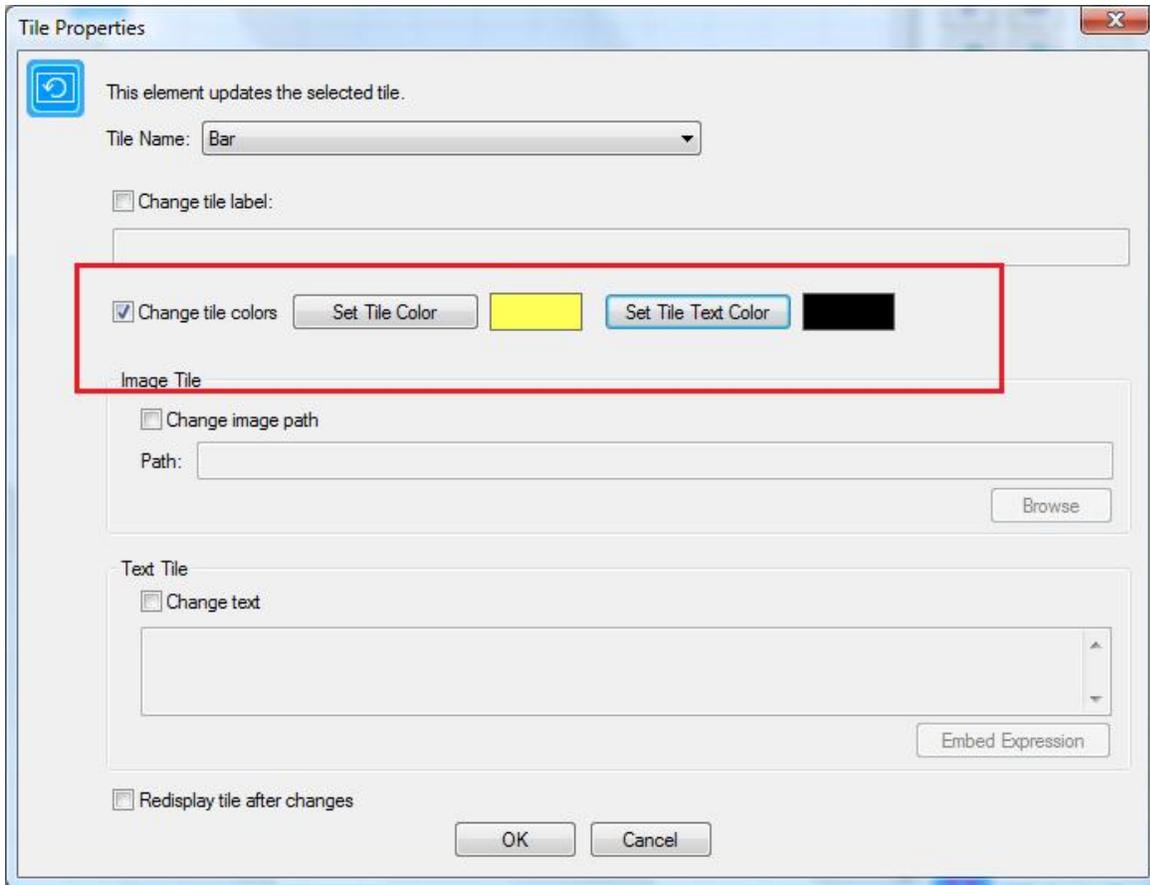
When this button is pressed then for any dynamic HTML displays, the client requests from the server the contents of the folder where the HTML result file resides. Any files in that folder are moved to the client computer with the same directory structure as was on the server computer.

Why would you use this feature?

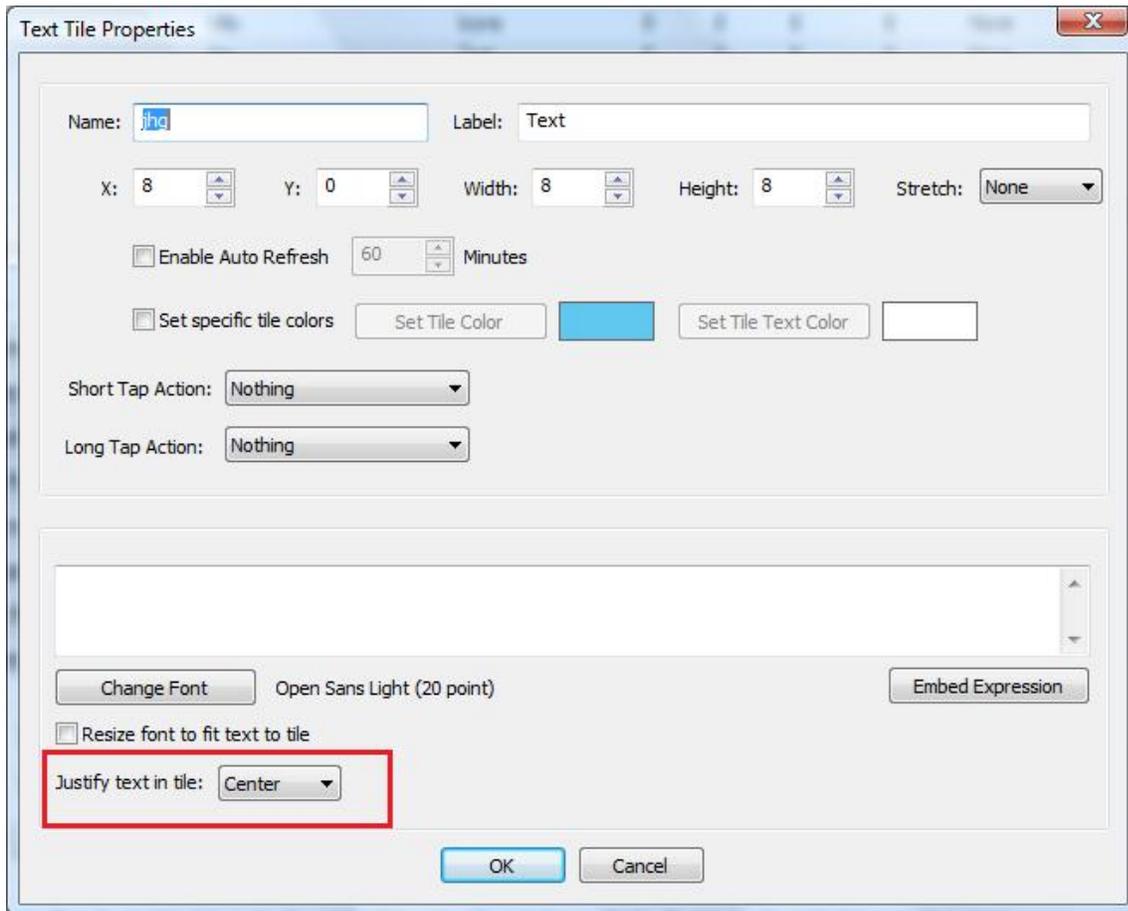
If you have HTML displays that are generate by HCA and that use associated files – style sheets and/or images- this can make it much simpler to configure clients.

Changes to the Tiled displays

The Update Tile VP element now lets you change the tile color



The Text Tile now has a property for setting the text justification within the tile.



Why would you use this feature?

Using these new features can improve displays you created.

New Icons

HCA and the clients now ship with these new icons.

- Flag Green
- Flag Blue
- Flag Red
- Insteon keypad 6 button

- Insteon Keypad 8 button
- Insteon Keypad 4 button
- UPB Keypad 6 button
- UPB Keypad 7 Button
- UPB Keypad 8 Button
- Window and Door icons now have open (ON) and close (OFF) representations. There are two sets for both door and window – “Window” and “Window A”, “Door” and “Door A”. They both are used in a similar manner. For example, in the “Window” set, the ON representation image is an open window, and the OFF representation image is a closed window. The “A” set reverses this. This lets you use the set that corresponds to your sensor reporting state.
- New holiday Lights choice
- The Thermostat icon has been updated

The flag icons are useful when you are using a program to place an icon on a display that represents some condition rather than a controllable device.

In addition, there are new grayscale icon themes:

- Theme36_36G
- Theme54_54G
- Theme110_110G

The advantage of these new themes is that all icons have both on and off representations. The off representations are in gray and the on representations have color. This makes the difference in the on and off images very clear.

It is important that you manually remove the old stateless Window and Door icons from the icon themes after installing HCA 13. These would be in your HCA Documents area in the “Icon Themes” folder. Look in the “Theme_36_36”, “Theme_53_54”, and “Theme_110_110” folders for the files “Door.png” and “Window.png”. Remove these files from each theme.

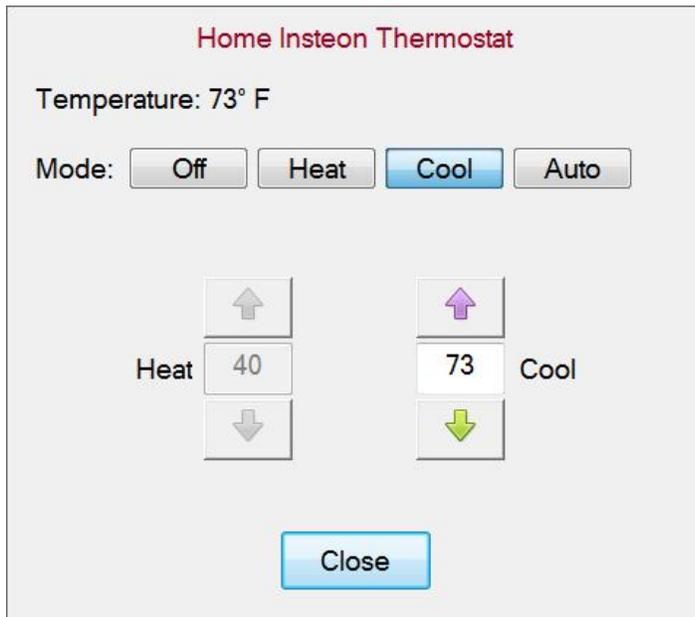
Why would you use this feature?

More and better icons are always useful.

Thermostat Support

Support has been added for the NEST thermostat. During this process many features of thermostat support have changed.

The Thermostat popup used in the Control UI is now the same as in the Development UI.



The sliders for the setpoints were replaced with simple up and down buttons.

The buttons in the popup for the Mode and Fan are now fully live – that is, as soon as you press them commands are sent to the thermostat to effect the change.

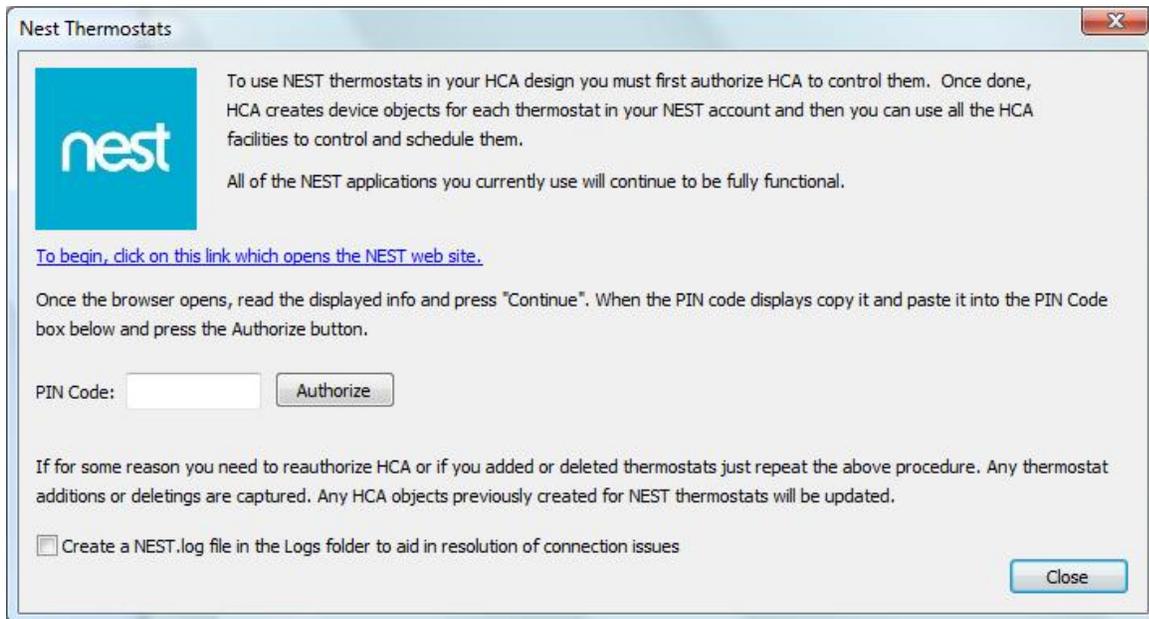
The up and down buttons move the set points by one degree. After a 1 second pause the set point change is sent to the thermostat. This lets you change the setpoint with multiple button presses and then when the desired setpoint is entered the command is sent to the thermostat.

NOTE: Each supported thermostat has different remote access capabilities and restrictions. Some types make the heat and cool setpoints available for modification at any time. Other thermostat types only allow for remotely changing the setpoints based upon the mode. For example, the Insteon thermostat only let the heat setpoint to be set when in heat mode, the cool setpoint only when in cool mode, and only the heat setpoint when in Auto mode. The UI now implements this restriction by enabling and disabling the setpoint controls as needed.

Nest support

Support for the NEST network connected thermostat has been added to HCA. This thermostat has a much more complex protocol for communication than any of the other supported thermostat types. Communication is not made with the thermostat directly but with the NEST servers which then control the thermostat. Before HCA can work with it you must first authorize HCA to access all the thermostats in your NEST account.

Select the “Nest” button in the Protocols category.



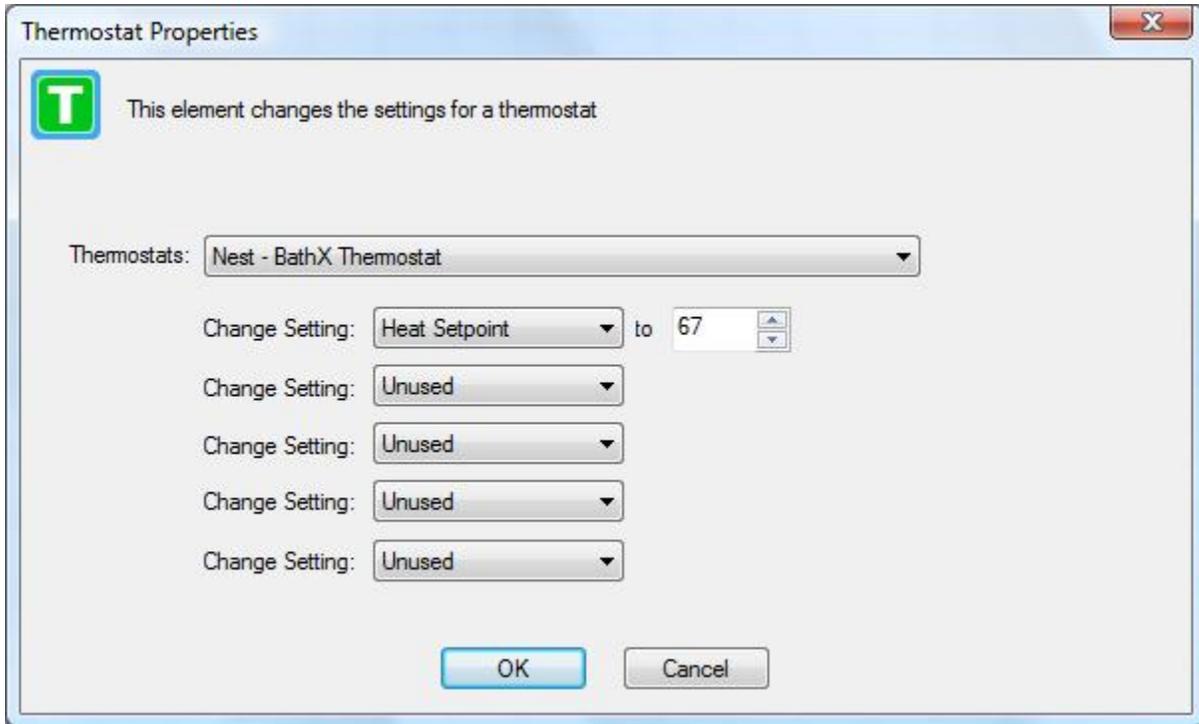
As the dialog says, you first click on the link to login to your nest account and then to authorize HCA. The web site displays a PIN Code that is entered in HCA. Once completed then HCA creates devices for each thermostat in your account. This need only be done once. If you add or remove thermostats just repeat this process and new thermostats are added and removed ones deleted.

CAUTION: When you click on the link your default browser is opened. Some users have reported problems with Internet Explorer in that the page comes up with a big gray rectangle and no controls or other display. Chrome seems to work fine.

NOTE: Providing HCA access to the thermostat in no way limits you use of all the other options for controlling your NEST thermostats – mobile clients, browser support, etc. In fact, you should view the HCA support of the NEST thermostats as an adjunct to all the facilities that NEST provides in their mobile applications.

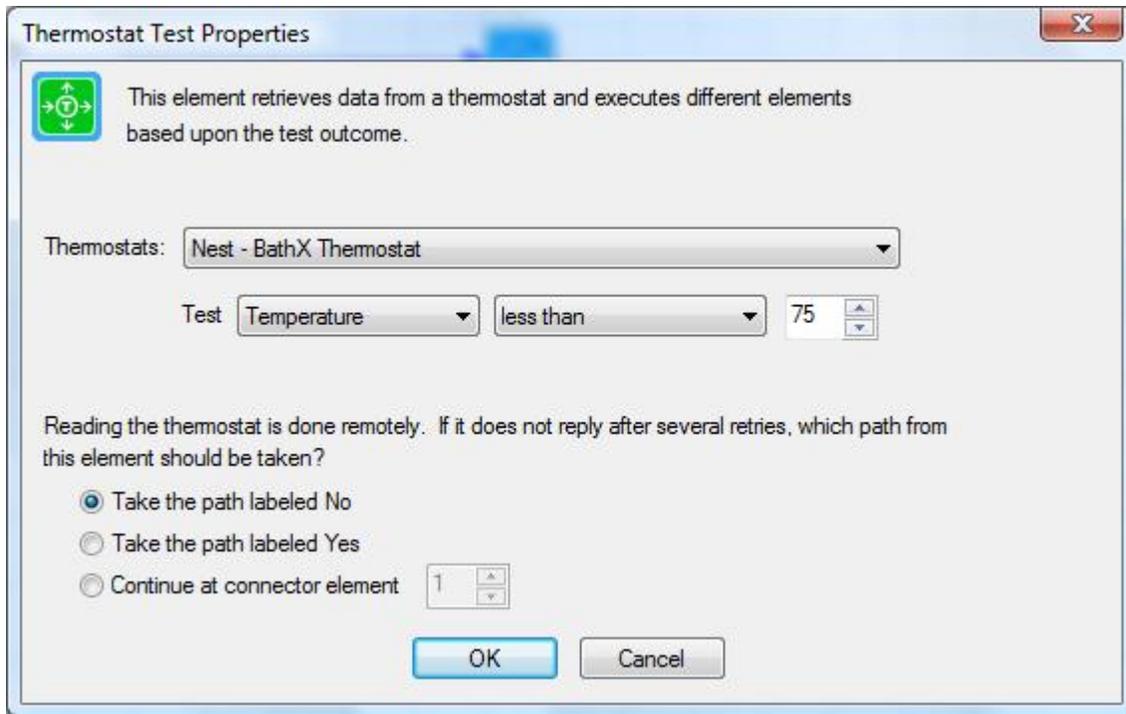
Thermostat VP element

The Thermostat VP element has been changed. In the element you can select up to 5 settings to change and the value of that setting. Only those settings for the thermostat selected that can be remotely changed are available.



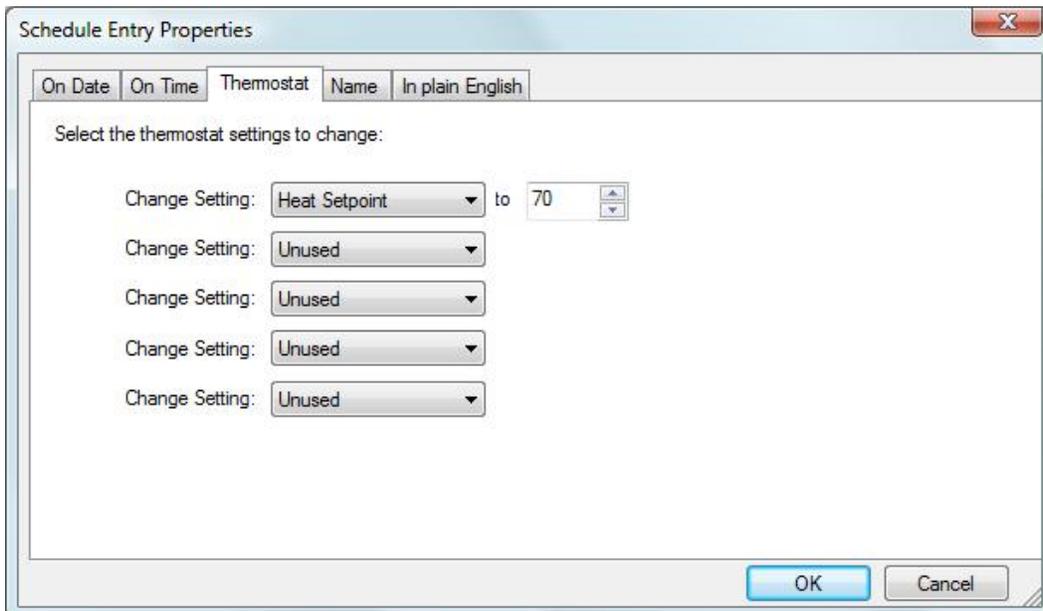
Thermostat-Test VP Element

The Thermostat-Test VP Element has been changed. The method of selecting the test has been changed. The settings available for test are only those available for the selected thermostat.



Schedule Entry for thermostats

The schedule entry properties have been changed to look more like the VP Thermostat element properties. The available selections and the effect of the schedule entry is the same as the VP element.



HCA Windows Client support

The HCA Windows client has been updated to show a similar thermostat control popup as in the HCA Development and Control user interfaces.

Compute and ComputeTest element support.

Two new functions have been added for thermostats. While the existing thermostat functions `_temperature`, `_setpoint`, `_mode`, `_isFanOn`, `_IsFanOff`, `_IsAuxHeat`, and `_Humidity` have not been removed they are now on legacy status.

New function `_GetThermostat`

Value = `_GetThermostat` ("thermostat name", code)

The "Thermostat name" is the two part name for the thermostat.

The code is the setting to be retrieved. These are:

Code	Setting	Returned value
0	Temperature	Integer value
1	Heat Setpoint	Integer value
2	Mode	Off = 0, Heat = 1, Cool = 2, Auto = 3
3	Fan	0 = On, 1 = Off
4	Economy	0 = On, 1 = Off
5	Aux Heat	0 = On, 1 = Off
6	Humidity	Integer value
7	Cool Setpoint	Integer value
8	Has Leaf (NEST only)	0 = On, 1 = Off
13	Nest Mode (NEST only)	0 = Away, 1 = Home

It is up to program that use this function to request only settings supported by the thermostat and for the setpoints only when in the correct mode.

The return value is the setting retrieved or an error. You should use the `_IsBool` on the result to determine if you have received the requested data or an error.

New function `_SetThermostat`

Bool = `_SetThermostat` ("thermostat name", code, value, code , value, ...)

You can supply up to 11 arguments. The 1st is the two part name for the thermostat device. The next two arguments are the code and value of the setting to change. The next arguments are up to 4 other code-value pairs.

The valid codes are:

Code	Setting	Data
0	Temperature	Integer value
1	Heat Setpoint	Integer value
2	Mode	Off = 0, Heat = 1, Cool = 2, Auto = 3
3	Fan	0 = On, 1 = Off
4	Economy	0 = On, 1 = Off
7	Cool Setpoint	Integer value
13	NEST mode (NEST only)	0 = Away, 1 = Home

Note: When changing the NEST mode it changes **all** thermostats in the structure associated with the thermostat being controlled.

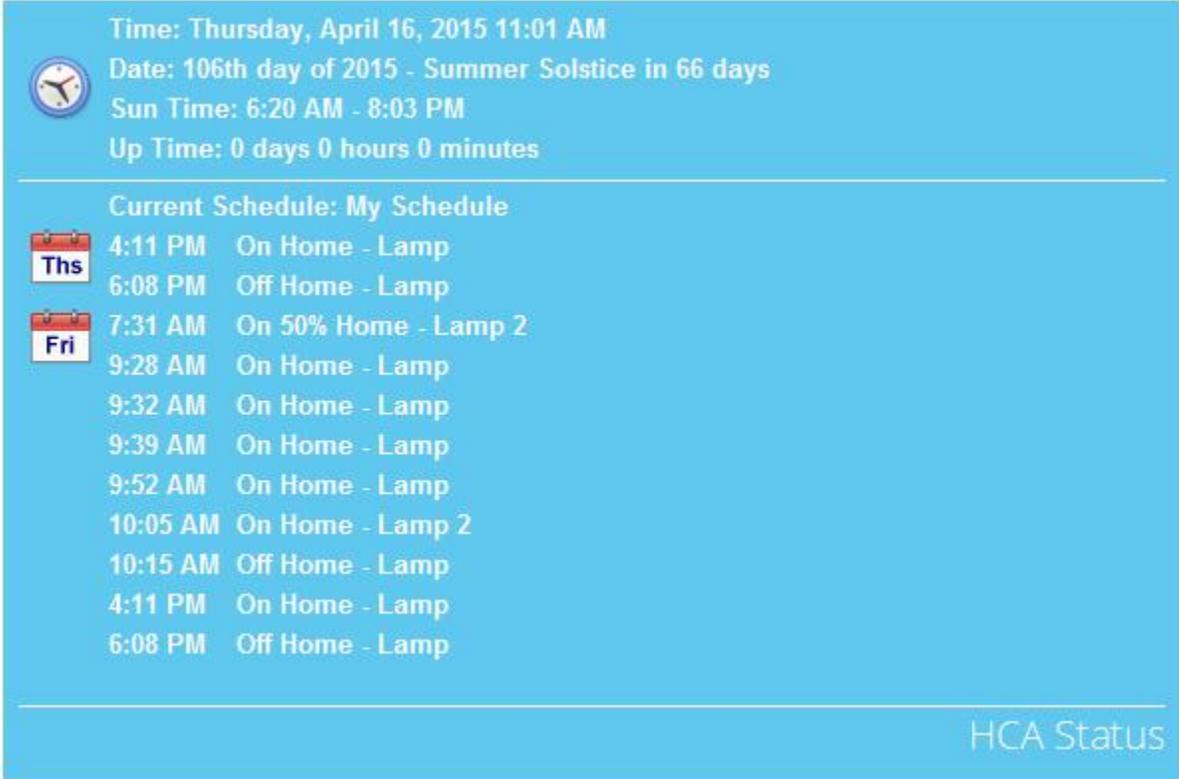
Why would you use this feature?

Improvements to thermostat support and the addition of support for the popular NEST thermostat makes HCA a much better product. Previous support was not up to standard.

Status Tile

A new tile type was added to the implemented tile types. The status tile show information similar to what the HCA Status dialog shows. The exception is it doesn't show the running program list.

The tile needs to be a reasonable size as the text isn't wrapped horizontally. The list of schedule items shown is limited by the tile height. There is no configuration of this tile type except for the standard tile options of colors and label. The status tile is supported in both HCA both stand-alone and client-server and in the Windows client.



The screenshot shows a blue-tinted status tile. At the top left is a circular icon with a clock and a gear. The text in the tile reads: 'Time: Thursday, April 16, 2015 11:01 AM', 'Date: 106th day of 2015 - Summer Solstice in 66 days', 'Sun Time: 6:20 AM - 8:03 PM', and 'Up Time: 0 days 0 hours 0 minutes'. A horizontal line separates this from the schedule section, which is titled 'Current Schedule: My Schedule'. The schedule is a list of time events: '4:11 PM On Home - Lamp', '6:08 PM Off Home - Lamp', '7:31 AM On 50% Home - Lamp 2', '9:28 AM On Home - Lamp', '9:32 AM On Home - Lamp', '9:39 AM On Home - Lamp', '9:52 AM On Home - Lamp', '10:05 AM On Home - Lamp 2', '10:15 AM Off Home - Lamp', '4:11 PM On Home - Lamp', and '6:08 PM Off Home - Lamp'. Small calendar icons for 'Ths' and 'Fri' are next to the first two items. The text 'HCA Status' is in the bottom right corner.

Why would you use this feature?

As with other changes, these allow for creation of improved displays.

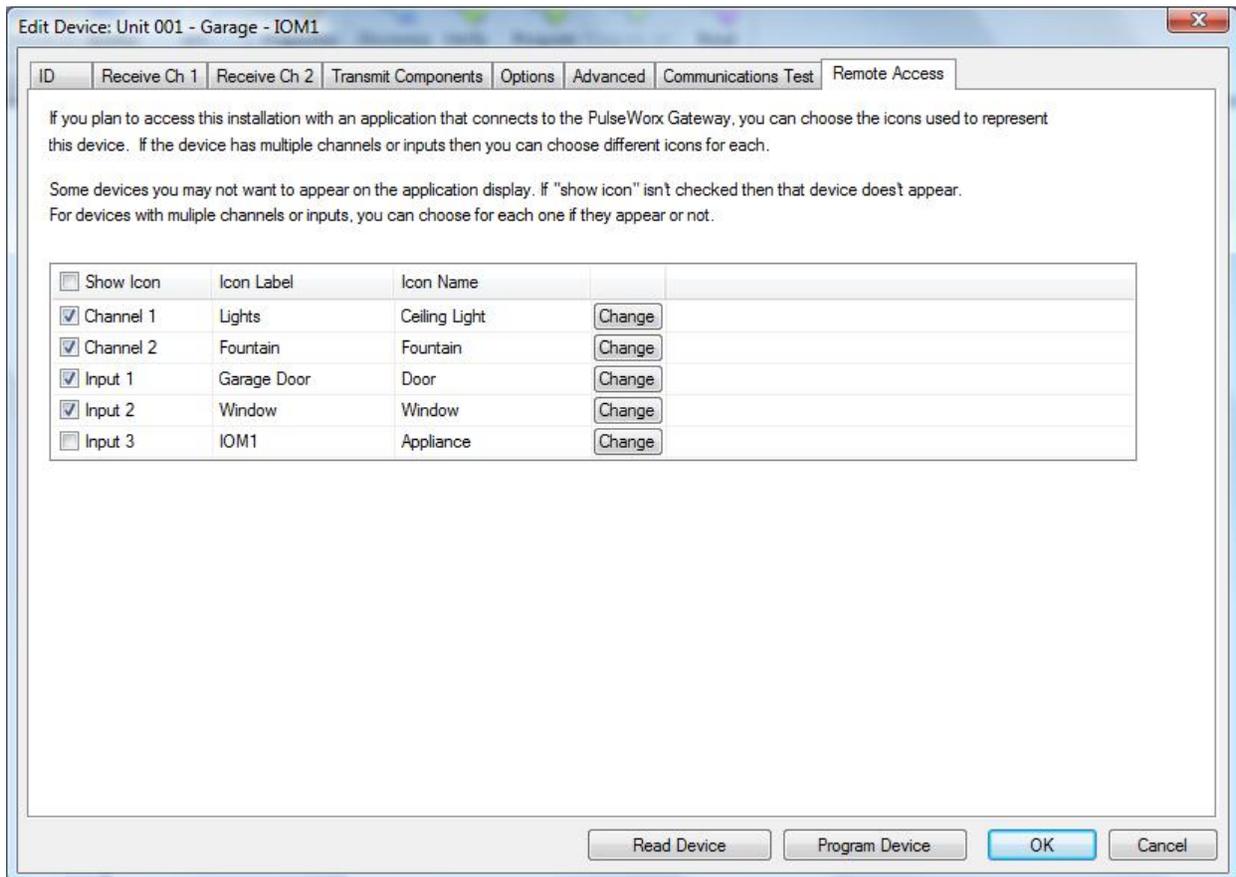
UPB Import

In previous versions of HCA, import of UPB networks was deficient in several areas related to "Input Control Modules" (ICM) and "Input Output Modules" (IOM) device types. HCA only created a single device for the ICM and didn't update the icon for it as its inputs opened and closed. For the IOM it always created 2 devices— one for each load but no devices for the inputs. So in both cases HCA can't use icons to show the state of the inputs.

In the latest version of UPStart – the UPB configuration programs – several changes have been made. In this new version of UPStart, for each channel of a device you can specify the icon to represent the device as well as the text below the icon. Also for each channel you get to choose if there should be an icon seen at all. For example, if you are only using 1 input channel of a 3 input IOM you don't need to have icons for the other 2 inputs.

HCA now imports this information and uses it when creating devices and controllers to represent the UPB devices.

Here is a screen image from UPStart. In this example, this IOM would cause 4 devices to be created in HCA – 2 for the loads and 2 for the inputs. No icon would be created for input 3.



Additionally these are now imported as well.

- In UPStart you can now specify the icon to use for the room. This is in the Application menu on the “Export” submenu.
- Keypad button names – specified on the “Engraving” tab in the properties for the PulseWorx keypads is now imported and used when showing the keypad in the Control UI popup.

Note that if you have previously imported your design, HCA does not change the HCA device name (the icon label) and icon selected for existing devices. The import doesn't want to override your icon choice you have made in HCA and also doesn't want to change the HCA device name which might break some HCA programs.

With HCA 13 you must use the new version of UPStart, export your design to a UPE file, and import it into HCA. Before doing the export from UPStart you may want to look at all the ICM and IOM device in your UPB design and update the information on the "Remote Access" tab.

The big advantage of these changes is that icons are now available in HCA that show the state of the inputs you are using on each ICM and IOM. Also no devices need be created for any unused IOM outputs that are not being used

However there are implications of this change.

TEST Element Problem

The problem comes in any TEST elements you have in programs. Previously you would test the input of an ICM as "Device:x" where "x" was 1 or 2. For an IOM it would get even stranger. HCA would always create 2 devices for the IOM. Each appeared to have the same 3 inputs. Suppose that the IOM was called "Garage Doors". HCA 12 would have created these devices.

1. Garage Doors channel (1)
2. Garage Doors channel (2)

In the test element you would see:

1. Garage Doors channel (1)
2. Garage Doors channel (1):1
3. Garage Doors channel (1):2
4. Garage Doors channel (1):3
5. Garage Doors channel (2)
6. Garage Doors channel (2):1
7. Garage Doors channel (2):2
8. Garage Doors channel (2):3

Obviously, this was not convenient.

With this new system, suppose that you were not using outputs 1 and 2 and only inputs 2 and 3. Also suppose you called input 2 "Washer" and input 3 "Dryer".

In the test element you would then see these devices as choices.

1. Washer
2. Dryer

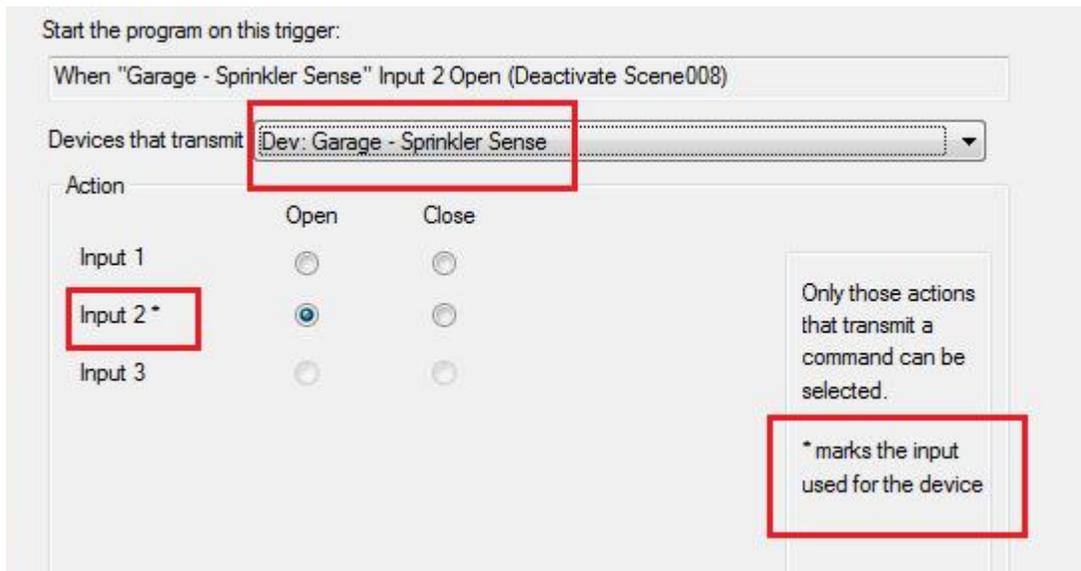
This is more convenient.

But you will have to find any test elements that tested the IOM inputs and adjust them.

HCA considers any CLOSED input to be ON and an OPEN input to be OFF.

Trigger Issue

The other “problem” isn’t so bad because HCA 13 didn’t want to break everything. You can still pick any of the devices created for an ICM or IOM and use that to establish a trigger. You will have access to all inputs regardless of which device you pick. Added to the trigger dialog was one change - it points out what input is associated with the device you select to establish the trigger.



We suggest that the first time you import your UPE file into HCA 13 you choose the “Match by name” (which matches by room and device name as in the UPB devices and not the HCA folder and device name) and also have the “Remove unmatched devices from the design” enabled. After the import then any of the IOM devices created for the outputs that are not used will be removed. Any ICM devices probably will not be removed since at least one of the channels was in use – but you may wind up with a second device for a second channel.

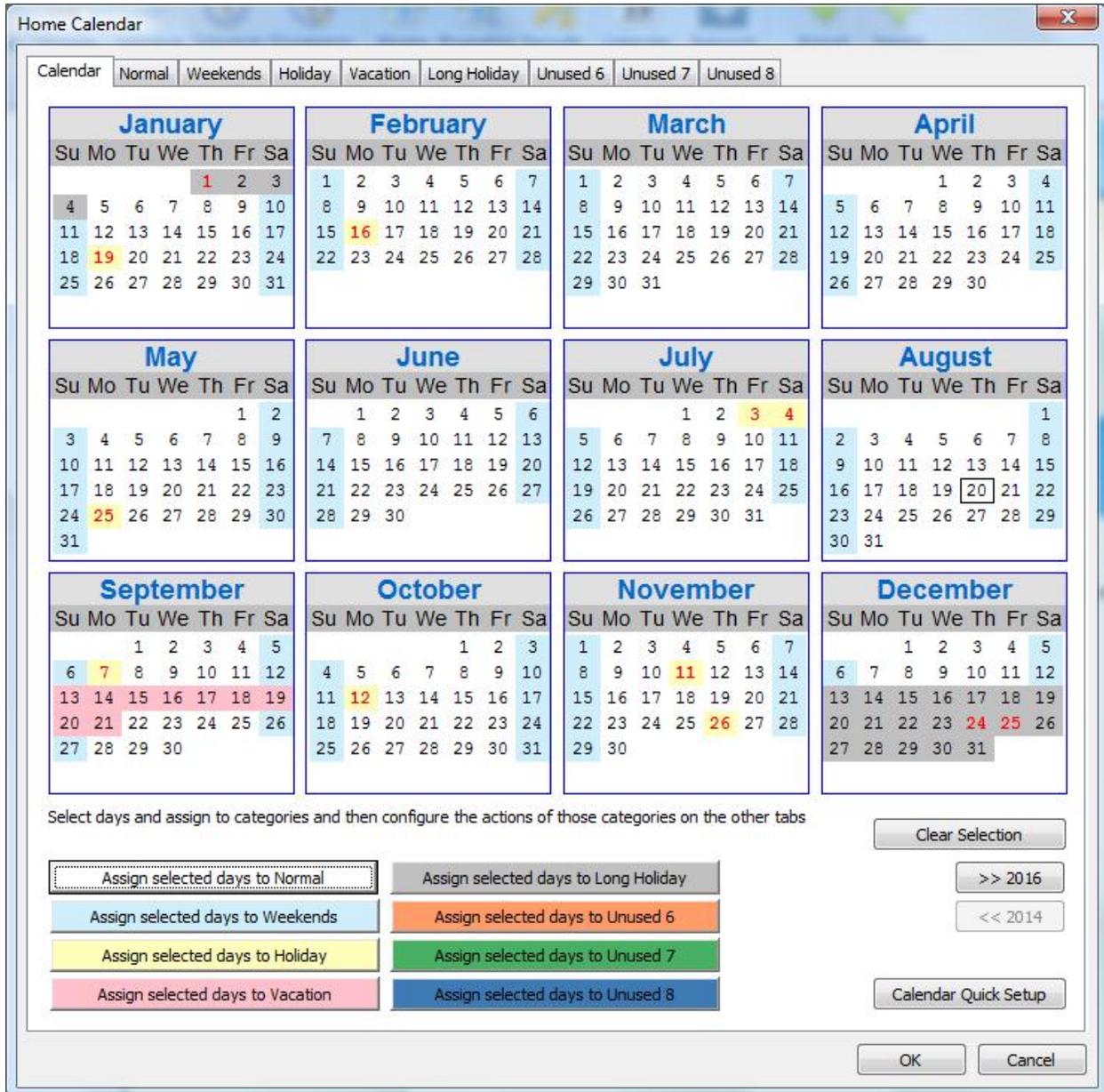
Obviously make a backup before you begin!

Why would you use this feature?

If you use UPB in your installation and have input or device with multiple channels this make it easier to work with those devices.

Date-Book out, Calendar In

The Date-Book has been removed from the HCA design and replaced by a different system using a 5-year calendar. Select form the ribbon Design category "Calendar".



The idea behind the calendar is simple: Each day is assigned to one of 8 possible categories. At midnight of each day, HCA looks at the calendar and determines what category the day falls into. Associated with each category is a name, a color, and actions that can take place when that category, based upon the calendar day, becomes active. These actions are:

1. Change to a different schedule

2. Execute a program when this category, based upon the calendar day, becomes the active category. That is, run the program upon entering the category
3. Execute a program when this category, based upon the calendar day, is about to no longer be the active category. That is, run a program just before leaving the category

In this example when the "Weekends" category becomes the active category the schedule is changed to "Weekends".

The screenshot shows the 'Home Calendar' configuration window. The 'Weekends' tab is selected in the top navigation bar. The 'Category Name' field contains 'Weekends'. Below this, there is a text box explaining that the calendar is checked at midnight each day. Three configuration options are listed:

- Select a schedule that becomes the current schedule on the days in category [Weekends]. The dropdown menu shows 'Special Weekends'.
- Select a program to run when changing from a different category to category [Weekends]. The dropdown menu shows 'Bedroom - Bath Motion Program'.
- Select a program to run when changing from category [Weekends] to a different category. The dropdown menu shows 'Bedroom - Bath Motion Program'.

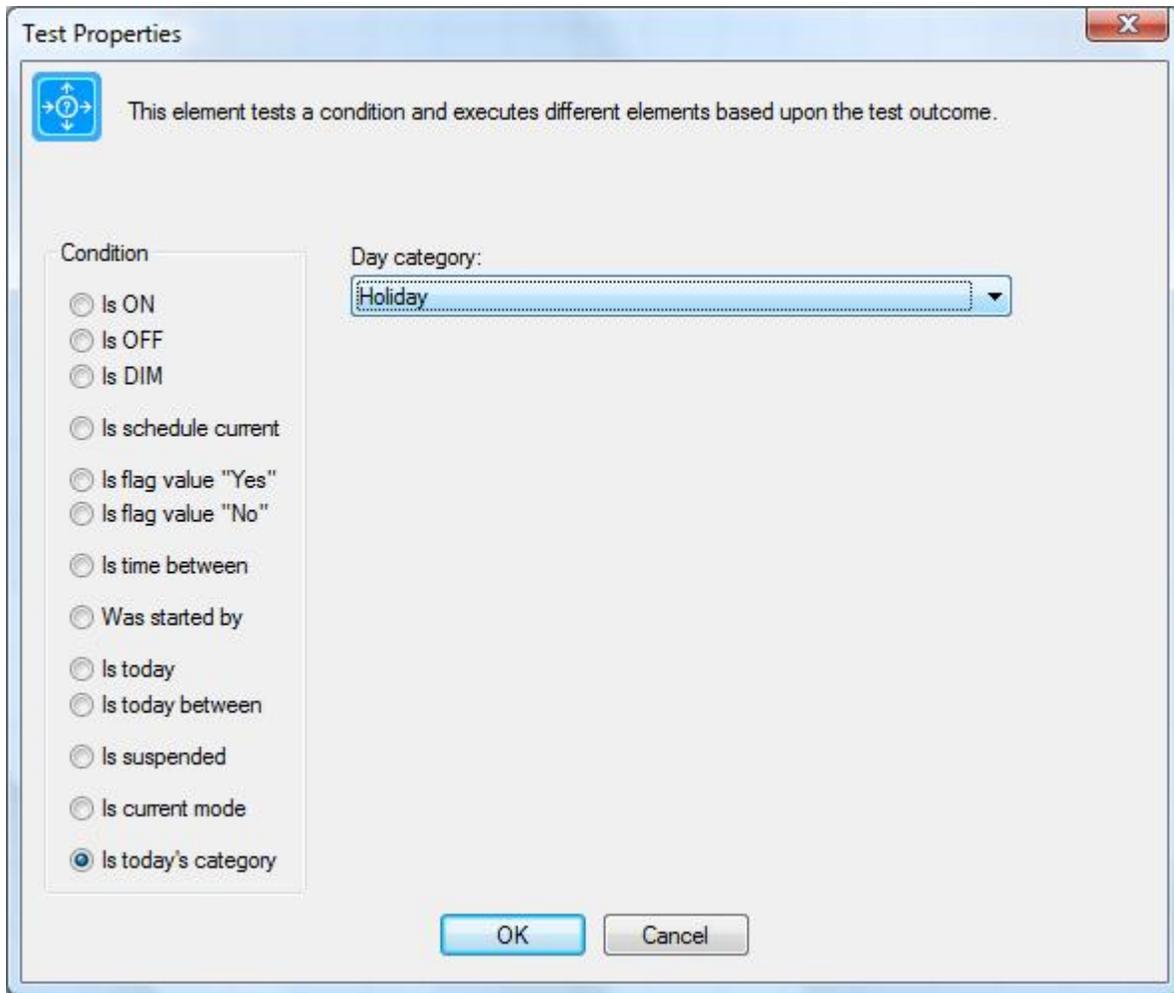
In this example, a program is run when the "Vacations" category becomes the active category and when the active category is about to no longer be "Vacations"

The screenshot shows the 'Home Calendar' configuration window. The 'Vacation' tab is selected in the top navigation bar. The 'Category Name' field contains 'Vacation'. Below this, there is a text box explaining that the calendar is checked at midnight each day. Three configuration options are listed:

- Select a schedule that becomes the current schedule on the days in category [Vacation]. The dropdown menu shows 'Special Weekends'.
- Select a program to run when changing from a different category to category [Vacation]. The dropdown menu shows 'Home Programs - Vacation Start'.
- Select a program to run when changing from category [Vacation] to a different category. The dropdown menu shows 'Home Programs - Vacation End'.

When the programs do is completely up to you.

In addition, a program can test to see what the current day category is.



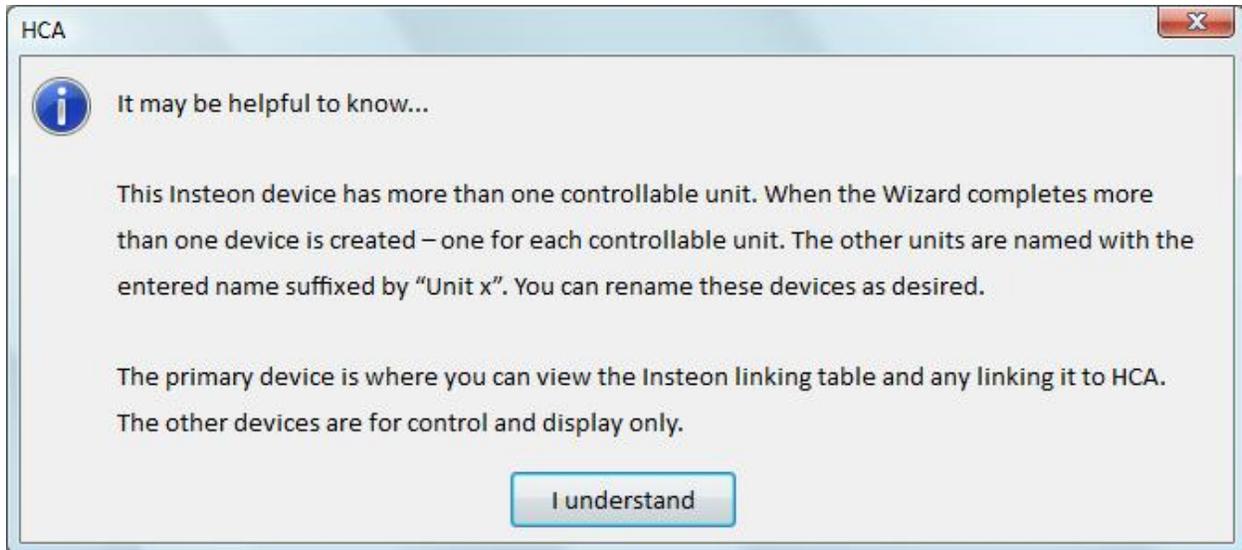
What use this: Since the calendar spans 5 years you can easily configure for pre-planned vacation or holiday periods.

Why would you use this feature?

The calendar makes it simpler to do different things on selects days of the year. Previously you had to create programs that ran everyday and then in the program test the day and see if that day was the day you cared about. This also makes it much simpler to switch from schedule to schedule based upon blocks of days like weekends, vacations, or holidays.

Support for Insteon On/Off outlet

The Insteon On/Off outlet has two controllable devices in the same physical device. When added to your design this popup appears:



As the dialog says two device objects are created. The one whose name is as you entered it is for the TOP outlet – let's call it the "primary device" and the second one for the BOTTOM outlet – let's call it the "secondary device". You can rename the secondary device if you want.

When you open the properties for the primary device the Insteon and linking tab are as usual. On the linking tab you can link the device – both top and bottom – to HCA. Once you do that then the buttons on the On/Off outlet will send messages to HCA that can be used as triggers.

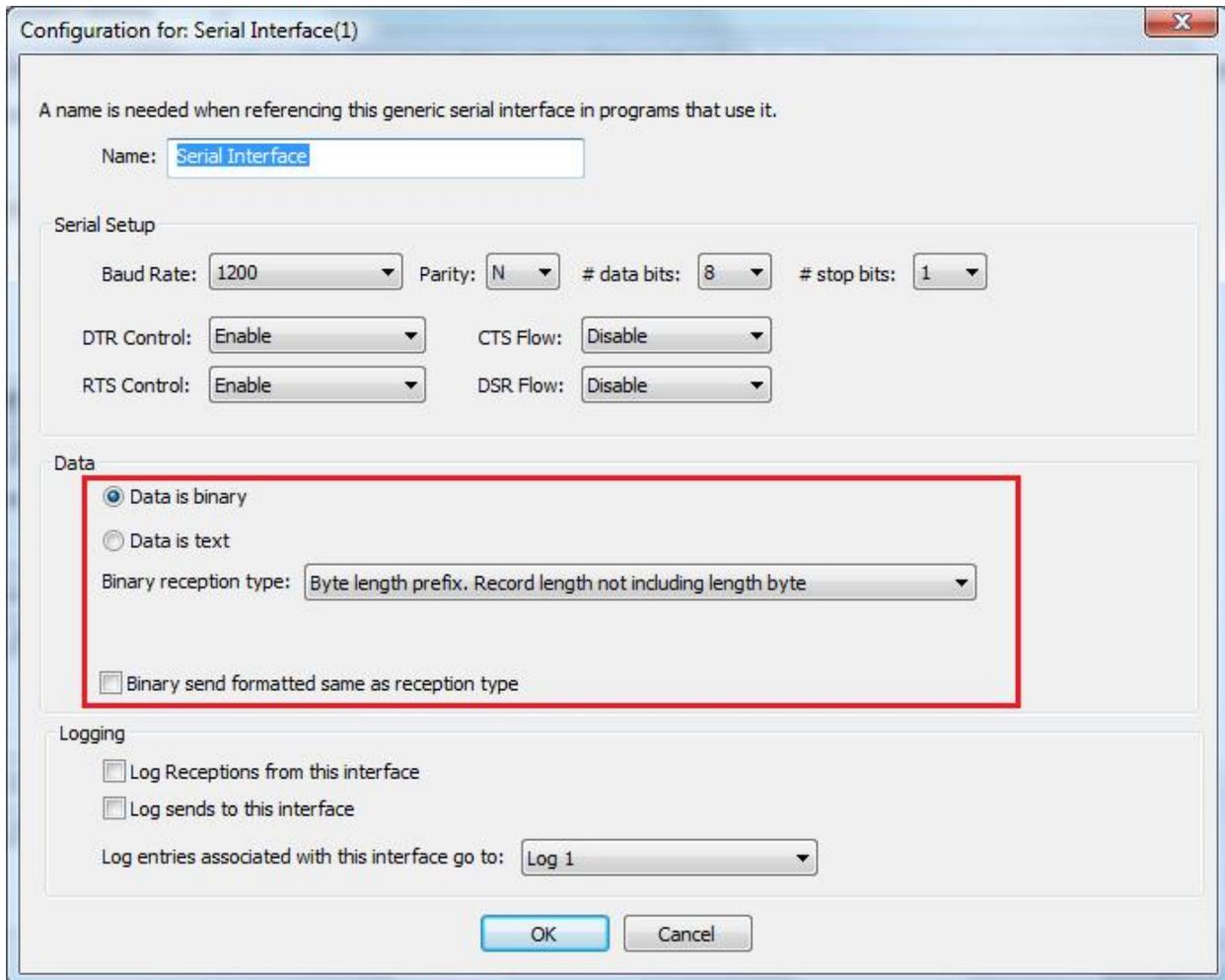
The properties for the secondary device doesn't have the Insteon and Linking tab as the primary device does. This is as expected. All linking and linking table reading are done from the primary device.

If you request the status of the ON/Off outlet, the status of both top and bottom are retrieved and the two devices are updated to show their state.

Other than that you can use the two devices as you normally would in schedules, programs, etc.

Generic Serial/Port I/O interface binary mode

The generic serial and IP interface now supports binary mode. The changes are in the interface configuration dialog.

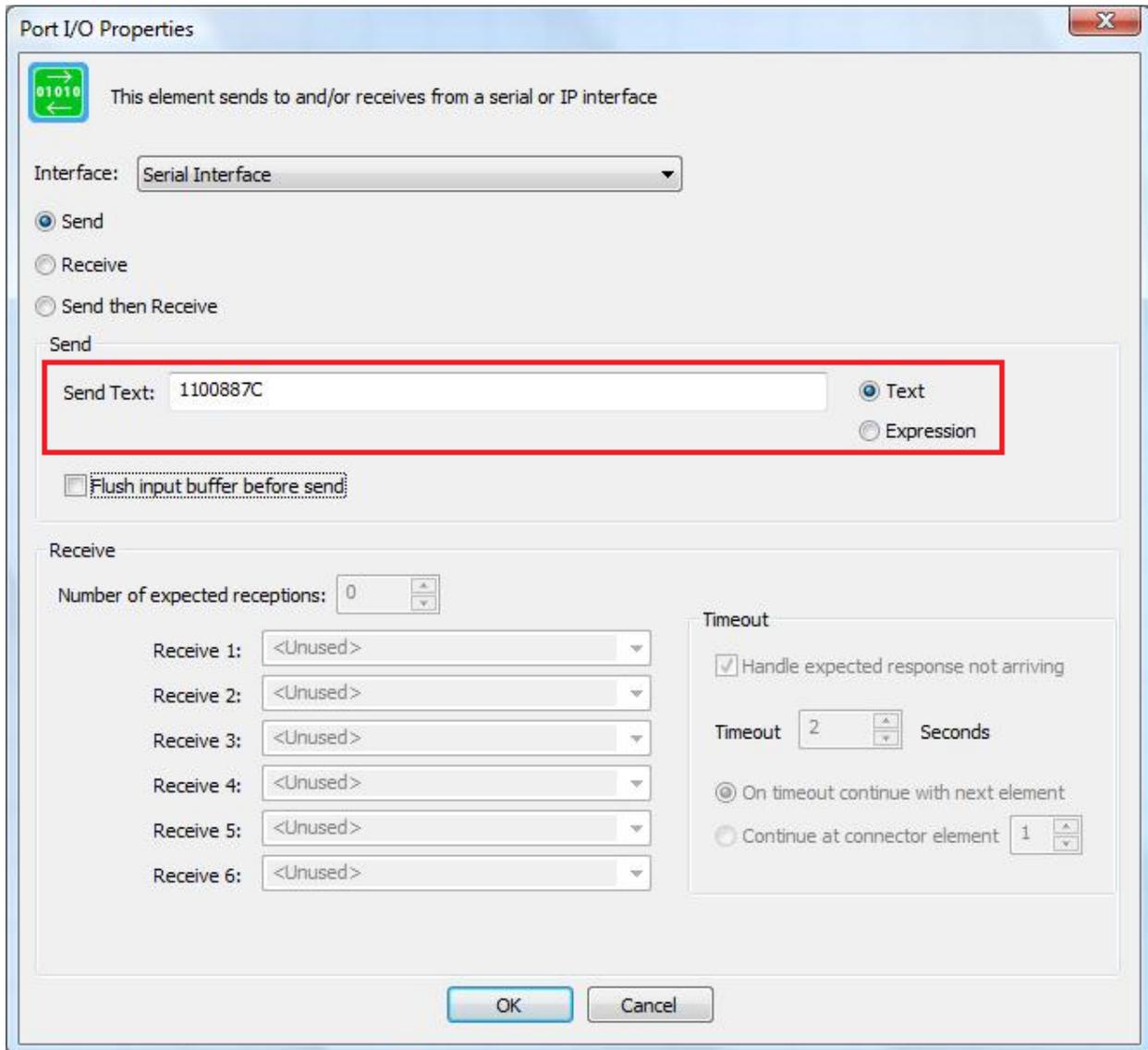


Unlike text mode where the receptions are split apart one from the other by the presence of delimiter characters in the data stream, in the case of binary mode there are several different formats of the data that determine a reception. These are:

- Byte length prefix. Each reception starts with a single byte that determines the number of bytes that follows. Once that many bytes have been received then that reception is determined to be complete. There are two variants of this: one where the length byte is included in the count of bytes and one where it isn't.
- Word length prefix. This is essentially the same as the byte length prefix except that instead of a byte there are two bytes – making a 16 bit quantity – that provides the length of the data to follow. There are 4 variants of this: If the count includes the 2 bytes for the length word, and variants for the order of the bytes – low-high or high-low.
- Fixed length: The number of bytes in each reception is given.

There is an additional configuration parameter related to binary transmissions from HCA: If the data should be formatted with the length byte or word in the same manner as chosen for receptions. If this option is not enabled then the bytes as given in the Port I/O program element are sent as is.

When using the Port I/O element, the data is given in hex characters. For example this element would cause 4 bytes to be sent out the port. The bytes sent are 0x11, 0x00, 0x88, 0x7C in that order.



The data provided in the "Send Text" must contain an even number of characters – to make up the correct number of bytes – and use only the characters 0-9, A-F, a-f.

Receptions are assigned to the receiving variables in hex characters. For example if these bytes made up a reception: 0x85, 0x76, 0x5D, then the variable would be assigned "85765D". The byte(s) that are used for the reception length - if that option is chosen - are not converted to hex text as part of the reception.

HCA Keypads

A new object has been added to HCA called a "Keypad".



The idea behind this new addition is that even with the client applications it is harder than it should be for the less technical users in a home to be able to control things. Currently, they have to start the app and then find the device or program – and understand the differences - they want and act upon it. That could be simpler.

The client applications work with these new keypads in the same manner as they already do with actual UPB and Insteon keypads. But there is no actual installed physical keypad in this case.

In the Control UI when you tap on the icon for a device that is a keypad, a popup window displays showing the buttons on the keypad. You press the buttons and the buttons do things.

Buttons appear in the keypad popup with a label and a background color. There is a color for "ON" and for "OFF". Buttons can either "toggle" or not. If the button toggles – it has two states - you can specify different actions – action "A" and action "B" - for if the button is ON when pressed and if it is OFF when pressed.

They key thing to know is that the color controls what action to execute when the button is pressed. If the button is configured as a toggle , action "A" is what happens when you press the button when it is showing the "Off" color and action "B" is what happens when you press the button when showing the "On" color.

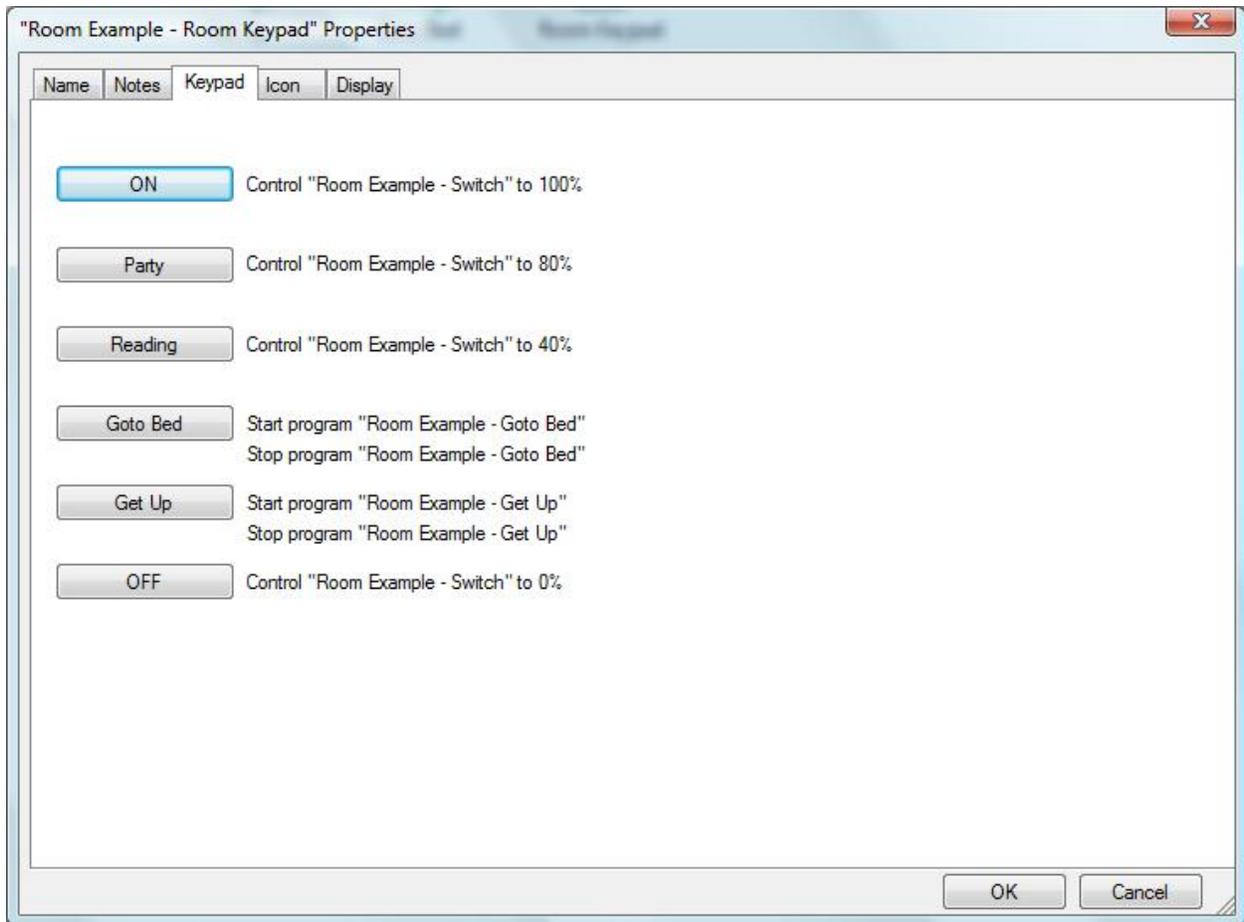
For a button that isn't configured as a toggle, the action that happens is always the same regardless of the color of the button.

What can a button do? Start a program, stop a program, activate or deactivate a scene, or control a device, group, or room. In addition to saying what action the button press does, you also configure the effect that press has on the "ON" or "OFF" appearance of the other buttons on the keypad.

You create a keypad using a "New Keypad" wizard. A keypad is a bit like a device: it has a name, lives in a room, has an icon, can appear on multiple displays. It has no triggers, can't be in a group, can't be scheduled, and has no power track or green settings.

The keypad wizard is like the program wizard. When you start the New Keypad Wizard (Button "Keypad" in the "New" panel of the "Design" category) you create the keypad object. Then open its properties and you can configure what the keypad does. The button actions are on the "Keypad" tab.

Here is the "Keypad" tab of the properties on an example keypad. The buttons show the label they will appear on the popup. The action they take when pressed is shown next to the button. If the button toggles then two actions show.



Note that keypads always have 6 buttons. In this example buttons 1, 2, 3, and 6 are not toggling buttons. Buttons that don't toggle perform the same action each time they are pressed. Buttons 4 and 5 are toggle buttons.

To configure a button press the button and a configuration dialog opens.

Button Configuration

Text: ON Toggle?

Action

Control device Dev: Room Example - Switch

To level: 100

Response

Button 1: Show On Button 3: Show Off Button 5: Show Off

Button 2: Show Off Button 4: Show Off Button 6: Show Off

Button appears when keypad displays

ON if level test passes Dev: Room Example - Switch

Level equal to 100

OK Cancel

This dialog is a little complicated but here are the parts:

The "text" is the label on the button. The "Toggle" checkbox says if the button toggles. For a non-toggle button there is a single action.

The possible "actions" are:

- Nothing
- Start Program
- Stop Program
- Control

- Activate scene
- Deactivate scene

The dropdown to the right of the action is loaded with different objects depending upon the action selection:

- Start or stop program: The dropdown is loaded with the programs in your design.
- Control: The dropdown contains all the devices, groups, and rooms in your design and the "Level" edit lets you enter the level to control the device/group to.
- Scene activate or deactivate: The dropdown shows all the UPB links and all Insteon scenes that you have created, named, and have HCA as the controller.

One important point when starting programs: When using the "Start program" option, the started program is passed 4 arguments that, if the program has the parameter option enabled, can be examined within the program. These parameters are:

1. Keypad Room Name
2. Keypad Device Name
3. Button label
4. Button state: "0" if the button is "Off" when pressed or "100" if the button is "On" when pressed.

In the "Response" section of the dialog you specify what happens after the button is pressed – how all the buttons on the keypad respond. The options are On, Off, and No-Change.

A key point to know: A button doesn't automatically go ON when pressed. You get to decide if it does and what the other buttons do by choices made in the "response" section of the dialog.

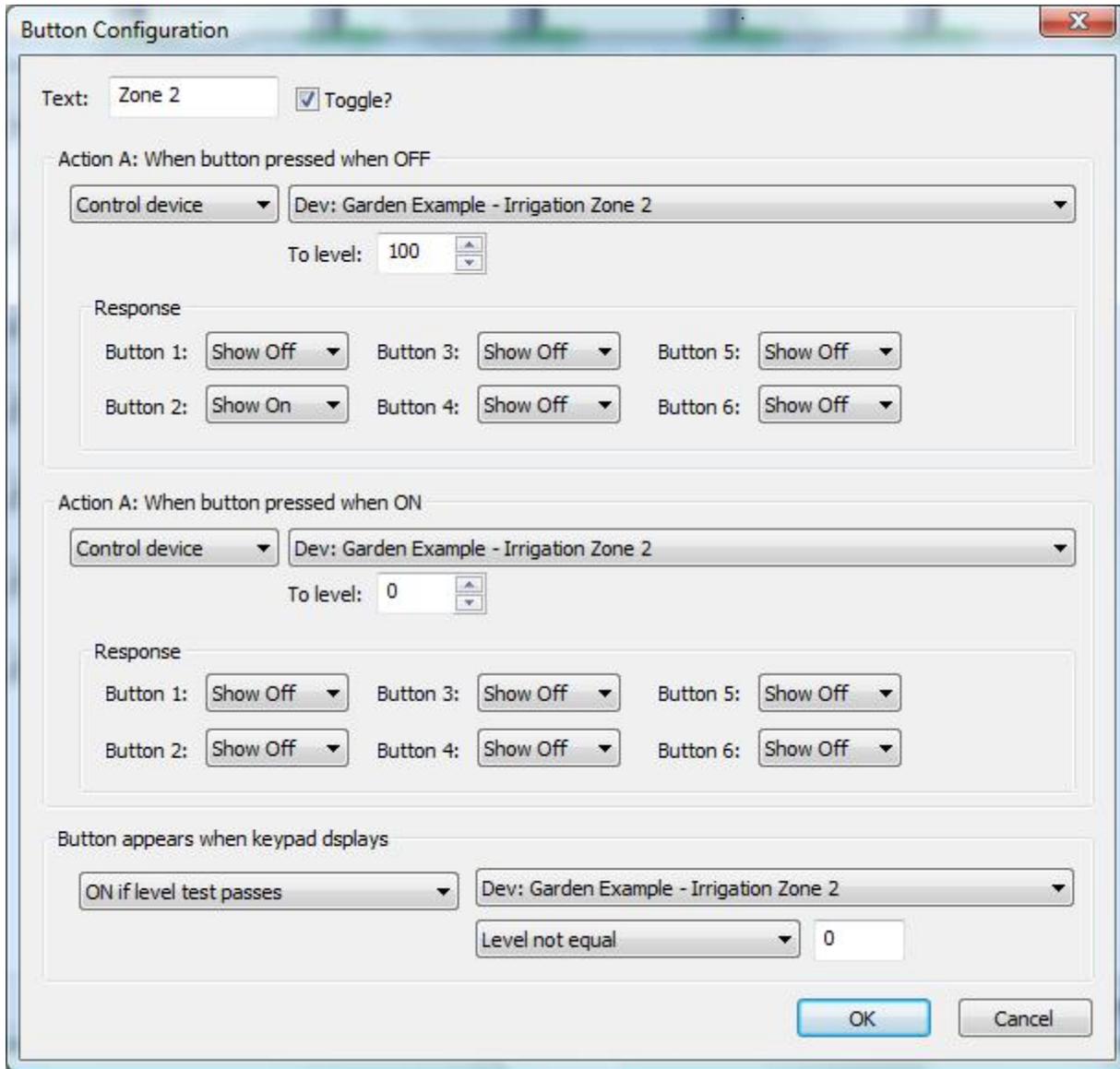
The last part of the dialog configures how the button appears when the keypad popup first comes to the screen. You can choose from these options:

- Always ON
- Always Off
- Same as when the keypad was closed
- ON if the selected program is running, OFF otherwise
- ON if the selected program is not running, OFF otherwise
- ON if the selected flag is YES, OFF otherwise
- ON if the selected flag is NO, OFF otherwise

An example might clarify this. Suppose that you have a keypad where the button starts a program. The program runs for some period of time and then terminates. You would configure the button to show as ON when pressed. That shows the user that they did indeed press the button. When next the keypad

popup appears it would be nice if the button showed if the program is currently running or not. That's what the configuration in the "Button appears" section is all about.

For a button that toggles the configuration is similar but there are now two actions.



In this example, the first press of the key turns the selected device ON – to 100% - and the second press turns it OFF – to 0%.

Note how the response section is configured. The button always shows if "Zone 2" is on or off. When pressed it turns off all the other buttons. In this example the device being controlled in the Insteon irrigation controller when only one zone can be on at a given time. The keypad buttons reflect that – press a button to control a zone the other zones now show as off.

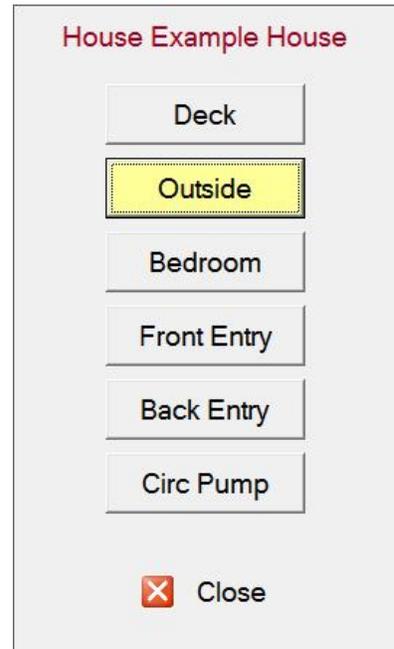
There are 4 examples in the keypads example file.

House controller

In this keypad each button acts independently of the other buttons. Each button controls a device. Each button toggles controlling the device ON and OFF. The state of the button shows if the device is ON or OFF.

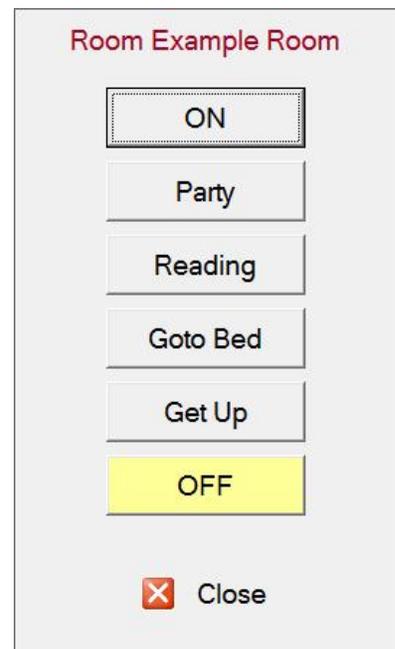
In this example some of the buttons control a single device and other buttons control a whole room using the HCA "room" operation.

The key feature of this use of a keypad is that more than one button can appear ON at any time.



Room or Scene controller

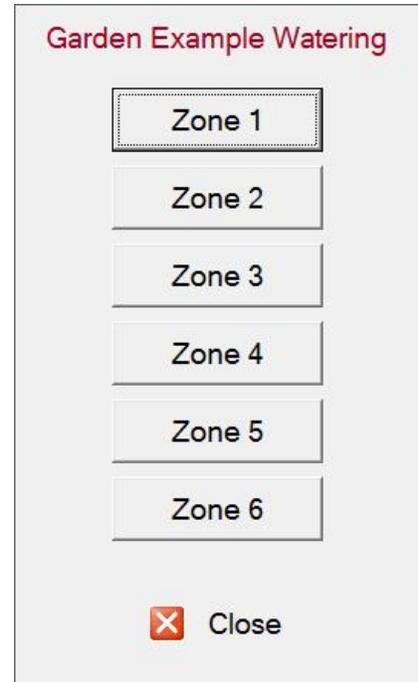
In this keypad each button controls the same set of devices. At any given time the room reflects one scene or another so only one button is ever on at a time. The keypad button that is ON shows what scene is the current scene.



Special Device Control Keypad

The keypad created for the Irrigation device is an example of this. Each button is tied to one part of the same device. In this case the keypad controls 6 zones of the irrigation controller.

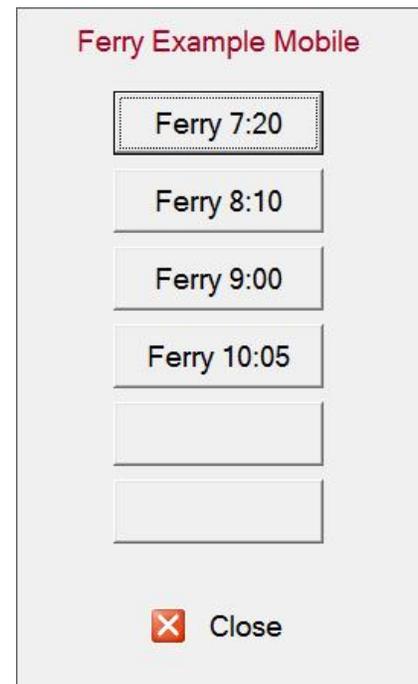
As explained above, since only one zone can be on at a time, only one button ever shows as ON.



Special Use Keypad

In this example a keypad was created for a family member to use when returning home. When they determine which ferry they will be on, they start the HCA application on their mobile, and press the button for that ferry they will take. Each button starts the same program – the program knows what button was pressed from the arguments passed to it – and the program determines when best to turn on lights in the home based upon the ferry time and the known travel time.

Without this keypad they had to navigate to the "room" that holds the 4 possible programs and tap to start the right one. Not difficult for some people but too difficult for them.



The whole idea of keypads is to let technical HCA users develop a user interface for non-technical users that gives them a limited set of actions that they can EASILY use.

Miscellaneous

The color used for the box around an icon to show suspend or green suspend has been moved to the registry so it can be adjusted for specific installations.

The UPB interface is no longer used in "Pulse Mode" and as such the UPB signal/noise meter has been removed.

Support for the older model Insteon hub is now on legacy status. It still can be used but only after selecting from HCA Options and on the "Legacy" tab ticking the "Support for older model Insteon hub" checkbox.

The editor for the compute and compute-test elements now handles "curly quotes". This is important only when you cut and paste text into the element from something like a word doc.

A problem when in client-server mode when selecting an IR keypad for an IR device was resolved

A potential problem of processing UPB commands sent by other software controllers was resolved. These commands are processed if they effect the state of devices.

The option to sort a schedule by time – right-click menu option in the design pane – now actually sorts.

Printing improved not that this feature is much used.

##end##