



**Home Control Assistant**

**Version 13**

**User Guide**

[WWW.HCATech.com](http://WWW.HCATech.com)

The information contained in this document is subject to change without notice. Advanced Quonset Technology, Inc. provides this information “as is” without warranty of any kind, either expressed or implied, but not limited to the implied warranty of merchantability and fitness for a particular purpose. Advanced Quonset Technology, Inc. may improve or change the product at any time without further notice; this document does not represent a commitment on the part of Advanced Quonset Technology, Inc. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the licensing agreement.

Windows is a registered trademark, and Windows NT is a trademark of Microsoft Corporation.

All other product names and services identified in this document are trademarks or registered trademarks of their respective companies and are used throughout this document in editorial fashion only and for the benefit of such companies. No such uses, or the use of any trade name, is intended to convey an endorsement or other affiliation with Advanced Quonset Technology, Inc.

© 2001-2016 Advanced Quonset Technology, Inc.

All rights reserved. Printed in the U.S.A.

March 1, 2016

Chapter 1 What is the Home Control Assistant? .....	1
About this guide.....	1
Quick tour of HCA .....	2
Opening and Closing HCA .....	2
Using the ribbon.....	3
The HINT System.....	4
Using the properties dialog boxes .....	4
Home Properties dialog box .....	5
HCA Options dialog box .....	6
The Design Pane and the Display Pane.....	7
The Design Pane.....	7
Design Pane.....	8
Two part names .....	9
Working within the design pane.....	9
Design pane information .....	10
Working with the design and display panes together .....	10
What doesn't appear in design pane.....	10
The conventions used in this guide.....	11
Procedures and operations common to HCA .....	11
Common Windows procedures used in HCA.....	12
Chapter 2 HCA Options .....	15
About Options.....	16
Startup .....	17
Client-Server .....	19
Hardware .....	19

Design Pane.....	20
Display Pane.....	21
Color and Theme.....	21
Visual Programmer .....	23
Web.....	24
Clock.....	24
Advanced .....	27
Legacy.....	27
Extra Features .....	28
Chapter 3 Your Home.....	29
Planning your home design.....	29
The HCA Designer .....	30
Adding Rooms .....	32
Adding an Insteon device.....	35
Adding an X10 device.....	36
Adding UPB devices .....	37
Configuring the Room Design .....	40
Configuring Home Mode Response .....	41
Configuring Auto Off .....	42
Designer Final Thoughts.....	43
Starting with a Blank or Empty file .....	43
Modifying your home file.....	44
Security settings.....	44
Observations about the security system .....	45
Light and Dark .....	46

Chapter 4 Devices .....	47
About devices.....	47
Planning your devices .....	47
Creating a new device.....	48
Modifying an existing device .....	51
Triggers Tab.....	52
Restart Tab.....	53
Display Tab.....	53
Groups Tab.....	55
Schedule Tab.....	56
References Tab.....	57
Green Tab.....	57
Protocol specific tabs.....	58
Deleting a device.....	58
Insteon Devices .....	60
UPB Devices .....	61
Final device topics.....	62
Chapter 5 Home Modes.....	63
Home Mode .....	64
Configuring devices for modes .....	64
Auto Off.....	65
Mode Change Triggers .....	65
Actions while in a mode.....	67
Putting it all together .....	68
Chapter 6 Rooms.....	69

How rooms are created .....	69
Rooms on and off.....	70
Working with rooms .....	71
Auto Off.....	71
Device Interaction with rooms.....	72
Device Auto Off.....	72
Chapter 7 Schedules and Schedule Entries.....	77
About schedules and schedule entries .....	78
Considering schedules and entries .....	79
Creating a new schedule.....	81
Creating a new schedule entry .....	83
Names for schedule entries .....	86
Modifying a schedule or entry .....	86
Modifying a Schedule Entry .....	89
Deleting a schedule or entry.....	89
Monitoring your schedules with HCA .....	89
Suspending device, programs, and groups from HCA control.....	90
The Occupancy Wizard .....	92
Chapter 8 Visual Scheduler .....	93
Starting the Visual Scheduler .....	94
Looking at the Visual Scheduler .....	94
Using the Visual Scheduler.....	95
Add time bars.....	95
Creating new schedule entries.....	95
Modifying existing schedule entries .....	97

Deleting existing schedule entries .....	98
Dim and scene schedule entries .....	98
IR schedule entries .....	99
Thermostat schedule entries .....	99
Configuring the Visual Scheduler .....	100
Other Views of your schedule .....	101
The Visual Scheduler and the Schedule Entry Wizard .....	102
Chapter 9 Groups .....	103
About Groups .....	103
A Group example .....	104
Planning your groups .....	104
Creating a group .....	104
Modifying an existing group .....	107
“Group” tab in device properties .....	109
Adding to a group using drag and drop .....	109
Deleting a group .....	109
Groups and Local Stored Scenes .....	110
Chapter 10 Programs and the Visual Programmer .....	111
Terminology .....	112
Using the New Program Wizard .....	112
Opening the Visual Programmer .....	113
Areas of the Visual Programmer .....	114
Toolbox .....	114
Element Palette .....	115
Validate button .....	117

Working with the programming canvas.....	117
About connecting elements.....	120
Snippet Wizard.....	121
Program Triggers.....	121
X10 Reception Trigger.....	124
Insteon Reception Trigger.....	126
UPB Action Trigger .....	127
UPB Powerline Message Trigger .....	128
Wireless Component Message Trigger .....	129
Global Cache Triggers .....	130
Weather Condition Triggers.....	130
Flag Value Change Triggers .....	131
Home Mode Triggers .....	131
Port Reception Trigger .....	133
ZWave Reception Trigger.....	134
Expression Triggers .....	135
Special Condition Trigger .....	136
Generic Trigger.....	137
Trigger Evaluation .....	138
Parameterized programs .....	139
Parameterized Triggers.....	144
Setting properties for program elements.....	147
Elements .....	147
Start Here.....	147
Add to log.....	147



Auto Off.....	148
Change icon.....	149
Change schedule.....	150
Compute and Compute Test.....	150
Day Night.....	151
Daily Message.....	151
Dim.....	152
Delay.....	153
Email / SMS.....	153
Exit.....	154
Get Status.....	155
Insteon.....	155
Make flag No/Make flag Yes/Not flag.....	156
Multi.....	156
On.....	158
Off.....	158
PING.....	158
Play sound.....	159
Port IO.....	159
Read Data.....	160
Repeat.....	161
Request Input.....	161
Resume and Suspend.....	163
Run.....	164
Script.....	164

Scene.....	166
Set Mode.....	166
Show Display.....	166
Show Message.....	167
Speak.....	167
Start Program.....	168
Status Export.....	169
Stop Program.....	170
Test.....	170
Update Tile.....	174
Wait Until.....	175
Specific Hardware Elements.....	176
IR.....	176
X10 All lights on / X10 All lights off / X10 All units off.....	177
Thermostat and Thermostat Test.....	177
UPB Blink.....	177
UPB Link.....	178
Weather Test.....	178
Send X10.....	178
Send ZWave.....	178
Legacy Elements.....	179
Constructing programs.....	180
Test.....	180
Repeat.....	181
The Validate button.....	182

Troubleshooting: Getting programs to do what you want.....	182
Program properties Advanced tab.....	183
Chapter 11 Visual Program Debugger.....	185
Using the Debugger: The basics.....	185
The debugger window .....	186
Debugger Buttons .....	186
Debugger Tabs .....	187
Breakpoints .....	188
Changing the current element.....	190
Fast and Slow elements .....	190
Setting the starting trigger.....	191
Modifying programs.....	191
Just in time debugging .....	191
Debugging Limitation .....	192
Chapter 12 Expressions.....	195
Introduction to expressions .....	195
Compute and Compute test visual programmer elements .....	196
Expression builder.....	197
Managing Flags .....	198
Other uses for expressions .....	198
Error Handling.....	199
Expression Syntax.....	199
String functions .....	200
General use functions .....	201
Time and date functions .....	202

Numeric formatting functions .....	205
Device / Program / Group / Schedule functions.....	206
Thermostat.....	211
Weather .....	213
File Operations.....	215
Miscellaneous .....	216
Chapter 13 Scripts.....	219
Script setup .....	219
The script element .....	220
Using scripts .....	221
Helpful Tool.....	222
Chapter 14 Displays .....	225
Types of Displays.....	226
Creating a Display .....	227
Text Displays .....	229
HTML Displays.....	230
Icon Displays .....	231
Power Track Displays .....	232
Tiled Displays .....	233
Tile Types .....	234
Alerts Tile .....	235
Blank Tile.....	235
Status Tile.....	235
Graph Tile.....	236
HTML Tile .....	237

Icon Tile.....	237
Image Tile.....	238
Power Meter Tile .....	239
Text Tile.....	239
Creating a Tiled display.....	240
Tile coordinates.....	241
Adding a tile .....	243
Modifying a tile .....	243
Repositioning a tile .....	244
Deleting a tile .....	244
Working with tiles in the Development UI.....	244
Updating Tiles .....	244
DXF files as backgrounds.....	245
Linking or importing.....	245
DXF files and layers .....	245
Picture files as backgrounds .....	246
Modifying a Display.....	246
Placing icons on the display .....	248
Tab Bar .....	249
Displays with Icons.....	249
Multi-Select Operations.....	252
HTML Folder Sync .....	252
Chapter 15 Insteon Visual Scene Editor (VSE) .....	255
Using the VSE: The basics.....	255
Adding responders to a scene.....	256

Modifying a device’s scene properties .....	257
Removing responders from a scene .....	257
Changing the scene name .....	258
Testing a scene.....	258
Programming a scene .....	258
Closing the VSE.....	258
Using the VSE: Starting, stopping, and creating scenes.....	259
Working in a “by scene” manner .....	259
Working in a “by device” manner .....	260
Deleting scenes .....	262
Scenes initiated by HCA and by other devices.....	262
VSE, the linking tabs, and the multi-way wizard.....	263
Using scenes.....	263
Chapter 16 Power Track.....	265
Enabling Power Tracking.....	265
Insteon Devices .....	266
UPB Devices .....	267
Current Power Graph.....	268
Configuring the current power graph .....	269
Historical Power Graphs .....	270
Focus the graph on a single device or room .....	272
Show a breakdown for a single bar.....	272
Displays and Graphs.....	273
Insteon Power Meter .....	273
Chapter 17 HCA Keypads .....	275

The Keypad Idea.....	275
Adding and configuring a keypad.....	275
Examples .....	280
Chapter 18 Network Devices .....	283
Configuration .....	283
Rules.....	284
Override .....	285
Logging .....	286
Chapter 19 Troubleshooter.....	287
Overview of the Troubleshooter.....	287
The Inspector .....	287
Inspector Configuration .....	288
Inspector Checklist.....	289
Alert Manager .....	291
Alert manager summary tab .....	291
Alert manager details tab .....	292
Configuring alerts.....	292
Device Overdue alerts.....	294
Per device alerts.....	295
Alert colors.....	296
Alert manager final issues.....	296
The Log.....	297
Log Setup.....	298
Log Viewer.....	300
Using the Log Viewer .....	303

Log Filters .....	303
Show Log .....	306
Log Focus.....	306
Chapter 20 Restart .....	309
Types of power supply .....	309
No uninterruptible power supply .....	310
Using an uninterruptible power supply .....	310
HCA really doesn't know that the power has failed .....	310
Controlling restart.....	311
Restart from a normal shutdown.....	312
Some final information about power failure and recovery .....	312
Chapter 21 Calendar .....	315
Calendar display.....	316
Category Setup.....	317
Pairing one or more days with a category .....	318
Programs and the Calendar .....	319
Chapter 22 Icon Themes .....	321
What are icon themes.....	321
Choosing a theme for a display and a default theme for your design.....	322
Adding icons to an existing theme.....	323
Modifying theme icons .....	323
Creating a new theme.....	323
Important facts about creating icons.....	323
Chapter 23 Control User Interface.....	325
What does the Control UI look like? .....	325



Configuring the Control Interface pages.....	327
HCA created pages.....	327
Custom pages.....	328
Controlling devices.....	330
Glass Keypads.....	332
HCA Options Control Interface Tab.....	334
Control Panel.....	335
Configuring the Control Interface for dedicated machines.....	336
Using HTML Displays with the Control Interface.....	339
Chapter 24 Design Tools.....	341
Program Flag Inventory.....	341
HCA Status dialog.....	342
Communications Monitor.....	344
Interface Viewer.....	344
File Inventory.....	345
Using the X10 House and Unit Code Inventory.....	346
Viewing settings.....	348
Changing settings.....	348
X10 Reception.....	349
X10 Test.....	351
Chapter 25 Status Export.....	353
Status Export Overview.....	353
Configuring Status Export.....	353
Using Status Export.....	354
Chapter 26 Design Import and Export.....	355

Design Import and Export Overview.....	355
Export Wizard.....	356
Importing .....	358
Chapter 27 HCA Client-Server.....	361
First some words about security.....	362
Using the HCA Server .....	362
Connecting a Client.....	364
What can you do with HCA Client – Server.....	366
Automatically Starting as a Client.....	367
Remote Access to the HCA Server .....	367
Other Clients .....	367
Chapter 28 HCA Clients .....	369
HCA Client for Windows – Quick Tour .....	369
Client Control Panel .....	370
Configuring the client.....	371
Connection tab.....	371
User Interface tab .....	372
Status Bar tab.....	373
HTML Folder Sync .....	374
HCA for Android – Quick Tour.....	374
Android Settings.....	376
Android Server Settings .....	376
Connecting and Disconnecting.....	377
HCA for iOS.....	378
Configuring the iOS application. ....	380

Primary and Secondary IP Addresses.....	381
Chapter 29 Web Component.....	383
Accessing your design from a web browser .....	384
Accessing your design from a mobile browser .....	385
Configuring the HCA Web Server.....	387
Security Considerations .....	388
HCA Design Passwords.....	388
Hardware Security.....	388
Web Server – More information.....	388
Chapter 30 Getting Help .....	389
Our web site.....	389
Other considerations .....	389
Technical support.....	389
Glossary.....	391



# Chapter 1

## What is the Home Control Assistant?

In today's complex world, busy people can benefit from a home environment that anticipates their needs and helps take care of itself. For instance, suppose your house could:

- Turn on outside and entryway lights at dusk every day, or before you get home from work on weekdays?
- Automatically run the hot tub filter during the day, or start the heater, filter, and turn on the lights all at the flick of one switch?
- Dim hall lights, turn off a ceiling light, and turn on the children's night light at a preset time every evening?
- Turn on kitchen lights, and start the coffee pot at the same time every weekday morning?
- Turn all the lights in the house on or off when you press one switch at your bedside?

The Home Control Assistant can help you program your house to automatically respond to your typical schedule on weekdays, or a weekend or vacation schedule. You can set up schedules to make your home looked lived in when you are away, and save energy by automatically turning lights off after you're in bed.

This chapter introduces the Home Control Assistant, and discusses the conventions of the guide and common procedures of HCA. This chapter includes sections on:

- About this guide
- Quick tour of HCA
- Using the menus
- Using the status bar
- Using the properties dialog boxes
  - Home Properties dialog box
  - HCA Options dialog box
- The conventions used in this guide
  - Procedures and operations common to HCA
  - Common Windows procedures used in HCA

---

### About this guide

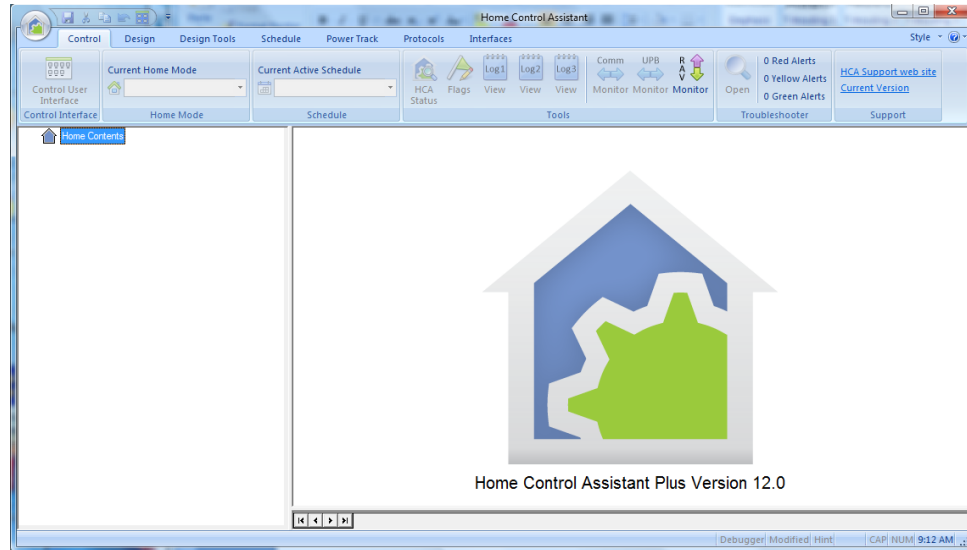
This user guide walks you through the process for setting up a typical device, and shows you how to set up more complex schedules and programs to help automate your home. This first chapter includes an introduction to and quick tour of the Home Control Assistant (HCA), showing what HCA looks like and discussing the use of ribbon, status bar, and the Properties dialog boxes. This is followed by a section on the conventions and typical Windows procedures used in this guide.

This guide is set up in chapters that discuss the major procedures that you can use in working with HCA. In addition to the main chapters, there are appendixes that describe specific features of HCA or the automation hardware supported.

## Quick tour of HCA

To start the Home Control Assistant once you have installed it:

1. From the Windows task bar, click the Start button.
2. Choose Programs.
3. Then click Home Control Assistant.



The Home Control Assistant window is split into two panes.

- The left pane, the design pane, shows the HCA design outline with items in the design organized into folders. Everything in the left pane has properties: simply right-click the element you want to access and choose Properties from the popup menu.
- The right pane, the display pane, is reserved for the displays in your design, and at different times also shows the Visual Scheduler, Visual Link Editor, and messages.

As with a typical Windows environment, you can re-size the panes by clicking the divider between the two panes, and moving it either left or right to achieve the relative sizes that you want. You can even close one side entirely to show just the other side.

Across the top of the Home Control Assistant you see the Ribbon which contains a number of categories – tabs – that control buttons to activate various features of HCA. Across the bottom of the display pane is a tab bar that shows a tab for displays in your design. Across the bottom of the whole window is a status bar.

## Opening and Closing HCA

When you first start HCA, its main window appears on the screen. Also in the system tray on the system task bar, a small HCA icon appears:



Unlike other Windows programs you may be familiar with (word processors, spreadsheets, email programs), HCA must be running 24/7 on your computer in order to control your home. To make it easier to keep HCA running all the time, and to help prevent you from inadvertently terminating HCA, HCA works a bit differently than these other programs.

There are five operations that are important to understand:

### Maximize the HCA Window

The HCA Window is maximized in the same manner as most Windows programs. Using the left mouse button, click on the maximize button on the upper right hand corner of the HCA window. The HCA window expands to fill the whole display.

### Minimize the HCA Window

The HCA Window is minimized in the same manner as most Windows programs. Using the left mouse button, click on the minimize button on the upper right hand corner of the HCA window. HCA is removed from the display and appears in the Windows Task Bar as a button labeled with your home design name.

### Hide the HCA Window

To hide HCA and have only the small icon in the system notification area, select *Exit* from the *File* menu. Unlike other programs this **does not** terminate HCA. This operation can also be done by clicking with the left mouse button on the close icon on the upper right hand corner of the HCA window.

### Open the HCA Window

To open the HCA window when HCA only appears as an icon in the system notification area, move the mouse over the icon and click the right mouse button. Select *Open* from the popup menu.

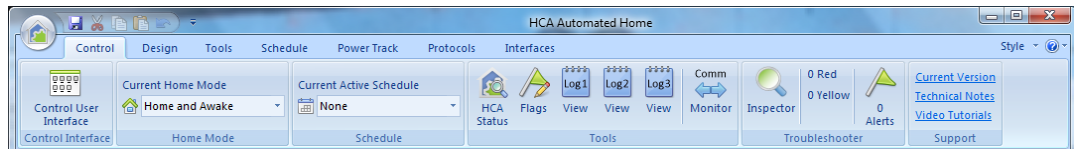
### Terminate HCA

Open the application menu by clicking on the application “Bubble” and then select *Shutdown* from menu

---

## Using the ribbon

Across the top of the Home Control Assistant window is a ribbon menu. There are seven categories – tabs - with various controls in each.



The **Control** category contains controls that show the current state of your design. For example, the current schedule and home mode and shown in dropdown controls. To change the mode of schedule all you need do is to change the selection in those dropdowns.

Also shown is the current inspector and alert report given in shorthand. The number of red and yellow problems identified and the count of alerts reported by the alert manager.

The buttons in the Tools panel open viewers for the Logs, flag inventory, and the communications monitor.

- **HCA Status.** Opens a dialog box that gives the current design path and name, current schedule, next scheduled event; a list of the programs running, and sunrise and sunset times.
- **Flags.** The Flag Inventory dialog box shows all the flags and their current values in your design. A Program Flag is used to represent various types of information including status, values, times, etc. which are used in HCA programs
- **Log1m, Log2, Log3.** Show a viewer for the each configured log. If the log is not configured the button is disabled.
- **Comm Monitor.** A window that shows the lowest level communication between HCA and the attached automation interfaces

The **Design** category contains controls for modifying your design. For example, adding new programs, devices, groups, and schedules. Also tools for accessing the properties of your design.

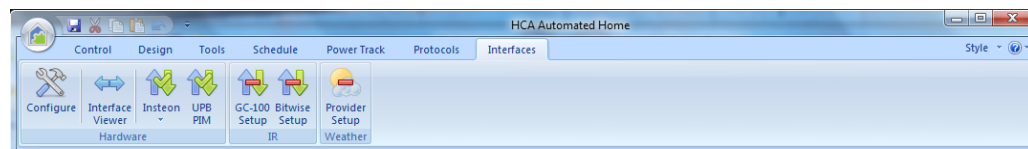
The **Tools** category contains tools for configuring the Control UI, Status export, etc. Also in this category are the tools for configuring various aspects of HCA like the Visual Programmer and debugger.

The **Schedule** category contains the controls necessary to work with the Visual Scheduler. This is described in the Visual Schedule chapter.

The **Power Track** category is to view both current power uses and to view historical power graphs that show power usage in your home over time.

The **Protocols** category contains tools customized for specific attributes of the various automation protocols supported by HCA. For example, special facilities for working with Insteon and UPB.

The **Interfaces** category contains tools for working with all supported automation interfaces and for weather providers.



One nice feature of the Interfaces category is that the icons show you which interfaces have been configured, which are connected and working, and which are not.

In the above picture the Insteon PowerLinc and the UPB interface has been configured and are working – green check. If they were not working the green check would have been replaced by a exclamation point. The IR interfaces have not been configured - red bars.

---

## The HINT System

HCA contains a system that may make learning all the features of HCA a bit simpler. When you are working with the HCA User Interface if your actions could be done in a better way, or there is a feature you may not be aware of, HCA displays a hint. This can be done in one of two ways.

First, the hint can just appear on the screen in a popup window. Or, the Hint Indicator in the status bar can change to show that HCA has a hint. In this case you must either click on it or press the F1 key to show the hint.

You can resize and move the hint window out of the way and leave it on the screen as you work. HCA updates it with whatever hint it has about what you are doing.

In the Help menu is an option to select your preference. If Auto Hint Display has a checkmark next to it, hints automatically appear. Without the checkbox the start bar method is used.

On each hint popup dialog is a checkbox that you can use to have HCA no longer show that hint. As you get more sophisticated you will have no need of some of the hints.

If you find yourself wanting to get the hints back that you turned off, select Enable All Hints from the Help menu.

---

## Using the properties dialog boxes

In HCA, nearly everything has properties. Most properties dialog boxes are available from a popup menu that you see when you right click an item in either pane of HCA.

- In the design pane of the HCA, right click an element word, and choose Properties from the popup menu. The properties dialog box for the item appears.



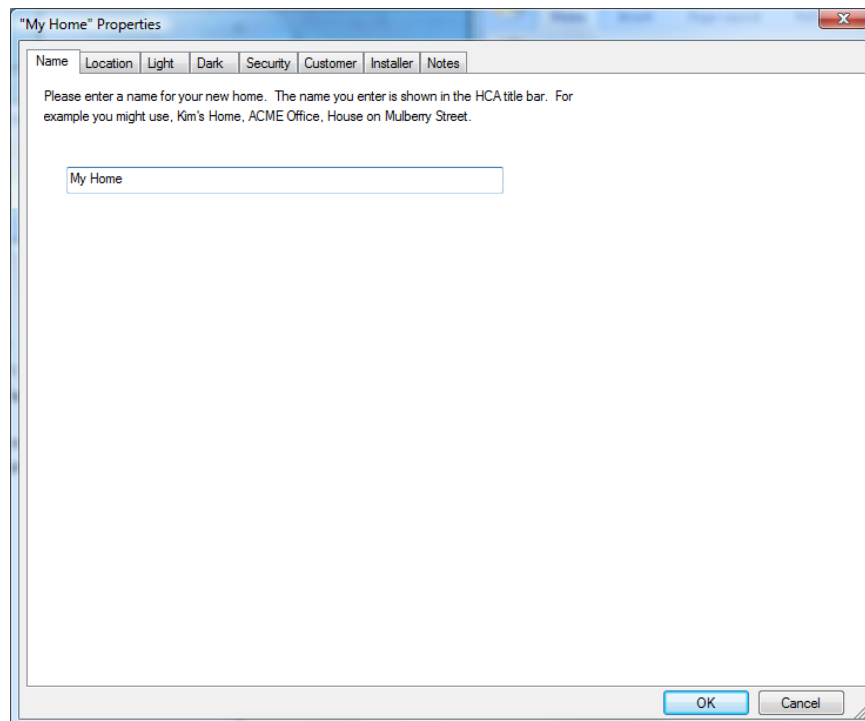
- In the display pane, right-click on the display background, or on an icon, or the visual scheduler background to display the popup menu with a properties dialog box for that item.

These properties dialog boxes generally have multiple tabs, and include items like the name, icon, restart choices, green settings, etc. There are usually other tabs with options specific to the object or the type of object.

---

## Home Properties dialog box

You can access your Home Properties dialog box by either right clicking Home Contents in the design pane, or by clicking the *Properties* button in the *Design* category.



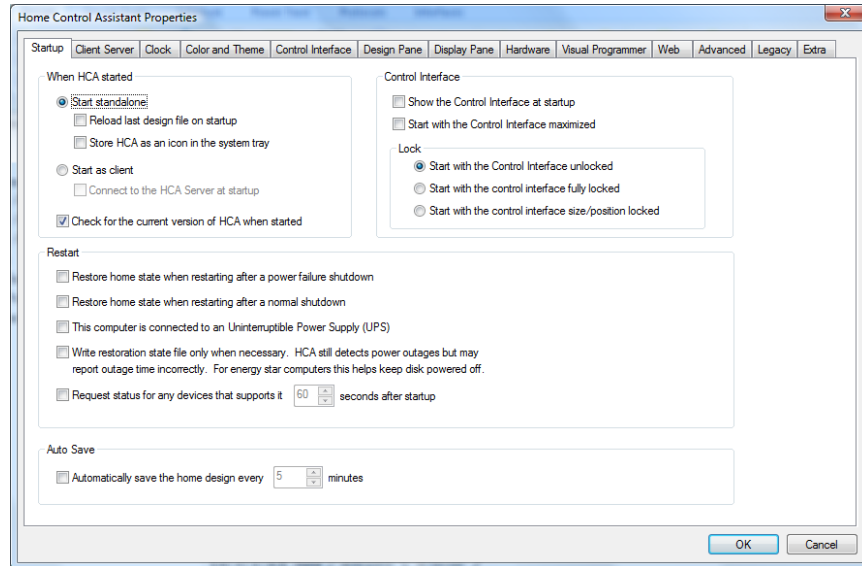
The home Properties dialog box shows the information specific to your home design that you entered when you first set up your HCA design. It provides tabs where you can modify or add settings later for these properties, including:

- Name of the design
- Location—city, state, longitude and latitude
- Light and Dark settings. How HCA determines if it is light or dark. There are many ways to do this.
- Security settings—passwords to modify the design, control programs and devices, and activate or deactivate HCA
- Identification info – name, address, email, etc – for both the home owner and installer
- Any notes about the design.

See the chapter on Home Properties for more information.

## HCA Options dialog box

You can access the Home Control Assistant Options dialog box by using the HCA Options button from the application menu.



The Home Control Assistant Properties dialog box is a very powerful tool where you can modify or add settings for the general properties for the HCA program. This is information about how to run HCA, including:

- Startup—design, size, power failure information, and auto save options
- Client Server – Opens for the HCA Server
- Clock—PC clock correction information, modem info if used, and Time Service options
- Color and Theme—select the default color and theme used for display background in the development and control interface.
- Control Interface – Options for the Control Interface
- Design Pane – design pane and time format options.
- Display Pane —what you'll see in the display pane of HCA
- Hardware—Automation interfaces and how they are attached to your computer.
- Visual Programmer – configure Visual Programmer options
- Web – Configuration for the HCA Web Server
- Advanced – other options
- Legacy – Selection for older legacy hardware and features.
- Extra – Extra feature codes that enable/ disable some special HCA features

Refer to the HCA Options chapter for more detailed information.

---

## The Design Pane and the Display Pane

As described above, the HCA main window is split into two parts. The display pane is described in the chapter on displays. The design pane is described here in the next section.

---

### The Design Pane

The left pane of the HCA Window, called the Design Pane contains a list of the elements of your design. There are a few items not shown and these will be described later.

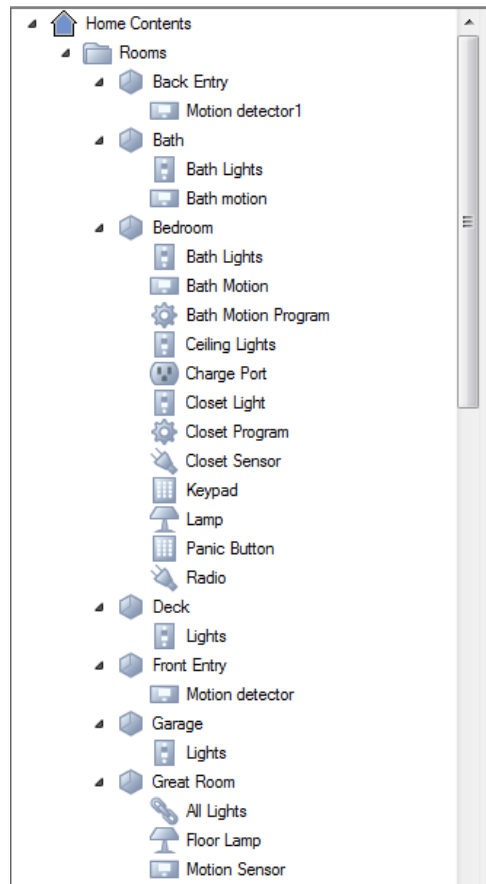
Every object in your design can have a name and most objects require a name. These names appear in the design pane. Each object is also contained within a room or folder. You can have as many objects as you need and place them in as many rooms or folders as you need. The only limitation is that each item within the room or folder must have a unique name, and no room or folder can contain another room or folder.

What's the difference between a room and a folder? While rooms are a good way to organize your device, programs, and groups, there are special operations that allow you to treat all the devices in a "room" as a single set of devices. This is all explained in the chapter on rooms. Folders are much simpler and are just an organizational method.

**Hint:** Rather than say "folder or room" everywhere, the term *folder* is used and means either folder or room unless otherwise stated.

Why would you want to organize your design info folders? One reason may be to help group those things together that work in unison. For example one room may have three devices, a group, and 6 programs. All of these work together as part of your automation solution so it makes sense to have them in one folder. And perhaps the folder name is the name of the room.

How do you organize things that don't fit into a simple room-by-room organization? That's up to you! One way would be to have a folder called Home that contains all those items that work with the house as a whole. Or have a folder for each floor that, in a similar way, contains things that handle that floor.



In this example, the design contains a number of Rooms. One room is for the Back Entry, another for the Bath, and so on.

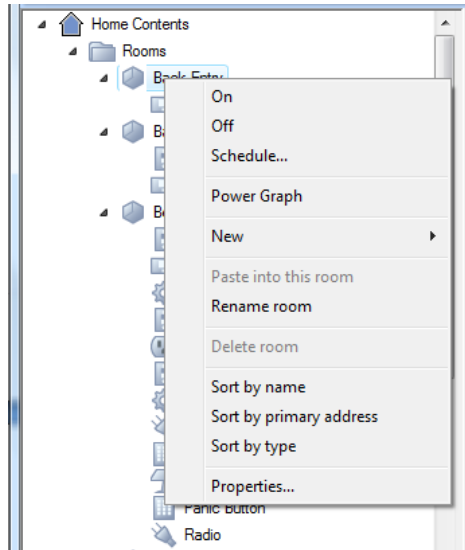
When you open the properties for each object type, on the first tab is the place where you can select which folder or room the object is placed in.

---

## Design Pane

There are a number of options for sorting the individual folder/room contents and the sorting for the order of the folders displayed.

Right click on the folder/room name and the sort options are in the popup menu.




---

## Two part names

In HCA everything has a name. It's always hard to come up with names for things in your design. It may seem that each room of your home has computer controllable lamps and ceiling lights. You could call them "Den Lamp" "Den Ceiling Lights", "Bedroom lamp", or come up with some other naming convention. But with the ability to assign objects to folders, HCA lets you call them the same name, as long as they are in different folders.

For example, assume you have two lamps. If you create two folders: one called *Den* and the other *Bedroom*, then you can create two devices and call each *Lamp*.

This can get confusing: which "lamp" is which? HCA always shows them as "Den – Lamp" and "Bedroom – lamp" whenever you have to choose a device.

When names are managed in this way, HCA calls them *Two Part Names*.

With two part names, where does the first part of the name come from? It comes from the name of the folder the object is stored in. Where does the second part of the name come from? It is the name you entered when you created the object. For example, when you used the "new device" wizard to add the device to your design you gave the new device a name.

Two parts names are enabled or disabled by selecting HCA – Properties from the menu and choosing the Design tab:

Keep these points in mind as you work with the design pane:

- Each object always is stored in a folder.
- The design pane can be organized to show objects within their folders or within their type.
- Names in HCA are *two part* where the name is the folder name and the object name taken together.

---

## Working within the design pane

There are a number of things you can do in the design pane to help create an organization that is best for your use.

1. You can rename a device, program, group, or schedule. Right click on the name and select Rename from the popup menu. You can then enter the new name.

2. You can rename a folder. This is done in the same way as described above.
3. You can reorder the folders, by grabbing the folder, dragging it and dropping it on the folder you want it to appear *after*.
4. You can drag an object from one folder to another. This changes which folder it is stored in. Each object must be stored in some folder, and can only be stored in a single folder. That is, one object can't appear in two different folders.
5. You can close or open a folder. To close the folder click on the small icon next to it. All the objects in the folder disappear. To have them reappear just click on the small icon next to it. HCA will remember which folders are open and closed when you close and reopen your file.
6. You can select more than one device, program, or group and operate upon them all at once. Not all operations are possible if you have more than one item selected. In general you can control them all on, all off, suspend, resume, delete or open their properties. You can also drag more than one object from one folder to another to change their location.

You can also control the order of the objects within a folder. To do this, right click on the folder name in the design pane and select one of the ordering options.

The small icon next to the name in the design pane can be changed in only one way. You must open the properties for the object and if there are any choices, select an icon from the choices displayed. There are a limited set of choices, if any, and you can't add you own icons to the design pane like you can for the display pane.

---

### Design pane information

In addition to a simple listing of your design elements by name and in folders, the design pane also shows the state of the object in two ways.

1. If the device, group, or controller is ON or if a program is running, the background of the icon in the design pane is colored yellow.
2. If the device, program, or group is suspended a red slash is drawn across it.
3. If the device, program, or group is disabled a black slash is drawn across it.

---

### Working with the design and display panes together

HCA supports drag and drop between the design pane and the display pane. To add an icon to a display (one that supports icons – see the Displays chapter for more information), just grab the device, program, or group name and drag it onto the display pane and drop it. You can select more than one object and drag them as a group onto a display.

In addition to this drag and drop method, you can also choose which displays an icon for an object appears by opening its properties dialog and making changes on the Display tab.

---

### What doesn't appear in design pane

The main elements of your design - devices, programs, groups, schedules, and schedule entries - are listed in the design pane.

Other elements of your design like the calendar, keypads, log filters to name just some are not listed in the design pane.

---

## The conventions used in this guide

The following are the common protocols used in this Guide.

Convention	Definition
<b>Arial font</b>	Text in this font indicates either entries that you type into HCA, or responses or messages from the HCA system for you to read.
<b>HCA</b>	An abbreviation for the Home Control Assistant.
<b>Hint</b>	A hint is a special trick, a neat way of doing something, or a reminder of other HCA operations.
<b>web tip</b>	A web tip points you to the Home Control Assistant web location for more information on a particular subject.

In this guide, we assume that you are familiar with the use of Windows, and the most common Windows conventions, such as click, drag and drop, double-click, right mouse click, use of the browse function, directories, and files. However, some of these common Windows conventions most often used in HCA are documented in the section following.

For instance, if a procedure says click, we mean to use the mouse button you most often use, generally the left. If you need to right-click an item, we'll specify that.

---

## Procedures and operations common to HCA

There are several common tasks that occur in multiple places in HCA, and that you may find yourself completing in different wizards, or for the various properties dialog boxes. Rather than repeating them or explaining them each time they are used, we have placed them here to be less repetitive.

### Selecting items for use in a particular area

This may be used to place icons for a device, group, or program on a display, to place members in a group, and in several other wizards. Regardless of the specific wizard or dialog box, there will be two boxes, separated by two arrow buttons.

You click a listed item in the left box, and then click the right arrow → pointing to the right box, and the item moves into that box. In some cases you can move one, some, or all of the items to the right column. If you find that you have mistakenly moved something to the right box, you can move it back. Select the item from the right box, and click the left arrow ← to return it to the left column.

### Object names

As described above, most HCA objects have names. You may use as many characters as you like to name your objects except for dash, left square bracket or right square bracket. There is no practical limit to the length of any name.

### Printing

HCA provides printing support for details on several elements of your design, using a typical Windows format. You can select any of the four print commands available from the Application menu:

- **Print**—provides the typical Windows print dialog box, allowing you to determine your print range, number of copies, and so on.
- **Print Preview**—shows you a reduced, full page layout of what you are printing.

- **Page Setup**—opens a dialog box where you can set the elements you want to print: home information, rooms, folders, displays, schedules, device address info, Insteon network description, protocol bridges, flags, and references.
- **Print Setup**—opens a dialog box where you can view and set the name and location of your printer, paper size and orientation.

The HCA printouts contain information about the elements in your design that you have selected using the Page Setup command. HCA formats and prints the information you chose in a vertical list, with headings for the different sections. The path for your design is included at the top of each page, along with the printout page number. Because the printouts can get long, you may want to specify only those elements that you are interested in seeing each time you print.

## Common Windows procedures used in HCA

There are also typical Windows procedures that are frequently referenced in HCA procedures. We have gathered a few of them here, for your reference in case you are not familiar with them.

### Right click for popup menu

Most objects in HCA have properties attached to them. You can generally get to the properties for the object by using the mouse, and right clicking on the object. To do this:

1. Move the mouse pointer to or over the object in question.
2. Click the **right** mouse button. (Unless you're using a mouse device set up for a left-hander, in which case, you would click the left mouse button.) This is the button that you don't normally use.

A popup menu appears.

3. With the mouse, move the pointer down the popup menu until you get to the menu command you want, and click the **left** (normal) mouse button.

In the case of properties, the properties dialog box opens.

### Drag and place

In HCA there are several things—icons on a display, entries on the visual scheduler—that you may want to move from one place to another. This is very easy to do with the typical Windows drag method.

#### To drag an icon to another location:

Move the mouse pointer to or over the object in question.

1. Click the **left** mouse button. (Unless you're using a mouse device set up for a left-hander, in which case, you would click the right mouse button.) This is the button that you use most of the time.
2. While you continue to hold the mouse button down, move the mouse pointer to the new location for this object. You will see the object moving with your pointer.
3. When you reach the correct location, release the mouse button.

The object “drops” into its new place.

### Viewing the hierarchy of your design

Several areas in HCA are organized in hierarchies like the design pane. These hierarchies use the Windows convention of small icons on the left margin to indicate the status of the hierarchy. On icon shows that the contents of the folder is hidden and another when it is seen.



This system is the same as used by the Windows File Explorer so you probably already know how to work it.



## Chapter 2

# HCA Options

You control the functions of the Home Control Assistant through various options. These options are accessible through the HCA Application menu *HCA Options* button. Because the HCA options apply to all home designs, you can review or modify them at any time, even when there is no design loaded.

This chapter discusses HCA options and how you set them to control the way the Home Control Assistant works for you. The first section in this chapter describes the similarities of all HCA properties, and tells you how to access them. The following sections describe each individual HCA property tab:

- Startup
- Client Server
- Clock
- Color and Theme
- Control Interface
- Design Pane
- Display Pane
- Hardware
- Visual Programmer
- Web
- Advanced
- Legacy
- Extra

## About Options

It is important to recognize the difference between options of HCA and the properties of a particular home design.

- HCA options are stored in the Windows registry and your choices are in effect regardless of what home design is loaded.
- Home design properties are saved in their .HCA files and affect only that design.

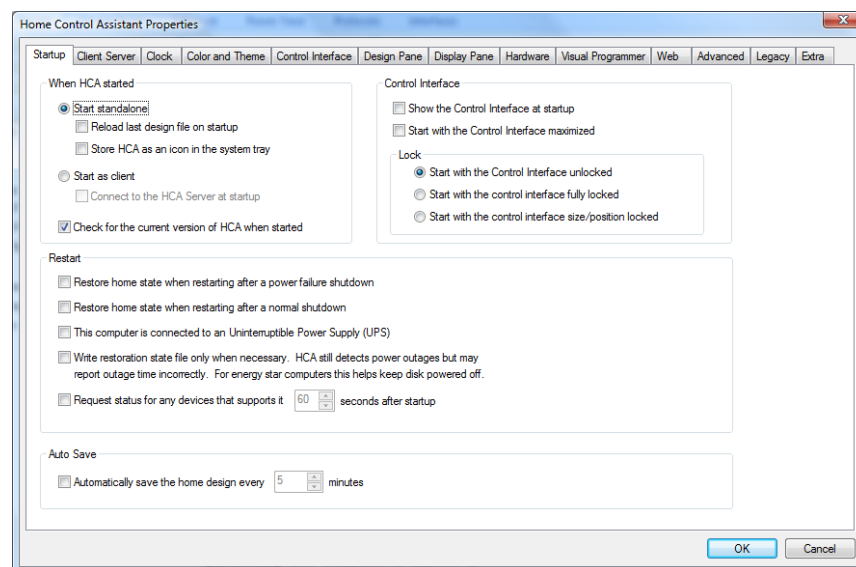
The following table provides a quick reference on the differences between the two sets of properties.

Home design properties	HCA options
Saved in their .HCA files	Stored in the Windows registry
Affect only that design	Affect HCA, regardless of what home design is loaded
Created as you complete wizards	Created with defaults when HCA is installed. Modified via HCA properties
Some are accessed from the Design category, most are available from popup menu on icon or design elements	Accessed from HCA Options
Review or modify them only when the design is loaded	Review or modify them at any time, even if no design is loaded

### Using or accessing properties

You get to all the HCA options in the same way:

1. From the HCA Application menu, press the *HCA Options* button.  
This opens the Home Control Assistant Options dialog box.

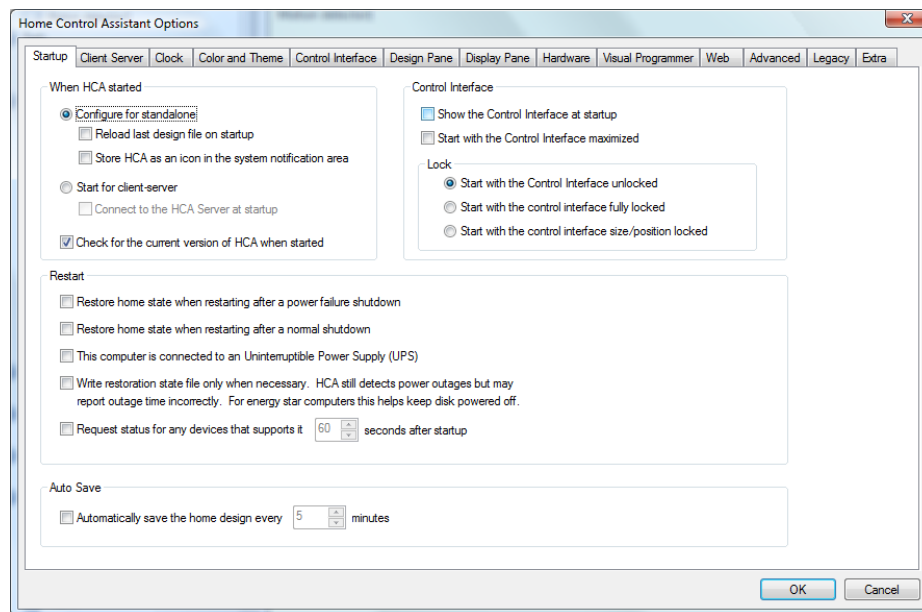


2. Click the tab for the properties that you want to view or change. This brings that tab to the front.
3. Make the changes that you want, and click another tab if you want to view or modify other properties.
4. When you are done, click OK to accept the changes and close the dialog box. The new properties will take effect as soon as you click OK.

---

## Startup

The Startup tab is the first tab on the HCA Options dialog. You use the options on this tab to control how HCA appears and functions when it is started, how it responds to a power failure, and whether it uses automatic file saving options.



### When HCA is started

There are several options designed to help you set up HCA when it first starts. These are:

- Reopen the last design loaded
- Have the HCA window hidden and only the HCA icon in the system notification area shown

The other option for starting HCA as a client to the HCA Server is fully explained in the Chapter on *Client Server*. Refer to that for a description of the Start as a client options.

The *Control Interface* options are described in the chapter on the Control Interface.

**Note:** The options that have HCA stored in the system notification area at startup and the Control Interface options only take effect if a file is automatically loaded at startup.

If you want these options to work, and HCA to be automatically started after a power failure, you must also add HCA to your Windows Startup group so that HCA starts when Windows starts. If you have done this, and select these options, HCA reloads your home design upon startup.

**Hint:** To find out how to add HCA to the Windows Startup group, search for Startup in Windows help.

### Restart

This area has five additional options dealing with restart. What these options do is covered in the chapter on *Restart*.

**Auto Save**

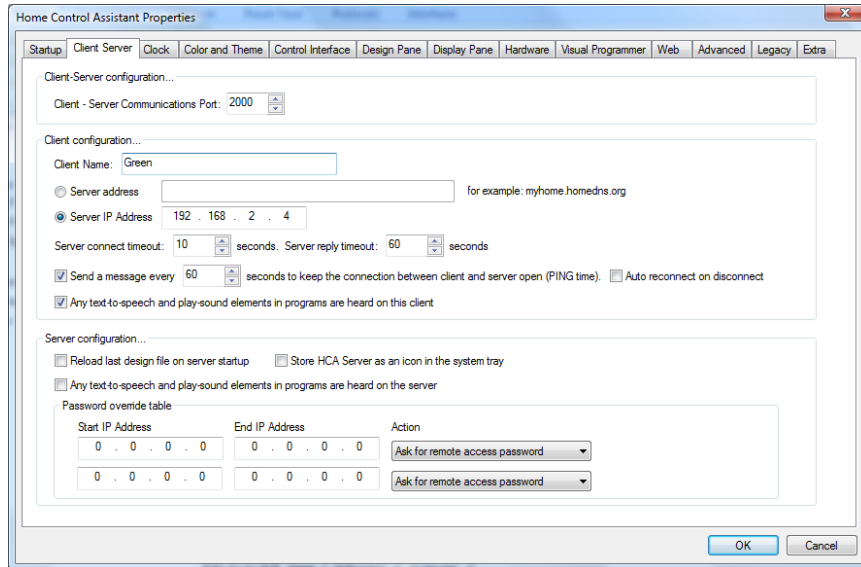
The last setting on this property tab tells HCA to automatically save changes (or not) at the time increment that you specify. This is useful so that changes to HCA are kept and not lost in the event of a power failure, or if you forget to save the file.

For example, you may work with HCA one day and make some changes to your design and forget to save them. Since HCA is always executing on your computer, it may be a long time (weeks or even months) until HCA terminates. That is too late to remind you to save the file.

Use this property to have HCA save your design periodically if the file has been modified and needs saving.

## Client-Server

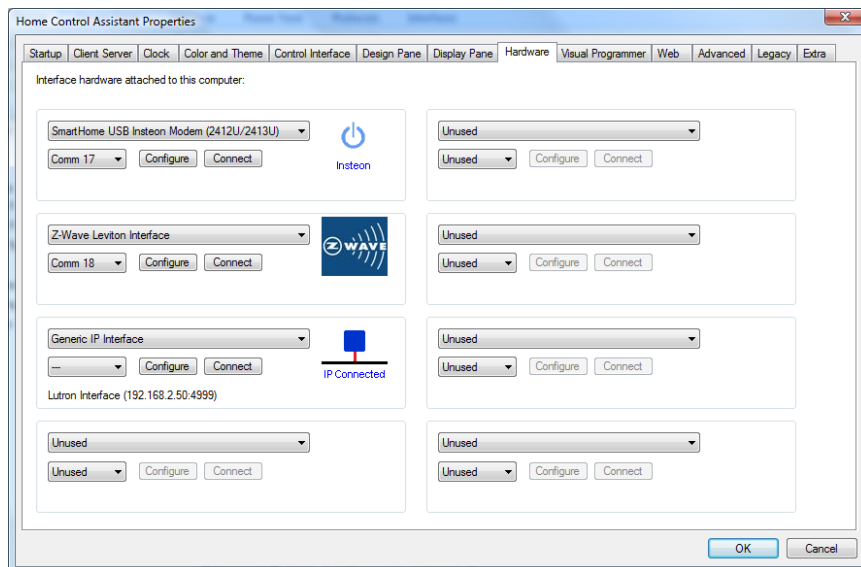
The client-server tab configures those settings necessary to have HCA operate in client-server mode. This tab appears as:



The client-server settings are fully described in the client-server chapter

## Hardware

The HCA properties tab for Hardware is the only place in HCA where you tell HCA about the type of automation hardware connected to your computer.



HCA supports from one to eight automation interfaces attached to your computer. Each must be attached to its own serial or USB port, or accessed via a network connection.

This can be used to get the best possible use from your hardware. For example you may have one interface to handle Insteon devices and another for UPB devices. Both can be operating simultaneously.

**Hint:** Not all versions of HCA support all interfaces. See the appendix on HCA versions for an explanation of what version supports which hardware. Also some of these interfaces are designated *Legacy Devices* and are not normally available unless requested on the Legacy Properties tab.

For each interface select the port it is attached to, and what the interface is. Next use the Connect button to confirm that the interface can be sent commands by HCA and process receptions from it.

If you have the hardware correctly attached, you will see a message box that says **Hardware test... OK**. The message box also provides any information from the hardware confirming that it is functioning. If the hardware is not correctly connected, an error message explains what the problem is.

If you ever need to disconnect and reconnect the automation hardware, be sure to return to this dialog box and use the *Connect* button to make sure that it is correctly hooked up and working.

Some interfaces support multiple protocols or require extra configuration. The Configure button opens a dialog appropriate for the type of interface.

**Hint:** See the appendix on the type of interface you are using for any specific information about that hardware.

---

## Design Pane

One tab of the HCA properties dialog is for the Design Pane. Most of these options are covered in Chapter 1.

The first section helps you control the way HCA displays times. There are three options for time format. You may choose only one, but you can change it at any time. The options are:

- AM/PM format
- 24-hour format
- Follow the control panel regional time settings

You can display all time values in either AM/PM format or 24 hour time format. The third option allows you to use the time format options set in the Windows control panel regional settings (which may be 12 or 24 hour time). HCA always uses the time punctuation and AM/PM strings from the regional settings, regardless of how you select the time format using the HCA properties.

HCA always follows the control panel regional settings for *date* formats, however *time* format is a bit more subjective. You may be comfortable working with 12-hour time, or with 24-hour time. You set this preference in Windows with the regional settings. But since HCA is so time-focused, you can have your needs met here, regardless of what is set up for other programs.

Note that the 24-hour time format is sometimes referred to as military time. In this format, 12:00 noon is 1200, 1:00 p.m. is 1300, 5:00 p.m. is 1700, and 12:00 midnight is 2400.

**Hint:** If you use the AM/PM format, you should remember that 12:00 p.m. is noon, and 12:00 a.m. is midnight.

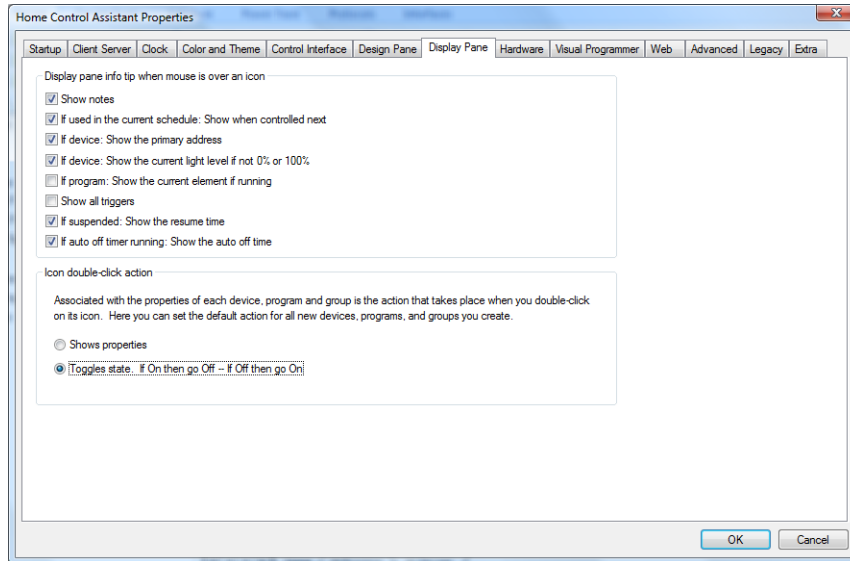
Also on this tab are options for HCA to automatically create and keep updated a display that shows all your devices and another than shows all your programs.



## Display Pane

Another tab of the HCA properties dialog is for the Display pane. While this tab is described here, refer to chapter on displays for a full explanation on how HCA works with displays.

Use the Display tab to set properties defining what you see in the display pane (right side) of HCA. The display pane shows the displays you have added to your design.



### Display Pane Info Tips

This section applies only to displays that show icons. When the mouse is placed over an icon and doesn't move for a few seconds, a popup message, called an info tip, displays. These info tips can show lots of useful information about the object the mouse is over. The seven checkboxes give you control over what is contained in the popup.

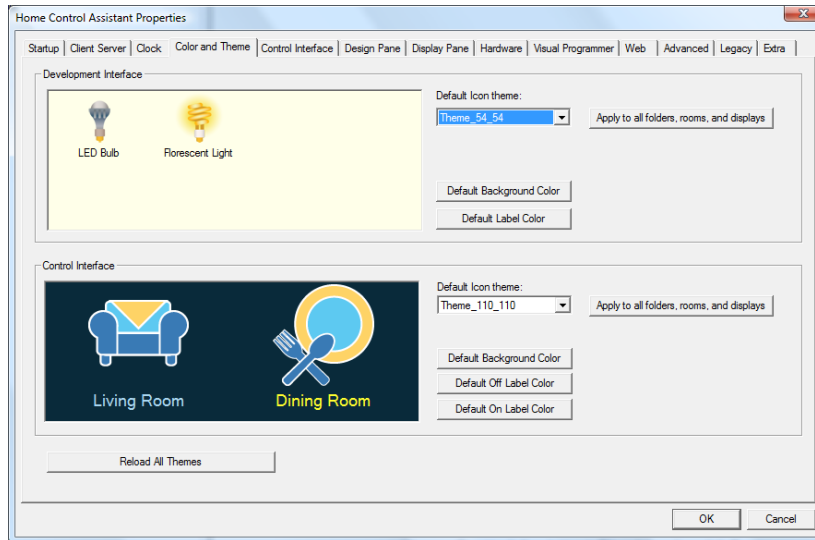
### Icon double-click action

As the dialog text describes, if you double-click on an icon in the display pane it can either control the object or display its properties. While you can set this property for each object, what you select here is the default setting for any devices added.

## Color and Theme

The Color and Theme is for selecting the default color and theme to be used for each display, folder, or room when it appears in the display pane.

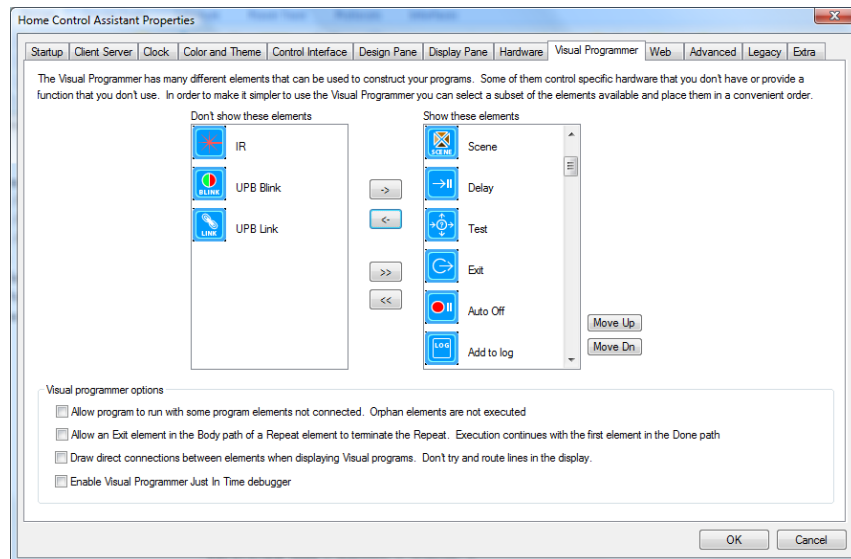
You can select the color and theme for each display, floor, and room independently of each other. This is the settings that are used when a new folder, room, or display is created.



If you decide you want to change the colors and/or theme for all your displays, folders, and rooms, you can make the changes and then press the “Apply” buttons to make that change.

## Visual Programmer

The Visual Programmer is the tool used in HCA to create programs. Programs are a sequence of actions that happen in response to some event. Visual Programs are composed of over 50 different types of actions. Most are general purpose and some are very specific to certain automation interfaces. Working with so many different actions can be daunting. This tab allows you to customize the list of elements that you want to work with and the order they appear in the Visual Programmer.



The left column shows those elements that aren't shown in the Visual Programmer, and the right column shows those elements that are and the order in which they appear.

To move an element between the two columns just select it and use the → and ← buttons. To move all from one column to the other use the >> and << buttons.

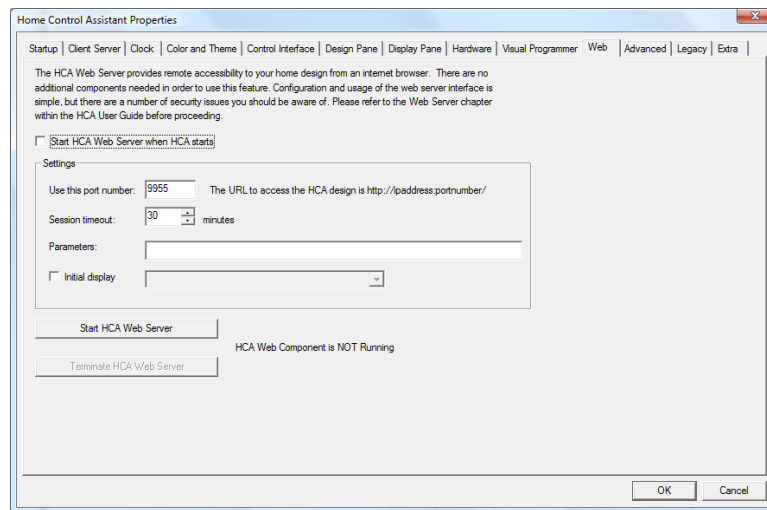
To change the order in the right column, select the element and use the *Move Up* and *Move Dn* buttons.

**Hint:** If you are new to the Visual Programmer and are wondering what this is all about, think of the Visual Programmer as a Paint program. The elements you have here are the colors in your palette. This tab of HCA properties allows you to choose which colors are on your palette from all available choices.

---

## Web

The web tab is for configuration and setup of the HCA web server.



This dialog and the parameters it contains are described in the Web Server chapter.

---

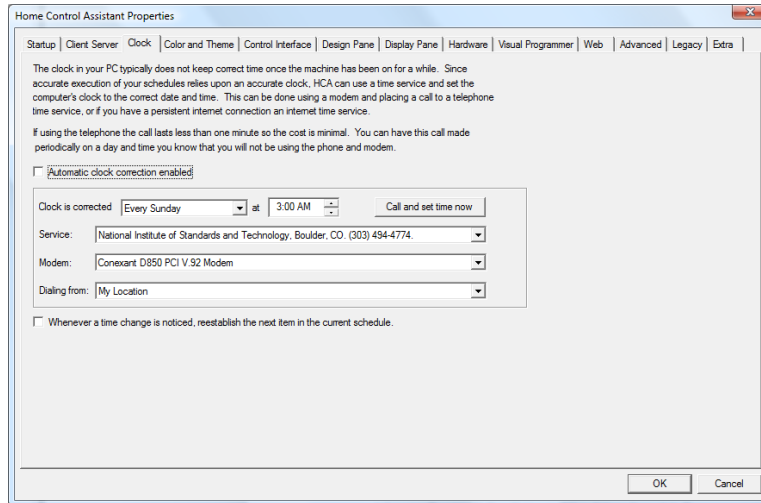
## Clock

The Clock tab in can help HCA stay punctual.

**The HCA clock setting facilities are used for those very few PCs that don't have access to a 24/7 internet connection. If you do have such an internet connection Windows already offers methods to keep the lock up to date and the HCA settings shouldn't be used.**

If there is one thing HCA does well, it's watching the clock. Inside every PC is a clock which HCA reads to determine the date and time. But, HCA must have access to the correct date and time in order for it to *correctly* control devices in your home according to the schedules you have created.

Unfortunately, the clock in a PC (particularly older PCs) keeps very poor time. Once the computer has been powered on for a while, you may notice that the time the PC reports may be off by several minutes, or even tens of minutes. If you have a connection to the internet or a modem in the PC that runs HCA, then HCA can use the internet or modem to connect to a service that reports back the correct date and time.



This dialog box has extensive instructions, and you may not need any other information on using it. However, here are a few details:

#### **Automatic clock correction enabled**

This checkbox must be checked, or no adjustments will be made to the PC's clock, regardless of the other settings.

#### **Clock is corrected \_\_\_\_\_ at \_\_\_\_\_**

Select the day of the month and the time that you would like the PC's clock set.

If using a modem, it typically takes under 10 seconds to check the time, from the time the telephone time service answers until the call terminates. It is important that you select a time of the day when the phone line HCA is connected to is not being used.

#### **Service**

HCA provides three options for time services:

- National Institute of Standards and Technology in Boulder, Colorado, at 303-494-4774
- United States Naval Observatory, Washington, D.C., at 202-762-1594
- The internet. This is done by having HCA start a windows program called NISTime32.exe

HCA comes with these options and phone numbers already listed; you only need to choose the one you want to use. If using a modem, you may want to choose the number of the service that is geographically closer to you. (We provide the numbers here for backup.)

#### **Modem – for VERY old PCs**

This is the modem used to make the call. HCA lists the options that are available on your PC. If your PC has more than one type of modem line, click the arrow on the list box, and select the one that you want to use. This is not available if using the internet.

#### **Dialing from**

This is the number you are calling from. HCA lists the choices available on your PC. You can select any option listed. This is not available if using the internet.

**Hint:** HCA uses the information from the Modems Properties dialog box (the Modems applet in the Windows Control Panel) to gather information and present options in the Modem and Dialing from boxes.

#### **Call and set time now**

You can use this button to override the pre-set time, and make the time check right now.

**Hint:** If you want to use the modem time set feature, after you select the modem and dialing location, use the “Call now” button to make sure that you have your modem configured correctly, and that all works well.

The checkbox option “**When a time change is noted...**” tells HCA to determine the next entry in the current schedule each time the clock on the PC changes. This could be due to an automatic time change if you have Automatic Clock Correction enabled, or Daylight Saving Time changes twice a year. Each time HCA notes that the PC clock has changed it creates a log entry.

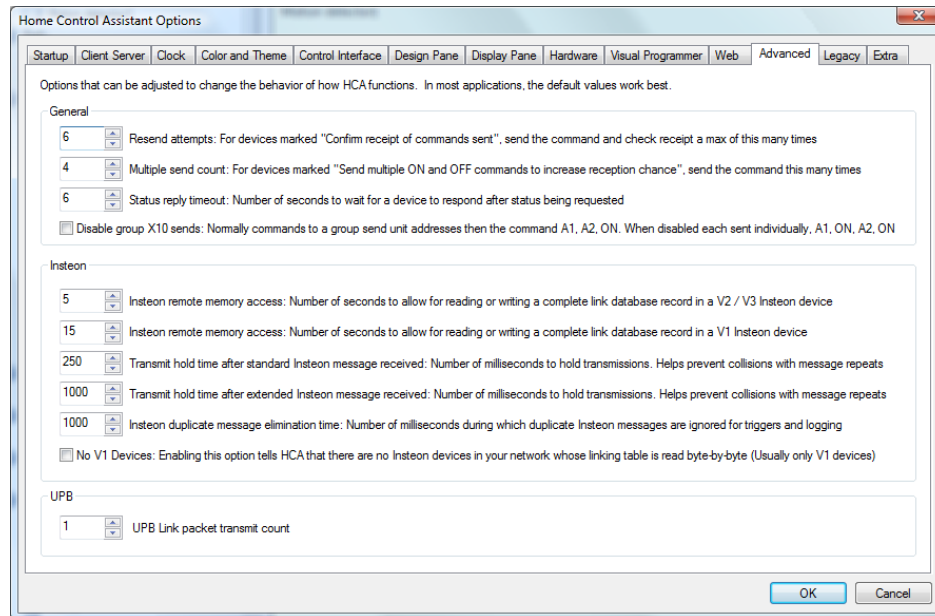
**Several more points about using the HCA clock options:**

There are several general tips that can help you make the best use of your PC time settings, and the Home Control Assistant clock options:

- You should use Windows Control Panel Date/Time option to check that your time zone is correctly selected. You should also select the option so that Windows can make changes to daylight saving time if you live in an area of the country that makes daylight saving time changes.

## Advanced

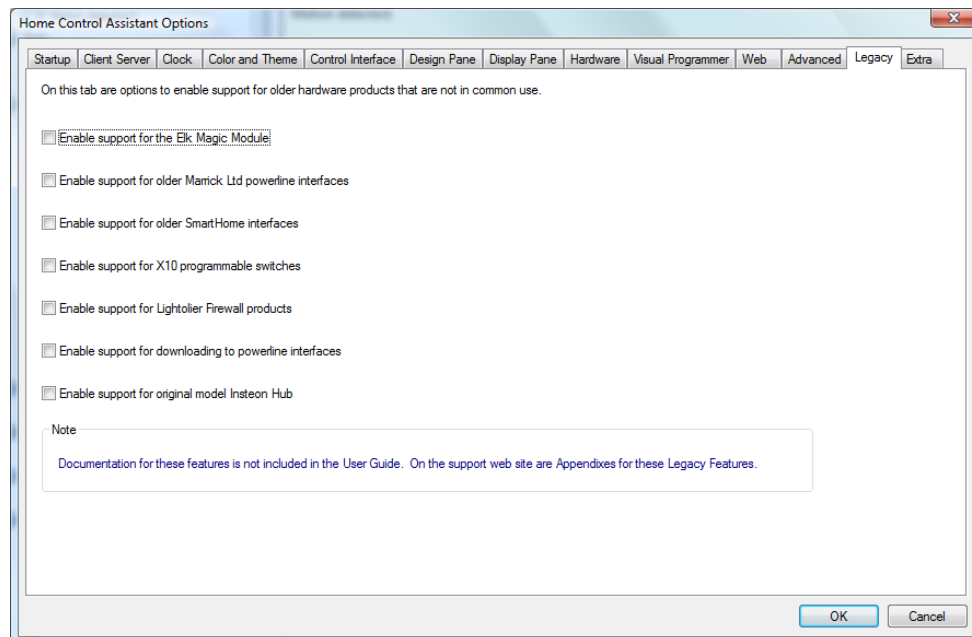
This tab is for several options that control specific features in HCA.



Hopefully the text in the dialog should be explanatory as to what these options do.

## Legacy

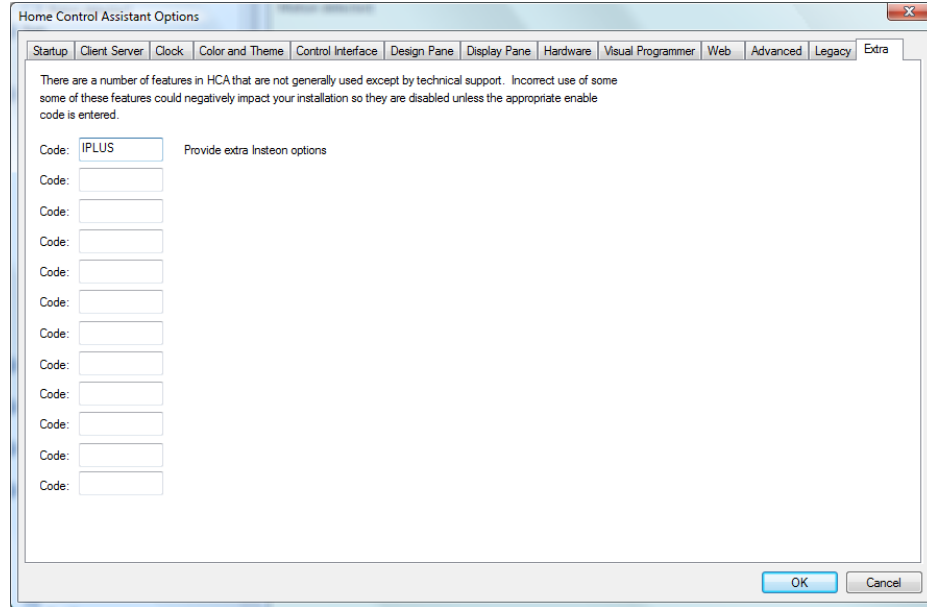
As time has passed, various automation gear supported by HCA has become less common. In order to reduce the amount of clutter in the User Interface these devices and the supporting user interfaces for their features has been hidden. Enabling options on this tab brings these options back into HCA.



---

## Extra Features

The last tab is a place to enter codes to enable or disable some HCA features. All of them are features that are normally not available as they are very special purpose or intended only for advanced users. Some of these codes are described in various User Guide chapters and Technical Notes.





## Chapter 3

# Your Home

You've already read the introduction to the Home Control Assistant and learned about HCA Properties. Now you're ready to create your own home design in HCA. This chapter tells you how to create a design, and save it in a file. This chapter also discusses simple modifications to your home properties, and security options for your design.

This chapter includes topics on:

- Planning your home design
- The HCA Designer
- Starting with an empty file
- Modifying your home file
- Light and Dark settings
- Security settings

---

### Planning your home design

There are two ways to begin a new home using HCA:

- The HCA designer is a quick way to get started by adding rooms and the devices in those rooms. It supports Insteon, UPB, and X10. Once the devices are added the most common configuration can be done.
- Starting with a blank or empty file.

Regardless of which of these methods is used to create a new design, there are two important pieces of information you'll have to configure.

#### **Design name**

You must supply a name for your design. This name appears in the HCA title bar when this design is loaded, but this need not be the file name that the design is stored in. If you like, you can create several designs, each with a unique name, and in a different file (you may want to do this if you're trying different approaches with your programs and schedules).

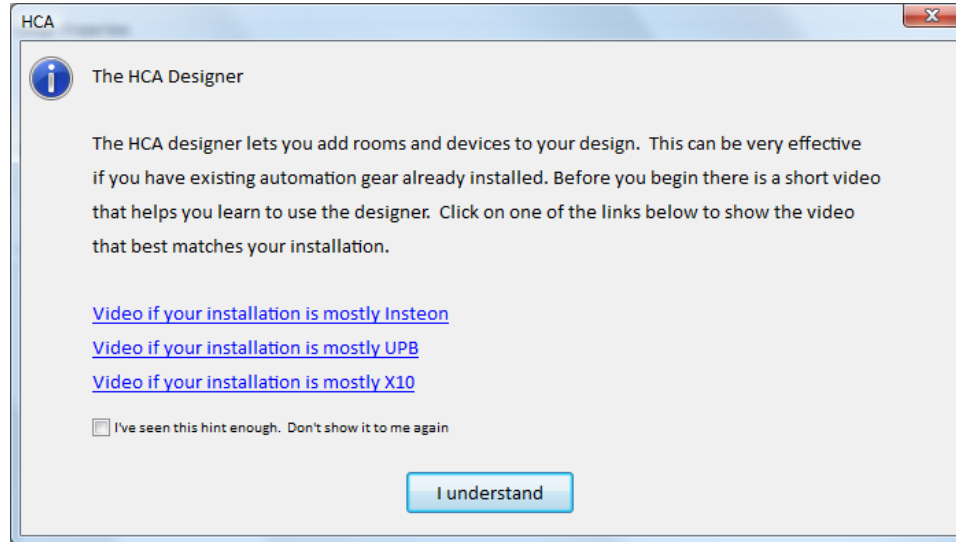
#### **Location**

You must also supply the location of your home. HCA provides a list of zip codes for United States locations. For outside of that area you can configure by entering in your location information.

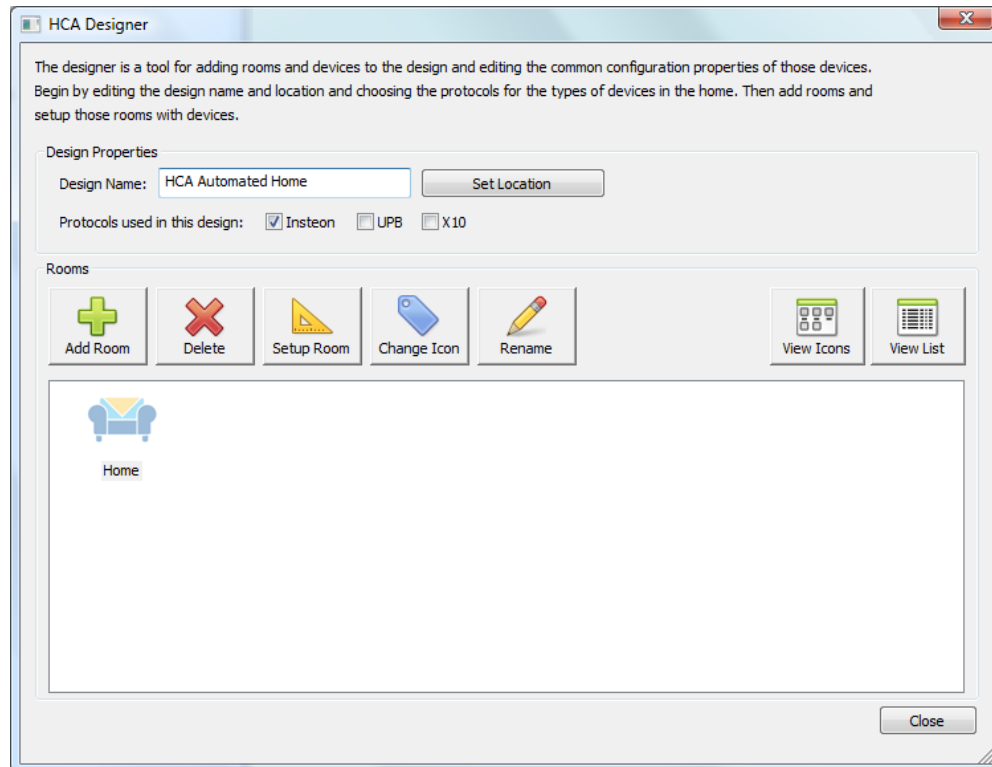
HCA uses your location to compute the sunrise and sunset times for your schedules. HCA also relies on the time zone you have selected in the Windows control panel Date/Time settings tool. If you select a location in HCA that is in a different time zone from the one you have set in the control panel, the sunrise and sunset times may appear to be off. So, remember to check your control panel time zone selection.

## The HCA Designer

The best way to begin working with HCA is by using the *HCA designer*. When it first starts, you are offered a link to a video that shows how it works depending upon the automation protocol you have. We urge you to watch the video.



The first steps in using the designer is to enter a name for your design, determine the location and select which automation protocols you are using.



Type in a name for your design, tick the appropriate boxes for which automation protocols you are using then select your location after you press the "Set Location" button.

If you are in the United States, selecting the location is done by zip code. Outside of the United States the location must be entered manually.

Location

For sunrise and sunset times to be determined it is necessary to provide the location of your home by latitude and longitude. For US locations enter your zip code and press the lookup button. For non-US locations use the manual location setup button.

Location

Zip Code: 92292

Manual Setup

This Location

City: Palm Springs  Computed Astronomical Sunrise today: 6:27 AM

State: CA  Computed Astronomical Sunset today: 4:37 PM

Latitude: 33.752886

Longitude: -116.055617

Time zone: Pacific

DST: Observed

Local Sunrise and Sunset

For this home sunrise is 0  minutes  the computed sunrise

For this home sunset is 0  minutes  the computed sunset

For manual setup, you need to know your location in Decimal Degrees. Most GPS devices can be configured to show this. Also you can get your location – perhaps with a smaller precision – from online mapping programs. For manual configuration, specify your time zone by the number of minutes from UTC and the rules for when Daylight Saving Time (Summer Time) starts and stops.

Installation Location

Location

City: Palm Springs

State / Country: CA

Latitude: 33.752886

Longitude: -116.055617

Enter in Decimal Degrees as shown on a GPS unit. South Latitude and West Longitude are entered as negative numbers  
For example, Palm Springs California USA has latitude 33.842984 and longitude -116.54344

Time zone bias: -480  Minutes to add to UTC time to calculate local standard time (Example: For US Pacific enter -480)

-420  Minutes to add to UTC time to calculate local daylight time (Example: For US Pacific enter -420)

Daylight Saving Time (Summer Time)

Installation location follows Daylight Saving Time

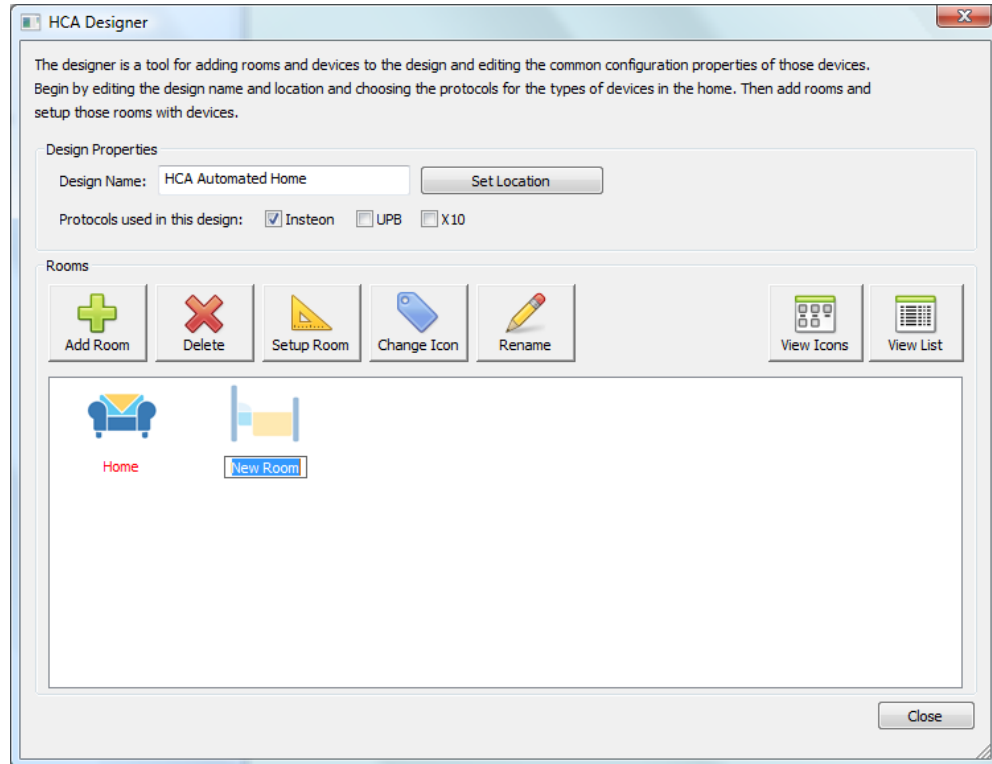
DST start day: Second  Sunday  March

DST end day: First  Sunday  November

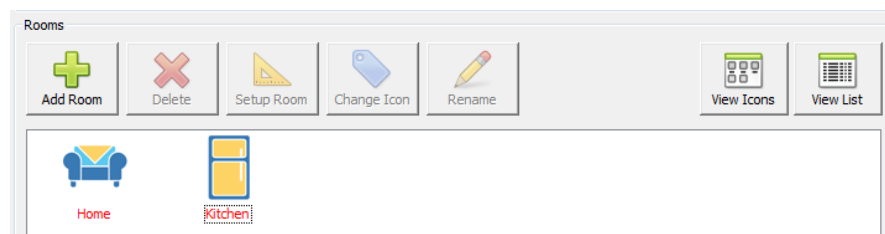
Regardless of how you enter your location, specify your local sunrise and sunset time as the number of minutes before or after the computed sunrise and sunset. This is useful if your home is on a hill – sunrise is earlier than computed and sunrise is later, or blocked by a building in the east – sunrise is later and sunset is as normal.

## Adding Rooms

Once the location is set, the next set is to add rooms. Each time the "Add Room" button is pressed a new room icon is added to the display:



Once the room icons appears, it is all set for you to type in the name you want for the room. When you complete entering the room name and have pressed the enter key, then HCA guesses as to an appropriate icon.



In this example, "Kitchen" was entered as the room name and an icon representing a refrigerator (Kitchens have refrigerators so that's the icon) was chosen.

If at any time you want to change the icon of a room, just select it and press the "Change Icon" button. To change the room name, select it and press the "Rename" button. And to delete the room, select it and press the "Delete" button.

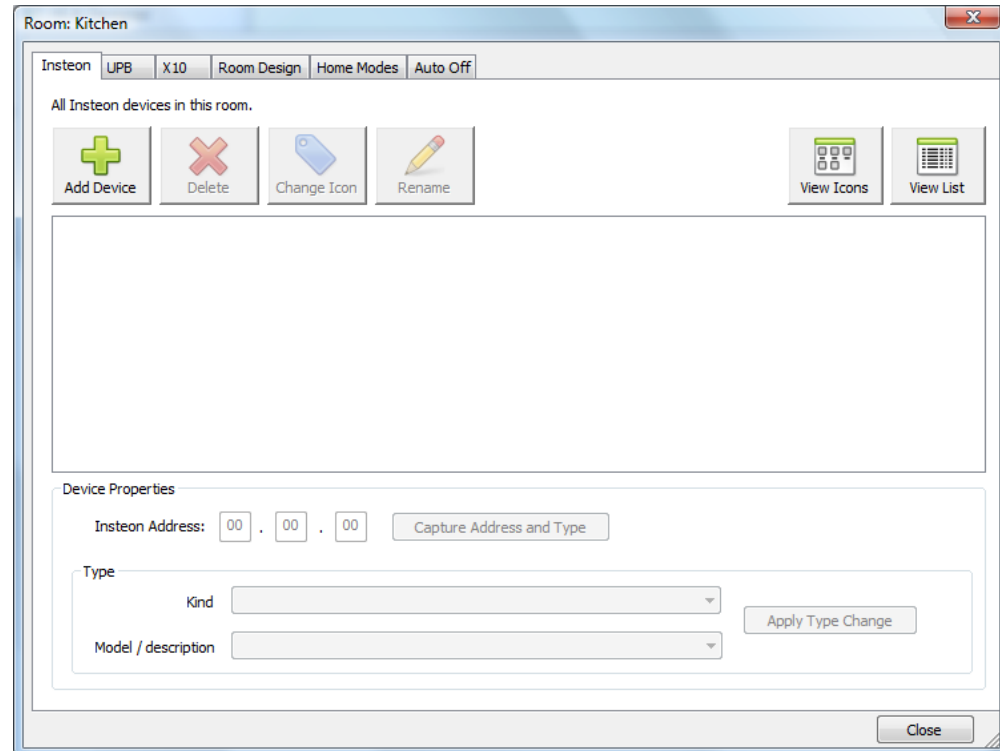
The text below the icon gives the room name. It displays in red if there are no devices yet in the room.

**Tip:** You can also right-click on the room icon and the popup menu contains these actions as well.

The design also lets you show the rooms by icon or by a list. The buttons at the right control this. Some users like to see things by icons and others prefer a table. It's up to you.

The next step is to add devices to the room and how that happens depends upon the protocol of those devices but they all start in the same way. Select the room icon and press the "Setup Room" button.

**Tip:** You can also double-click on the room icon to have the same action occur.

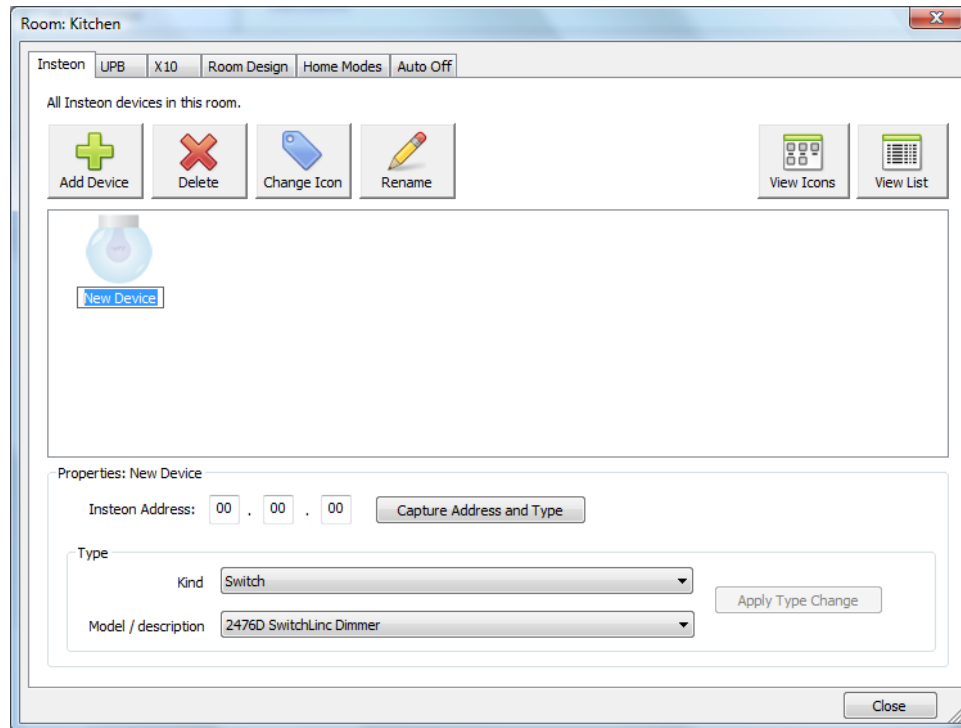


Note that the top of this dialog there are six tabs. These are:

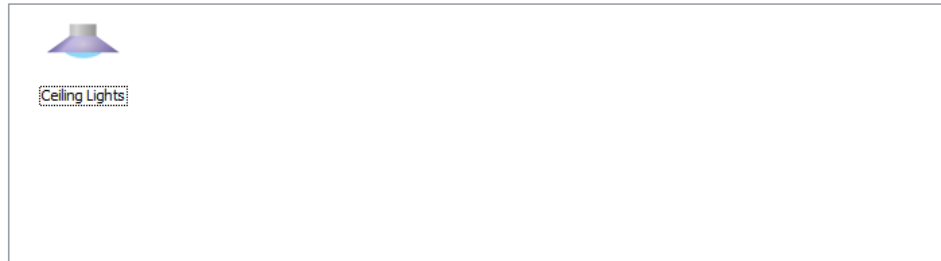
- Insteon: For adding an Insteon device
- UPB: For adding a UPB (Universal Powerline Bus) device
- X10: For adding an X10 device
- Room Design: How devices interact with the room they are in
- Home Modes: How devices act depending upon the mode the home is currently in
- Auto Off: If the device automatically goes off a specified number of minutes after it is turned on

Depending upon the checkboxes on the main Designer dialog, the three protocol tabs may not all appear.

Once the room is open, to add a new device select the tab for the protocol of the new device and then press the "Add Device" button. It works in the same manner as adding a room. An icon for the room is added and you can enter the name for it. Once the name is entered, HCA guesses at the appropriate icon for it.



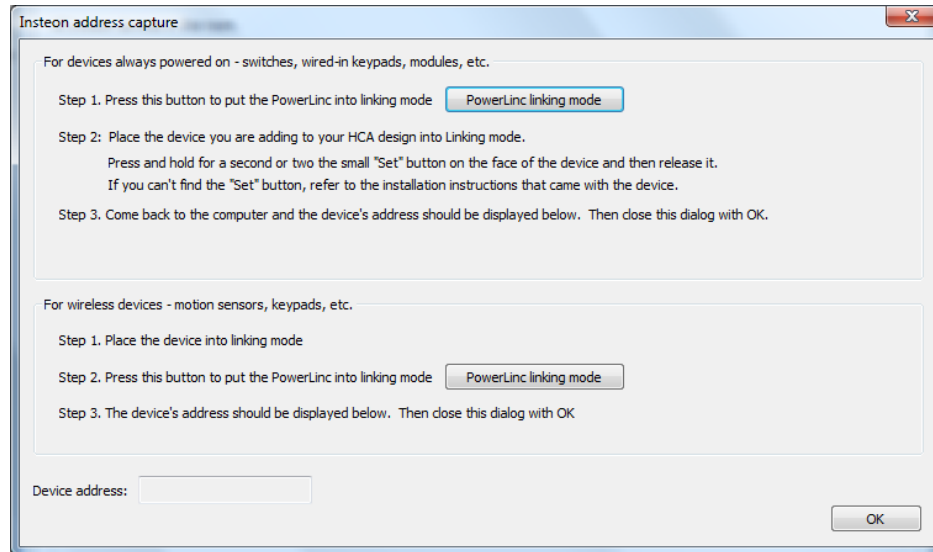
In this example, the device name was changed from "New Device" to "Ceiling Lights" and the icon changed to reflect that:



Similar to working with rooms, you can delete a device, change the icon for it, or rename it by selecting and pressing the button for the action you want to perform.

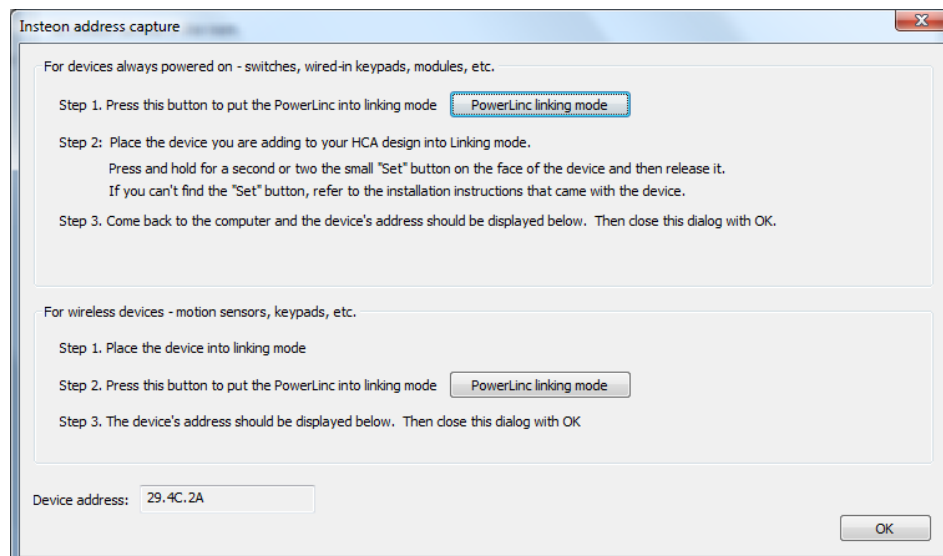
## Adding an Insteon device

For Insteon devices the next step after adding the device is to capture the address and type for it. Select the device and then press the "Capture Address and Type" button. This dialog opens.

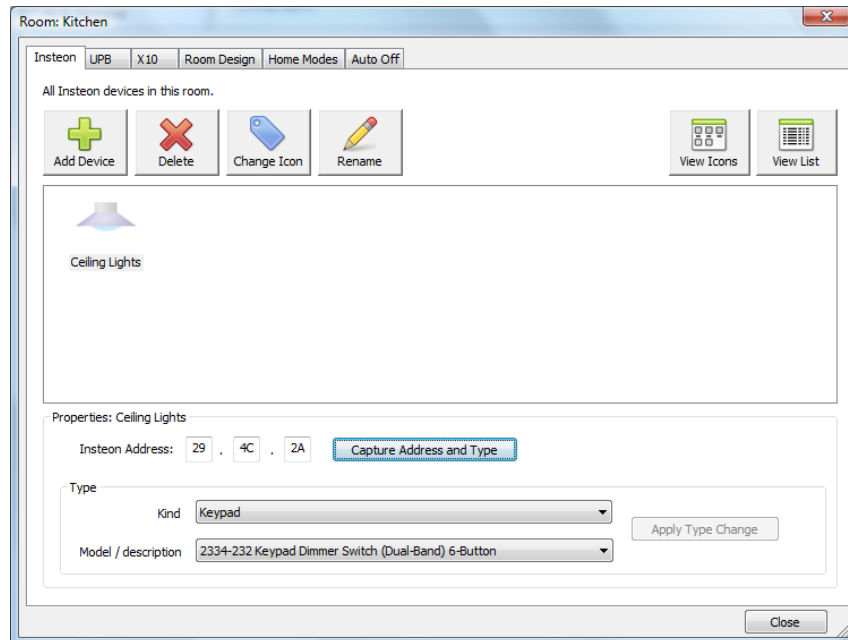


Follow the instructions in the dialog. For devices like switches and modules that are always powered on, press the top button in the dialog to place the PowerLinc into linking mode and then at the device, press its "Set" button to complete the link. For wireless devices like the open/close sensor, wireless thermostats, etc use the second section of the dialog. In this case you place the device into link mode first then press the dialog button to put the PowerLinc into link mode to complete the link.

When the link is complete the address of the device is shown at the bottom of the dialog.



Close the dialog with OK and the address and model/description of the device shows in the room contents display.

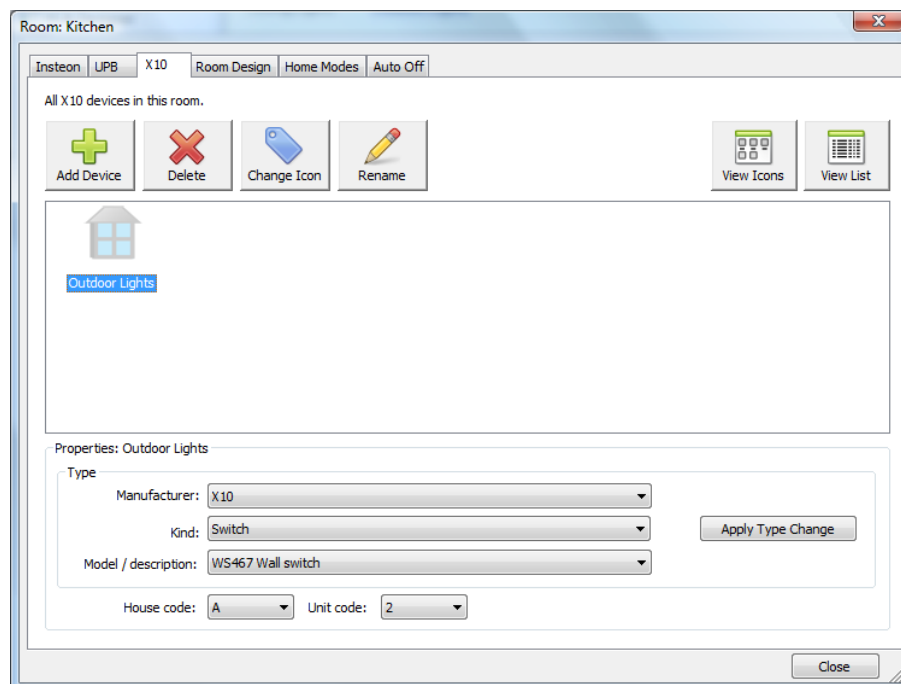


In general, there should be no reason to override the type determined unless the device is very old or a very new type, but if you need to then select the new type with the "Kind" and "Model/description" dropdowns and then press the "Apply" button.

---

### Adding an X10 device

Adding an X10 device starts the same as adding an Insteon device: On the X10 tab, press the Add Device button, and enter the name of the device. Once added then select the type of the device and its X10 address in the lower part of the dialog.





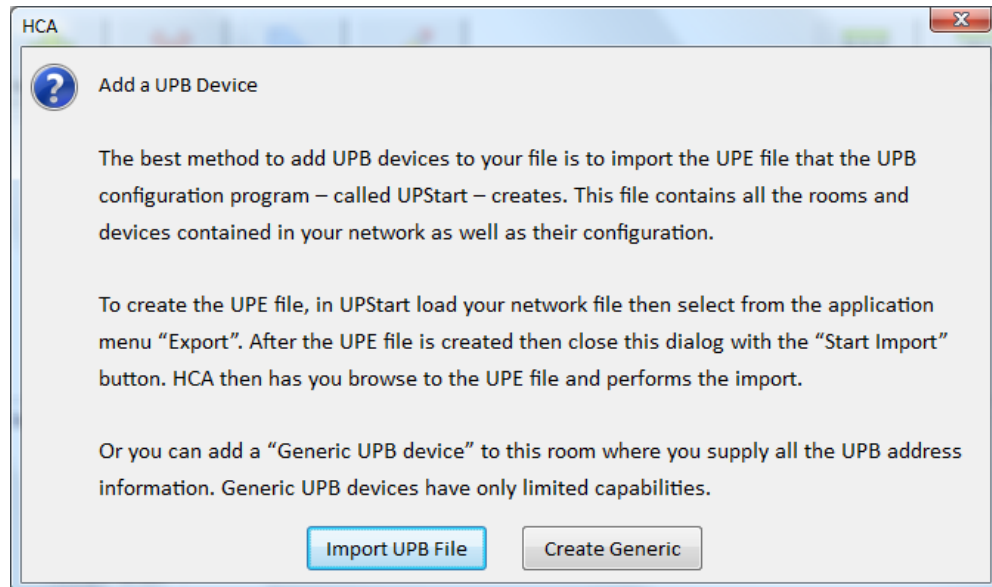
After selecting the Manufacturer, Kind, and Model, choose the X10 address and then press the "Apply Type Change" button to capture the information.

If at any time you want to change the type or address, just select the device, make the changes, and press the "Apply" button again.

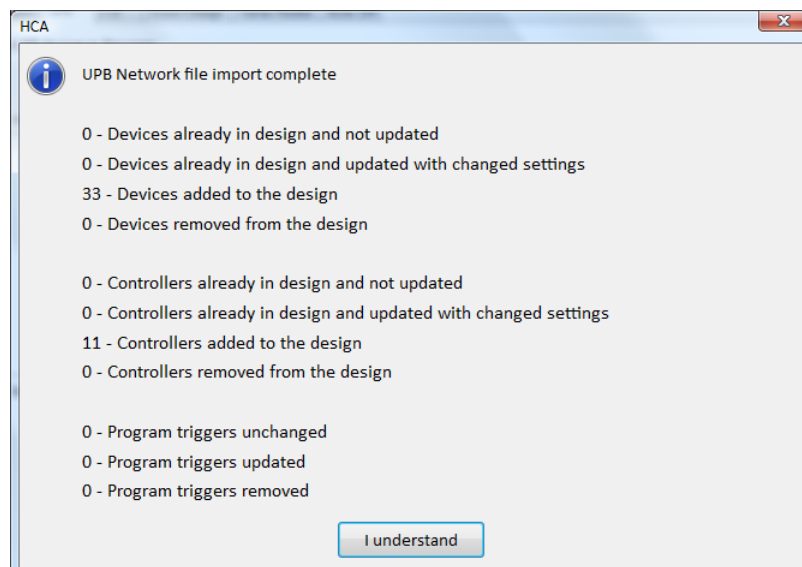
---

## Adding UPB devices

To add UPB devices to your design proceeds differently than X10 or Insteon as all the devices in the UPB network are added at one time. When the "Add Device" button is pressed on the "UPB" tab this popup explains:



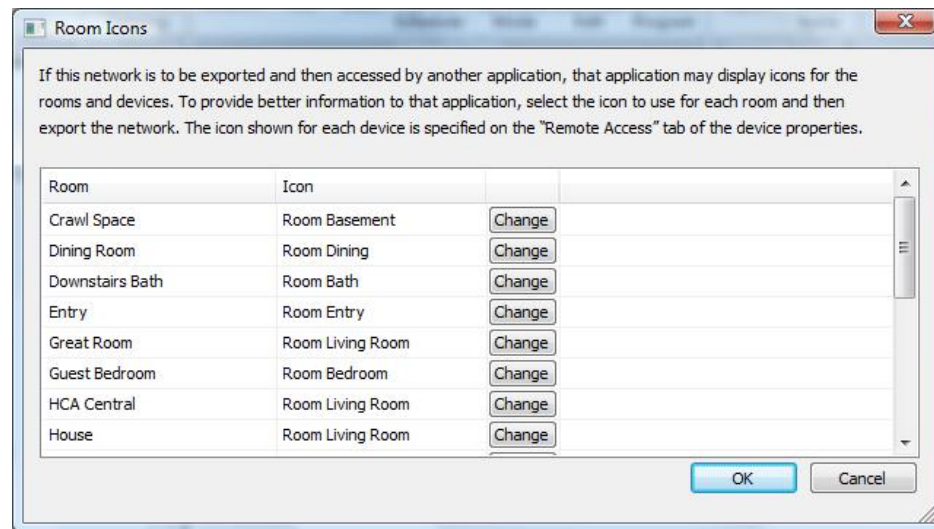
If you press the "Import UPB File" button a File-Open dialog lets you select the UPStart Export File (UPE) and then the devices in that file are added to you design. When complete a popup shows you what has been imported:



## Using UPStart

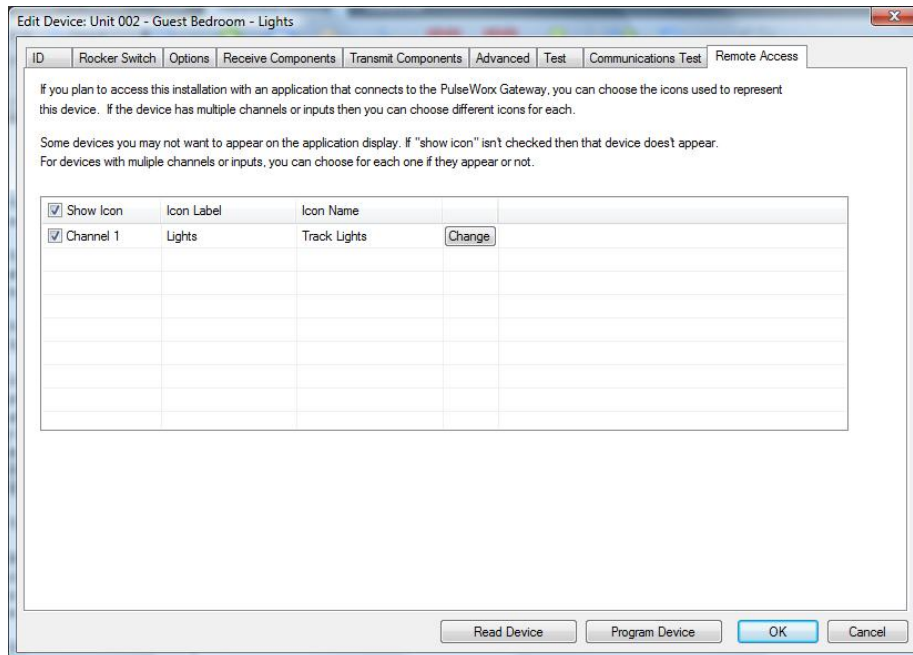
UPB devices are configured by a separate program called UPStart. Before creating the UPE file from UPStart, you should take a few minutes to configure how icons appear in HCA for rooms and devices.

When running UPStart with your UPB file loaded, in the “Gateway Export” ribbon category there is a button labeled “Room Icons”. Press that to open the room icon selection tool. It shows each room in your UPB network and what icon is used for that room. You can change the icon selected by pressing the “Change” button on the line for the room. This opens the icon selection tool. The currently selected icon is shown in the upper left. Just select the icon you want and then close the icon selection tool with OK.



For device icons you need to open the properties of the device. In the properties for a device is a tab labeled “Remote Access”. For simple devices like switches and one-channel modules there is only a single icon to select. The icon choice is displayed and changed in the same manner as room icons. From this tab you can also change the text that shows below the icon. Normally this is just the device name but you can change it if you want. In the icon selection tool, the icon label is shown at the top of the dialog for editing.

Additionally, if you don’t want this device to be imported into HCA – maybe it is something you never want to work with in HCA - you can disable the “Show Icon” option by “un-ticking” the checkbox.



For a two channel device like the PulseWorx Output Control Module you can select an icon and icon label for each channel. If you are only using a single channel of the device, you can un-tick the unused channel so it doesn't appear in your HCA design.

For an Input module (ICM, TCM, DBM) you can name each input and those appear on the application display to show the state of the input. Again, you can choose to show 1 input, both inputs, or no inputs.

<input type="checkbox"/> Show Icon	Icon Label	Icon Name	
<input checked="" type="checkbox"/> Input 1	Driveway Sensor	Appliance	Change
<input type="checkbox"/> Input 2	Driveway Sensor	Appliance	Change

Also, make sure that any keypads in your design have engraving info specified that gives the name for each button. This text is exported into the UPE file and HCA uses it in popup keypads where you can virtually “push” keypad buttons”. For this purpose, the button size and font choices don't matter.

Small Buttons

Font Style:

Font Size:

Text:

Large Buttons

Font Style:

Font Size:

Text:

ON

A

B

C

D

OFF

Once you have all the devices configured in UPStart, to create the UPE file select from the UPStart application menu, "Export" and then "Export to file".

## Configuring the Room Design

Once all the devices are added then next step is to decide how devices within a room interact. This is more fully described in the User Guide Rooms chapter, but here is the general idea.

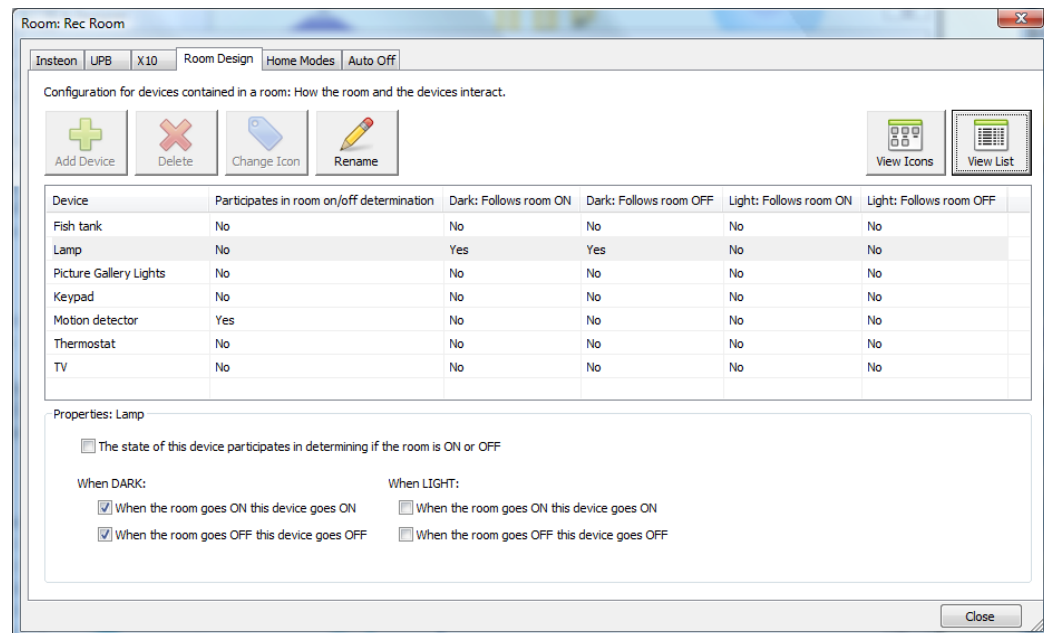
A room in HCA can either be ON or OFF. When a room is ON one or more of the devices in the room are ON. When a room is OFF then all of the devices in the room are OFF.

For each device in the room you specify how it and the room interact:

- When the room goes on does the device go on as well?
- When the room goes off does the device go off as well?
- Is the state of the device – on or off – help determine if the room itself is on?

While this may seem complex it can make implementing some basic "room logic" simple. Suppose that you have a motion sensor and some switches. The motion sensor can be marked as determining if the room is on or off and the switches configured so that when the room goes on they go on, and when the room goes off, they go off.

By just a few selections you can easily have the lights come on when the motion sensor sees motion. In the Rooms chapter you can read how this system can be additionally configured so that the motion sensor and switches work better together to make sure the lights don't go off if you just happen to stop moving.



In this example, note that the Motion Detector determines if the room is on or off and the Lamp follows the room..

Also note in this example, that there are two classes of settings – when light and when dark. HCA has extensive configuration for how the concept of "light" and "dark" is determined in your home. But the general idea is that you may want to do different actions depending upon if it is daytime or nighttime.

---

## Configuring Home Mode Response

Home Modes are a very simple yet powerful system to control devices in your home as your home goes through its day.

Let's assume a normal day to see what you do and how your home can respond.

Midnight	Asleep
6am	You wake up
7:30 am	You leave home
6:30 pm	You return home
10:30 pm	You go to bed

OK, so that's the schedule now let's see how your devices should respond.

### TV

Most televisions these days are never really off. They are in a lower power mode when they appear off but still are using power. This makes it possible for remote controls to turn them on. When you are not at home or asleep the TV might as well be truly powered off. You can do this by plugging it into a controllable outlet.

### Battery chargers

Anything with a "power brick" uses power even if nothing is attached to it. When you are not at home it should be off. When you are home – even when you are asleep – it should remain powered on so things can charge. Again, a controllable outlet can be used here.

### Interior lighting

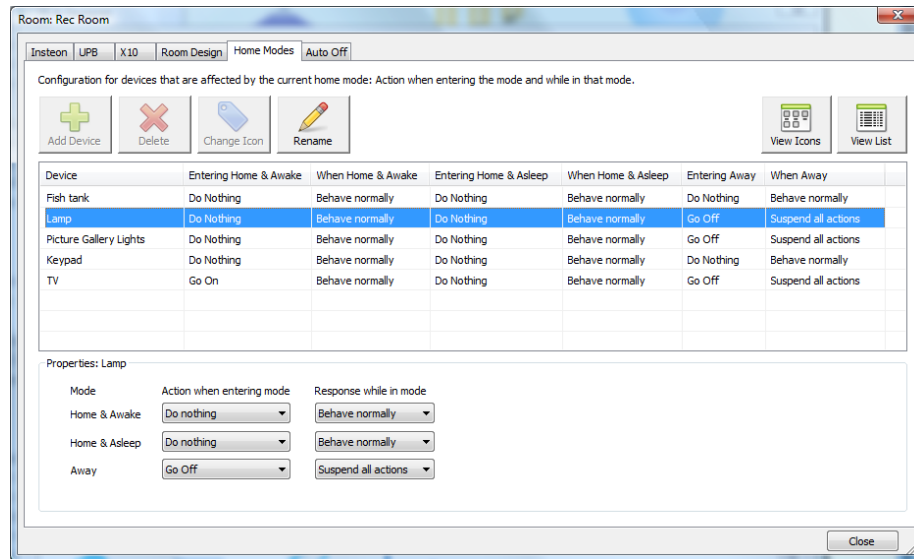
Interior lighting should be off when you are not at home and should go off when you leave or when you go to bed. While you are home, nothing should be automated.

### Circulation Pump

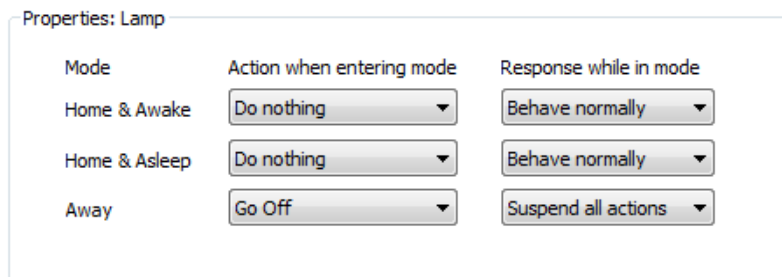
There is no point in having hot water distributed through your home when you are away but should come on when you get up and when you return home.

These are just some of the devices in your home that can be controlled based upon what *Mode* your home is in. The intent here is to have off what should be off when it should be off. This results in energy savings and reduced electricity use.

Some very simple options in the properties for your devices can make all this happen and the Designer can configure these. Select the "Home Modes" tab.



In this example, you can see the configuration for each device: What action to take when the house is going into a mode and when it is leaving a mode. Let's focus on the Lamp device for this example.



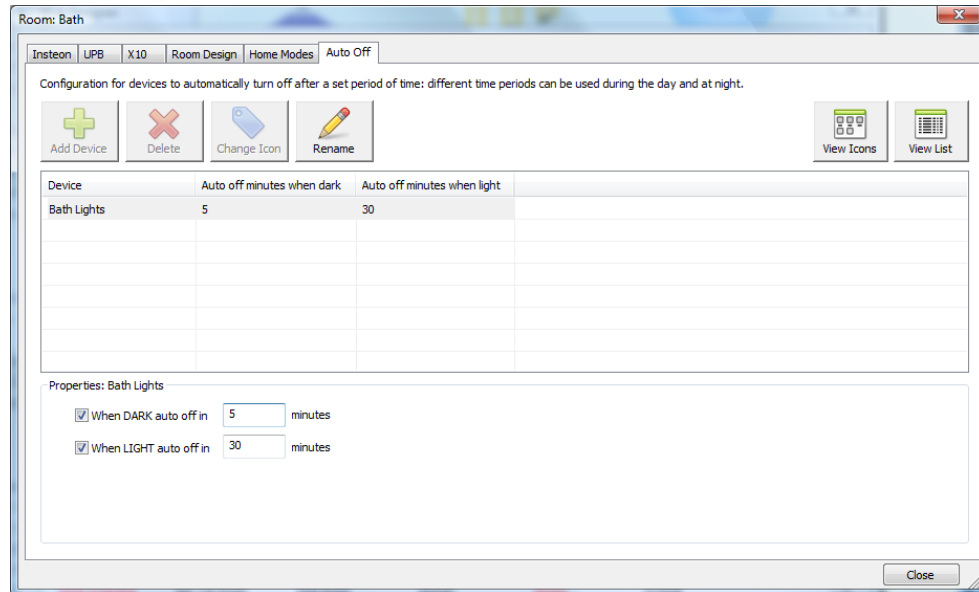
When going away the lamp is turned off. But even more importantly when you are away even if the lamp is scheduled to go on or off at some time, none of those scheduled actions happen. Why would you want to do that?

You may want to schedule the lamp to go on every day at sunset so that the "Rec Room" isn't dark. But you don't want to do that when you are away. Using home modes you can just schedule the light to come on every day and then the home mode concept suppresses that when you are away.

---

## Configuring Auto Off

Auto off for a device is exactly what it says: After some length of time after the device is turned on, it is automatically turned off. This is configured on the "Auto Off" tab for a room.



In this example, the bath lights automatically go off after 5 minutes at night and 20 minutes during the day.

One important point about this: This auto off only works if HCA controlled the device ON – from a schedule or program- or HCA receives a message that the device was locally controlled – you walked up to the switch and tapped the paddle on. Most modern devices can send messages when locally controlled but how that is done depends upon the protocol and device type.

---

## Designer Final Thoughts

The Designer is a simple way to make changes to your rooms and devices. All of what the Designer does can be done using other tools in HCA. The idea behind the designer is to bring all these tools together into one place.

You can use the designer to create a new design file but you can also use it subsequent to that to modify an existing file.

---

## Starting with a Blank or Empty file

In addition to the HCA Designer you can also start with an empty file – one that has nothing added to it.

To start like this just start HCA. If you don't load a file then the design is empty and you can start adding devices and programs.

There is one important action you should also take when starting a HCA design in this manner and that is to set the location so HCA can correctly determine sunrise and sunset.

Select from the ribbon "Design" category, "Home Configuration" panel, the "Properties" button. On the location tab you can set the location either by zip code for US locations or manual location for non-US locations in the same manner as described above when using the Designer.

## Modifying your home file

If you want to change the details of your home file you can open the properties dialog. Press the *Properties* button in the ribbon *Design* category.

From this multi-tabbed dialog you can easily change:

- The Home design name
- Location
- Light and Dark determination
- Security settings
- Customer, Installer and Notes. These are for reference only and can be used in any manner you want.

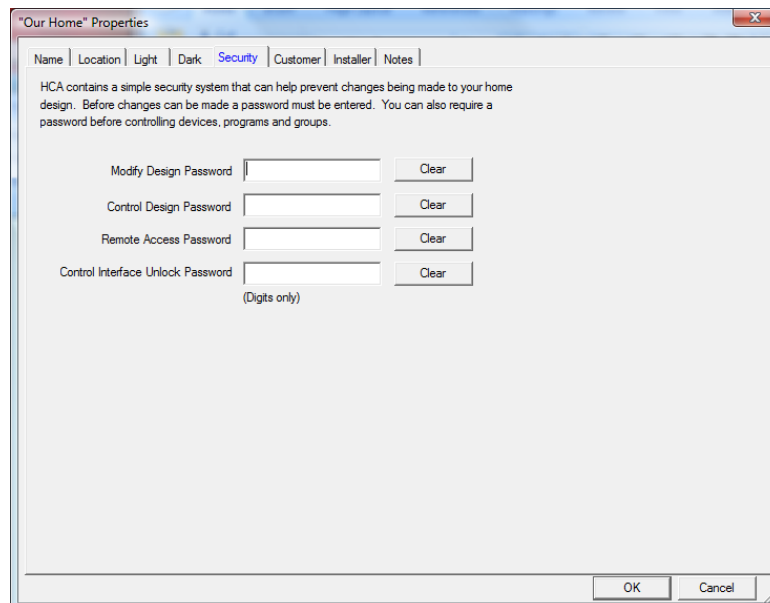
## Security settings

If you have HCA maximized on the display, with the display pane taking the entire HCA window, you have a pretty nice display. It may be nice enough so that you want to leave the HCA computer out, and not buried in the basement or in a closet. If you want to have the HCA computer “front and center” in your home, visitors or inquisitive children may feel compelled to play with it. The security system is for just this reason—to prevent unintended changes.

The options in this system can help you avoid making inadvertent changes to your design, and can guarantee that only people with the password can change programs, devices, and groups.

Remember that you need to leave the computer on all the time—it would be a shame if just anybody could walk up to the computer and start changing your design. This is when you use the home design properties to assign passwords to various actions and protect your design.

To create security settings click on the security tab.



You can use the options on this tab to set password security for four different actions: modify design, control the design, and remote access from clients or the HCA Web Component via a browser. Also you can specify the control user interface password which must be only digits.



Once set, when anyone attempts to make changes to that element of your home design, a message box appears, and the password must be provided before the properties dialog box displays.

As an example, let's say you entered a password for the design. The next time you want to make changes to the properties for this design, a password message box appears, and you need to type your password before you can change the properties.

---

## Observations about the security system

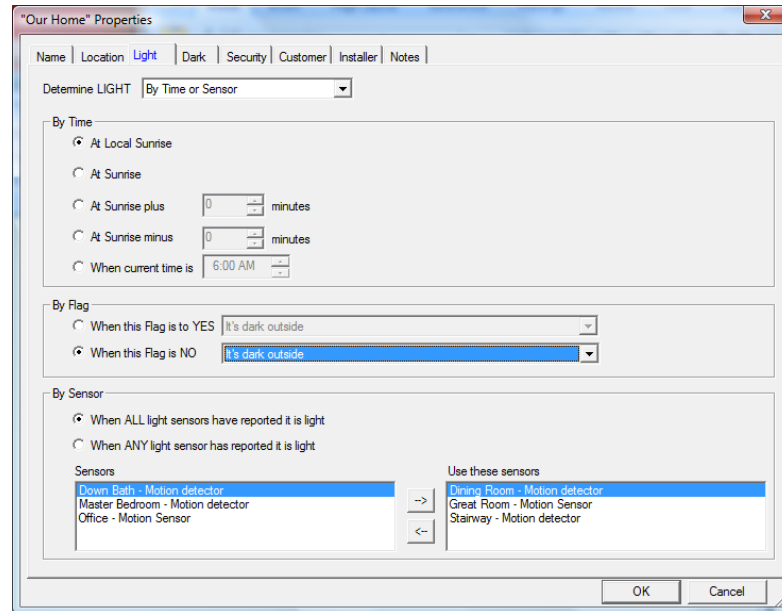
This section discusses a few points about the security system that may help you.

- The security system is only moderately secure. The file is not encrypted but the passwords are stored in the .HCA file in a hashed form. This means that once you enter a password you can't recover it if lost.
- Much of this isn't needed when using Client-Server. You can use one of the read-only clients and in that way no one can make changes to your design. In client-server the "Remote Access" password is the most important since it guards access to your home from anywhere if using the internet. Refer to the client-server user guide chapter.
- Regardless of the password settings you create, anyone can still turn off the HCA computer with the power switch, or stop the HCA task by using the Windows task manager.
- If you don't provide a password for *modify design*, anyone will be able to see the other passwords by using the Properties choice from the Home menu. Once you have set up a design password, no one can access the home design properties without first entering that password.
- If you have lost the password there is a way to load the file and remove all password protection. Follow this procedure:
  1. Select from the HCA application menu HCA – Options. On the startup tab make sure that all the options in the *When HCA Started* group are clear. That is, they are not checked.
  2. Shutdown HCA
  3. Restart HCA
  4. Select from the menu HCA – Properties. Select the Extra tab. Enter the code PSWD and close the dialog with OK
  5. Load the file that you lost the password for. During the load the password protection is removed. Also the extra feature PSWD code is removed.
  6. Save the file.

## Light and Dark

One of the important things HCA does is to control lighting. It is often useful to know if your home is light or dark to determine if lights should be turned on and at what level.

The question “Is it light or dark” can be answered by sunrise and sunset time but there are other options.



There is the same dialog for Dark and well as Light.

While this dialog has many options the general idea is that you determine light from three possible sources.

- By Time
- By the setting of a flag – determined by programs you have created
- By s selection of motion sensors which contain a light sensor

At the top of the dialog is a dropdown where you pick what sources determine light or dark and how those sources interact. In the above example, it will be considered Light at the local sunrise or when all of the selected motion sensors report that it is light.

In later chapters of this user guide you will see two places where you can make use of this information: In the auto off settings of rooms and devices and in programs where you can test the flag “\_Light” or “\_Dark”. HCA keeps those flags up to date during the day as it reevaluates the light and dark settings specified here.

**Hint:** Because there are separate criterion for Light and Dark it is possible for both to be true or both to be false. This can get confusing so there is a "Check" button at the top of the dialog that shows you the determination of light and/or dark and how HCA reaches its decision.

## Chapter 4

# Devices

A “device” is the name HCA gives to real world items – like a switch or module – that it can control.

First you “create” the device in HCA using the New Device Wizard. When you create the device in HCA, you give it a name, and other parameters. The New Device Wizard helps you through this process for each device that you want to create for your own home. Then if you need to, you can later modify the properties that you set up when you created each device.

This chapter concentrates on creating devices and how to later modify the properties for any device. Sections in this chapter include:

- About devices
- Planning your devices
- Creating a new device
- Modifying an existing device
  - Using the Options tab
- Deleting a device
- Insteon devices
- UPB devices

---

### About devices

“Device” is the term used in HCA to represent any item in your home that is controlled with power line signals -Insteon, UPB, X10, IR. A device can be a lamp or appliance, in any room or location of your house. Each device has some way to identify it from all others. For X10 devices this is by a house code and a unit code which is set in some way at the switch or module. For UPB devices this is the unit code assigned to it when it was set up. For Insteon this is the address that is assigned when it is manufactured. HCA must know the device’s address so that HCA can control it.

In HCA, a device can be independent, it can be part of a group, it can be controlled by a schedule and programs. You may want to review the chapters on groups, schedules, and programs before you begin to create devices, as how it is controlled may affect how you name it, or what notes you want to include with the device. You will also want to read the following section on planning your devices before you get down to the creation of the devices.

UPB devices are added to your design by importing a UPB network file. This is explained in the UPB appendix.

How IR devices are set up and controlled depends upon the IR interface being used. See the IR appendix for the Global Cache and Bitwise which are interfaces for controlling IR devices.

---

### Planning your devices

Before you start creating devices in HCA, it’s a good idea to consider which devices you want to create, where they are, how they work, and how they might work together. When creating devices, there are two possible scenarios; either you already have controllable switches and modules in your home, or you don’t. The way you plan for creating HCA devices differs slightly, depending on how you start.

If you already have modules and switches in your home, you need to make a list. You can check each one and write down:

- the location
- the type of light or appliance it controls
- the manufacturer and model number
- The address if it has a user settable address

If you do not yet have controllable switches and modules in your home, it may be easier to create the devices in HCA, and then print out a list of what you plan to add. You can take this list with you and use it as you setup the modules and wire in the units in different parts of your home.

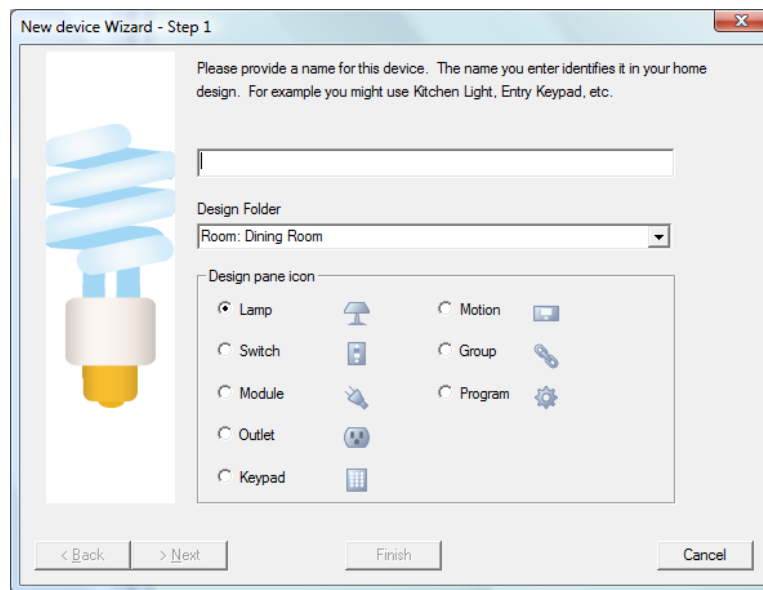
---

## Creating a new device

Now, if you know which devices you want to create, and have made an installation plan, you're ready to begin creating devices. There are three ways to start the New Device Wizard:

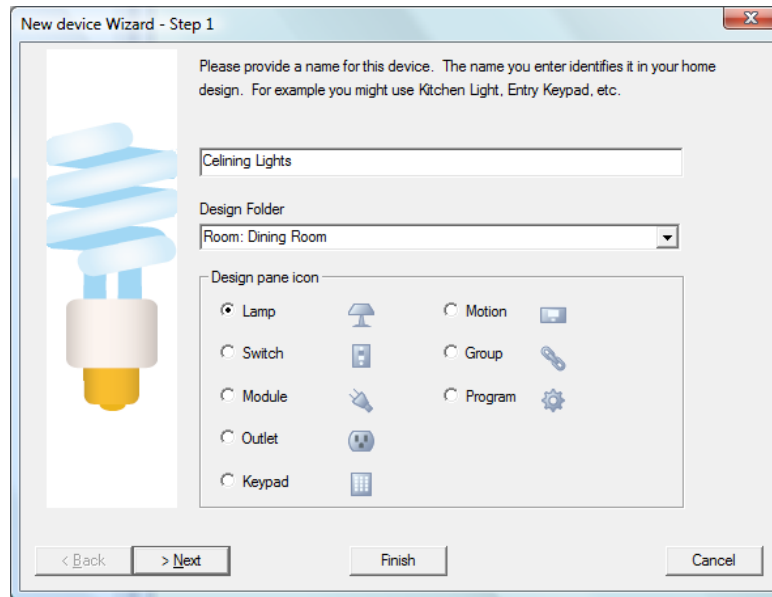
- Right click on the background of the display pane and select New – Device.
- Right click on a folder name in the design pane and select New – Device
- Press the *New Device* button in the ribbon *Design* category.

Any of these options opens the New Device Wizard.

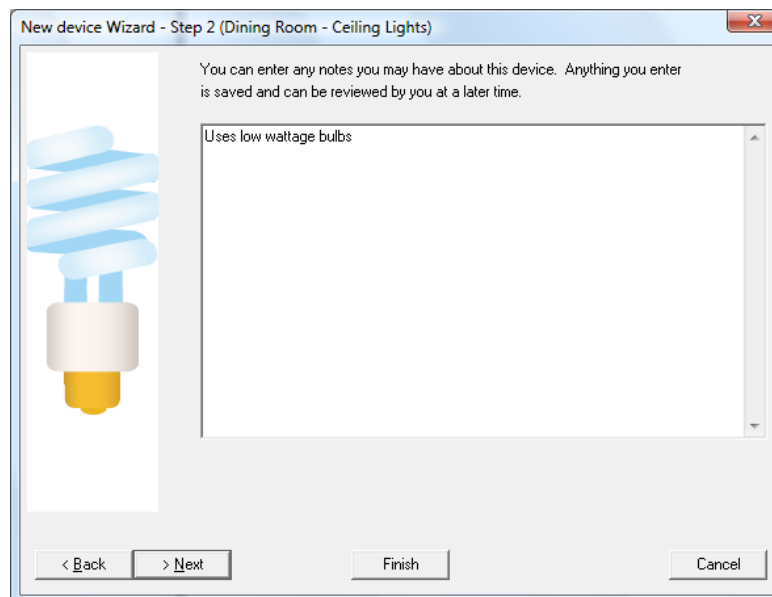


Type in a name for the device and type in or choose one of the already existing rooms or folders to store this device. The device name must be unique in the chosen folder. The name you enter can contain most characters, and can be of almost any length.

To aid in recognition, you can name devices with descriptions, or location identifiers. For instance, you could name devices: Katie's night light, dining room chandelier, or Bedroom Lamp 1, 2, and 3.

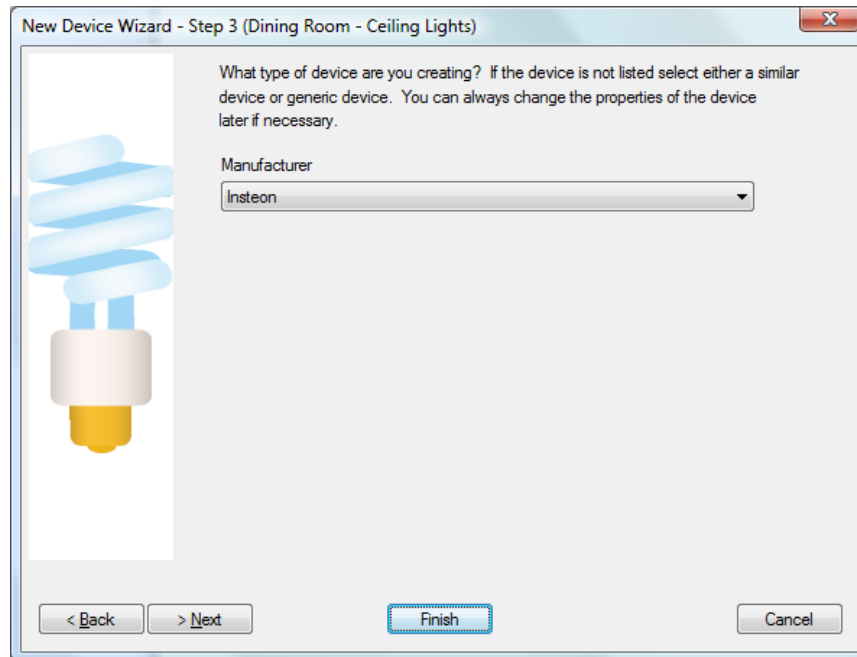


Click Next and add any notes you might want to remember about this device. The notes can tell you where the device is (hidden in the attic), what it looks like, or what socket or circuit it is connected to. You can review, and revise or delete these notes later at any time. These notes can be displayed in a popup window when the mouse is moved over the icon for this device.



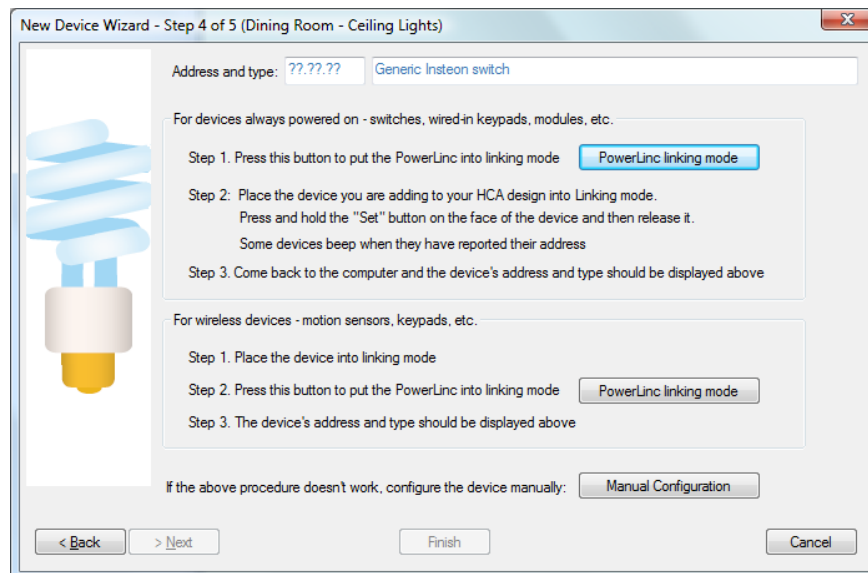
In the next step you specify what type of device being created. The choices you have depend upon the manufacturer you select. First select a manufacturer and then select the type of switch or module.

**Hint:** Don't be worried if you don't see exactly the product you have. Many times two parts will have different part number (for example a 600w or a 100w switch) but work the same way. If you don't see exactly what you want just pick something close. In most cases that will be fine.

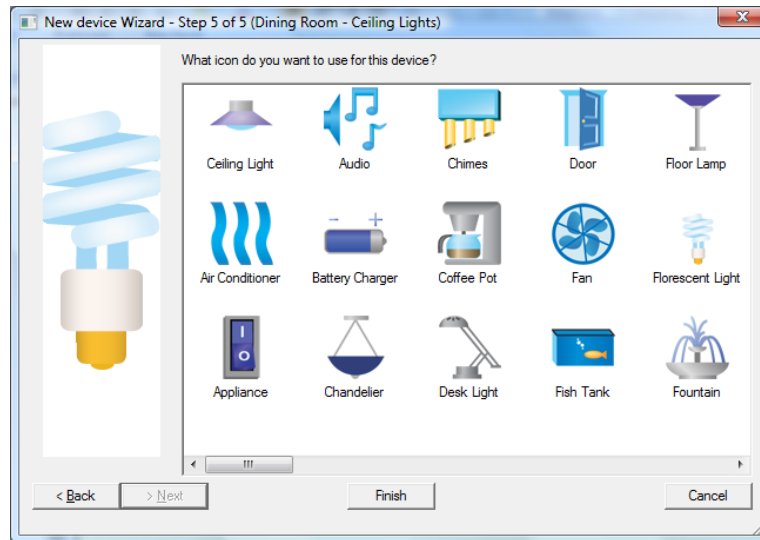


Make your selections and then click Next.

The next step in the Wizard depends upon what type of device you are creating. For an Insteon device, in this step you choose how HCA determines the device address and type. For other device types the address entry may be done in other ways.



Now you get to choose the icon that you want to represent this device on a display. The listing includes all device icons provided in the default Icon Theme. See the chapter on Icon Themes for more information.



When you first get to this step, HCA tries to choose an icon based upon the name you supplied and the type of device. Sometimes it chooses well and other times not.

Click the icon that you want to use, and click Finish as this is the last step in the wizard

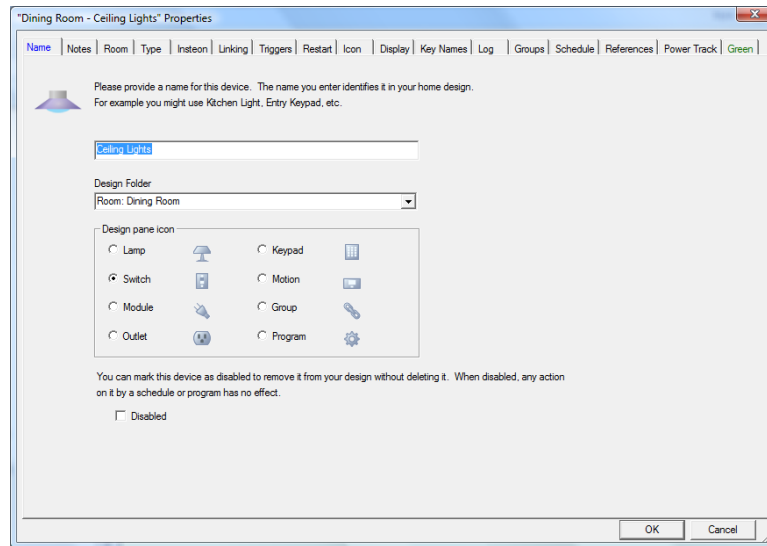
---

## Modifying an existing device

Once a device is created, you can modify any of the properties that you set in the New Device Wizard. You do this in the Properties dialog for the particular device. There are two ways that you can get to the device properties:

1. Select either:
  - the device **icon** on a display or
  - the device **name** in the design pane
2. Click the right mouse button.
3. From the popup menu, select Properties.

The Properties dialog title reflects the name of the device you selected. The Name, Notes, Type, and Icon tabs have the same information as the corresponding Wizard steps.



4. Click the tab you want to change properties on, or click each to review the settings for this device. You can change whatever you like, and click OK to save your changes.

In addition to the tabs that are similar to the Wizard steps there are many others. Some of these are specific to the type of device and others always appear.

---

## Triggers Tab

Each device has a single address that it responds to. For Insteon it is set at the factory. For UPB it is automatically selected when added to a UPB network. For X10 devices this is something you choose. But the common thing is that each device has exactly one address and this can be a problem.

You may have a situation where you would like to give a device multiple addresses. Perhaps even a different address on a different protocol. For example, to control an X10 device with a UPB keypad.

This is what the triggers tab is for. This can be a complex topic so an example with some simpler X10 devices may help make the concept clearer but the idea is the same with all protocols.

Suppose your home contains a device called “Den lamp” and is assigned an address of B1. Also imagine that you have a keypad someplace in your home set to housecode B. You can use this keypad to control this lamp.

But what about other keypads in your home? Perhaps you have a wireless keypad that you want to be able to take throughout the house and be able to control devices in various rooms. For that to work you would need to set the wireless receiver to some housecode and have all the devices you want to control set to the same housecode. This may be ok, but maybe not.

This is where triggers come in handy. Suppose that wireless keypad is set to housecode K and you want to use K3 to turn on and off the den lamp. If you add a trigger of K3 to the den lamp device then here is what happens.

1. You press the K3 On button on the wireless keypad. It is received by the wireless transceiver, put on the powerline and then received by HCA.
2. HCA finds that K3 is a trigger for the Den lamp so it sends an On command to B1.
3. The light comes on.

If you had sent a K3 Off command, HCA would have sent out B1 Off.



You can create any trigger you want – from any protocol – and if received by HCA it will determine the command associated with the reception – on, off, dim, bright, etc – and retransmit that to the device using the device’s address and the command – translated across protocols if necessary.

To remove a trigger for a device, right-click on the trigger and select *Delete* from the popup menu.

To change a trigger for a device, right-click – or double-click – on the trigger and select *Edit* from the popup menu.

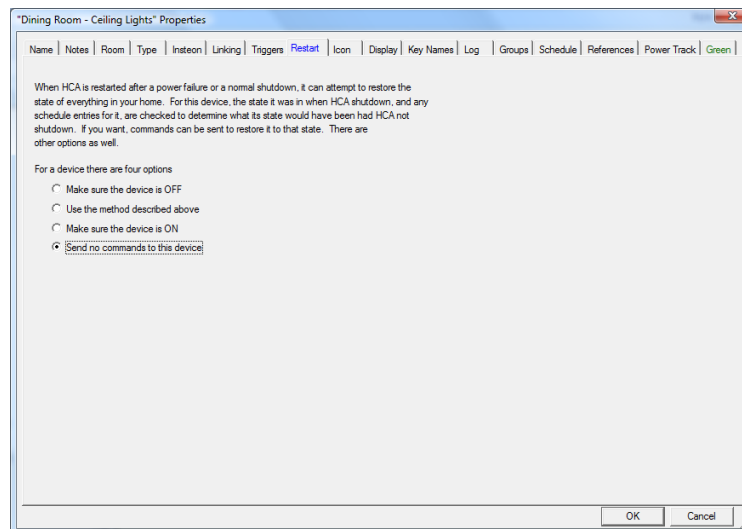
---

## Restart Tab

On this tab you determine what happens to this device when HCA is started. You should carefully consider the selection for this step, in light of the device that you are creating.

- If you have an appliance that is best not left unattended (one that generates heat), the best choice is "Make sure the device is OFF".
- The device can be set to follow the schedule you have for the device,
- Only in very special circumstances should you ever choose *Make sure the device is ON*, and then only for devices like lights.

For some devices, like the X10 chime, you may want to have HCA send no commands.



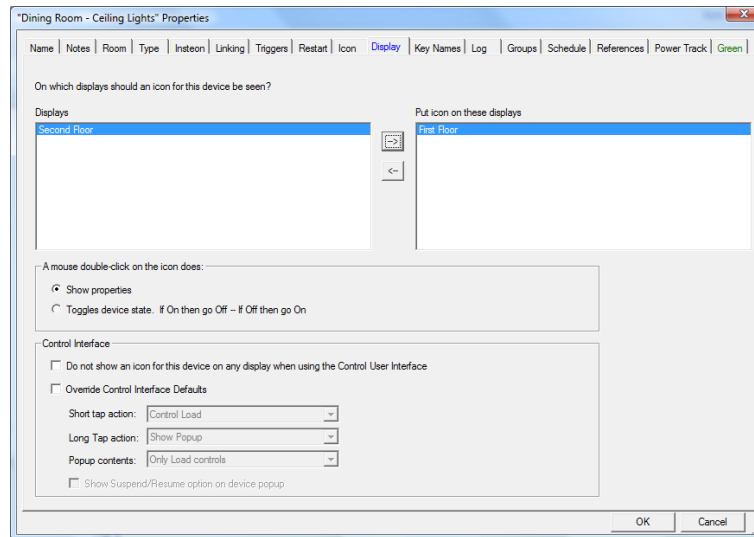

---

## Display Tab

While an icon for this device always appears in the display for the folder than contains it, you can, if you want, place an icon for it on other displays as well.

On this tab you can add an icon for the device onto one or more displays. Click the display where you want the icon to appear, click the right arrow →, and the display appears in the right list. You can move one, two, or all of your displays to the right column. If you find that you’ve listed a display in the right column mistakenly, you can select it, and click the left arrow ← to return it to the left column.

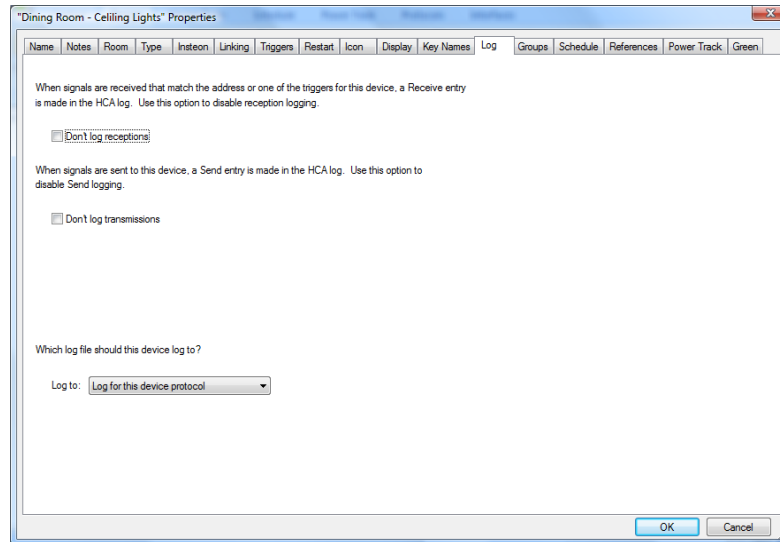
To move the icon to where you want it on the display, select it with the pointer, click the mouse, and drag it to its intended location. When you release the left mouse button, the icon is dropped at that location. If the Auto Arrange option is enabled, you will be limited in where you can drop the icon.



The other sections of this dialog are explained in the chapter on the Control User Interface.

## Log Tab

The Log tab contains the log for this device. Normally a log entry is added whenever a reception from this device is received or HCA transmits to this device. You may not want to log some devices that generate a lot of traffic.



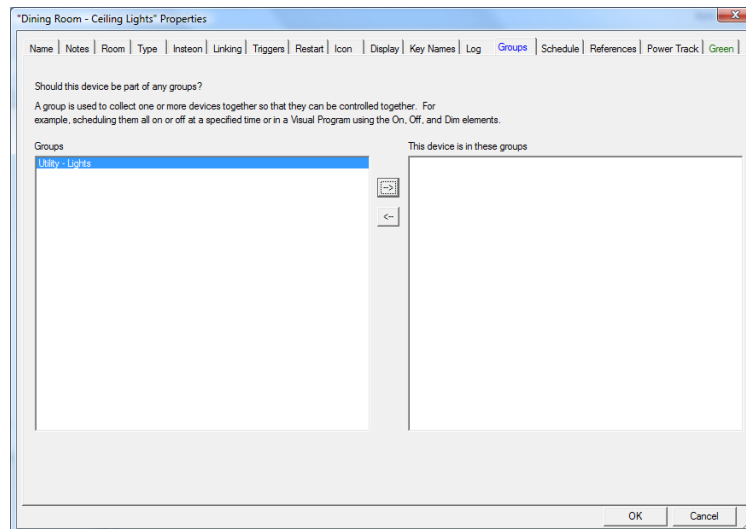
Transmissions and receptions for a device can create entries in the log used for the device's protocol or to a specific log. Each interface can route log entries for the protocols it supports to one of the three logs. This is specified in the interface configuration options in *HCA Options*.

---

## Groups Tab

On this tab you can add this device to one or more groups. Groups are more fully covered in their own chapter but, in general, a Group is a set of devices that HCA controls as a unit. If you have a group with three devices, then you can control that group with a program or schedule and all three devices are affected.

You can add devices to groups using the properties for the group but you can also add or subtract devices from groups using this tab.



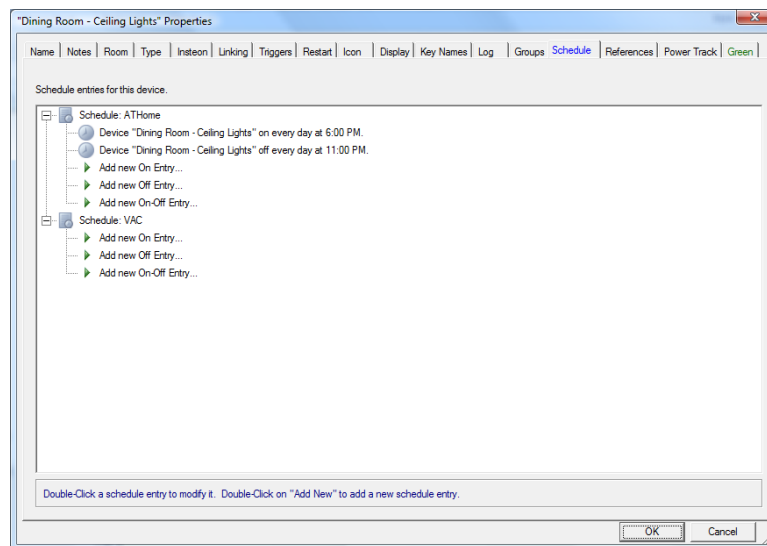
---

## Schedule Tab

HCA contains many different scheduling tools and these are covered in later chapters. In addition to those tools, you can add or remove a schedule entry for a device using this tab. To add a new schedule entry for this device to a schedule, double-click on the “Add new” branch below the schedule. You can add a schedule entry that makes this device go on, go off, or go on then off at a later time.

To remove a schedule entry for a device, right-click on the schedule entry and select *Delete* from the popup menu.

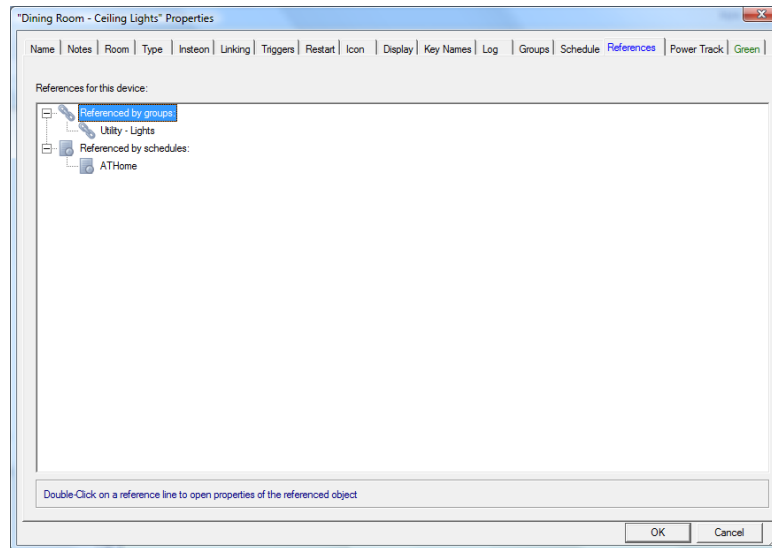
To change a schedule entry for a device, right-click – or double-click – on the schedule entry and select *Edit* from the popup menu.



## References Tab

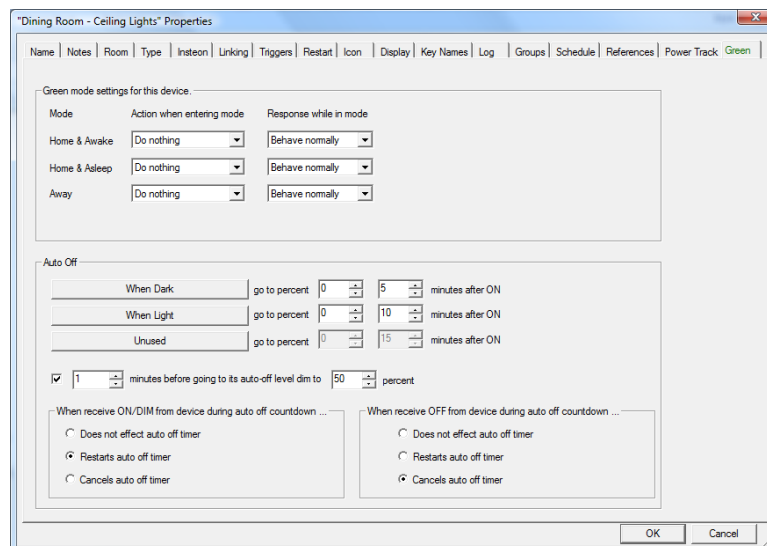
The references tab brings together into one place a report on the use of this device in your design. Devices can be used in many places: schedules, programs, groups, protocol bridges, etc.

This provides a quick way to tell where this device is used in case your design becomes complex.



## Green Tab

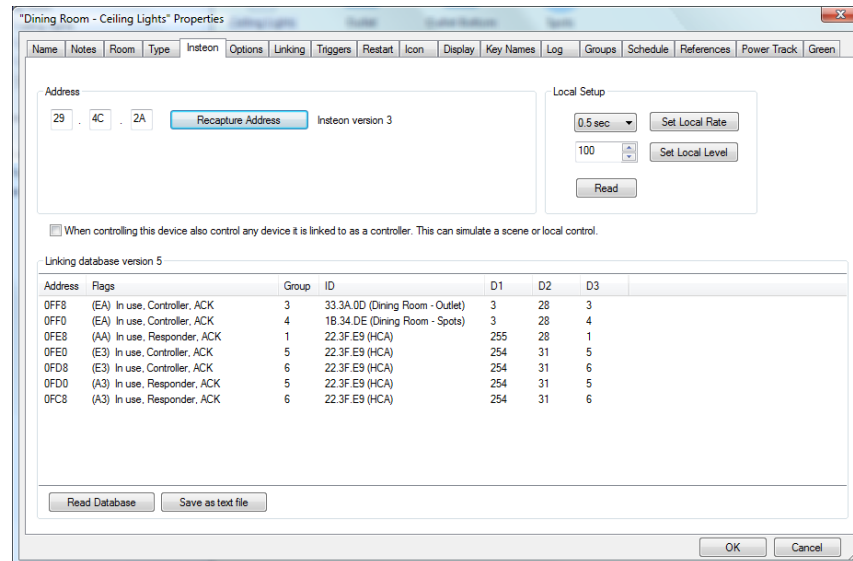
The Green Tab is where a device is configured in its respond to what the current home mode is. This topic is full cover in the *Home Modes* chapter.



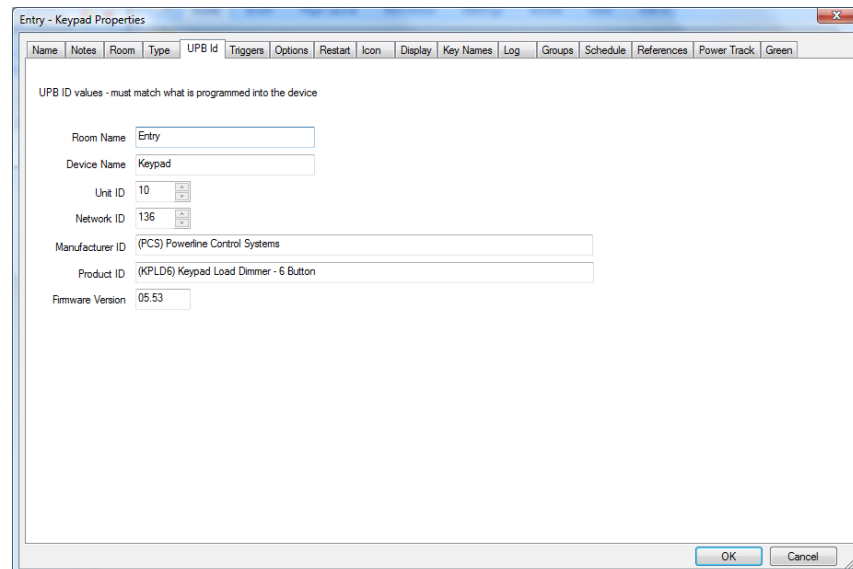
## Protocol specific tabs

Depending upon what protocol the device supports you may see additional tabs. These tabs configure or examine options specific to that protocol. For example for Insteon devices you will see an Insteon tab, for UPB devices a UPB tab.

For example, for this Insteon device, the Insteon tab shows the address and the linking table.



For a UPB device:



For older style X10 devices there are lots of configurations options since each device type may operate differently from different manufacturers.

## Deleting a device

If you find that you have created an incorrect device, or if you get rid of an existing appliance, you have two options.

- You can modify the properties for the device. For instance, changing the name and notes (but retaining the house and unit codes) of a lamp that has been replaced with a newer model, or changing the location and address (house and unit codes) of a lamp that you've moved.
- You can delete the entire device.

To delete a device:

1. Use the right mouse button to click the device name in the design pane of HCA.
2. Click Delete.

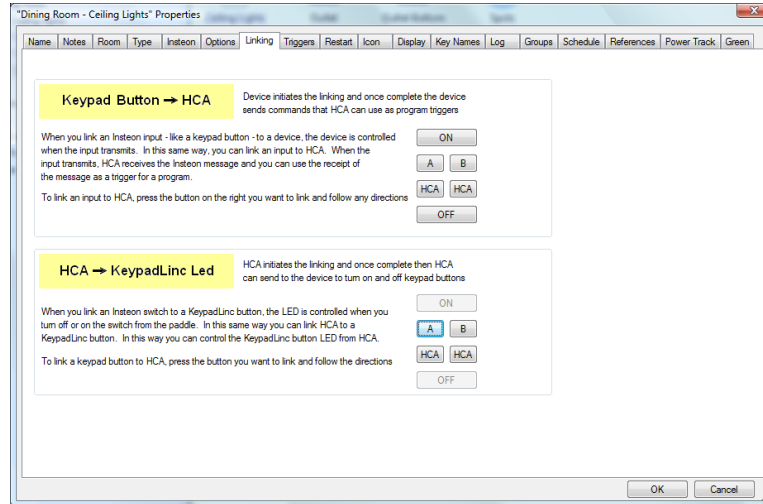
HCA removes the device name from the design pane, and removes the icon from any displays it appears on.

**Warning:** Be sure to check all aspects of a device before you delete it. If you delete a device, it is gone, any group it was a part of still works, and any schedule it was listed in is still valid, **but** any program it was a part of will not start.

**Hint:** However, you can use the Undo command on the Edit menu to restore the deleted device. It will undo just the last change you have made—so if you delete the device for a toaster and then a lamp, you can undo deleting the lamp, but not the toaster.

## Insteon Devices

Once an Insteon device has been added to your design, when you open its properties dialog Insteon devices have a *Linking* tab.



This tab is fully described in the Insteon appendix.



## UPB Devices

UPB Devices are **not** added to your design using the New Device Wizard. They are imported from a UPB Network definition file.

Once added, they do have properties like all other devices. On the UPB Id tab is the information about the UPB device as read from its setup memory.

Library - Keypad Properties

Name | Notes | Room | Type | UPB Id | Triggers | Options | Restart | Icon | Display | Key Names | Log | Groups | Schedule | References | Power Track | Green

UPB ID values - must match what is programmed into the device

Room Name: Library

Device Name: Keypad

Unit ID: 46

Network ID: 136

Manufacturer ID: (PCS) Powerline Control Systems

Product ID: (KPLD6) Keypad Load Dimmer - 6 Button

Firmware Version: 05.53

OK Cancel

The Options tab for UPB devices is described in the UPB Appendix

Library - Keypad Properties

Name | Notes | Room | Type | UPB Id | Triggers | Options | Restart | Icon | Display | Key Names | Log | Groups | Schedule | References | Power Track | Green

Transmission

Confirm receipt of commands sent by ACK      Transmit count: Send twice

Confirm receipt of commands sent by Status Request

Level and Ramp rate

Select defaults on level and ramp rate used when controlling a device from the UI and from a schedule. These could be the same defaults stored in the device but they need not be.

When using the Visual Programmer you can choose any of the available levels and rates. What is chosen here applies only to schedules and the User Interface.

When controlled ON use the last light level

When controlled ON use this light level: 100

When changing light levels use the default fade rate

When changing light levels use this fade rate: Snap!

To better work with linked keypad indicators, control this device ON using this link: Do not control by link

... and OFF using this link: Do not control by link

Icon

When any command from this device is received momentarily show the icon background as ON then OFF

Show the icon background based upon the last command received

Show the state of the attached load for devices that directly control a load

OK Cancel

---

## Final device topics

There are two additional topics to discuss with devices.

### Icon placement

As part of the device properties dialog you can select one or more displays to show an icon for the device. If you are using a display with a DXF or picture background you may want to show more than one icon for the device on the display. Perhaps this will make a more realistic appearing floor plan.

To place more than one icon for a device on a display, right-click on the single icon placed by the wizard and select *Icon* and *Add* from the menu. At this point you can select the icon picture to use. An additional icon for the device is created and placed on the display. You can then drag it to whatever place you want.

To remove an icon, right-click on the icon and select *Icon* and *Remove* from the menu. The icon is removed from the display.

To change the picture of the icon, right-click on the icon and select *Icon* and *Change* from the menu.

### Retrieving the status of a device

If a device supports status reporting you can have HCA request its status by using the right mouse button to click the device name in the design pane or on an icon in the display pane and select *Get Status*.

### Locally Stored Scenes

If the device supports scenes (illumination level and in some cases the rate at which the light changes up or down to that level), any scenes that you have identified to HCA are listed in the popup menu. If you select one of those scene names, the command to set the switch to that scene is sent.

HCA contains features from programming scenes in some device types. For example, Insteon device scenes can be created by using the *Visual Scene Editor*. This is described in the *Insteon* appendix.

## Chapter 5

# Home Modes

Home Modes are a very simple yet powerful system to control devices in your home as your home goes through its day.

Let's assume a normal day to see what you do and how your home can respond.

Midnight	Asleep
6am	You wake up
7:30 am	You leave home
6:30 pm	You return home
10:30 pm	You go to bed

OK, so that's the schedule now let's see how your devices should respond.

### TV

Most televisions these days are never really off. They are in a lower power mode when they appear off but still are using power. This makes it possible for remote controls to turn them on. When you are not at home or asleep the TV might as well be truly powered off. You can do this by plugging it into a controllable outlet.

### Battery chargers

Anything with a "power brick" uses power even if nothing is attached to it. When you are not at home it should be off. When you are home – even when you are asleep – it should remain powered on so things can charge. Again, a controllable outlet can be used here.

### Interior lighting

Interior lighting should be off when you are not at home and should go off when you leave or when you go to bed. While you are home, nothing should be automated.

### Circulation Pump

There is no point in having hot water distributed through your home when you are away but should come on when you get up and when you return home.

These are just some of the devices in your home that can be controlled based upon what *Mode* your home is in. The intent here is to have off what should be off when it should be off. This results in energy savings and reduced electricity use.

Some very simple options in the properties for your devices can make all this happen.

HCA supports this up to four modes. Three of these are defined by default:

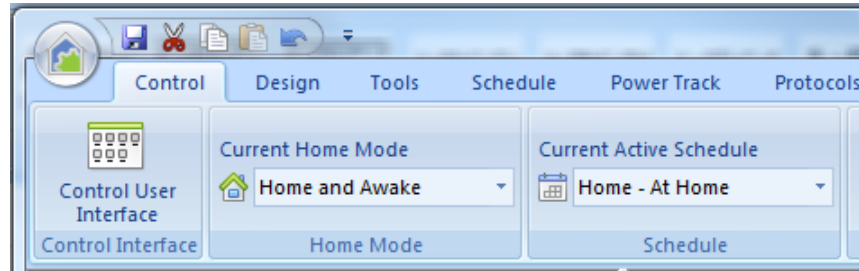
- Home & Awake
- Home & Asleep
- Away

You can change the names of the modes, remove the pre-defined ones or add a 4th.

The remainder of this chapter will show you how to set your home mode and how to manage devices based upon it.

## Home Mode

HCA displays the current mode in the ribbon *Control* category.

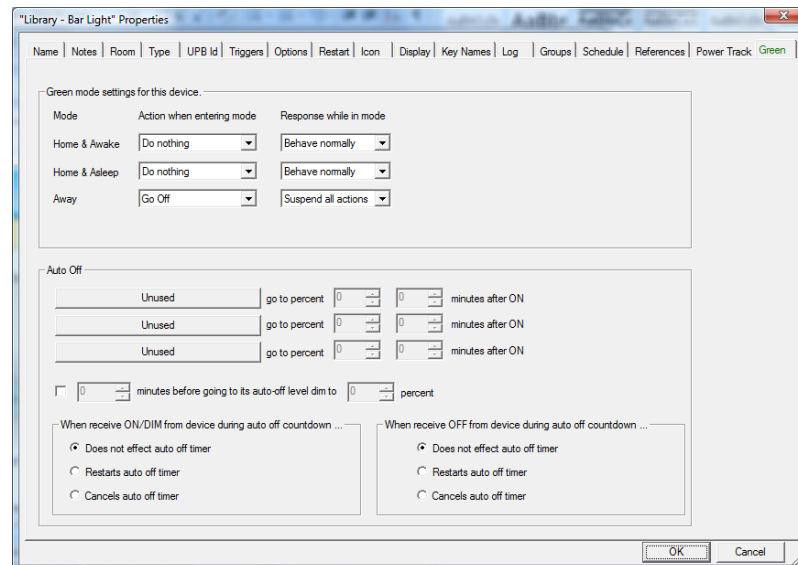


In this portion of a screen image, the current mode is Home & Awake.

To change the current mode, change the selection in the dropdown.

## Configuring devices for modes

To configure devices for what actions are taken when the mode changes and how they respond while in the mode, open the device's properties and select the Green tab.



There are two settings for each mode: “Action when entering mode” and “Response while in mode”. It’s important to understand the distinction. The “Action when entering mode” only happens when HCA shifts from mode to mode. If the current mode is *Away* and is in that mode for days, nothing happens. But as soon as the house changes from *Away* to *Home & Awake*, then each device is examined to see what should happen when entering the *Home & Awake* mode.

The choices for what happens upon entering a mode are:

- Do Nothing
- Go Off
- Go On

Let’s first examine the options for what happens when a mode is entered by looking at the examples given at the start of this chapter.

In this table each device shows what action should be taken when your home enters the mode.

Device	Home & Awake	Away	Home & Asleep
TV	Go On	Go Off	Go Off
Charger	Go On	Go Off	Go On
Lighting	Do Nothing	Go Off	Do Nothing
Pump	Go On	Go Off	Go Off

Focus first on the “Away” column. When you tell your home that you are leaving everything goes off. That’s good as it says a lot of energy.

When you return home, or when you wake up in the morning, your home goes into the Home & Awake mode. The TV is powered on so you can use your remote to actually turn it on – the picture doesn’t appear because we are just powering on the outlet it is plugged in to. The circulation pump goes on to make sure hot water is distributed. Nothing is done with Interior lighting – you turn on what you want when you want it.

When you go to bed, energy is again saved by shutting lots of things off, except for the mobile and tablet chargers which stay powered on so they can charge at night.

This is quite a feature! With just a few checkboxes you can control every device in your home to respond to the most common actions of your day.

We will cover the second set of options – what happens to the device while in the mode - a bit later in this chapter.

## Auto Off

Also on the Green Settings tab are some options to automatically turn a device off after HCA sees it on for a number of minutes. Additionally, you can also configure it to dim ‘n’ minutes before it goes off.

For this to work, HCA has to know the device is on. This could be because HCA itself turned it on or, more usually, the device transmitted when you manually controlled it – like tapping the paddle ON for a switch. UPB devices can be configured to report state upon local control. Insteon devices can be linked to HCA so when they are locally controlled a message is sent to HCA in the same way that you can link Insteon devices together.

For UPB Devices you must use the UPB configuration program to enable the state reporting option.

For Insteon you can use the Visual Scene Editor or the Link button on the linking tab to link the device to HCA so HCA knows when it is locally controlled.

Auto off is more fully covered in the chapter on *Rooms*.

## Mode Change Triggers

While the actions upon entering each mode, as described above, are a powerful tool, how does HCA know when you come home, leave home, get up and go to bed?

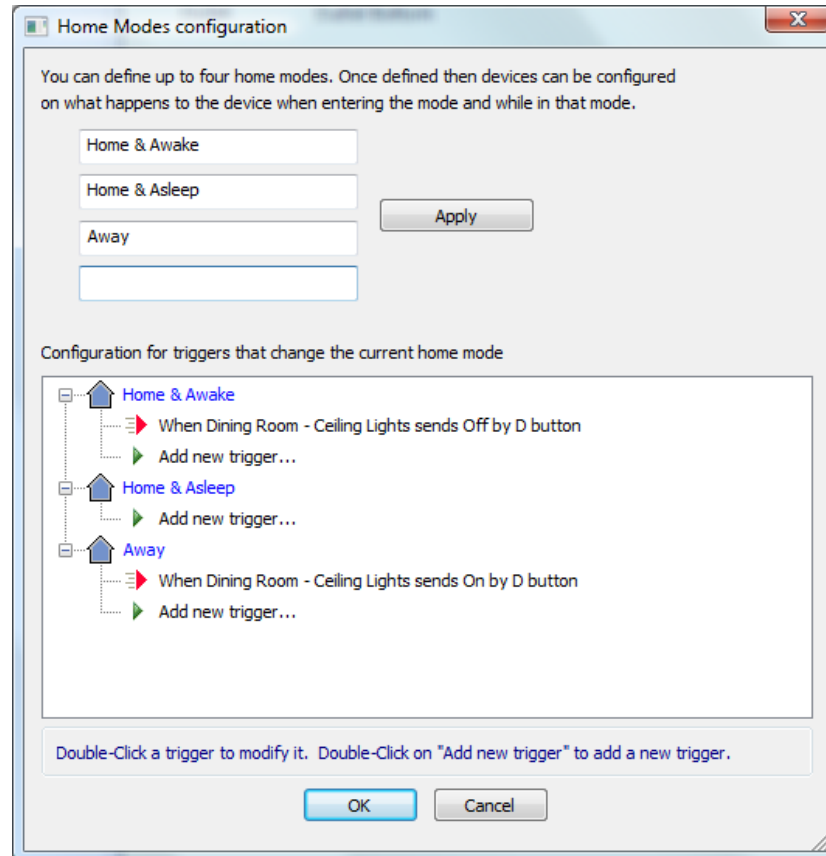
That’s up to you. A common method is to have a keypad that you designate for this. A keypad by the door where you press one button when you leave home and another when you come home. A keypad in the bedroom that you press a “Go to bed” button at night and an “I’m up” button when you get up.

But it doesn't have to be that way. There are many different methods. You could use a motion sensor that when triggered switches the home into "Home & Awake" mode.

In fact, all the features of HCA triggers can be used for this. Triggers, more fully covered in the User Guide Chapter on Programs, is a method for you to describe the contents of a message reception from the powerline, wireless, IR, or other methods. Each time HCA receives a message it matches it up with all the triggers in the system and when it finds a match it responds. This could be to start a program, or to change home modes.

You may want to skip ahead a bit in the User Guide and review the Programs chapter for a lot of information on triggers then come back here.

Press the *Home Modes* button in the ribbon *Design* category.



This is also the place where you can define or change the home modes. Just edit the mode names and press the Apply button.

For each mode you can provide one or more triggers that tell HCA the home is now in that mode.

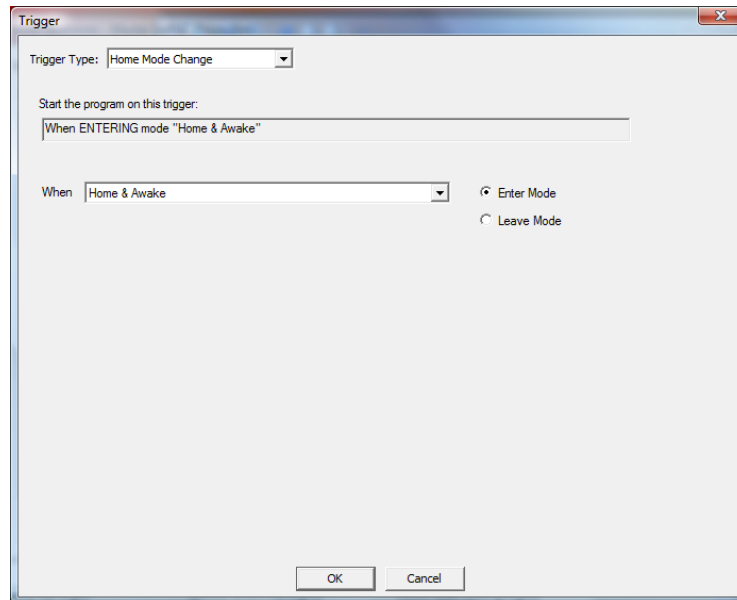
To create a new trigger, just double-click on the "Add new trigger". In this example, a UPB keypad in the dining room has been used to switch from home mode to away mode.

Pressing the 'D' button when leaving has the button LED show yellow – meaning no one is home. When returning home press the button again and you are home.

A similar keypad could be used in the bedroom to have a button for go-to-bed and get-up.

What if you want to do something more complex? You can always create a program with whatever triggers you want and in that program perform any tests you want and use the *Set Home Mode* program element to change the mode.

For even more control there are program triggers that can be used to start a program when a specified mode is entered or left.




---

## Actions while in a mode

In the device properties tab shown above, there was a second column of configuration parameters: Response while in mode.

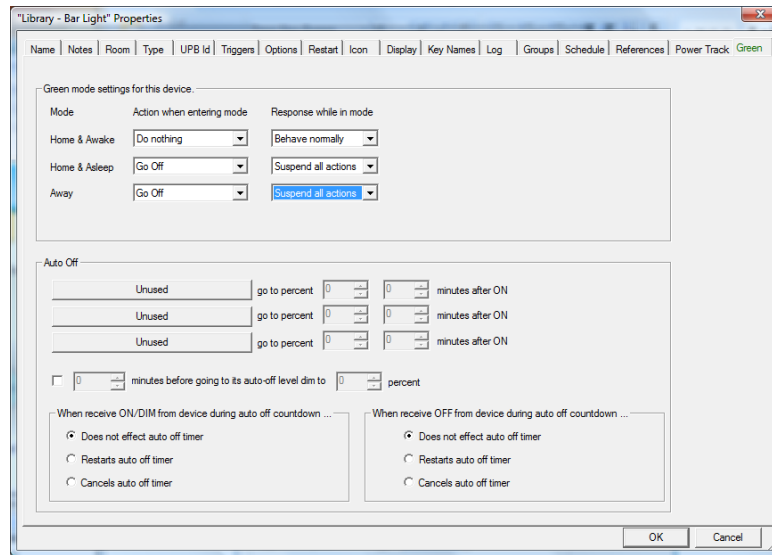
The purpose of this will become clearer when you review subsequent chapters of this User Guide - when you read the chapters on Schedules and Programs but we will press on.

To look into these settings, let's take another example. Suppose that every day you would like some lights to come on automatically at dusk. That's easy - you create a simple schedule that controls those lights at that time. The problem is you don't want it to happen when you are not home.

There are many ways to solve this problem. One method would be to have two different schedules – one for when you are home and one for when you are away. Another method would be to have a program control the lights and in that program make a test if you are home or not. Then schedule the program to run every day at dusk.

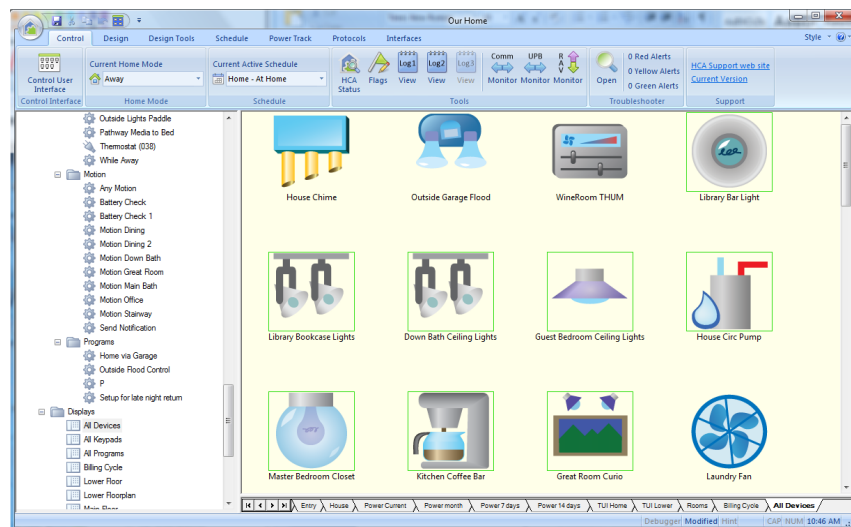
With the Home Mode concept it is much simpler. All you need to is to schedule the lights to come on everyday at dusk. That's all!

On the properties for those interior lights, mark them like this:



When a device is marked as “suspend all actions” for *response while in mode*, as long as your home is in that mode that device will not be controlled by a schedule or any programs regardless of what the schedule says or the programs does. This is why you can schedule the lights to come on every day. The home mode determines if the schedule entry has an effect or not.

When you home is in a mode and a device’s actions are suspended, in the design pane the device is marked with a slash through it, and in the display pane shown with a green box.



## Putting it all together

Now that we have covered the pieces of Home Mode, you may want to consider all that you can do with just home modes. While the powerful tools of scheduling and programs to be covered in later chapters can do much with your home design, just using modes can affect many actions without them.

The four example devices – TV, Chargers, Interior Lights, and Circulation Pump, can be configured to act as you need just using home modes. Add a few simple schedules and you have a complex automation design.



## Chapter 6

# Rooms

Each device, program, and group in your design is stored in a folder. That folder can be a “folder” or a “room”.

A folder is simply for organizational purposes. It allows you to have a device, for example, call “Lamp” in one folder and a different device called “Lamp” in another folder. That is really all a folder does.

A room is in many ways similar to a folder except it comes with extra meaning. In many ways a room is similar to a device in that:

- It can be on or off. It can be scheduled.
- It can be controlled by a program.
- An auto-off condition can be established for it.
- It can participate in a power track graph.

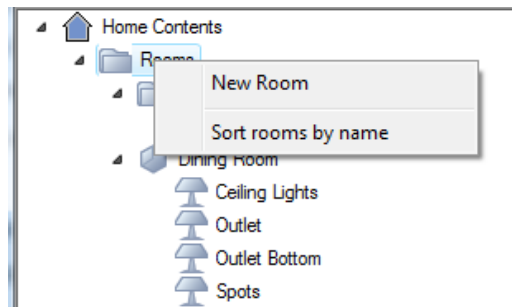
This chapter describes how Rooms work in HCA and includes these topics:

- How rooms are created
- What circumstances cause a room to be ON or OFF.
- Working with rooms: Turning on and off, scheduling, using in programs
- Device interaction with rooms

---

### How rooms are created

A room is created using a similar mechanism as a folder is created. Right-click on the Rooms root in the design pane and select New Room from the popup menu.



The room is added then to the design with the default name “New Room” and you can then change the name to something more meaningful.

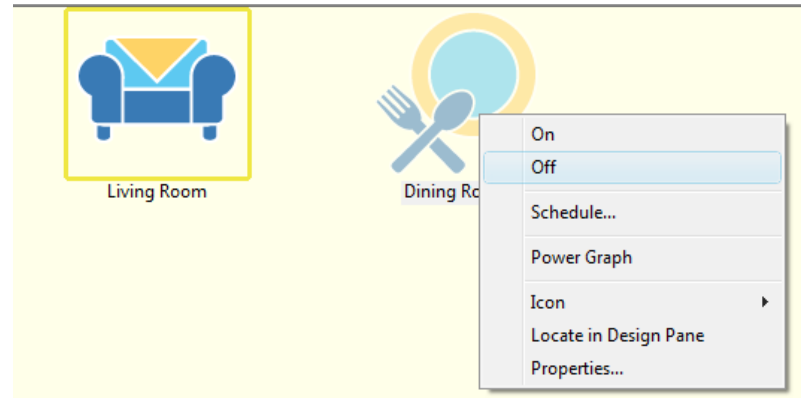
You can also create a new room when you add a new device, program, or group to your design. In the first wizard step you can choose an existing room or folder name or you can enter a new name. If you enter a new name then a room with that name is created.

**Hint:** If you have a folder, perhaps from a design created using a previous version of HCA, and you would like it to work as a room, you can drag the folder icon in the design pane and drop it into the room section. It then becomes a room.

## Rooms on and off

A room, like a device, can be ON or OFF. To turn a room on or off, you can right-click the room name in the design pane and select On or Off from the popup menu. When the room is On it shows with a yellow background in the design pane.

If you have an icon for a room on a display you can right click the icon to access these menu choices:

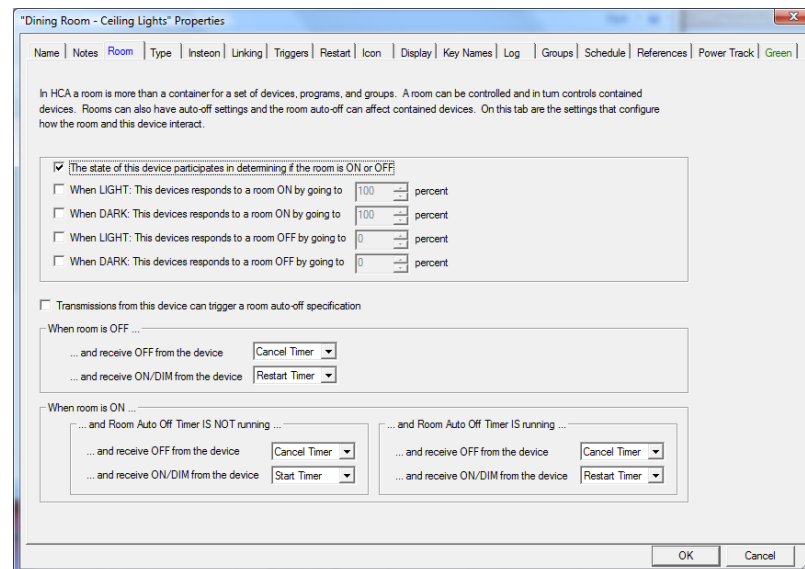


But what does it mean for the room to be on or off?

A room is ON when **any** of the devices in the room that participate in determining the room state are on.

A room is OFF when **all** of the devices in the room that participate in determining the room state are off.

What determines if a device “participates in determine the room state”? It is a property of the device. One of the tabs in the property dialog for a device is the *Room* tab.



There are a lot of options on this dialog and they will all be covered in the chapter. For now only the first checkbox is important. If that checkbox is ticked then the device participates in determining the room state as described above.

## Working with rooms

Like a device, a room can be scheduled to go on or off at a certain time or, in a program it can be controlled by the On and Off elements.

What happens when a room goes ON or OFF as the result of a program, schedule, or your interaction with the user interface?

What happens is that any device in the room that is configured to go on or off with the room is sent on or off commands.

In the device's property dialog on the room tab are four checkboxes that control this. The picture above shows these checkbox options.

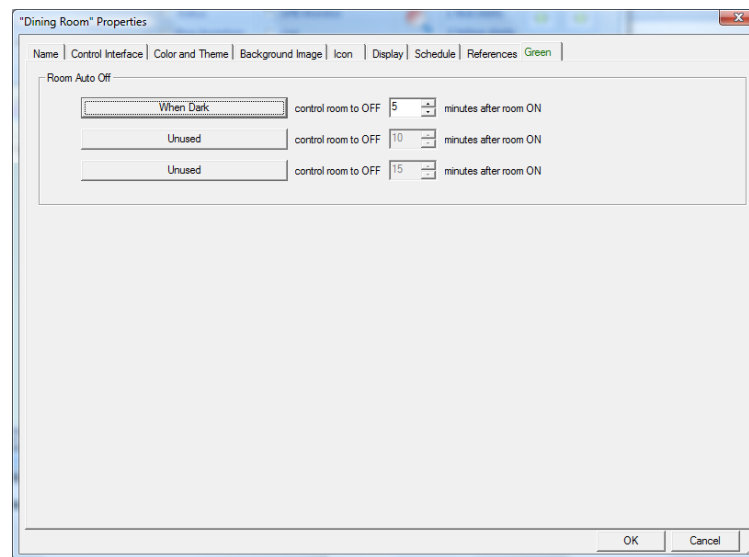
As explained in the *Your Home* chapter, HCA has a concept of Light and Dark - that chapter describes all the different ways that can be determined by time, astronomical time, or by one or more light sensors.

When a room goes on or off, you select in the device's properties how the device responds. And this can be different depending if the room is going on or off, and if it is light or dark. For each of the four conditions you specify if the device responds – tick the box if it should – and how it responds – choose a level.

**Hint:** It can be very useful to set different light levels for at night when you want the lights to go on at a lower level than during the daytime.

## Auto Off

A room, like a device can have an auto-off mechanism. This is configured on the Room properties Green tab:



In the same way that a device auto-off is configured, you can have up to three different auto-off specifications. In the above example, 5 minutes after the room is tuned on – when it's dark – the room goes off. This means that any devices in the room configured to respond when the room goes off, are sent a command to go off.

**Hint:** Even if the devices in the room have their own auto-off specifications, the room auto-off operates independently of device auto-off specifications. This can get confusing so be careful.

---

## Device Interaction with rooms

Devices and rooms can interact in a relationship that allows you to build complex actions with only the settings in the device and the room. Here is a simple example.

Suppose you have a room with two lights and a motion sensor.

The motion sensor is configured to participate in the room state.

One device is configured to turn on when dark at 50% and the other is configured to turn on when dark at 80%. Neither device is configured to participate in the room state nor are they configured to turn off when the room goes off.

Here is a time line of what happens:

1. The motion sensor detects motion so it sends a command that HCA receives.
2. Since the motion sensor is now on, the room becomes on.
3. Since the room has become on, the two lights are sent commands to turn on to the levels chosen and that depends on if it is light or dark.

And all of this is accomplished without any additional work on your part except configuring the devices and the room.

You can also configure the room with an auto-off so that the devices turn off after a preset time.

---

## Device Auto Off

One problem with auto-off is that it can be very unyielding at times. Yes you want an auto off to happen but sometimes you really wish it could be controlled.

Devices and rooms interact to make this possible. As part of the device properties you can configure how auto off can be controlled. This is best shown by some examples.

### Example 1

\*Suppose a room contains a motion sensor and a switch. Here is what we want to have happen:

- You walk into the room. The motion sensor turns on the light and starts an auto-off timer.
- If you leave the room then when the timer expires the light goes out.
- If you stay in the room then each time the motion sensor sees you the timer is restarted
- If you change the light level using the switch, the timer is canceled and not restarted until the room is off and the motion sensor again sees something.

Here is the configuration for the motion sensor and the switch:

## Motion sensor settings

The state of this device participates in determining if the room is ON or OFF

When LIGHT: This device responds to a room ON by going to  percent

When DARK: This device responds to a room ON by going to  percent

When LIGHT: This device responds to a room OFF by going to  percent

When DARK: This device responds to a room OFF by going to  percent

Transmissions from this device can trigger a room auto-off specification

When room is OFF ...

... and receive OFF from the device

... and receive ON/DIM from the device

When room is ON ...

... and Room Auto Off Timer IS NOT running ...

... and receive OFF from the device

... and receive ON/DIM from the device

... and Room Auto Off Timer IS running ...

... and receive OFF from the device

... and receive ON/DIM from the device

## Switch settings

The state of this device participates in determining if the room is ON or OFF

When LIGHT: This device responds to a room ON by going to  percent

When DARK: This device responds to a room ON by going to  percent

When LIGHT: This device responds to a room OFF by going to  percent

When DARK: This device responds to a room OFF by going to  percent

Transmissions from this device can trigger a room auto-off specification

When room is OFF ...

... and receive OFF from the device

... and receive ON/DIM from the device

When room is ON ...

... and Room Auto Off Timer IS NOT running ...

... and receive OFF from the device

... and receive ON/DIM from the device

... and Room Auto Off Timer IS running ...

... and receive OFF from the device

... and receive ON/DIM from the device

Here is a timeline of what happens:

- The room is off.
- The motion sensor sees you and starts the timer. This is because the motion sensor has the setting enabled to trigger an auto-off specification and that specification says when the room is off and an ON command is received, the timer is started.
- Since the room goes ON, the light in the room is sent a command to go on. This is because the motion sensor is marked as participating in determining if the room is on or not. The motion sensor is the only device so marked and since it is ON the room is ON. The light is marked as responding to the room going on or off so it is sent a command to go on.
- Each time you move, the motion sensor restarts the timer. This is because the motion sensor options say that if the room is ON and a command is received and the auto off timer is running then the auto off timer is restarted.
- If you change the light level at the switch, the timer is canceled. This is because the switch is marked such that if the room is on and the timer is running and a command is received then the timer is canceled.
- Since the room remains on – the light is on and that participates in the room state - and the timer not running, the motion sensor now has no effect. The room is still ON since the light is on. And when a command is received from the motion sensor it is ignored.

Wow! That is a lot of function from just a few checkboxes.

### Example 2

In this example there are two lights in the room. Here is what we want to happen:

- You walk in the room.
- You turn on one switch
- An auto off timer starts
- You turn on a second switch
- The timer should be canceled because by turning on multiple lights you are saying you want to stay in the room.

And since you may turn on either light first, the settings of each light must be the same. Here are the settings for both devices:

The screenshot shows a configuration window for a room's auto-off timer. It contains several sections:

- Room ON/OFF Participation:** A checked checkbox "The state of this device participates in determining if the room is ON or OFF". Below it are four unchecked checkboxes for "When LIGHT" and "When DARK" responses to room ON/OFF, each with a "100" percent value.
- Transmissions from this device can trigger a room auto-off specification:** A checked checkbox. Below it are two rows of settings:
  - "When room is OFF ...":
    - "... and receive OFF from the device": Ignore
    - "... and receive ON/DIM from the device": Restart Timer
  - "When room is ON ...":
    - "... and Room Auto Off Timer IS NOT running ...":
      - "... and receive OFF from the device": Ignore
      - "... and receive ON/DIM from the device": Ignore
    - "... and Room Auto Off Timer IS running ...":
      - "... and receive OFF from the device": Ignore
      - "... and receive ON/DIM from the device": Cancel Timer

The action of turning on either light causes the room to go ON. Since the room was OFF and an ON was received then auto-off timer is started. Controlling the other light causes the timer to be canceled.

If you had a third light in the room, since the room is ON and the timer is not running controlling that light has no effect.

### Example 3

- This example is simpler than the first two. Here is what we want to happen.
- You walk into the room
- The motion sensor detects you so the lights come on and starts an auto off timer.
- You change the light level at the switch. You want the auto off timer to restart but not to be cancelled.
- Each time the motion sensor sees you the timer should be restarted.
- If you turn off the lights manually then the timer should be cancelled.

## Switch settings:

The state of this device participates in determining if the room is ON or OFF

When LIGHT: This device responds to a room ON by going to  percent

When DARK: This device responds to a room ON by going to  percent

When LIGHT: This device responds to a room OFF by going to  percent

When DARK: This device responds to a room OFF by going to  percent

Transmissions from this device can trigger a room auto-off specification

-When room is OFF ...

...and receive OFF from the device

...and receive ON/DIM from the device

-When room is ON ...

...and Room Auto Off Timer IS NOT running ...

...and receive OFF from the device

...and receive ON/DIM from the device

...and Room Auto Off Timer IS running ...

...and receive OFF from the device

...and receive ON/DIM from the device

## Motion sensor settings:

The state of this device participates in determining if the room is ON or OFF

When LIGHT: This device responds to a room ON by going to  percent

When DARK: This device responds to a room ON by going to  percent

When LIGHT: This device responds to a room OFF by going to  percent

When DARK: This device responds to a room OFF by going to  percent

Transmissions from this device can trigger a room auto-off specification

-When room is OFF ...

...and receive OFF from the device

...and receive ON/DIM from the device

-When room is ON ...

...and Room Auto Off Timer IS NOT running ...

...and receive OFF from the device

...and receive ON/DIM from the device

...and Room Auto Off Timer IS running ...

...and receive OFF from the device

...and receive ON/DIM from the device

## Timeline of what happens.

- You walk into the room and the motion sensor sees you. Since the motion sensor participates in the room state, the room is now on. When the room goes on, the switch is sent a command to go on.
- As long as the motion sensor sees you the timer keeps getting reset.
- You change the light level of the switch. The timer is restarted. When the motion sensor next sees you, the switch is not sent a command. Why? Because the switch is only sent a command when the room goes from off to on.
- You turn off the switch at the paddle. While the room may still be on – the motion sensor is the only determinate of that – the timer is cancelled.

The interaction of devices, rooms, and auto-off can be complex but there is a lot of power in the concept. Hopefully the above examples will get you started.





## Chapter 7

# Schedules and Schedule Entries

Now that you've created some devices, you might want to set up a schedule or two for the control of them.

Controlling devices in your home at various times throughout the day is one of the central uses for the Home Control Assistant. With HCA you can create multiple schedules, each one fitting a different aspect of your home life. For example, you may want a schedule that turns lights on and off in your home while you are away, to provide the house a lived-in look. But having that schedule current while you are home could be annoying. Regardless of whether you are home or not, you might like to have your outside lights and a few interior ones come on at dusk. You may also want a schedule that you use when you entertain, so lights stay on a bit longer than normal at night.

This chapter discusses schedules and schedule entries, how to use the Visual Scheduler and a Wizard to automatically create schedules with entries that make your home look lived in. Sections in this chapter include:

- About schedules and schedule entries
- Considering your schedules and entries
- Creating a new schedule
- Creating a new schedule entry
- Modifying a schedule or entry
  - Deleting a schedule or schedule entry
- Monitoring your schedules with HCA
  - Suspending and resuming schedules
- The Occupancy Wizard

---

## About schedules and schedule entries

Creating schedules and schedule entries is a very powerful aspect of HCA. In order for you to take full advantage of the possibilities, you should be familiar with a few important terms.

Term	Definition
Schedule	A schedule tells when things will happen: when devices will turn on and off, when programs will start, and run, what the “plan” for HCA is. A schedule is composed of a list of schedule entries. HCA can contain more than one schedule.
Schedule Entry	Each schedule entry contains the date and time that one device or group is turned on or off. A schedule entry can also be used to start a program.
Current Schedule	The current schedule is the one schedule in your design that HCA uses to control devices at the times specified by its entries. Only one schedule can be current at any time, but you can change which schedule is current.

Out of all the schedules present in your design, you can mark **one** of them as the schedule HCA makes current when your design is loaded. Of course, your design can be loaded, or HCA started on your computer, in one of two situations: either normally (you started HCA or loaded a design), or restarted after a power failure (when HCA automatically restarts as you have set it to). You can also choose to have a schedule marked to become the current schedule when HCA restarts after a power failure. Normally when HCA starts the last schedule that was current when HCA shutdown is made the current schedule, but there are options to modify this.

**Hint:** The name of the current schedule is always shown in the ribbon in the *Control* category.

When you begin creating schedule entries, you will see that there are seven kinds:

- Entries that send an **On** command at a certain time to a specific device or group
- Entries that send an **Off** command at a certain time to a specific device or group
- Entries that send an **On** command at a certain time, and later send an **Off** command to the same device or group
- Entries that start a program with an **On** or **Off** command at a certain time
- Entries that initiate a scene stored in a device
- Entries that change the setpoint and other properties of a thermostat
- Entries that send sequences of IR commands

In addition to these seven kinds, if the schedule entry sends an On command and the device supports dimming, then you can also specify the illumination level of the device. If the device supports stored scenes, you can specify the scene name.

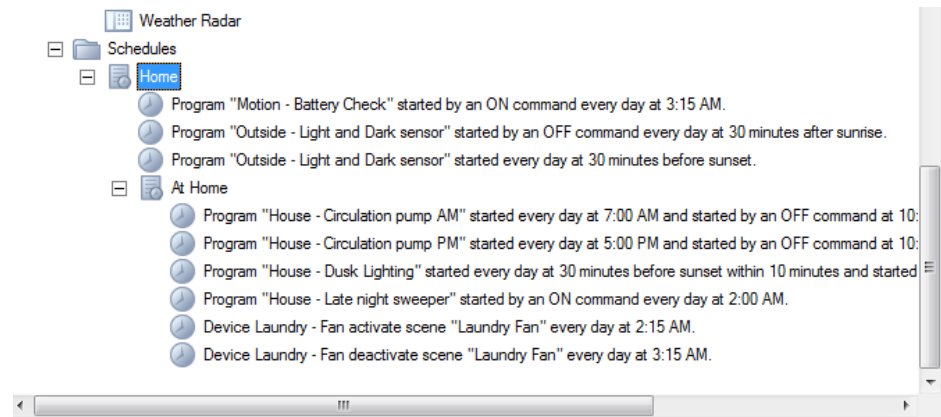
You can see that in HCA you can not only schedule devices but also groups and programs. If a group is scheduled to go on (or off) at a given time, each of its members goes on (or off) at that time. Before you start creating schedules and entries, it is important to think about what actions you want to happen in your home and when they should happen.

## Considering schedules and entries

Before creating schedules, it's best to think a bit about how to organize them. In HCA schedules can be organized in what is called a *parent* and *child* relationship. What this is all about is best shown by an example. Suppose in your home you have both outside and inside lights that you want to control. Every day you would like to have the outside lights come on at dusk. Also, when you are away from home you would like the inside lights to turn on and off to make the house looked lived in.

You could create two schedules: one for when you are home that controls only the outside lights, and another for when you are not at home and controls outside and inside lights.

If you do this the schedule entries for the outside lights need to be added to both the *at home* and *away* schedules. Rather than duplicate your work, HCA can organize your schedules like this:



Using the terminology introduced above, *Home* is the parent and *Away* and *At Home* are children. In HCA it is possible to have children of children and as many parent schedules as needed. But be careful as it is very easy to get confused!

Organized this way, there are three schedules. *Home*, which contains the controls for those things that you want done regardless if you are home or not. *At Home* contains those things to be done when you are home (just the one entry in this case), and *Away* for when you are not at home and making the home looked lived in.

As described above, the *current schedule* can be any one of these three schedules. When a child schedule is the current schedule, all the entries in it **and in any of its parents** are used by HCA to control your home. In the above example, if *At Home* is the current schedule, both the schedule entries for the fish tank and the outside lights happen. If *Away* is the current schedule, all the schedule entries for the inside lights and outside lights happen.

It's also important to consider what type of schedule entries you might need before you start creating them. You will want different types of entries for different purposes. Here are some examples:

1. You like to wake up in the morning to music, so you want the radio to come on at your wake up time. Just before you leave home in the morning for work (varies somewhat each day) you'll turn it off manually.
2. You often forget to turn lights off when you leave home in the morning or when you go to bed at night.
3. You want to turn your outside lights on at dusk and off at 11:30 p.m.

These three examples illustrate the needs for each type of schedule entry: example 1 is a good use for an On entry; for example 2 you would need an Off entry; and for example 3 an On-Off entry would be appropriate.

### Selecting days

When creating a schedule entry, you may be asked to select the day, date, or type of days that you want this schedule entry to apply to. There are several options. This schedule entry can apply:

- Every day
- Only on Monday, Tuesday, Wednesday, Thursday, and Friday
- Only on Saturday and Sunday
- Only on the days you select  
If you choose the *These days* option, you must select at least one day.
- Only on a particular date each month  
This option creates a schedule entry that applies only once each month, on the same date.  
The choices for this option are 1 to 31. If you choose a day that the current month does not contain (like choosing 31 and the current month is November) this schedule entry won't happen this month.
- Only on the date you select  
This option creates a schedule entry that happens only once. Once the date you select is in the past, this schedule entry will not be used again until next year.

Each date type has its use. *Every day* entries are useful for things like turning outside lights on at dusk. *Monday - Friday* entries are good for controlling your house on days that you are off to work or school. *Saturday and Sunday* entries are for those days when you are home. *These days* entries may be useful for things like automatic watering systems that you want to run only a few selected days a week. The last two entry types are used for special purposes that you may have.

### Selecting times

You may also be asked to enter the time you want the on (or off) command to be sent. This step provides many different options for specifying the time. You can create an entry that happens:

- At a specific time (like 6 p.m., 4 a.m., 12:32 p.m., etc.)
- At Sunrise
- At Sunset
- Before sunrise by so many minutes
- After sunrise by so many minutes
- Before sunset by so many minutes
- After sunset by so many minutes

If you choose one of the options based upon sunrise or sunset, the time at which this schedule entry happens changes each day as the times for sunrise and sunset change. These times are calculated based on the location properties – latitude and longitude - that you set when you created your home file.

### Using Vary

If you choose a specific time (say 6 p.m.), then the command is sent at that time each day. If you are creating schedule entries in order to make your house appear lived in (perhaps for security reasons), you may not want that time to appear so exact. The last options in this step are for that purpose.

- If, for example, you have chosen 6 p.m. as the time, and you select *Exactly*, then this entry happens at 6 p.m.

- If you choose *Vary within 10 minutes*, then this entry can happen anywhere between 5:50 p.m. and 6:10 p.m.

You can make the time vary by any amount from one to fifty nine minutes. The *Vary within* amount HCA uses for one day will never be the same as the amount chosen for the previous or next day.

### On and Off entries

A few additional points about creating On-Off entries:

- You can only select one date (which is the date for the On portion of the schedule entry) and the On time and the Off time. You can't create a single entry that has two different date selections.
- The Off portion of the schedule entry always follows the On portion in time. For example, suppose you have a schedule entry where the On portion is for 8 p.m. and the Off portion for 1 a.m. The On portion of the schedule entry will happen at 8 p.m. and the Off will happen five hours later (which is 1 a.m. the next day). This is very useful for creating a single entry where On happens at sunset and where Off happens at sunrise the next day.

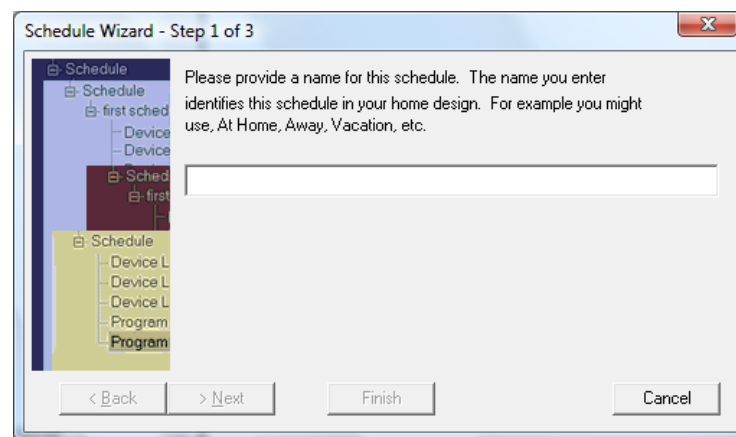
---

## Creating a new schedule

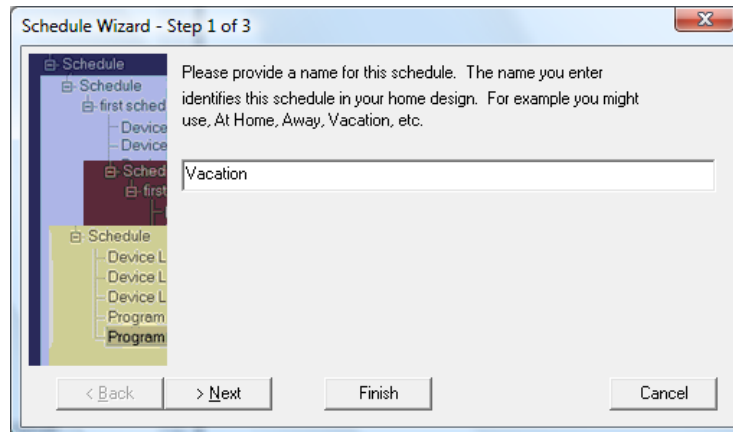
Creating a new schedule is very easy. All you do is open the Schedule Wizard, give the schedule a name, and then make a few decisions about the schedule.

1. On the Home Control Assistant menu bar, click New, and then choose Schedule.

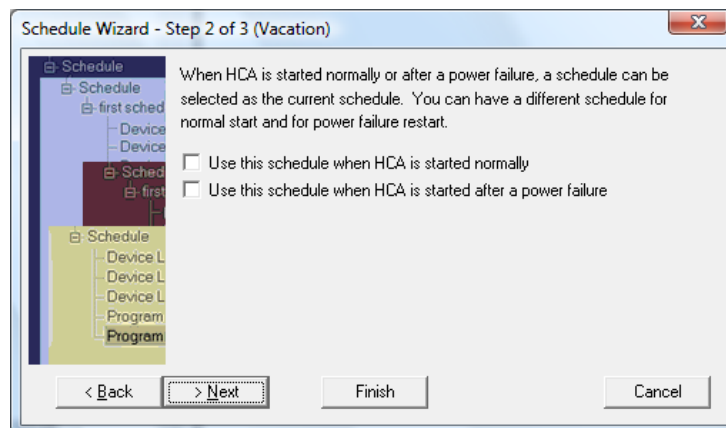
This opens the Schedule Wizard.



2. Give the schedule a name and click Next.



3. Decide whether you want this schedule to be the current schedule when HCA is started normally. If so, click the appropriate check box.  
You can designate only one schedule in your design as the normal start schedule, however, you can change this designation whenever you like.

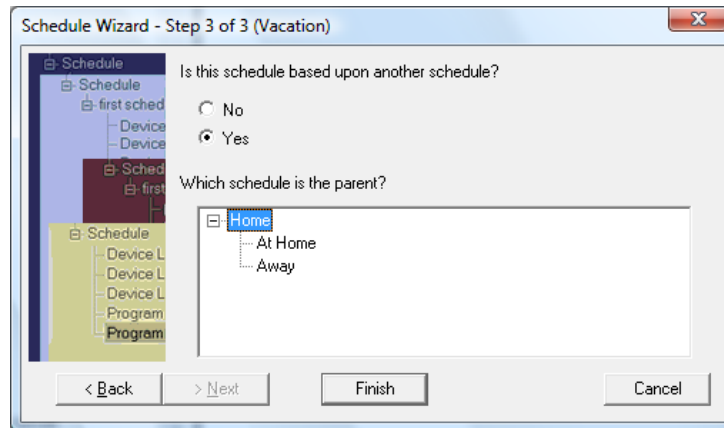


When you have made your choice, click Next.

If you attempt to designate more than one schedule as the normal start schedule, the last one you choose becomes the selection.

**Hint:** The schedule you are creating does not have to be the default schedule, although you should probably choose one of your schedules to be the current schedule when HCA is started normally (that is, not from a power failure).

- The next step of the Schedule wizard is to select if this new schedule is to be a parent schedule or a child of another schedule.



Once you have created a schedule, you can modify its properties by first selecting the schedule in the design pane, then right click with the mouse and choose Properties from the popup menu. The properties are almost the same as those you created using the wizard. Refer to *Modifying an Existing Schedule* later in this chapter, for details.

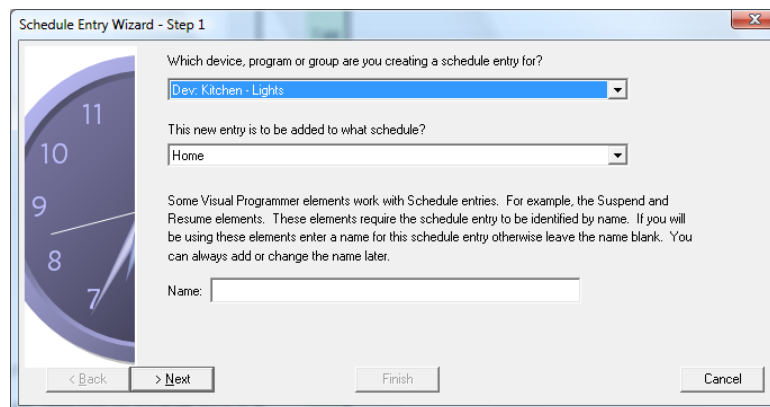
Now you are ready to create schedule entries.

## Creating a new schedule entry

Once you have one or more schedules, you can add entries to the schedules. An entry contains the date and time that one device or group is turned on or off. This is done in either of two ways: you can use the Schedule Entry Wizard to create any schedule entry, or you can use the Visual Scheduler, which allows you to graphically create new schedule entries. This section discusses the Schedule Entry Wizard. Refer to the Visual Scheduler chapter for details.

- In the ribbon click the “Schedule Entry” button in the “New” panel..

This opens the Schedule Entry Wizard.

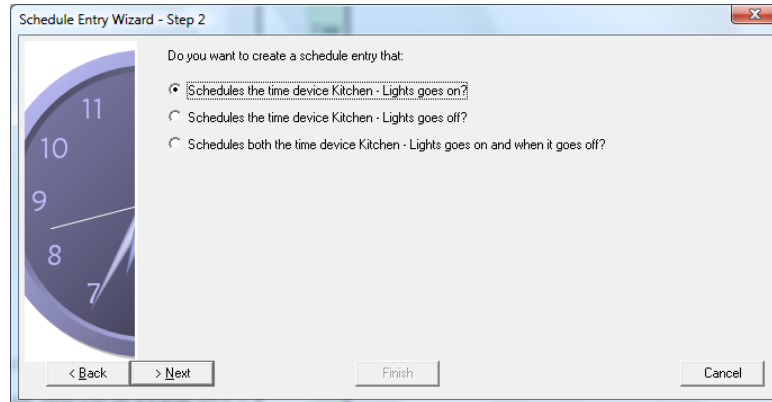


The upper dropdown contains the devices, programs, and groups in your design. The lower list contains all the schedules in your design.

- Click the arrow at the right of the upper dropdown, and select the device, program, or group that you are creating a schedule for.

Click the arrow at the right of the lower dropdown, and select the schedule to add the new entry to. Then click Next. Use of name you can enter for this schedule entry will be explained later.

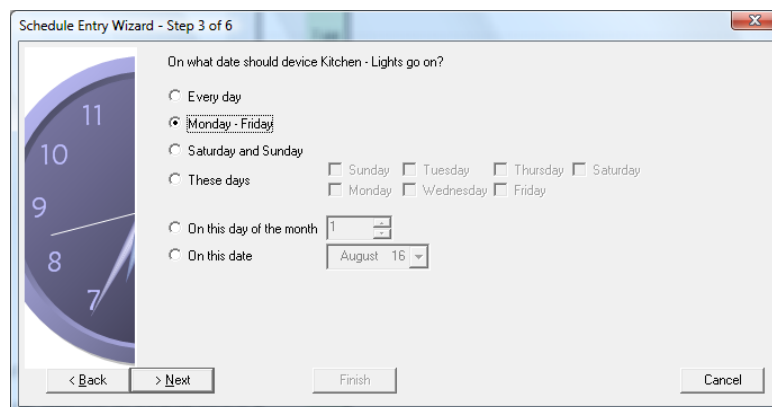
The next steps of the wizard depend upon what type of object you are creating a schedule entry for. If you choose a device or group, the next wizard step prompts you to choose the type of entry you are creating.



If in the first step of the wizard you selected a *program* to create a schedule entry for, you may or may not see this wizard step. If the program can be started only by an On command or only by an Off command, then this wizard step is skipped. If the program can be started by both On and Off commands, then this step is shown.

Once you have completed these first two steps of the wizard, the next steps you see depend upon the type of schedule entry you are creating.

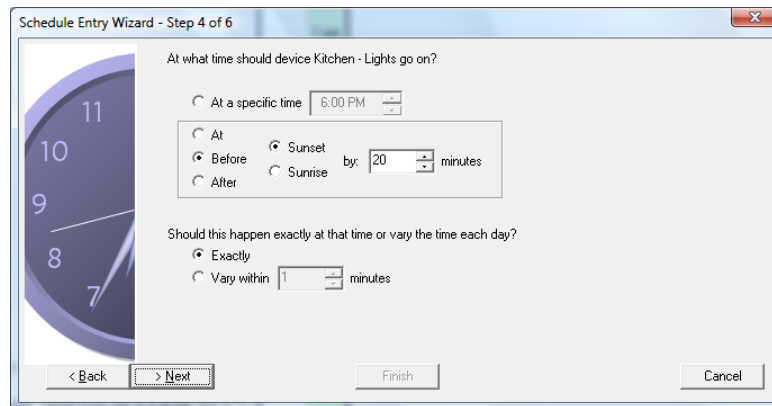
3. Click the type of entry that you are creating, and click Next.
4. If you are creating an On entry, in this step of the wizard, you select the day, days, or date that this schedule entry applies to. You can choose only one option, but if you choose *These days*, *On this day of the month*, or *On this date*, you must also provide additional information.



Click your choice, and then click Next.



5. Now you need to specify the time you want the on (or off) command to be sent. This step contains many different options for specifying the time.

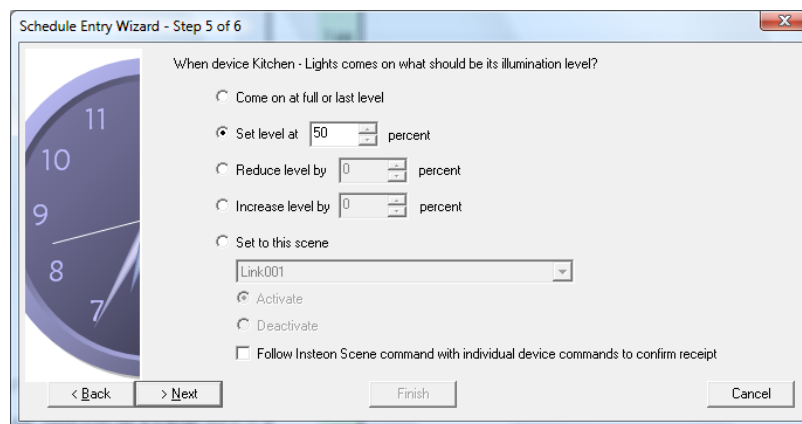


You can create an entry that happens:

- At a specific time (like 6 p.m., 4 a.m., 12:32 p.m., etc.)
- At Sunrise or Sunset
- At so many minutes before or after either sunrise or sunset

If you choose one of the options based upon sunrise or sunset, the time when this schedule entry happens changes each day as the times for sunrise and sunset change. See *Considering your schedules and entries* and *Using Vary* earlier in this chapter for more details about these options.

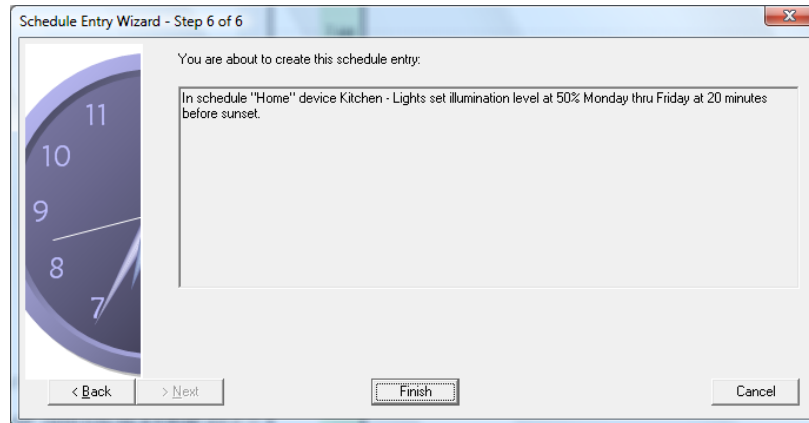
6. The next step of the Schedule Entry Wizard depends upon if you are creating an ON entry and if the device can be dimmed, or supports scenes. If so, then the next step allows you to specify the illumination level or scene.



There are five options. The first has HCA send an On command to the device. The second sets a specific level, for example come on at 50% illumination. The third and fourth choice changes an existing level either up or down. The last option is for devices that have the capability to store scenes – preset illumination levels – in their local memory.

Make your choice then click Next.

7. The final step of the Schedule Entry Wizard is the most helpful of all. This step presents the schedule entry you created in a text form.



After going through so many wizard steps, you may have lost track of exactly what you are creating. This wizard screen shows you exactly what you have created, as a sentence. This gives you a chance to check your work, and see if the schedule entry is going to do what you really want it to.

After you check the schedule entry, if it is what you want, use the Finish button to create the entry and add it to a schedule. If you need to change it, you can use the Back button to go back to previous wizard steps and make your changes.

Any schedule entries you create are displayed in the design pane under the schedule they are part of. If you later find that you need to change a schedule entry you can modify it. Select it in the design pane and then choose Properties from the popup menu you get when you right click the mouse. Refer to *Modifying a schedule or entry*, later in this chapter, for more details.

---

## Names for schedule entries

In the first step of the schedule entry wizard you can optionally supply a name for this schedule entry. The name is like the names used for devices, programs, groups, etc. It identifies this schedule entry from all other schedule entries.

Later on in this chapter is a discussion of suspending a device, program, or group from schedule control. In this same way, you can also suspend a schedule entry. In this way the schedule may still be the current schedule but when the time for the schedule entry occurs the schedule entry is not processed if it is suspended.

You can only suspend schedule entries that have a name. If you suspect that you may want to do this, give the schedule entry a name. If you don't give it a name during the wizard you can always add a name to an existing schedule entry by opening its properties and entering or modifying the name.

---

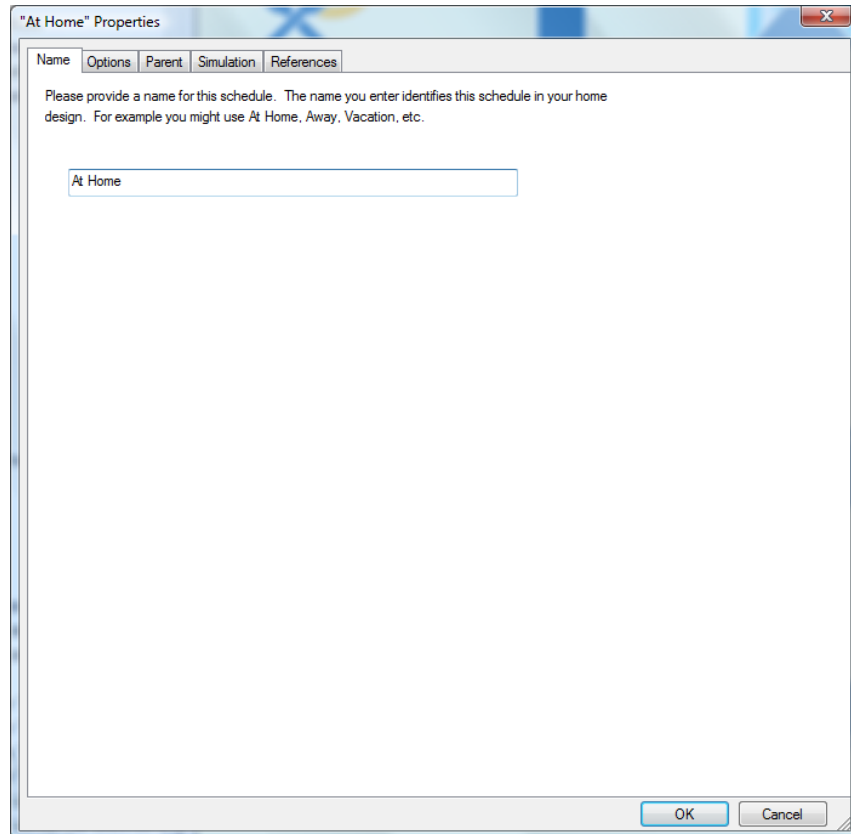
## Modifying a schedule or entry

Once a schedule or schedule entry is created, you can modify any of the properties that you set in the Schedule or Schedule Entry Wizard. You do this in the Properties dialog box for the particular schedule or entry. To get to the properties:

1. Select either:
  - the **schedule** name in the design pane of HCA
  - the **schedule entry** name in the design pane of HCA

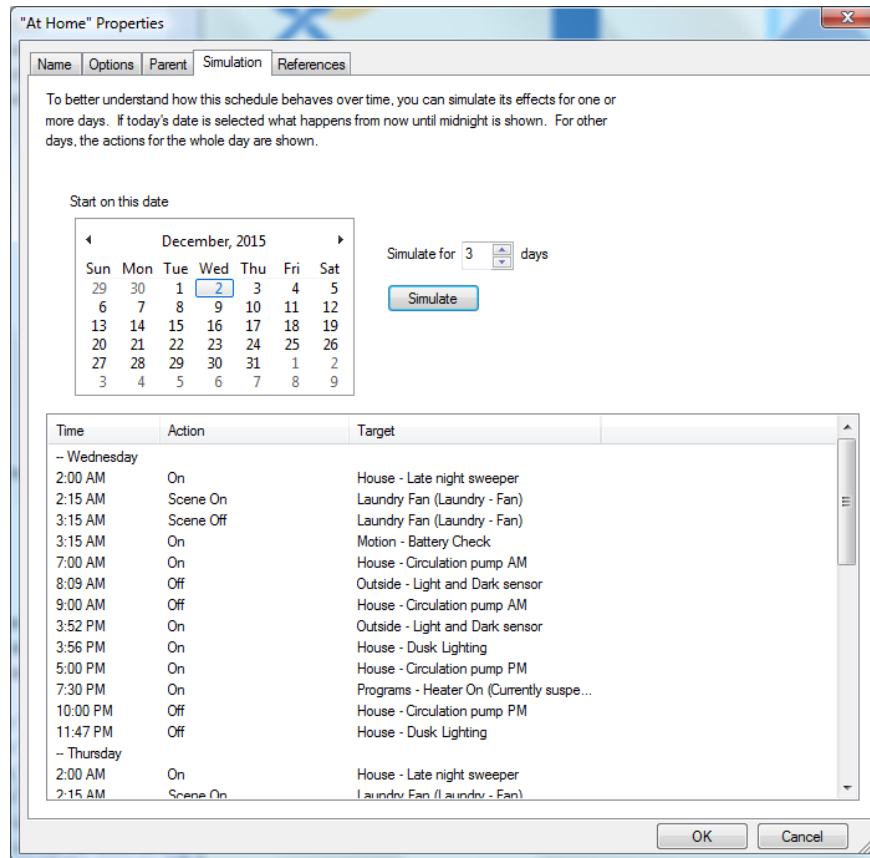
2. Click the right mouse button.
3. From the popup menu, select Properties.

The Properties dialog box title reflects the name of the schedule (but not the schedule entry) you selected.



In the Schedule Properties dialog box, you can change the schedule name, the options for normal or power failure start up, and which if any parent this schedule belongs to.

There are two additional tabs in the schedule properties dialog that are not found in the wizard. The "Simulation" tab doesn't change any properties of the schedule but is very useful for making sure that your schedule does what you expect it to do.



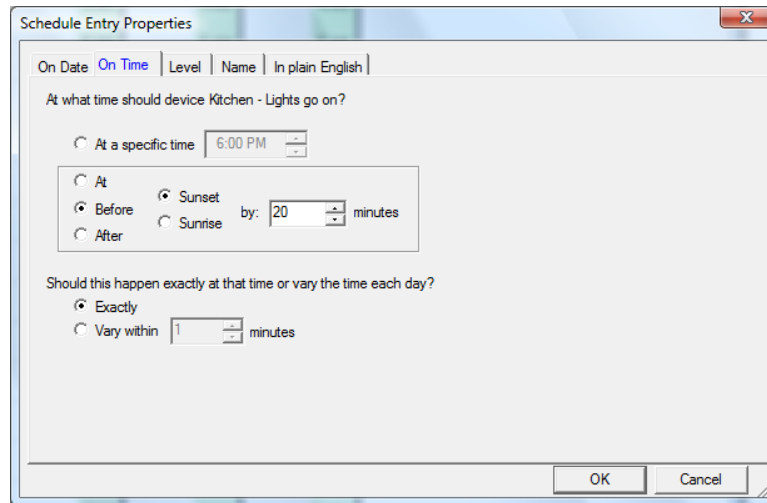
This tab is titled *Simulation*. In a complex schedule, especially one that is a child of a parent schedule and one that contains many entries, it can be hard to see just what the schedule does over time. The Simulation tab will help you with that.

To use the simulation facilities, select a date in the calendar and the number of days to simulate and press the Simulate button. The dialog then shows, sorted by date and time, how each device in your home is controlled.

---

## Modifying a Schedule Entry

In the Schedule Entry Properties dialog box, you can change the schedule entry date or time, or review the *In plain English* text version of what your entry does.



For either properties dialog box, click the tab you want to change properties on and make changes. You can change whatever you like (except the text version of your schedule entry), and click OK to save your changes.

---

## Deleting a schedule or entry

If you find that you have created an entry you don't need, or a schedule that doesn't work for you, you have two options.

- You can modify the properties for the schedule or entry. For instance, you can change the name or the schedule, or change the date or time for an entry.
- You can delete the schedule or entry.

### To delete a schedule or schedule entry:

1. Use the right mouse button to click the schedule or schedule entry name in the design pane of HCA.
2. Click Delete.  
HCA removes the schedule or schedule entry name from the design pane.

---

## Monitoring your schedules with HCA

Once you have created schedules and schedule entries, HCA can use them and act on them once the schedule is the current schedule.

At any given time, there can be only one schedule selected to be the current schedule. The current schedule is the schedule that HCA monitors to determine when it is time to turn a device (or group or program) on or off. HCA chooses a schedule to be the current one based upon the schedule properties when your design is first loaded.

- If HCA is starting normally, it chooses the schedule with ...*when HCA is started normally* checked.
- If HCA is re-starting after a power failure, it makes the schedule that was current when the power failed current again, unless you have a special power failure restart schedule.
- If HCA can not find the correct option for the circumstances, then it doesn't make any schedule current.

**Hint:** The name of the current schedule is always shown in the ribbon when the "Control" category is selected.

You can choose a schedule to become current at any time by using the Schedule dropdown in the ribbon.

---

## Suspending device, programs, and groups from HCA control

Another aspect of HCA to discuss under schedules is your ability to suspend items from control by HCA.

Devices, programs, and groups can be marked as suspended in various ways. If you do this, depending upon the type of suspend, the normal actions of HCA may not apply to a device, program, or group.

If something is suspended from schedule control, HCA acts as if there are no schedule entries for it in the current schedule. The schedule entries still exist, they just have no effect.

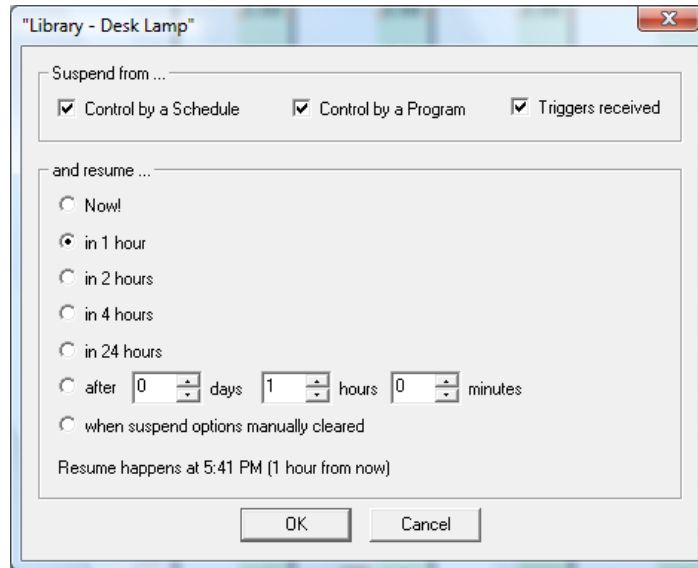
### Why would you want to suspend something?

Generally, the reason is to account for some unusual happening. Suppose for example that you schedule your living room lights to go off at 11 p.m. But tonight you are entertaining some friends and will be using the living room after 11 p.m. You could just delete the schedule entries that control the living room lights, but then you would have to put them back tomorrow. You could deactivate HCA but that would shutdown automatic control of the rest of your home—and you don't want to do that! In this case, suspending the living room lights from schedule control prevents them from automatically going off at 11 p.m., but leaves the rest of your home functioning as usual. If you later remove the schedule suspend from the living room lights then tomorrow they will function as normal.

### To suspend a device, program, or group:

1. Select either
  - the **icon** on the display or
  - the **name** in the design pane
2. Click the right mouse button.

From the popup menu, select Suspend and this dialog appears:

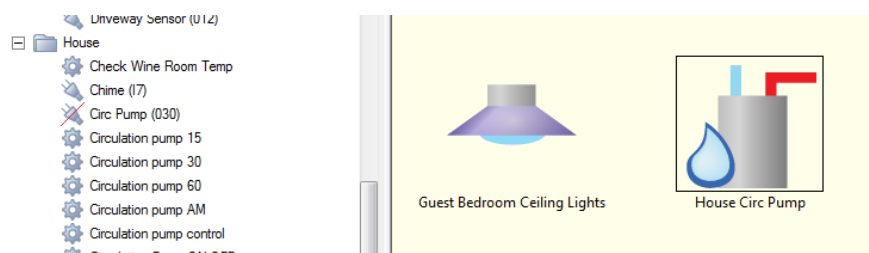


This dialog allows you to both suspend and to resume a device, program or group. In the *Suspend from* section of the dialog you can select what kind of actions you are suspending. These are:

Schedule	Prevent any HCA schedule from controlling this device, program, or group regardless of any schedule entries for it.
Program	Prevent any HCA program from controlling this device, program, or group using ON, OFF, or DIM elements.
Triggers Received	Prevent receipt of any commands that are the trigger for a device, program, or group from starting the program or controlling the device or group.

**Hint:** If you find that a device, program, or group doesn't seem to be responding as it should, check to see if it is suspended. This is one of the conditions that the Inspector looks for.

When something is suspended, the design pane icon shows this by a red slash though the icon and in the display pane the icon shows a black border. In the picture below, House – Circ Pump is suspended.



The resume part of the dialog tells when the suspended functions are resumed. There are two major choices: To suspend until some action you take resumes it or to have HCA automatically resume it at a given time. If you are using automatic resumption, several useful time limits are given as options. If you open the suspend dialog when something is already suspended, the time to resumption shows in the bottom of the dialog.

**Hint:** There are visual programmer elements for Suspend and Resume that make it simple for you to create programs that perform suspend and resume options as part of their actions. This way you can control suspend and resume without having to interact with the HCA UI. Just give the program a trigger and send that trigger from a keypad in your home. You can also use these elements to suspend and resume schedule entries. This only is possible if you gave the schedule entry a name as you use that name in the element.

---

## The Occupancy Wizard

The Occupancy wizard is designed to produce a schedule that controls lights in your home to make it appear that someone is home even if they are not. It does this by asking a series of questions about the rooms in your home, where you spend time, when you get up, and when you go to bed. While the schedule it produces is far from perfect, it does provide a good enough schedule that is more realistic than just turning on a few lights at the same time every day.

To start the wizard select Occupancy Wizard from the ribbon “New” panel. There is a lot of information in the wizard dialogs that should guide you through the process.

Once you have produced this schedule you can use the Visual Programmer to create a program that makes that schedule the current schedule when you leave home – just press a button on the way out and the schedule becomes current. Another button press on the way in, and your normal schedule is back in control.



## Chapter 8

# Visual Scheduler

So far you've created a design with some devices and schedules. Now you may want to see another way to create schedules for your home.

The Visual Scheduler is a graphical way to add, modify, or delete entries in a schedule in the Home Control Assistant. It can be used to quickly create schedule entries and also to look at multiple devices to see how their schedules compare. Although it is very powerful, it does not take the place of the Schedule Entry Wizard in constructing the schedules for your home.

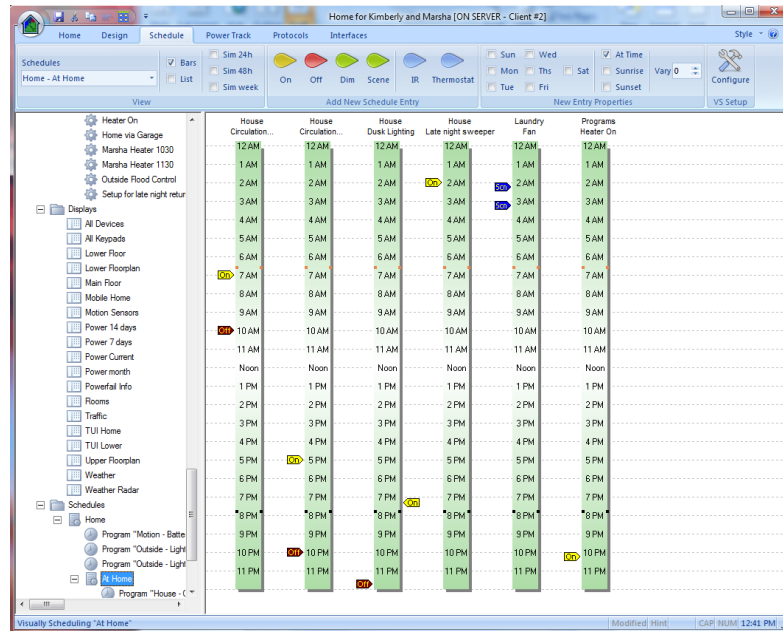
This chapter discusses the Visual Scheduler, including topics on:

- Starting the Visual Scheduler
- Looking at the Visual Scheduler
- Using the Visual Scheduler
  - Creating new schedule entries
  - Modifying existing schedule entries
  - Deleting existing schedule entries
- Additional features of the Visual Scheduler
  - Sun-relative times
  - Every day and not-every day schedule entries
- Characteristics of the Visual Scheduler
- The Visual Scheduler and the Schedule Entry Wizard

## Starting the Visual Scheduler

Before you can start the Visual Scheduler, you must already have one or more schedules in your home design.

To start the Visual Scheduler, choose the *Schedule* category in the ribbon. The Visual Schedule appears as:



All the tools necessary to work with the Visual Scheduler are in the ribbon.

The schedule you are viewing is named in the dropdown at the left end of the ribbon. In this example, the “At Home” schedule is being seen.

You can’t work on multiple schedules at one time. To work on another schedule, use the dropdown in the ribbon to change schedules.

To close the Visual Scheduler, chose another category in the ribbon.

## Looking at the Visual Scheduler

When you first open the Visual Scheduler, you might want to take note of its main features.

- The Visual Scheduler uses a time bar to represent each device, group, program, and room in your design. This bar is a vertical column with 12 a.m. (midnight) at the top and 11:59 p.m. (almost midnight) at the bottom.

HCA uses the times for sunrise and sunset (for the current day) to mark the sunrise and sunset times with small black blocks. If you look at the Visual Scheduler during the summer, the location of the sunrise marks is different than what shows when you open the Visual Scheduler in the winter.

- Above each bar is the name of the device, program, group, or room that the schedule bar represents.

Attached to the bars are time markers representing schedule entries. For example, in the above image there is an On marker in the bar for “Circulation Pump” at about 7 a.m. In the “Dusk Lighting” bar there is a marker at about 30 minutes before sunset. Remember, specific times are shown by markers on the left side of the bar, times relative to sunrise and sunset are on the right.

Each schedule entry has a date component – the days of the week, and a time – either a set time or a time plus or minus sunset or sunrise, and the number of minutes to vary each day. When a new schedule entry is created, the current settings in the ribbon are used for that new schedule entry.

- At the bottom of the display pane is a scroll bar. If you have so many devices, programs, and groups in your design that the display pane cannot show all the time bars at one time, you can use the scroll bar to display the rest of the time bars. It operates as you would expect, and can scroll the Visual Scheduler left and right to see the other time bars.
- In the ribbon are a number of tools for creating new schedule entries as well as changing how the Visual Scheduler appears.

Each schedule entry – represented by time marker on a bar - has a date component – the days of the week, the time – either a set time or a time plus or minus sunset or sunrise, and the number of minutes to vary each day. When a new schedule entry is created, the current settings in the ribbon are used for that new schedule entry.

Also in this ribbon are the markers themselves – on, off, dim, scene, thermostat, or IR. To add a new schedule entry, click on the marker picture and then move the mouse to where it should be and click again to drop it. As you move the mouse the marker “clicks” onto the nearest bar.

**Hint:** Remember that 12:00 p.m. = noon, and 12:00 a.m. = midnight.

---

## Using the Visual Scheduler

Now that you are familiar with the basic look of the Visual Scheduler, you probably want to know what you can do with it. The next sections describe how to

- Add time bars
- Create new schedule entries
- Modify existing schedule entries
- Delete existing schedule entries

The following sections discuss these operations.

---

### Add time bars

In order to schedule a device, program, group, or room, there must first be a time bar for it. Select the name of device, program, group, or room in the design pane and drag it onto the Visual Programmer and drop it. A new time bar is created.

The Visual Programmer normally only shows time bars for those devices, programs, groups, and rooms that have schedule entries in the current schedule. This behavior can be modified. See the section below on Configuring the Visual Scheduler.

---

### Creating new schedule entries

To create a new schedule entry, you must first decide two things:

- Do you want to set a new schedule entry with a specific time or with a sun-relative time (based upon sunrise or sunset)?

Suppose you want a lamp to come on every day of the year at 6 p.m. In this case, you want a specific time—6 p.m. This might be what you want, but remember that in the summer the light will come on when it is still light outside, and during the winter the light will come on

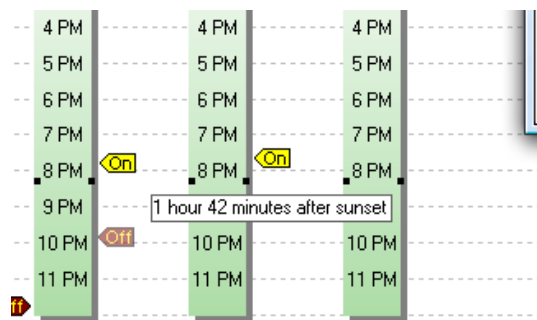
well after dark. In this instance, you may want to create a sun-relative time, say at 15 minutes before sunset.

- Do you want to turn something on, off, change its illumination level – dim or brighten, or activate a scene?

Once you know the type of schedule entry you want, it is easy to create the entry. As an example, let's create an On entry for 6 p.m that happens on Saturday and Sunday only.

1. Locate the time bar for the device you want.
2. Make the settings you want in the ribbon. For this example, make sure that Saturday and Sunday are checked and the other days are not. Also make sure that the *Clock Time* option is selected.
3. In the ribbon click the On marker. The cursor of the mouse changes to an *On* marker. Move the cursor to where you want to drop the marker and click again to add that time marker to the bar. Since the new marker is set for a clock time, the marker will “click” onto the left side of the bar. Had the new marker options been set for a sun relative time, the marker with “click” onto the right side of the time bar. If you drag an IR or thermostat marker they will only “click” onto a time bar for the appropriate type of device.

As you drag the marker, a small popup window shows the time you are pointing at. For example, in this case an *Off* marker is being dragged and is currently at 1 hour and 42 minutes after sunset.



**Hint:** Time markers on the left side of the bar always represent the same time and appear at the same place regardless of what day you use the Visual Scheduler. Time markers on the right represent times relative to today's sunrise and sunset and these positions vary throughout the year. Suppose you have a schedule entry that is for sunset. If in May you open the visual scheduler, that schedule entry marker will appear pointing to 7pm. If you open the Visual Scheduler again in December, that same schedule entry would appear but now pointing at 4pm. Still sunset in both cases just that the Visual Scheduler shows it at a different place.

If you drag a Dim, Scene, IR, or thermostat markers, once you drop the marker a dialog opens that allows you to select its properties. More on this later.

As you drag time markers around, you may see that you can't quite get the exact time you want. You want to get something to happen at 10:15, but no matter how carefully you drag, you can get 10:13 or 10:16, but not 10:15.

This is due to the fact that the Visual Scheduler translates the position of the pointer on the time bar into a percentage of 24 hours. Since the pointer moves only in single pixel increments, this percentage calculation may not always yield schedule entries to the exact minute. You will need to use the Schedule Entry Properties for this device to set the time exactly should you need to the minute operation.

**Hint:** If you can't see all of the time bar details and all the time markers, make the HCA window bigger or maximize it.

One final point about using the Visual Scheduler: When creating schedule entries that control **programs**, you may be able to drag On markers and have them “click” onto the time bar for the program but not Off markers. As the chapter on the Visual Programmer describes, programs can be started by a lot of different trigger types. Using the Visual Programmer you can only start programs using simple ON and OFF triggers.

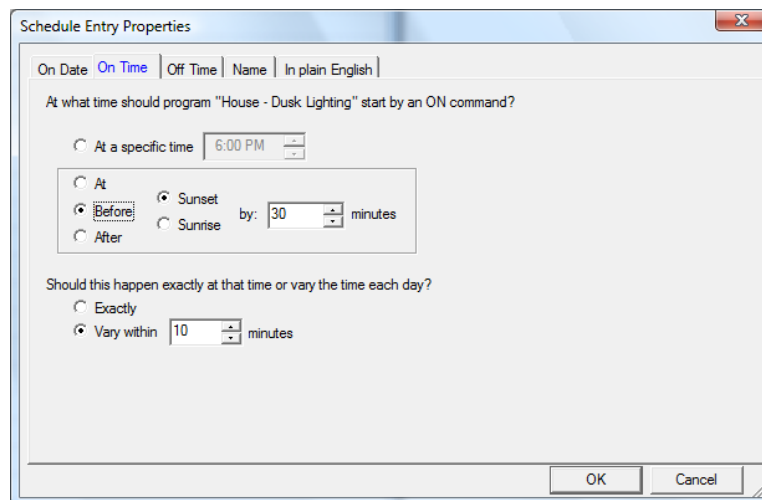
---

## Modifying existing schedule entries

It is very easy to change an existing schedule entry using the Visual Scheduler:

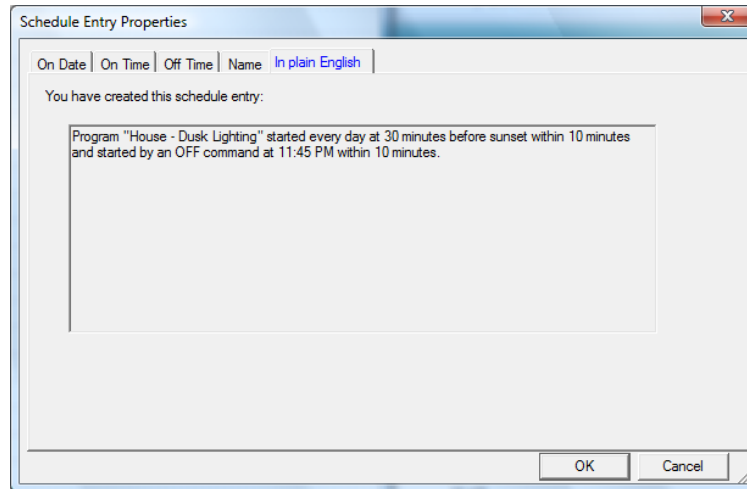
1. Move the mouse pointer to the time marker for the schedule entry.  
As soon as the pointer is over an existing time marker, the status bar displays the corresponding schedule entry.
2. Click the marker and drag it to another location.  
When you complete the drag, the schedule entry updates to reflect the new position.

This only changes the time. To change the date or the vary amount, right-click on the time marker and select Properties from the popup menu.



Using this dialog you can change all aspects of the schedule entry: Its date, time, action and name.

The last tab shows you what the schedule entry does in a simple text statement. With all the options available, this is a good way to check your work.




---

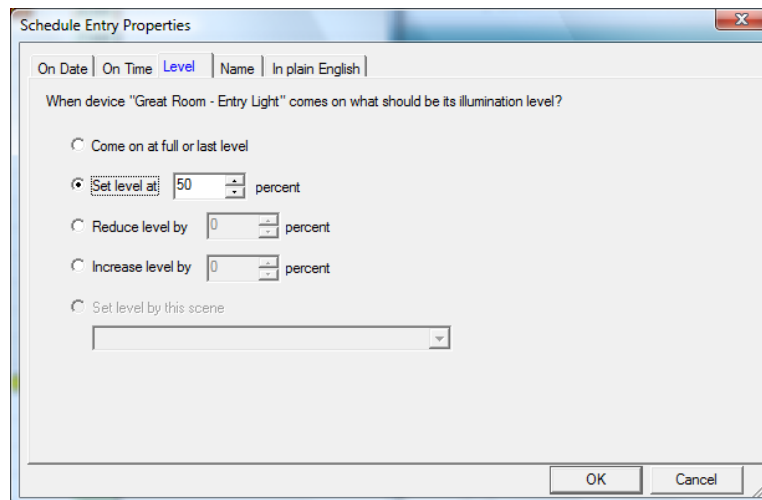
## Deleting existing schedule entries

There may be times when you want to remove an existing schedule entry from a time bar. To do so right click on the marker and select Delete from the popup menu.

---

## Dim and scene schedule entries

When creating entries for a device that supports dim or supports scenes, the Dim markers are dragged and dropped. When dropped, the schedule entry properties opens:

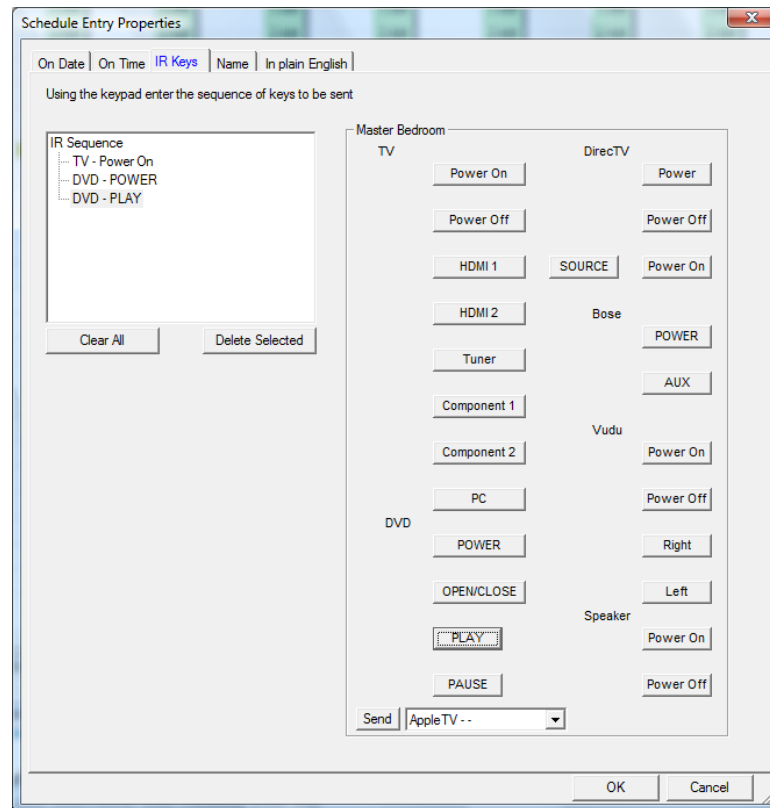


In this example, the device doesn't have any stored scenes but if it did those would be listed in the dropdown.

For devices that support dim, you can select the illumination level to go to, to increase to, or to decrease to.

## IR schedule entries

When creating schedule entries for an IR device the marker at the bottom of the time bar says “IR”. When this marker is dragged and dropped someplace on the time bar this dialog opens.

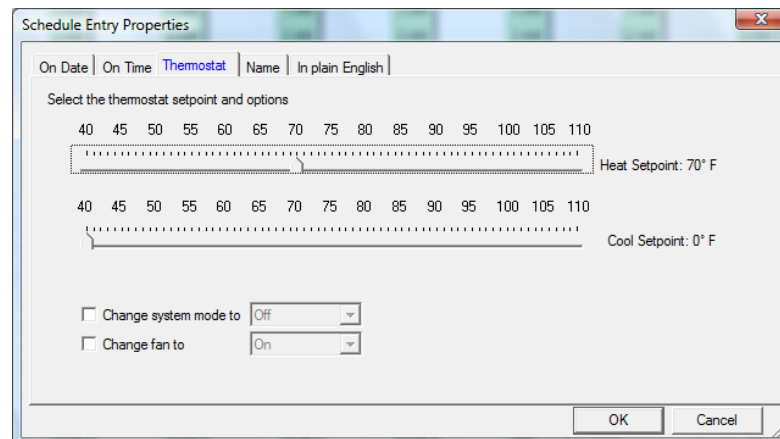


The contents of the dialog depend on the keypad used by the IR device. In this case a custom keypad for this television was created with the keypad builder. The keycodes sent when this schedule entry happens are shown in the box at the left side of the dialog

**Hint:** You may have to drag the dialog larger to see your whole keypad.

## Thermostat schedule entries

When a time marker for a thermostat is dragged and dropped this dialog opens:

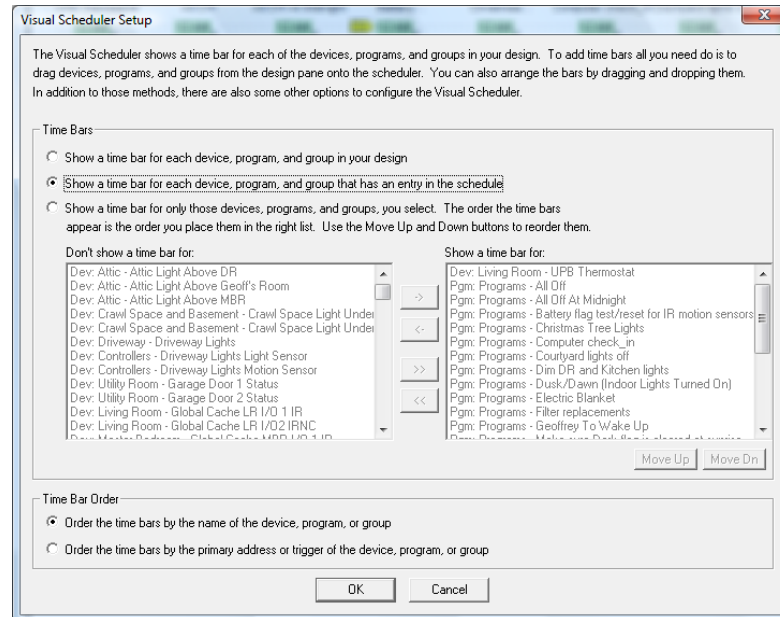


Depending upon the model thermostat in use the options may be different. To change the setpoint drag the thumb of the slider to the desired setpoint. To change the other options enable the option with the checkbox and select the option value from the dropdown.

## Configuring the Visual Scheduler

One of the best features of the Visual Scheduler is that you can visualize a schedule by looking at the markers on the time bars. By having two bars next to each other you can see how they both are affected over time. The problem comes if you have a lot of devices. HCA gives you quite a few ways to help you control which times bars you see and in what order.

To open the configuration dialog press the *Configure* button in the ribbon.



As you can see from this dialog you can choose to have the Visual Scheduler show time bars for only those devices you have in the schedule, or you can have it show a selected set of devices. In addition, there are options for controlling the order you see the bars.

Regardless of which display you choose, you can always add a time bar for a device you want to add to the schedule. There are two ways to do that.

First, and the simplest way, is to select the device's name in the design pane and drag it on to the Visual Scheduler. Drop it on or before the bar to want it to appear before.

The other method is to use the VS setup dialog – shown above - and use the second display option to select the devices you want to see and the order they should appear in.

Finally, if you use the display option where you pick the bars and their order, you can reorder the bars by drag and drop within the VS. Click on a bar and drag it on or before the bar you want it to appear before.



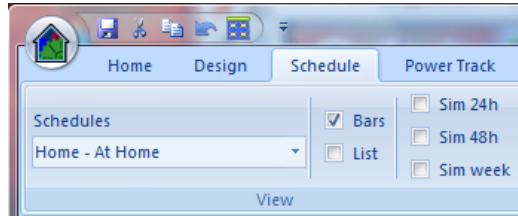
## Other Views of your schedule

In addition to the time bar view of your schedule, you can also view it in a tabular format and in a simulation.

The tabular view shows the schedule entries in a form that may be easier to see if you have many schedule entries.

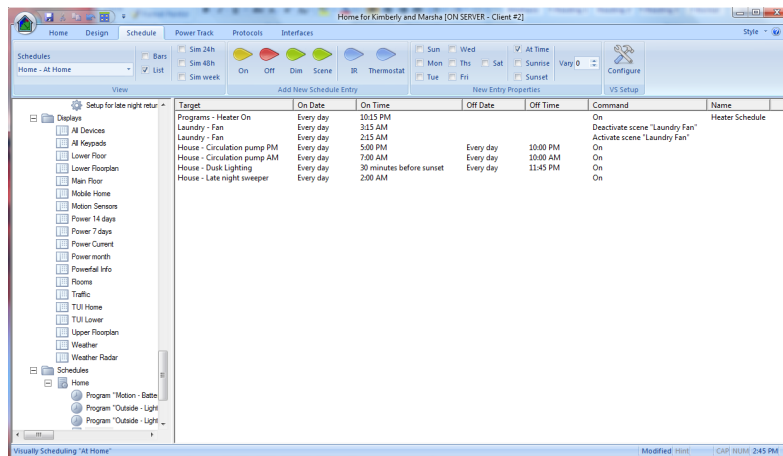
The simulation view shows what the schedule does in the order that it will happen for today, tomorrow, and this week.

To change the Visual Scheduler view, tick one of the view checkboxes in the ribbon.



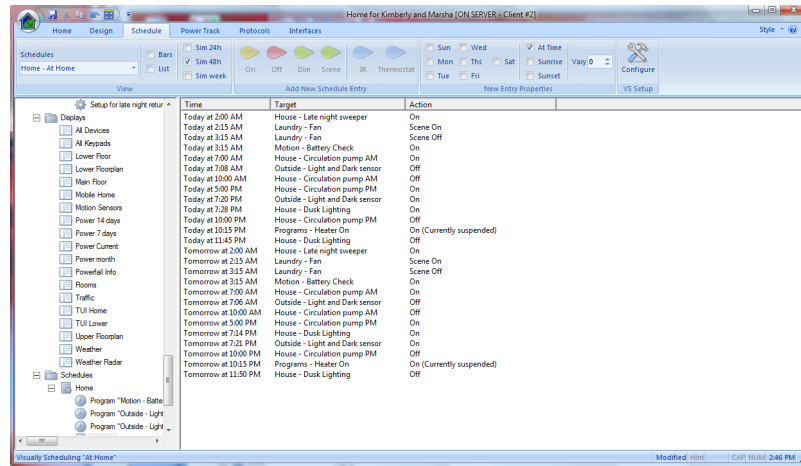
In the List View the schedule is broken down into target, on date, on time, off date, off time, and action columns.

One nice feature of the List view is that you can sort the columns using the standard list view column header click method. This may make it simpler to focus in on a part of your schedule.



The List View isn't a static representation of the schedule. You can delete or modify any entry as well as add new schedule entries. To delete the schedule entry right-click on the entry and select Delete from the popup menu. There is also a Modify option on that popup menu.

The Simulation View shows the schedule sorted by when entries happen and what happens at that time.



You can change what the period that the simulation shows, choose one of the three simulation option checkboxes in the ribbon.

## The Visual Scheduler and the Schedule Entry Wizard

The last topic to discuss is not exactly about the Visual Scheduler alone, but rather how the Visual Scheduler interacts with the Schedule Entry Wizard. While the Visual Scheduler is active, you still have access to all HCA commands. You can use the wizard to create new schedule entries. If the entries you create with the wizard are for the schedule that the Visual Scheduler is displaying, new time markers appear when the wizard is finished. You can create new devices, and new time bars appear. If you delete schedule entries, devices, programs, or groups, time bars may disappear.

### Example

Suppose you are using the Visual Scheduler to look at schedule entries that happen every day.

1. You use the Schedule Entry Wizard to create a new schedule entry for a device that has a time bar shown in the Visual Scheduler. (Remember that you do have access to all the HCA commands while using the Visual Scheduler.)
2. Using the wizard, you select Every Day for the date part of the new entry.

When you complete the wizard, a new time marker is drawn on the device's time bar at the location that corresponds to the time in the schedule entry you created. If you created an On-Off entry, two new time markers are drawn.

## Chapter 9

# Groups

Now that you've created devices you may be interested in setting up groups for the control of the devices. Groups are a convenient way of having several devices act together in response to the same command, and still maintain individual control over each device. For example, if you have set up groups in HCA, you could turn on five living room lights (as a group), and turn off one of them independently. Rather like having your cake, and eating it, too!

This chapter discusses groups, provides an example, and walks you through creating a group. Sections in this chapter include:

- About groups
  - An example
- Planning your groups
- Creating a group
- Modifying an existing group
  - Deleting a group
- Groups and Switch Stored Scenes

---

### About Groups

Groups are very useful concepts in some circumstances. The best way to describe what a group is, and why you would want one, is by an example, which follows. However, it might help to have some of the facts about groups up front.

- You can control a group, and also control separately the devices in a group.
- The devices collected in a group are called the members of that group.
- The members of a group can be devices or programs, but not another group. Groups can contain device of different protocols.
- Groups can have icons that appear on displays like devices and programs.
- You can create schedule entries that control the group (and each of its members) lie you do for devices.
- Not all devices types can be added to a group. For example, IR and thermostat devices can't be added to a group.

As with devices, groups can have icons (that are placed on displays), restart options, and so on. You can inspect or change the properties for a group using the same methods as for devices.

There are two reasons to use groups. The first is just to control a set of devices more conveniently than just device by device. You can schedule the group or use Visual Programmer elements rather than work with each device independently.

The other reason is more complex and has to do with providing a trigger for the group that sends commands out to each group member when the trigger is received. This is best shown by an example.

---

## A Group example

Much of what Groups do can be done by Insteon scenes or UPB links. But for older style devices, groups can be useful. Here is an example from older style X10 devices.

Suppose that in your home you have a number of outside lights. One light is in the front, one by the driveway, and another by the back door. There is a separate wall switch for each one, and you are going to replace the manual switches for each of these with controllable wall switches.

After you install the wall switches, you need to know what settings to use for them. You might want to turn them all on, off, or dim at the same time. So, you could set them all to the same house code and unit code. For this example, assume that setting is B-4. Using a control panel, if you press B-4 ON, all three lights come on. If you press B-4 OFF, then all go off.

Now, while this is very convenient, there are still some times when you want to control each light independently. So you set them all differently. For this example, assume the front light is set to B-4, the driveway to B-5, and the back light to B-6. Now you can control each light from a control panel independently, but you can no longer turn them all on or off with one button.

An HCA group is designed to solve this, allowing you to control each device independently, or all three as a group, with one control.

- First, you set each light to have a separate address (B-4, B-5, and B-6).
- Then, you create a group that has these three devices as members.
- Next, you assign a house code and unit code to the group, so that you can control it (C-1).

When HCA receives a command with the trigger for the group, that command is routed to each member. In the example, we have created a group called “outside lights” and assigned it to C-1. When you send an ON command from a control panel to C-1, then HCA receives it and sends an ON command to each group member, and the three outside lights go on. If you send a C-1 OFF command, all the lights go off. Since each light that is a member of the group also has a different house code and unit code, you can also control each one independently. The best of all worlds!

Don’t forget that groups can contain devices using different technologies – X10, Insteon and UPB. HCA does the best it can in translating what you are doing with the group into the appropriate commands for each technology.

---

## Planning your groups

Before you start creating groups in HCA, there are several decisions you need to make, aside from the name of the group and where to place it. You should decide why you want a group and what you want the group to do. This may include deciding:

- What is the purpose of the group?
- When will you use the group control?
- Will you typically control the group, the individual members, or both?
- Do you want programs included in the group?
- Will you control the group from a control panel, or only through an HCA schedule?

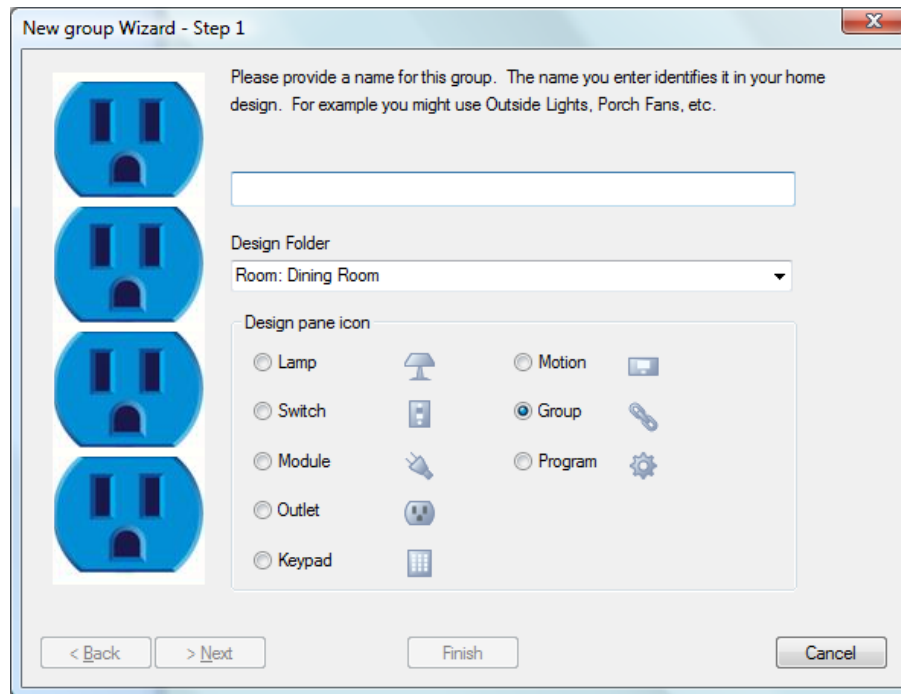
Using devices, schedules, groups, and programs together may get complicated. Some of these choices involve careful planning.

---

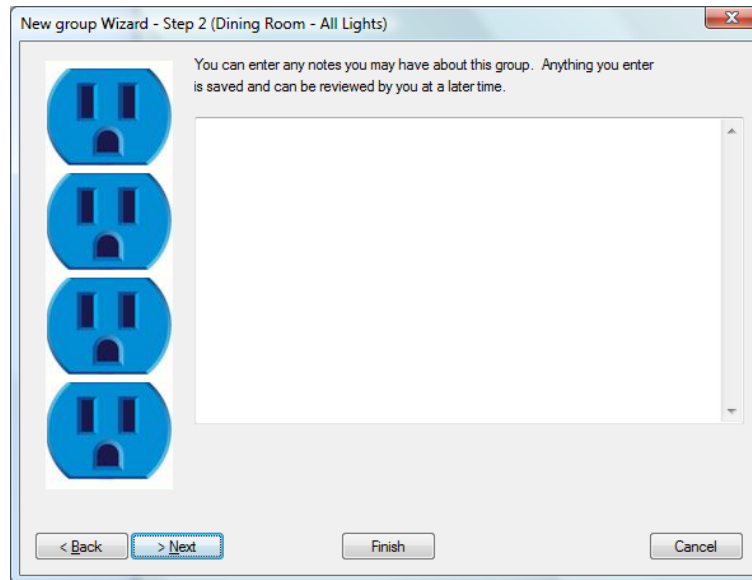
## Creating a group

You create a group using the New Group Wizard. Most of the steps are exactly the same as when you create a device. However, there is one additional step specific to the New Group Wizard—specifying which devices and programs are members of this group.

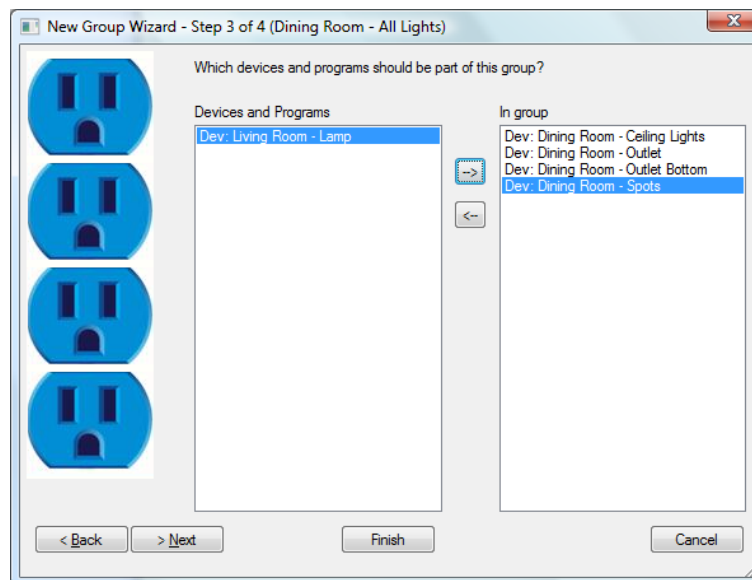
1. On the Home Control Assistant menu bar, click New, and then choose Group. This opens the New Group Wizard.
2. Type in a name for the group and type in or choose an existing folder to store the group. The name you choose must be unique within the folder. If you don't have two-part names enabled, then the group name must be unique within your design. To help you later, you can name groups with descriptions, or function identifiers. For instance, you could name a group: Entry lights, My Room, or Welcome Home.



3. Click Next and add any notes you might want to remember about this group. The notes can tell you what the group controls or why you created it. You can review and revise or delete these notes later at any time. These notes also popup in a window when you move the mouse over a group icon on a display.



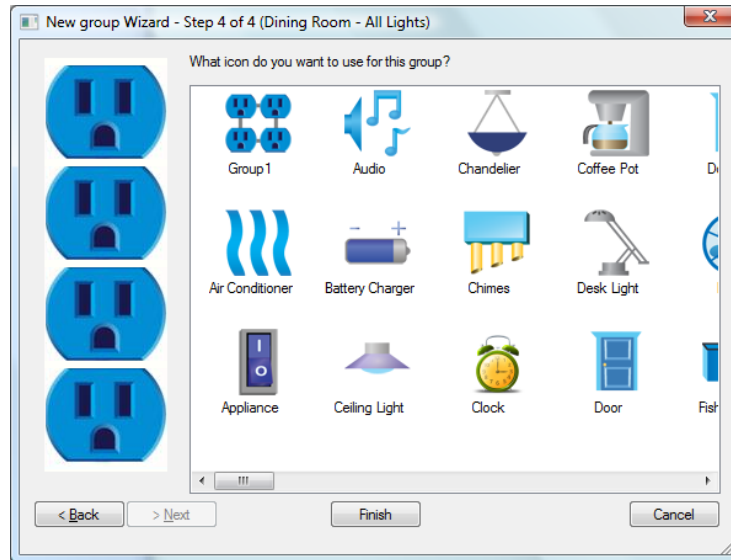
4. This is the step where you specify which devices and or programs are part of the group. There are two lists—*Not in group*, and *In group*. You can move device and program names between lists (that is, in or out of the group) by selecting the device or program name and clicking the appropriate arrow.



Group members can be devices or programs, but not other groups; so there are no groups listed in the left column for you to choose.

When you have the correct devices or programs included in your group, click Next.

5. Now you choose an icon to represent this group.



This screen lists all the icons that can be used for a group. Click the icon that you want to use, and then click Finish.

A last note on groups is that for a group you can create schedule entries that control the group (and each of its members). Refer to the Schedule and Schedule Entries Chapter for specific details.

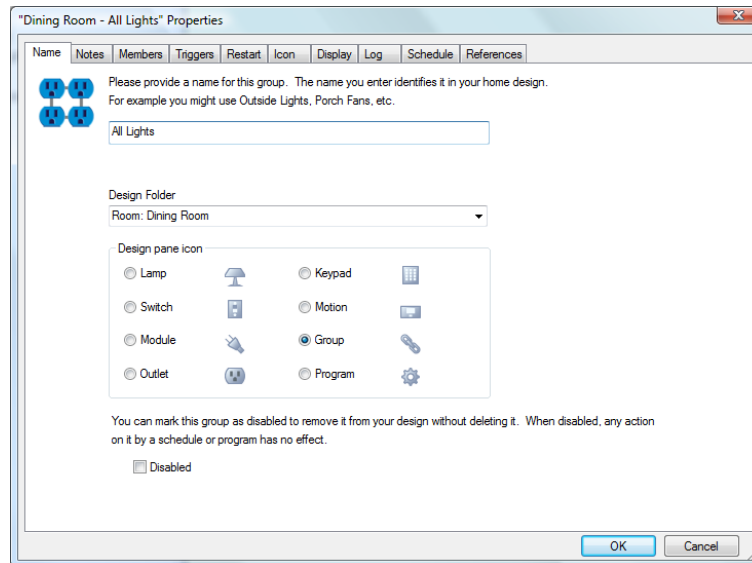
---

## Modifying an existing group

As with devices, you can inspect and change the properties for a group using the same methods as for devices. You do this in the Properties dialog for the particular group. There are two ways that you can get to the group properties:

1. Select either
  - the group icon on the display
  - the group name in the design pane
2. Click the right mouse button.
3. From the popup menu, select Properties.

The Properties dialog box title reflects the name of the group you selected. Each tab is like one step in the New Group Wizard.



4. Click the tab you want to change properties on, or click each to review the settings for this group. You can change whatever you like, and click OK to save your changes.

There are additional things you can do with a group using its property dialog that you can't do in the wizard. These are to establish triggers, select restart parameters, add a group icon to one or more displays, set log properties, show and modify schedule entries, and see where the group is referenced in your design. These properties are the same as for devices so refer to that chapter for information.



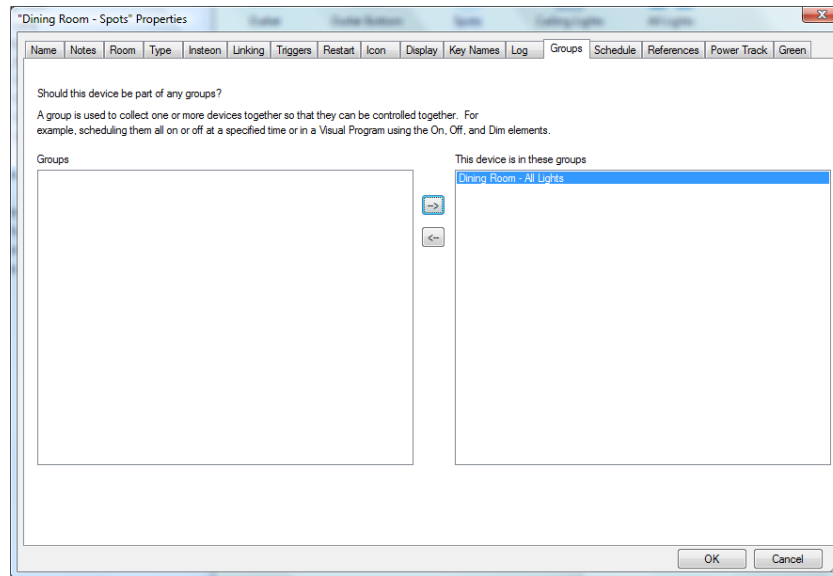
---

## “Group” tab in device properties

You can always open a group’s properties and change the group members but there is another way to modify a group.

The “members” tab of a group’s property dialog provides you an answer to the question: *What are the members of this group?*

When you open the properties of a device there is a “Group” tab that answers the question: *What groups is this device a member of?*



You can remove the device from a group or add it to a group using this tab.

**Hint:** Which way should you modify a group? By using the group properties “members” tab or the “Group” tab on devices? Either method accomplishes the same task – adding or removing members. Sometimes you want to focus on the device and sometimes on the group. It’s up to you.

---

## Adding to a group using drag and drop

You can always open a group’s property dialog and change the group members but there is another way to modify a group.

You can also add a device to a group by dragging the device name or icon and dropping it on a group name or icon. Using this method you can only add and not remove from the group.

---

## Deleting a group

If you find that you have created a group that doesn’t work the way you want it to, you have two options.

- You can modify the properties for the group. For instance, you can change the devices that it controls.
- You can delete the entire group.

### To delete a group:

1. Use the right mouse button to click the group name in the design pane of HCA.

2. Click Delete.

HCA removes the group name from the design pane, and removes the group icon from the display.

**Warning:** Be sure to check all aspects of a group before you delete it. If you delete a group, it is gone, and schedule entries you created for it are destroyed, and any programs that control it are put into a state where they will not start.

**Hint:** You can use the Undo button on the Quick Access toolbar or on the Design ribbon category to restore the deleted group. It will undo just the last change you have made—so if you delete two groups in succession, you can undo the deletion of only the second one.

---

## Groups and Local Stored Scenes

If any devices in the group are capable of storing scenes then the group can be thought of as containing those scenes as well.

For example, suppose that your group contains three devices: Light1, Light2, and Light3. Further, suppose that the three lights have these scenes defined:

Light1: Late Night, TV, Dinner

Light2: Late Night, Welcome Home, Dinner

Light3: Late Night, TV, Welcome Home

When you right click on the group, in addition to the ON, OFF, and Dim menu selections in the popup menu it also contains, Late Night, TV, Dinner, and Welcome Home.

If you select any of those scene names then the commands to activate that scene is sent to any of the group members that have that scene defined for it.

In the same way if you schedule the group and select a scene name, or create a program with a *Scene* element using the group and a scene, then HCA sends the scene activation command to those members with the scene. If you select a scene that is not defined in each group member, the inspector makes a note about the schedule or program because you may want to know that you are controlling the group with a scene that is not defined in each member.

## Chapter 10

# Programs and the Visual Programmer

Once you have created your HCA design with devices and groups, schedules, and schedule entries, you may want to take the next step and create programs to help control your home. The Home Control Assistant makes it easy for you to create programs by using two tools that work together: the New Program Wizard, and the Visual Programmer.

HCA programs are very capable. They provide a method of constructing simple sequenced programs: do this, do this, do that, done. The New Program Wizard makes it easy to begin a program, and the Visual Programmer provides a user interface that eliminates the need for a programming language with all its inherent difficulties. HCA also provides the ability to introduce conditionals into programs, thus allowing more complex programs to be created.

This chapter focuses on creating programs, the next chapter focuses on the Program Debugger.

Sections in this chapter are:

- Terminology
- Using the New Program Wizard
- Opening the Visual Programmer
- Areas of the Visual Programmer
  - Tool palette
  - Element list
  - Control buttons
  - Snippet Wizard
- Working with the programming canvas
  - About connecting elements
- Triggers
  - X10 Reception triggers
  - Insteon triggers
  - UPB triggers
  - Generic triggers
  - Magic Module triggers
  - Wireless triggers
  - Global Cache Sensor Triggers
  - Weather triggers
  - Flag triggers
  - Expression triggers
  - Special condition triggers
  - How are triggers evaluated
- Setting properties for program elements
  - Elements
- Constructing programs

- Test
- Repeat
- The Validate button
- Troubleshooting, or Getting programs to do what you want them to do
- Program properties Advanced tab—Examples

## Terminology

Although the Visual Programmer was designed to be usable for non-programmers, and does not use a lot of esoteric symbols and punctuation, there are still some terms used that you may want to become familiar with before you start using the Visual Programmer.

**Start**—When a program is *started* it begins running. Programs can be started in several ways.

The first is from the HCA display by using the popup menu from right clicking either a program icon in the display pane or the program’s name in the design pane. Programs can be also started when a trigger for it is received.

**Run**—Between the time a program is started and the time it finishes, it is said to be *running*.

**Execute**—While a program is running, each element is *executed*, that is, whatever the element is supposed to do, is done. If it is an element that turns on a light, when it is *executed* the light comes on.

**Element**—Each action that the program executes (does) is an element.

You draw programs by placing *elements* in the programming canvas, and link them together by drawing connecting lines. The program begins with the “Start Here” element and flows from element to element following the connecting lines in the direction of the arrows.

**Test**—A test is an element that allows the program to examine a condition and execute different elements based upon the outcome of that test.

**Trigger**—A condition that when it occurs starts a program running. This could be the receipt of a Insteon message, a UPB message, a weather condition, the change in value of a flag, etc

**Repeat**—An element that allows one or more elements to be executed (done) a number of times.

**Flag**—A *Flag* can hold values – numbers, strings, dates and times, or simple yes or no.

A program can use flags like pieces of note paper. Each flag has a name and a value, such as Yes or No. In a traditional programming language, a Flag is a variable that has no fixed type.

In the way that you might make notes to yourself while doing a complex task, programs can use flags to record things. A program can set a flag to a value and that value remains as long as HCA is running. For example, one program can set a flag to Yes while it runs. Much later another program can test the value of that flag to see if it’s Yes or No.

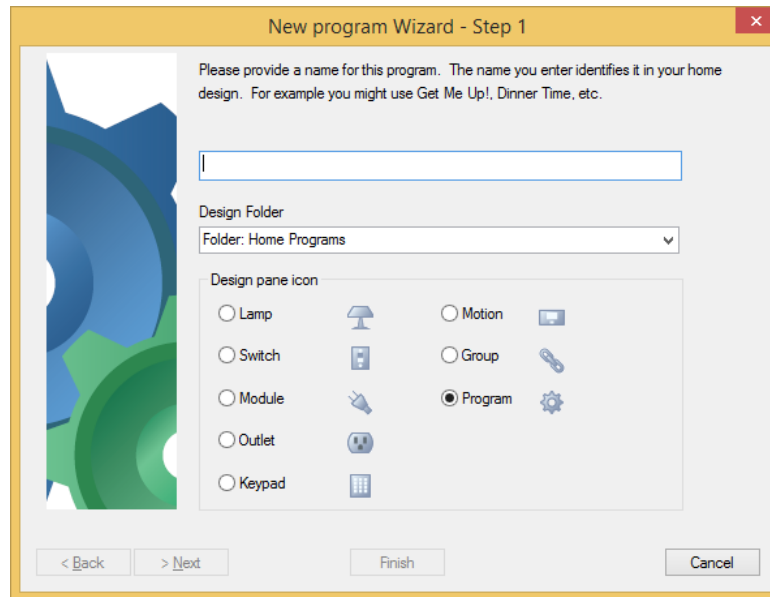
Now that you know the terminology, let’s look at HCA programs, what you can do with the Visual Programmer, and how you do it.

## Using the New Program Wizard

In the Home Control Assistant, you begin a new HCA program using the New Program Wizard. However, while the wizard creates the program, it doesn’t create what the program does. To create the program, after completing the wizard, you use the Visual Programmer.

The first step in creating a new program begins with the New Program Wizard.

In the *Design* ribbon category, click *Program* in the *New* panel. This opens the New Program Wizard. This wizard is very similar to the New Device and New Group wizards. The wizard prompts you through a series of steps as it collects the program name, notes, icon, etc. You can click Back any time to check or change a previous step.



Go through the steps as the wizard prompts you. You will fill in several types of information. Click Next to go each successive step.

1. Type in a name and type in or select an existing room or folder name. Each program in the folder must have a unique name
2. Add notes if you like. Like devices and groups these notes can appear in a popup window when you move the mouse over a program icon.
3. Choose the icon you want to represent this program.
4. You may go back and check any steps now, or click Finish to complete the definition of the program.

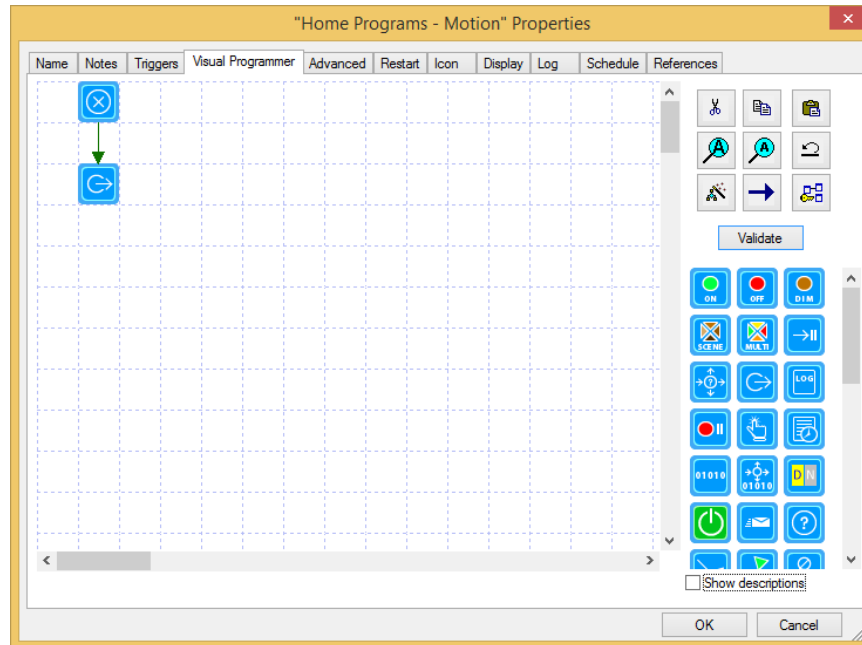
At this point, you need to switch to the Visual Programmer, and complete your program using it.

---

## Opening the Visual Programmer

The Visual Programmer is a tab on the properties dialog box for your program.

1. In the design pane or in the display pane select the program that you just created.
2. Right click and select Properties from the popup menu.
3. On the properties dialog box for your program (it will have the name of the program in the title bar), click the Visual Programmer tab.



There is one very important thing to note about this dialog. Look in the lower right hand corner. Note the standard Windows mark for a dialog that can be expanded. To make the dialog bigger, left click on that mark and drag the mouse to expand the dialog. Doing this makes it much simpler to see large programs.

---

## Areas of the Visual Programmer

The Visual Programmer tab contains a number of areas.

- The large area marked in a grid pattern is the *programming canvas*. This displays the programs that you create.
- At the upper right is a *Toolbox*, which provides a number of actions that you can take with the Visual Programmer.
- Directly below the toolbox is a button used to check the program to make sure it is all defined correctly.
- Below the validate button is a palette showing all the different elements that you can use to construct your program.

This section discusses each of these areas, except the programming canvas, which has its own section, following.

---

### Toolbox

The toolbox contains buttons for a number of operations that you might use when creating your programs. These operations are listed below (going from left to right, and top to bottom).

**Cut**—removes the current selection from the programming canvas and places it on the clipboard.

**Copy**—duplicates the current selection and places it on the clipboard.

**Paste**—Inserts the clipboard contents onto the programming canvas.

**Zoom in**—Makes program elements in the canvas appear larger.

**Zoom out**—Makes program elements in the canvas appear smaller.

**Undo**—Reverses your last change.

**Snippet Wizard** – A wizard that can create useful sequences of elements that can be pasted into your program.

**Connect program elements**—Enters into a mode to connect elements.

**Label elements**—Place text next to each element that shows what it does.

The first six of these tools are for activities that should be familiar to Windows users. The other operations – the snippet wizard, connect, and label, are described below.

## Element Palette

Below the toolbox is a palette containing all the different elements that you can use to construct the program.

What you see in the list may be different than the screen images presented here. As described in the *HCA Options* chapter you can configure which elements show in this list and in what order they appear.

There are two ways to view the Element Palette. The first method - shown in the screen image above – shows only icons for the elements. If you check the Show Descriptions checkbox below the Element Palette, the elements are shown with descriptions.

+ -

If you don't use "Show descriptions" you can hover the mouse over an element in the palette and an info tip names the element.

The Element Palette contains many different elements. These are the common elements:

Element Name	What it does
Start Here	This is the element that starts a program running. Each program has only one Start Here element.
Add to log	Add a message to the log.
Auto Off	Configure auto off settings for a device
Change icon	Change the icon for this program seen on displays
Change schedule	Make a different schedule the current schedule.
Compute	Assign values to flags
Compute Test	Test a flag's value
Day Night	Program selected keypads and switches for LED illumination level
Insteon	Read Insteon devices linking tables
Delay	Delay for a while. The time to delay is given in hh:mm:ss.
Dim	Control a device or group to Dim.
Email /SMS	Send email or SMS message
Exit	Close, stop, or end the program.
Get Status	Poll one or more devices that support status requests
Make flag No	Make the value of a flag be "No."
Make flag Yes	Make the value of a flag be "Yes."
Multi	Control multiple devices in one element

Not flag	Reverse a flag. If its current value is Yes, make it No. If its current value is No, make it Yes.
Off	Control a device, group, or room to Off. Start a program with an Off command.
On	Control a device, group, or room to On. Start a program with an On command.
Ping	Send a message to a network device to see if it replies or not
Play sound	Play a sound file through the computer's sound system.
Port I/O	Send or received with a serial or IP interface
Read Data	Read data from an external file
Repeat	Repeat a sequence of elements a specified number of times.
Request Input	Display a dialog to request user input
Resume	Remove the suspend condition from a device, program, or group.
Run	Start a Windows program
Scene	Send a command to activate or deactivate a scene
Script	Execute a text based script
Set Mode	Change Home Mode
Show Display	Change the display pane to show a named display
Show message	Show a message in the display pane for a few seconds or in its own window.
Speak	Uses a Text to Speech engine installed on your computer to say a phrase
Start program	Start another program and continue on with the current program when that program finishes.
Status export	Perform a status export
Stop program	Stops a running program
Suspend	Suspend a device, program, or group
Test	Test a condition, and execute different elements based upon the outcome of that test.
Update Tile	Update a tile in a tiled display
Wait Until	Wait until a time in the future. Time given as hh:mm or as sunset or sunrise.
Element Connect	Provides a way to join two elements without drawing a line between them.

These elements are for specific hardware that you may or may not have.

Element Name	What it does
Send IR	Sends a sequence of IR commands
Send X10	Sends X10 commands
Send ZWave	Sends ZWave commands



Thermostat	Controls a thermostat device
Thermostat test	Tests data retrieved from a thermostat
UPB Link	Send a UPB Link command
UPB Blink	Causes a UPB device to blink at a specified rate by sending the Blink command to the device
Weather Test	Tests data retrieved from a weather provider
X10 All lights on	Send an X10 All Lights On command to a house code.
X10 All lights off	Send an X10 All Lights Off command to a house code.
X10 All units off	Send an X10 All Units On command to a house code.

There are other elements besides those shown above for specific hardware that has been designated Legacy Hardware. To see these elements, select HCA – Properties from the menu and look on the Legacy tab.

**Hint:** It is suggested that you use the Visual Programmer tab of HCA Properties to configure the list of elements you see in the Element Palette. It makes no sense to show elements for hardware you don't have or elements you plan never to use.

---

### Validate button

Also in this area of the dialog is the Validate button. This performs a number of checks on your program making sure that all element properties have been set and all elements are connected correctly.

---

### Working with the programming canvas

Before discussing the details of the programming elements, let's look at ways you can use the programming canvas to add, delete, select, and set the properties of elements. You can use the programming canvas for any of the following functions:

- To select an element or more than one element
- To add a new element or delete an element
- To move an element
- To change the properties of an element
- To cut or copy an element to the clipboard
- To paste an element from the clipboard
- To connect two elements together
- See what an element does
- Add a note or comment about an element

Following are procedures for accomplishing each of these tasks.

**To select an element:**

1. Place the mouse over the element and click.

You can use the same method to select the arrow lines connecting elements.

**To select more than one element:**

1. Place the mouse over the first element and click.
2. Hold down the Ctrl key and move the mouse over another element and click.

Or

1. Left mouse button down someplace on the canvas – not on an element - and drag. Let up the mouse button to complete the rectangle. This creates a selection rectangle. Any element fully contained in the rectangle will be selected.

You can use either of these methods to select as many elements as you need.

**To add a new element:**

1. From the element list, select the element you want to add to the canvas and drag in on to the canvas and drop it where you want.

When you add a new element, the properties dialog for that element type automatically displays.

**To delete an element:**

1. Select an element or several elements – using either of the methods described above - and press the Delete key on the keyboard or right click and select Delete from the popup menu.

Whatever you have selected is deleted. You can also delete connecting lines in this way by selecting them and pressing Delete.

**To move an element:**

1. Select the element or elements to be moved.
2. Click on one of the selected elements and drag the element(s) to the new location. Whatever you have selected: element, elements, or connecting lines, are moved. Connecting lines attached to the moved element(s) are stretched to maintain the connections.

**To change the properties of an element:**

1. Right click on the element and select Properties from the popup menu.

Or

1. Double click on the element.

The popup menu shows the *properties* choice only if the element has properties.

**To cut or copy an element to the clipboard:**

1. Select the element(s)
2. Use the tools palette Cut/Copy buttons.

or

1. Right click on one of the elements in the selection, and choose Cut/Copy from the popup menu.

**Hint:** The operations to delete, move, cut, and copy all work whether you have one or several elements selected. The action that you choose happens to each element you have selected.

**To paste an element from the clipboard:**

1. Click the paste button from the tools palette. The element is pasted into the canvas on the first unused row - starting from the top and moving down.

or

2. Right click when over a blank cell on the programming canvas and select paste from the popup menu. This will paste the clipboard contents at the location.

If you have more than one element on the clipboard (that is, you selected multiple elements and then cut or copied), all the elements are pasted into the canvas. Any connecting lines between elements that were present when they were placed on the clipboard are retained when pasted.

### To connect elements together:

There are two methods of connecting elements.

#### Method 1:

This method for connecting elements requires a steady hand with the mouse, but is worth learning since it can be quick once you get used to it.

1. Select the *From* element.
2. Carefully move the mouse pointer to the edge of the element. If you get the pointer in just the right place, it changes to the small circle with a dot.
3. Left click and drag a connecting line to the *To* element. Release the mouse when over the target element.

#### Method 2:

This method is useful for hooking up a whole series of elements.

1. Click the line tool in the toolbox and the cursor changes to the "circle with dot".
2. Now click on an element (let's call it A) you want as the source for the line. Just click. Don't click and drag!
3. Now click on the element you want to have the line end in (let's call this B). The line is added from A to B. Note that the cursor is still the "circle with dot".
4. Now click on another element (called C). A new line is created from the B to C.
5. And you can keep going, connecting C to D, D to E, etc.
6. To end, just click anywhere not on an element.

It's important to remember to click someplace on the canvas to end this style of connect. Until you do, you will not be able to, for example, press the Validate button, OK, button, change to another tab, etc.

**Hint:** There is another way to connect two elements without a direct line between them. Refer to the About Connecting Elements section.

### See what an element does:

Once you have added an element to the canvas and set its properties, you can quickly see what it does.

1. Move the mouse pointer over the element.  
At the lower left corner of the display, directly below the programming canvas, you will see a description of what the element does when executed. If you have not yet set the properties for the element, the description reads *Element not ready*.

Or

1. Press the Label Elements tool in the tools palette. This places text next to each element that shows in a very brief form what the element does.

**Hint:** It is a good idea to leave some space between elements so the label text doesn't overlap other elements.

**Add a note or comment about a program element:**

Use the right mouse button when positioned over an element and select *Comment* from the popup menu. Enter any notes you have about this element into the dialog box that appears. If an element has any notes, these display in a popup window when the mouse is moved over the element.

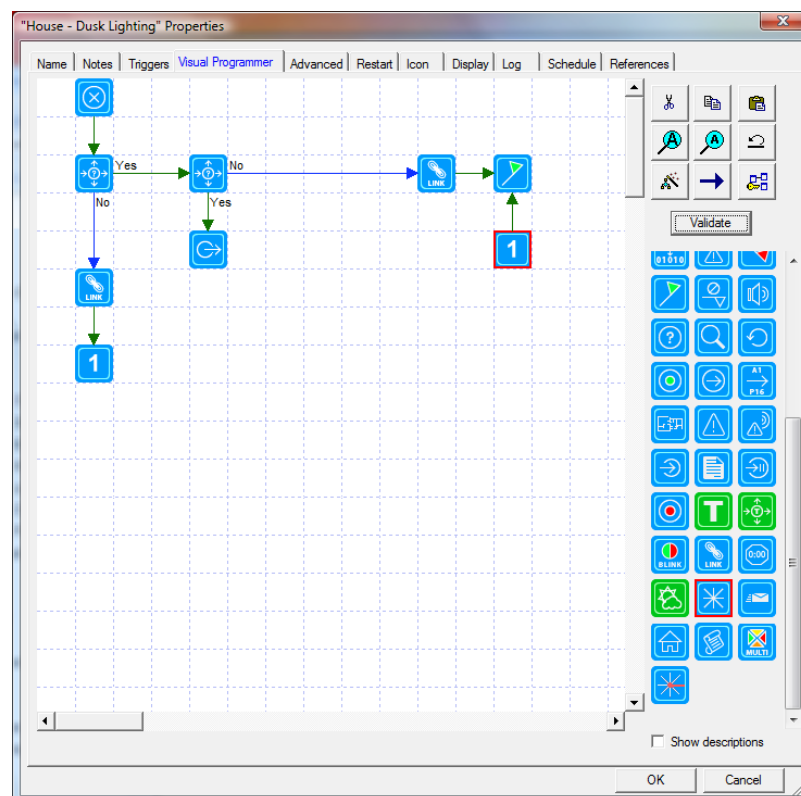
**About connecting elements**

You control the sequence in which your program executes its elements by connecting the elements. The program begins running with the *Start Here* element, and continues from element to element by following the connecting lines in the direction of the arrows. If you have two elements with a connecting line between them, the *From* element is where the line starts, the *to* element is where it ends. The *To* element has the connecting line arrowhead pointing at it.

If you try to draw a connection that the Visual Programmer does not support, the connecting line will not appear when you complete the drag. Here are the connections that Visual Programmer considers invalid, and that it does not support:

- You can't connect to the "Start Here" element from any element. It is the beginning—no element can be in sequence before it.
- You can't connect to any other element from an Exit element. It's the end—no element can come after it in sequence.
- You can't have more than one connecting line from any element to another, except for the test and repeat elements.

If you create complex programs you may find that you have lines crossing elements and that can be hard to read. One way to solve this is to use the connector elements. These act to join two elements together without a direct line. For example:



In this example when execution completes the link element then it continues to the make flag yes element.

When you add a connector element you set the connector number from one to nine. In this way you can use connector elements to connect different parts of your program.

---

## Snippet Wizard

The snippet wizard is started from the tool palette button, the one to the left of the arrow button. What it does is to create sequences of elements and places them on the clipboard. Once the wizard is done, you can paste these into your program where they are needed.

There are three different snippets that the wizard creates:

- Control lights over time. A sequence of elements to increase or decrease the illumination level of a device or a given time.
- Establish Scene. This captures the current illumination level of a set of devices you select. Elements are created to send commands to set those lights to those same illumination levels.
- Test for start conditions. Constructs a sequence of test elements to test for all the possible triggers for this program

Depending upon the snippet, the wizard may ask a series of questions or not. Once complete the snippet is on the clipboard. Select the cell where it should go, right click and select Paste from the popup menu.

---

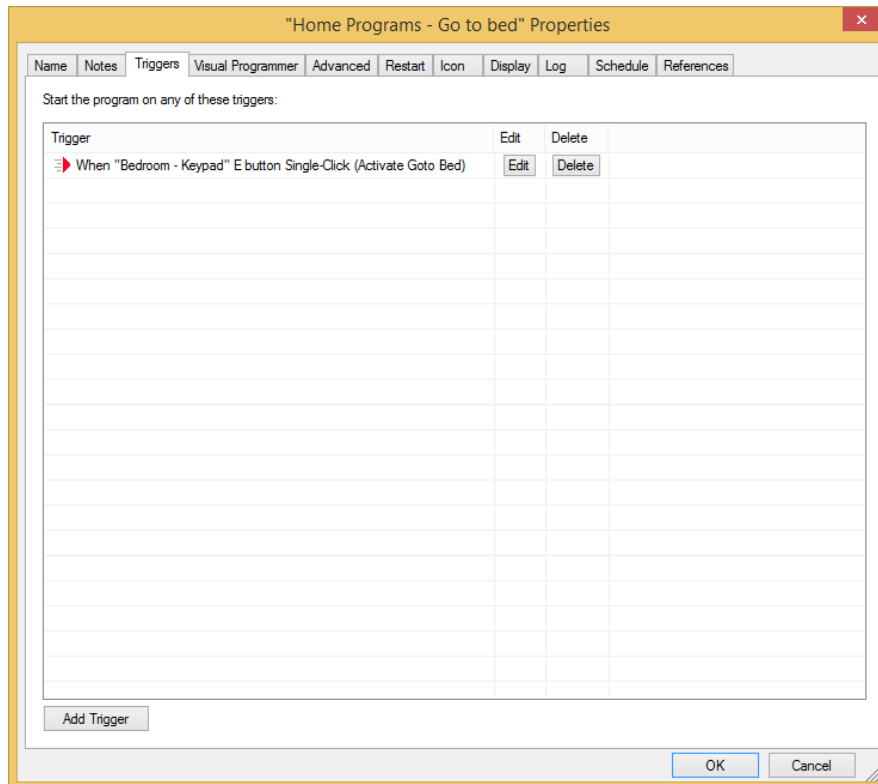
## Program Triggers

Before discussing what programs can do, let's first examine how program can be started. Programs, unlike schedule entries, start when something happens in your home and usually not at the same time each day. There are several possible kinds of triggers that start programs. These are:

- Expression
- Flag Value change
- Generic trigger
- Global Cache Sensor reception
- Home mode change
- Insteon reception
- Port reception
- Special condition
- State change
- UPB Action
- UPB Powerline message
- Weather condition
- Wireless Component Message
- X10 reception
- Zwave reception

There may be other trigger types if legacy hardware support is enabled.

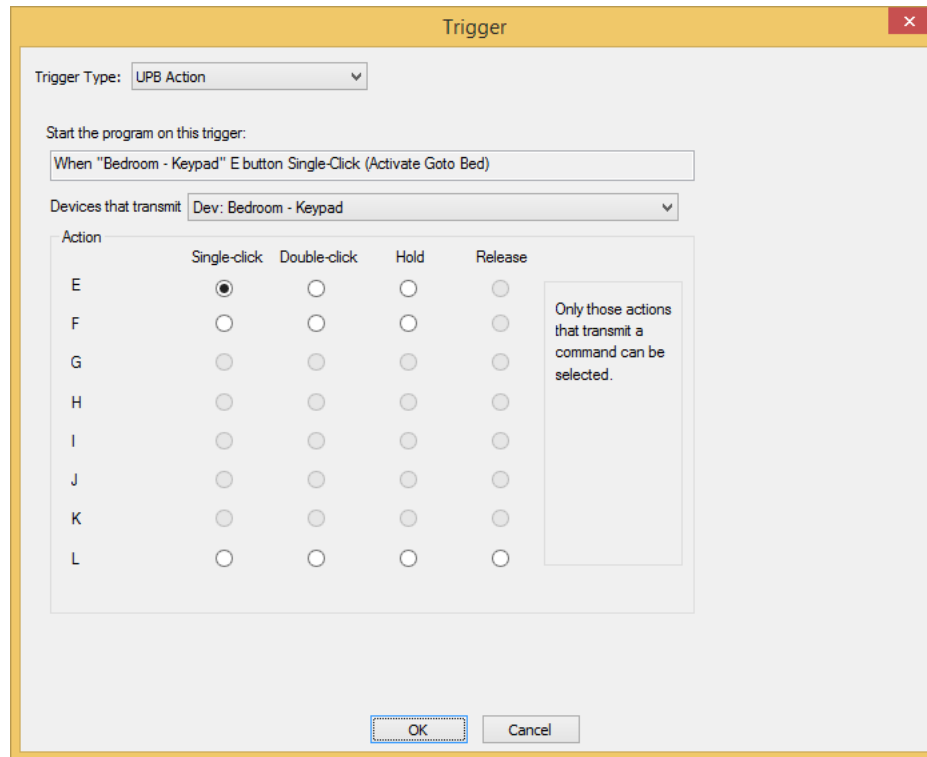
When you open the program properties and select the Triggers tab this page appears:



In the big list at the center of the tab is a list of all the triggers that can start the program. In this dialog you can perform three actions:

- To delete a trigger, press the Delete button on that row.
- To modify an existing trigger, double-click it or press the Edit button on that row.
- To add a new trigger, double-click on an empty row or press the *Add Trigger* button.

When adding a new trigger, a dialog opens where the configuration of the trigger is set. Each trigger type has configuration appropriate to its type. The example below shows a UPB Action trigger:



At the top of the dialog is a dropdown where you can select the type of trigger you want to add. In this example, a UPB Action trigger is being added.

Each trigger type has different properties and is shown in the next sections.

**Hint:** The Snippet Wizard can construct a series of Test elements to see what trigger started the program.

---

## X10 Reception Trigger

This is the Add Trigger dialog when adding an X10 trigger.

In the Address portion of the dialog is specified the house and unit code. This can be done in four possible ways:

- Select the house and unit code
- Select the house and unit code assigned to a device. If the device has multiple units, for example a multi-button keypad, select which unit of the device to use. The individual units are shown as the name of the device followed by a colon then a number.
- Select the house and unit code assigned to another device.
- Select a group

The first case is simplest. All you need do is to pick a house and unit code.

The next two options have a special behavior. If the X10 address of the device is changed, then this trigger for the program also changes.

For example, assume you created a trigger and selected the “Use the address of this device” option. The device selected was “Keypad Entry:2” and “Keypad Entry” has a X10 address of L7. This would assign to this program a trigger of whose address is L8. If the keypad was later changed to D2, the trigger address automatically changes to D3.

The last case is a lot like assigning a trigger address using a device. In the case of selecting a group, the address of any of the device members of the group is the address of the trigger. For example, assume you have a group called Lamps. Also assume that in this group are three devices whose addresses are C1, D1, and E9. If you create a trigger and chose the group address option and select group Lamps, then the address part of the trigger are the addresses C1, D1, and E9. If you change the address of any of the group members, or add or remove members from the group, the addresses for this trigger changes also.



In the lower section of this dialog you specify the command that must follow the address to trigger the program. Listed here are all the X10 commands. You can also choose a specific preset dim level or a range of levels that trigger the program. The final option of the command list is, in effect, no command. If you choose this option the program triggers when the address selected is received regardless of what command, if any, follows.

---

## Insteon Reception Trigger

This is the Add Trigger dialog when adding an Insteon trigger.

The screenshot shows a dialog box titled "Trigger" with a close button (X) in the top right corner. The dialog is configured as follows:

- Trigger Type:** Insteon Message
- Start the program on this trigger:** When Bath - Bath Lights sends On by Device Transmit
- Device sending command:** Dev: Bath - Bath Lights
- Component:** Device Transmit
- Command:** On (selected), Off, Dim, Bright, Fade Start, Fade Stop, All On, All Off, Fast On, Fast Off

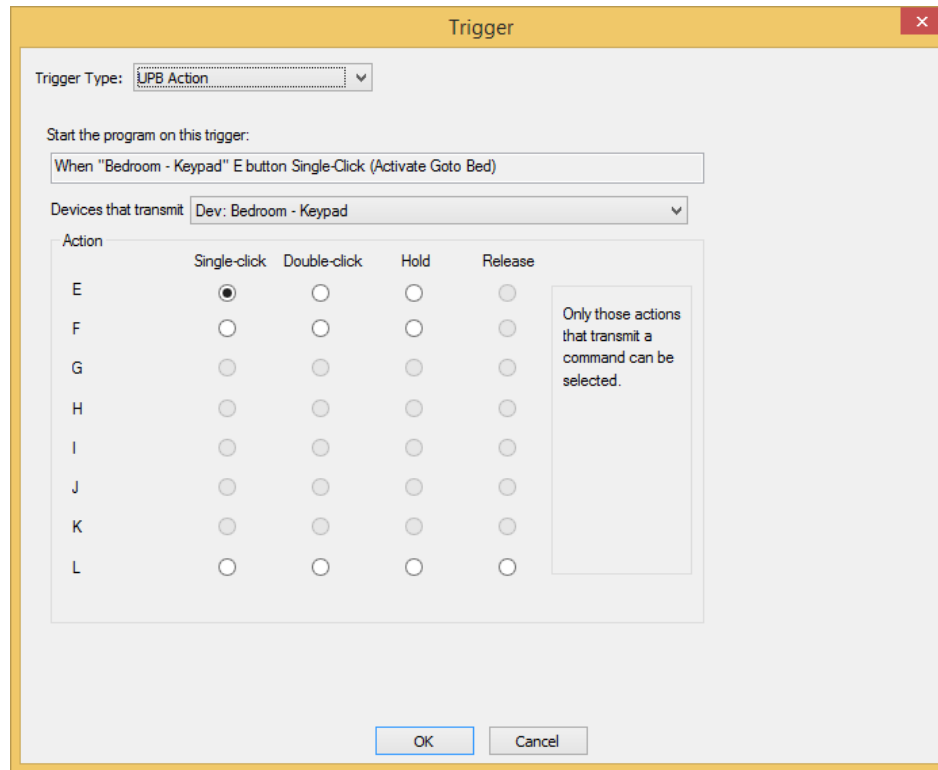
At the bottom of the dialog are "OK" and "Cancel" buttons.

This type of trigger is used to respond to an Insteon powerline message.

Insteon power line message triggers are described in the Insteon Appendix.

## UPB Action Trigger

This is the New Trigger dialog when adding an UPB Action trigger.



This type of trigger is used to respond to an action taken at a UPB transmitter – a keypad, input module, or switch rocker.

UPB Action triggers are described in the *UPB Appendix*.

## UPB Powerline Message Trigger

This is the New Trigger dialog when adding a UPB power line message trigger:

The screenshot shows a dialog box titled "Trigger" with a close button (X) in the top right corner. The dialog is configured as follows:

- Trigger Type:** A dropdown menu set to "UPB Powerline Message".
- Start the program on this trigger:** A text box containing the text: "When 'Home - Garage Keypad' sends [Goto 000 xxx.xxx] to all receivers of 'Any'".
- Source:** A dropdown menu set to "Dev: Home - Garage Keypad".
- Destination:**
  - Direct: A dropdown menu set to "Any" and a "Unit id:" field with a spinner set to "0".
  - Link: A dropdown menu set to "Leave Home" and a "Link id:" field with a spinner set to "1".
- Message:** A dropdown menu set to "Goto Off".
- Custom:**
  - Match Command: A dropdown menu set to "Goto".
  - Match argument 1: Two spinner boxes, both set to "0", with a "to" label between them.
  - Match argument 2: Two spinner boxes, both set to "255", with a "to" label between them.
  - Match argument 3: Two spinner boxes, both set to "255", with a "to" label between them.

At the bottom of the dialog are "OK" and "Cancel" buttons.

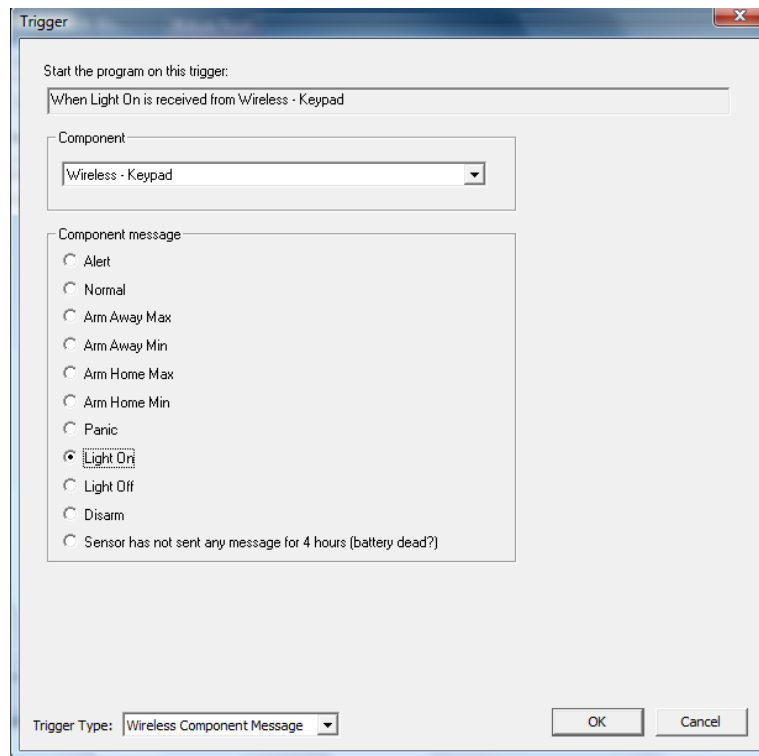
This type of trigger is used to respond to any UPB power line message.

UPB power line message triggers are described in the UPB Appendix.

---

## Wireless Component Message Trigger

This is the New Trigger dialog when adding a Wireless Component trigger:



In conjunction with a Wireless interface, messages from wireless door/window open/close sensors, non-X10 motion sensors, and security keypads, can be received and processed by HCA.

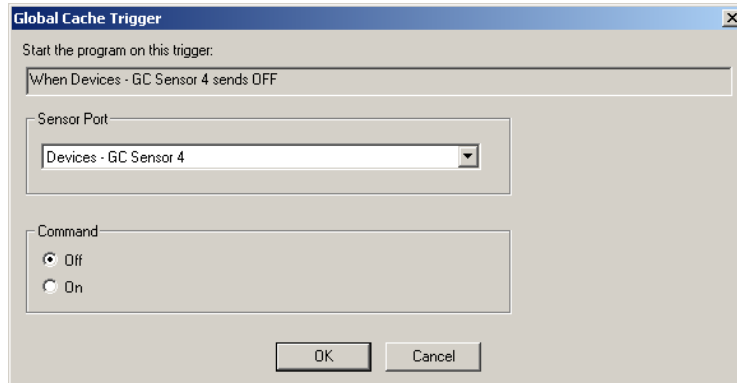
Once you have added wireless components to the design, programs can start when message are received from them.

Wireless Component Message triggers are fully described in the appendix on *Wireless Interfaces*.

## Global Cache Triggers

A Global Cache trigger is used to start a program when a Global Cache port configured as a *Sensor Notify* detects a change.

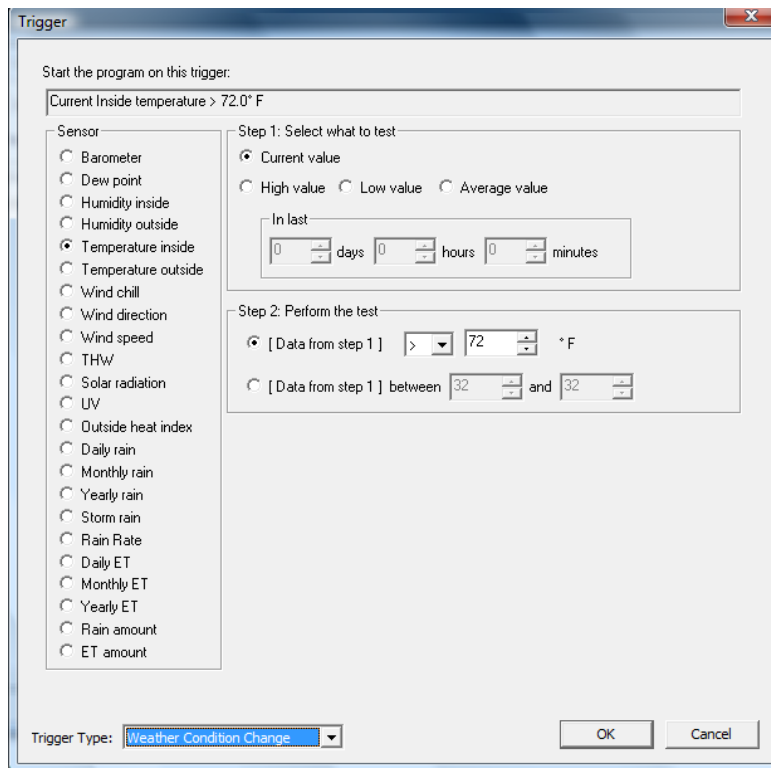
This trigger type dialog appears as:



Global Cache triggers are described in the *IR* appendix.

## Weather Condition Triggers

The Weather Condition trigger dialog appears as:



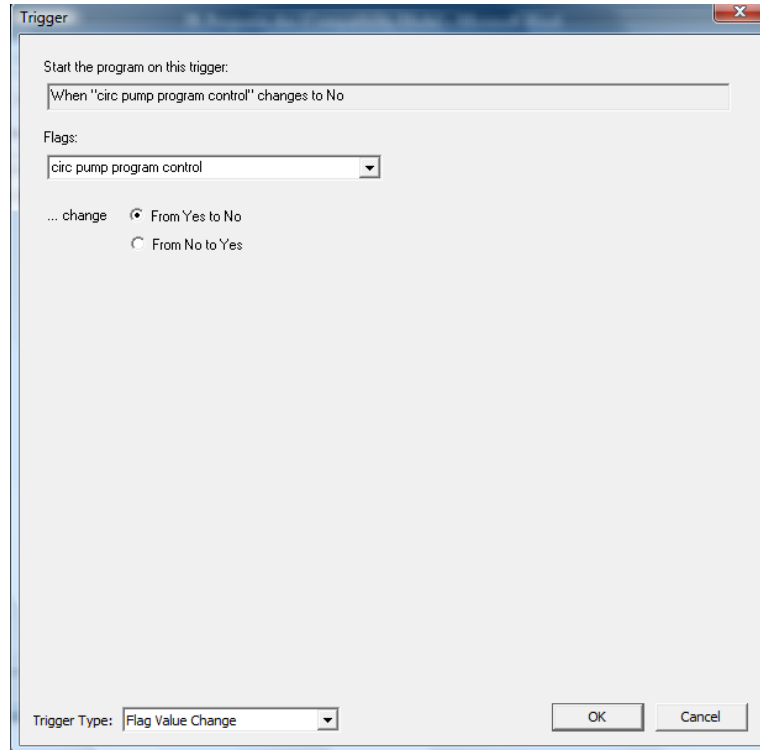
The Weather Condition trigger provides a means for you to create programs that start when data from a weather provider passes some test. For example, you can create a program that starts when the outside temperature goes over 80 degrees.

Weather Condition triggers are fully described in the appendix on Weather Providers.

---

## Flag Value Change Triggers

The Flag Value Change trigger dialog appears as:



Flags can hold Yes and No values and can be created with the Make Flag Yes, Make Flag No, and the Compute elements. This type of trigger doesn't test what the flag value is, just what it changes to. For example, suppose you have these three programs:

- Program A. Sets Flag "It's Dark" to No.
- Program B. Sets Flag "It's Dark" to Yes.
- Program C. Has a trigger that says: When "It's dark" changes to No

When program A runs nothing happens. When program B runs, the change of the Flag to NO causes Program C to start.

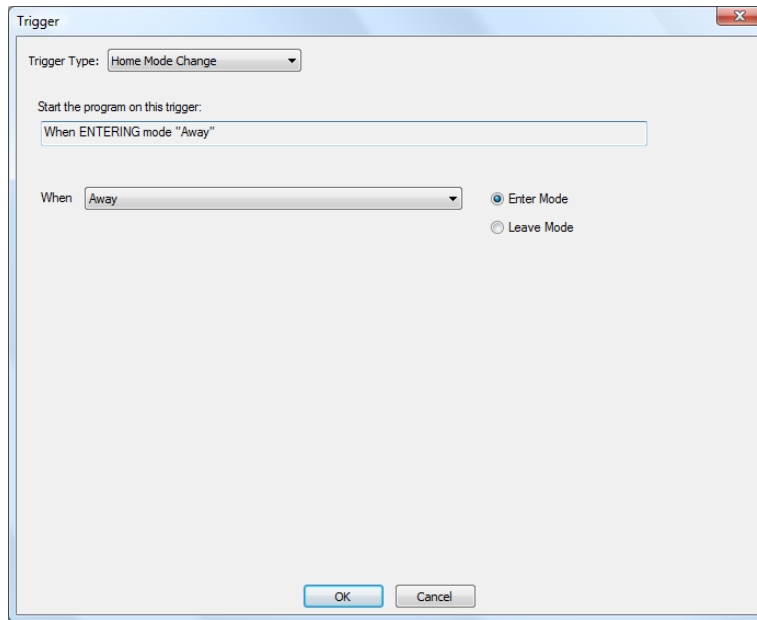
Why wouldn't program B just start program C directly instead of relying on the flag change trigger? Well it could but it just may be more convenient to do it this way.

This is one place where the Flag Change Event is the best method to use. In the date book (see the chapter on the Date Book), you can specify that when a selected date becomes "today", that a flag is assigned a value. Using the date book and flag change triggers you can have programs start when that date book entry becomes "today's" entry. Again, why not just schedule the program to start on this day? Well it could be done that way but this may be more convenient.

---

## Home Mode Triggers

This is the Add Trigger dialog when adding an home mode change trigger:



This type of trigger happens when the current home mode changes. Any mechanism that resulted in the mode change, a program changed the mode or it was changed by user action, can cause programs with this type of trigger to start.



## Port Reception Trigger

This is the Add Trigger dialog when adding a port reception trigger:

The screenshot shows a dialog box titled "Trigger". It contains the following elements:

- Trigger Type:** A dropdown menu set to "Port Reception".
- Start the program on this trigger:** A text input field containing "When \*.\* matches a reception from Lutron Interface".
- Serial or IP Interface:** A dropdown menu set to "Lutron Interface".
- Reception matches:** A text input field containing the regular expression \*.\*. Below this field is a "Validate Pattern" button.
- Assign matching reception to:** A dropdown menu set to "<Unused>".
- Trigger on a reception not otherwise handled:** An unchecked checkbox.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

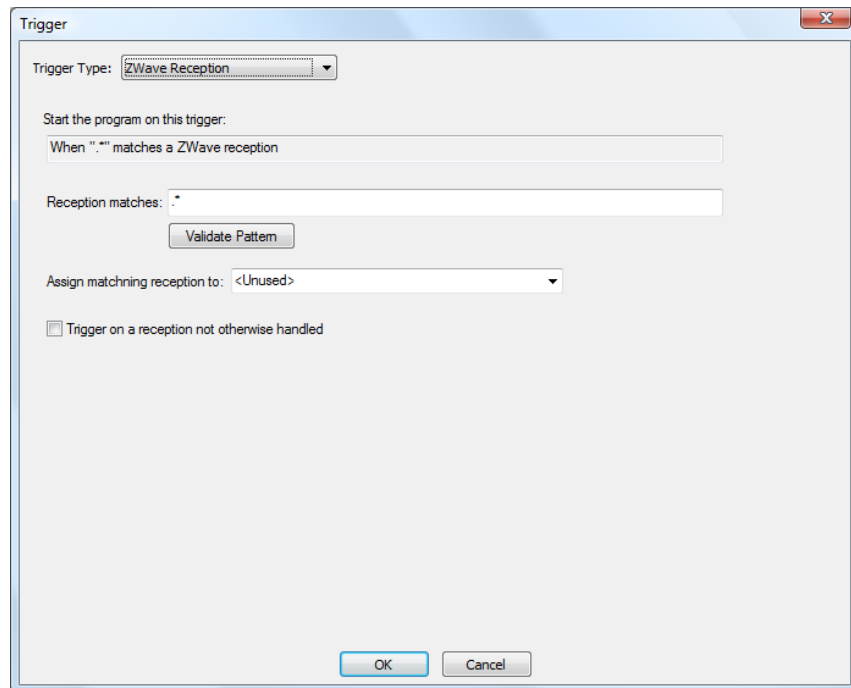
This type of trigger happens when a reception from a Generic Serial Interface or Generic IP interface occurs.

If the regular expression in the trigger matches the reception then the program starts. A program can also be designated as the program that starts if no other program triggers on the reception. This could be because no other trigger on any program matches the reception or because no other program has any triggers for this interface.

If the trigger does match then the reception message is optionally assigned to the flag entered or selected in the dropdown.

## ZWave Reception Trigger

This is the Add Trigger dialog when adding a zwave reception trigger:



This type of trigger happens when a reception from a ZWave interface occurs.

It is configured similarly as the Port Reception trigger except that you don't special an interface since it uses the Zwave interface and there can be at most one of those in use.

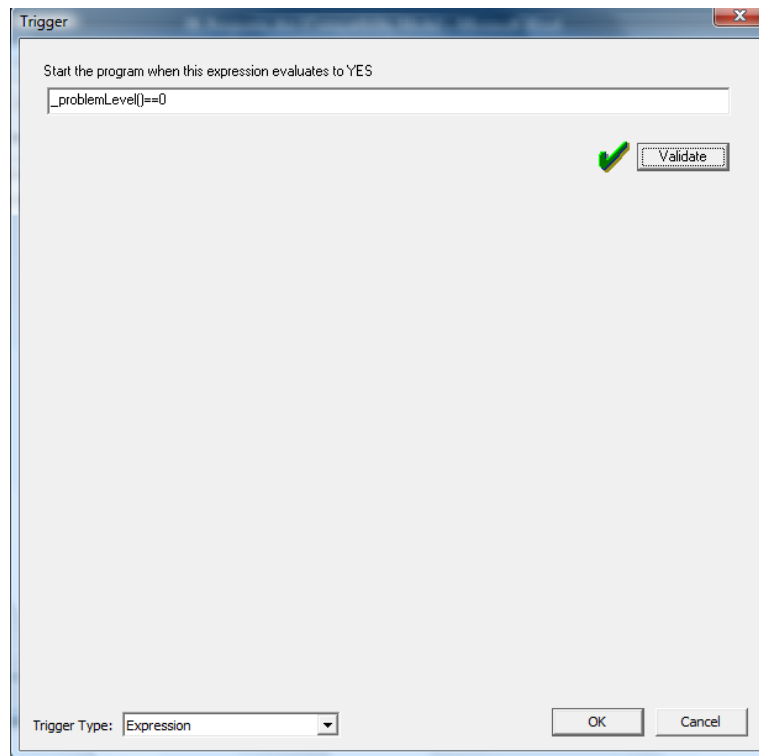
If the regular expression in the trigger matches the reception then the program starts. A program can also be designated as the program that starts if no other program triggers on the reception. This could be because no other trigger on any program matches the reception or because no other program has any triggers for this interface.

If the trigger does match then the reception message is optionally assigned to the flag entered or selected in the dropdown.

---

## Expression Triggers

This is the Add Trigger dialog when adding an expression trigger:



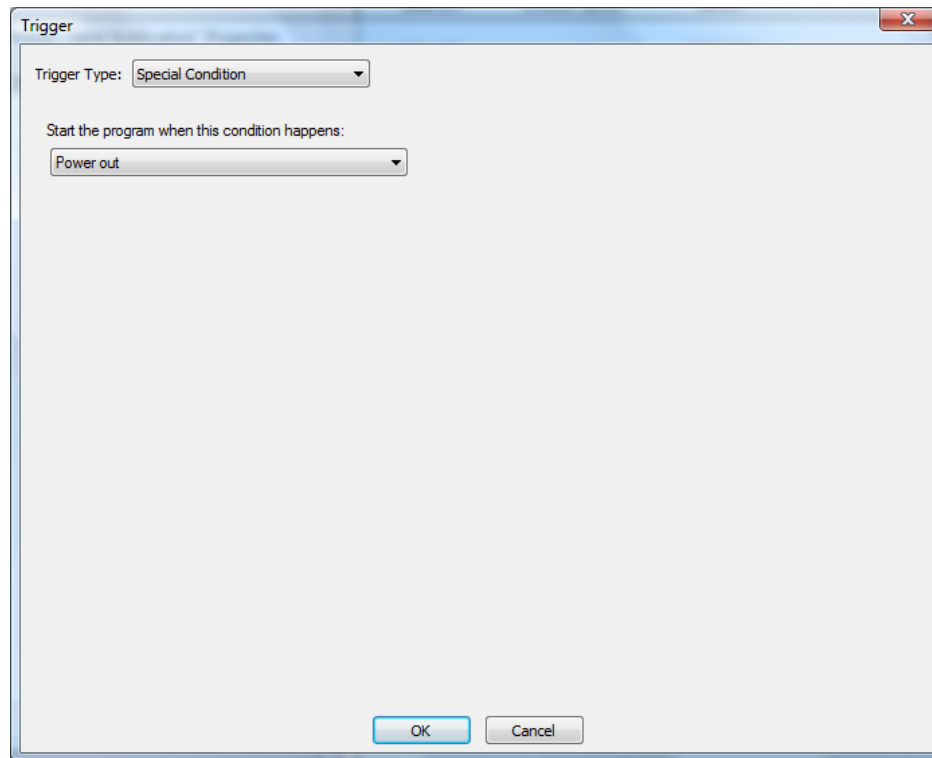
The expression trigger is the most general type of trigger. What you enter into this dialog uses the same expression language used in the ComputeTest element.

Whenever this expression evaluates to Yes, the program starts.

---

## Special Condition Trigger

The Special Condition trigger dialog appears as:



In this dialog you create trigger for special conditions that occur. These are:

- HCA started normally
- HCA started from a power failure
- Power Out
- Power restored

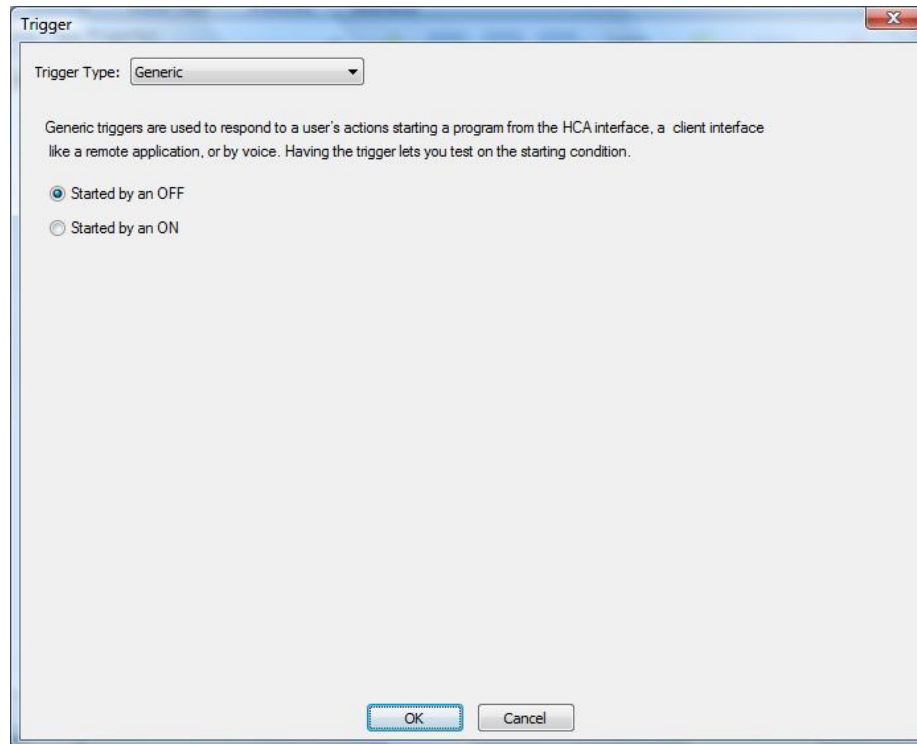
The first two conditions give you the means to have a program run when HCA first starts. The second two conditions are only useful when your computer is on a UPS backup system. When HCA detects that the house power is off, and programs with a "Power Out" trigger start. When Power returns, any programs with a "Power restored" trigger start.

**Hint:** The Power Out and Power Restored triggers are for legacy designs. The better method to handle these conditions is using the Alert mechanism of the Troubleshooter. See chapter 11 for information on the Alert Manager.

---

## Generic Trigger

The Generic trigger dialog appears as:



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 for when a program is started from a client.

Generic triggers provide the ability to create triggers that help you get the action you want when a program is started from the user interface of a client.

---

## Trigger Evaluation

There are a few important additional points to consider when using Weather triggers, Flag triggers, and Expression triggers.

The properties of the trigger define when the program starts. For example, the trigger "When the outside temperature is over 80" starts the program when the weather provider tells HCA that the outside temperature has risen to 80. But when does the program start again? The temperature may stay above 80 for several hours. Does the program start each time a weather observation is made?

No. The program only starts the first time the temperature is reported over 80. The temperature must drop back below 80 and then rise above 80 before the program starts again.

Think of the trigger being in two possible states: *Ready to trigger* and *triggered*. If the temperature is under 80, then the trigger is *ready*. Once it goes over 80 the program starts because the trigger has *triggered*. It can't trigger again until it first goes back to the *Ready* state.

This is the same logic used by the Flag triggers and expression trigger. The expression must evaluate to NO to be *ready*, then evaluate to YES to *trigger* and then back to NO to become *Ready* again.

The last important point is that HCA evaluates expression and weather triggers about every 60 seconds. This means that the expression trigger is only looked at once a minute. If the condition the expression is based upon changes more rapidly than once a minute, HCA may miss some of those changes.

## Parameterized programs

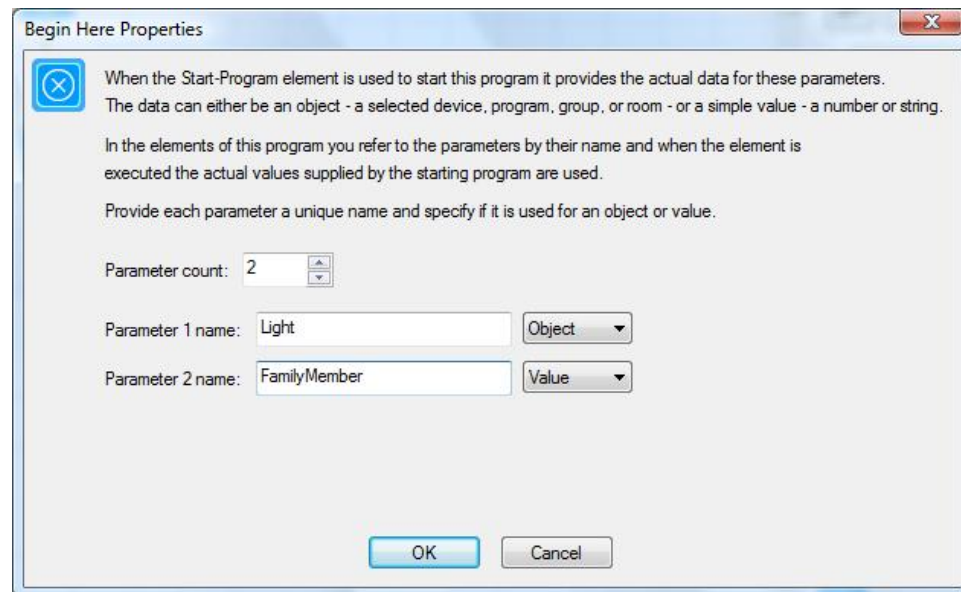
In previous versions of HCA one program could start another program but, unlike a more traditional programming language, there was no way to pass “parameters” to the started program. You could use flags (variables) for this but they are always global to the whole design.

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.

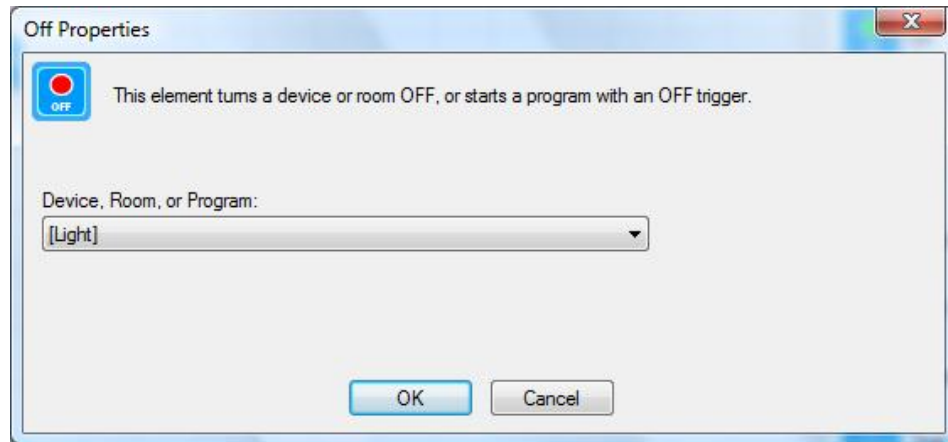
In traditional programming language, Program B is a “subroutine” of A.

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:



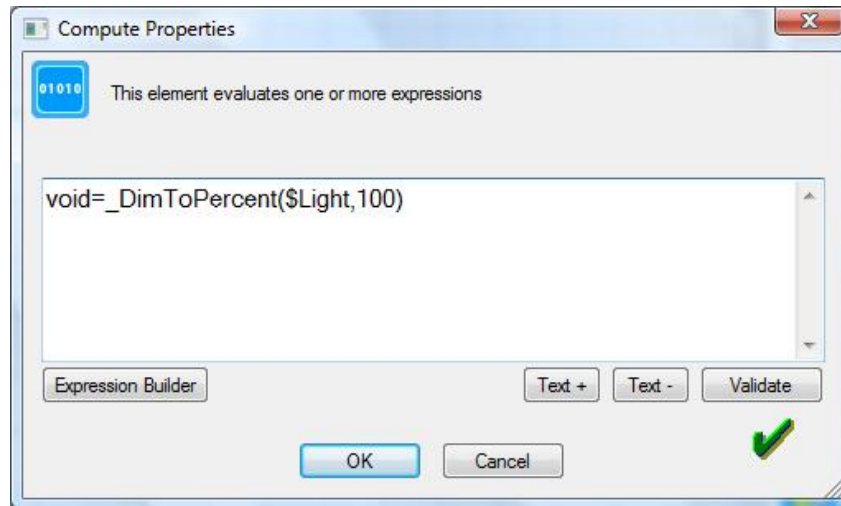
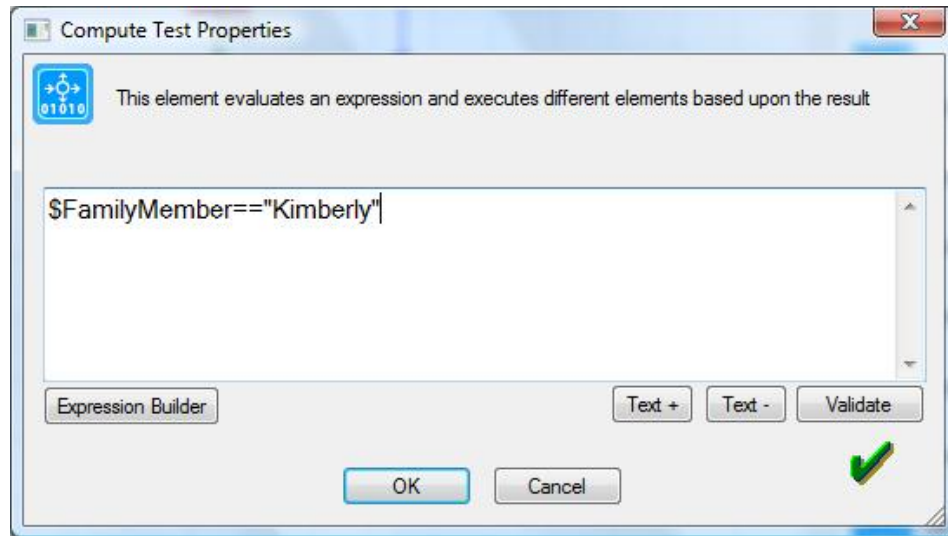
Note how the selection, in addition to all the usual “on-able” things are the parameter names. It added to the dropdown those parameters specified in the Begin-Here element that are used for objects. In this example, in the drop down you will not see “[FamilyMember]” since it is a parameter for a value.

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

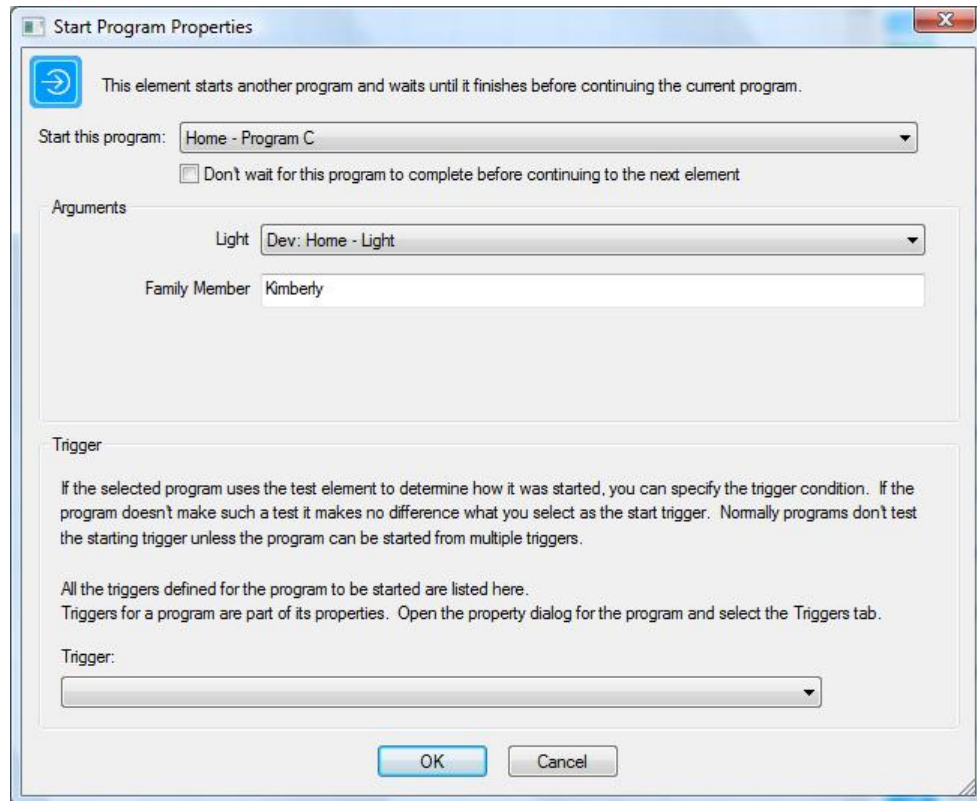
In a compute or compute-test element you can also use the value parameters as well as the object parameters. Here are two elements from this program. The first is seeing which Family Member is being targeted and the second controls the light to a level.





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%

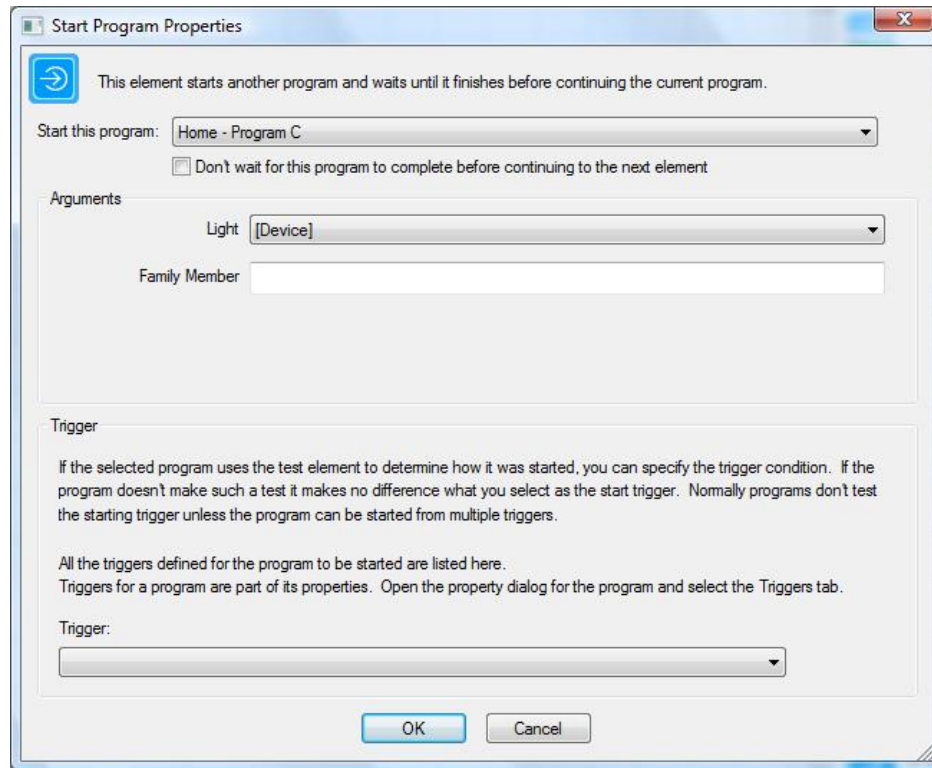
The started program would receive a parameter of text "Count is 24"

### 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.

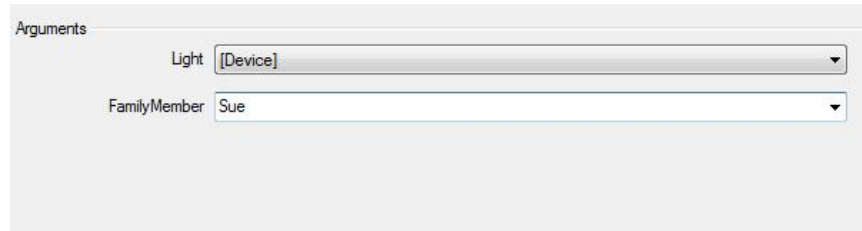
Things get a bit more complicated for value parameters due to the UI. When you are filling in the Start-Program properties for a value parameter you want to be able to enter in text and also to be able to select one of the program parameters.

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”



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



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?

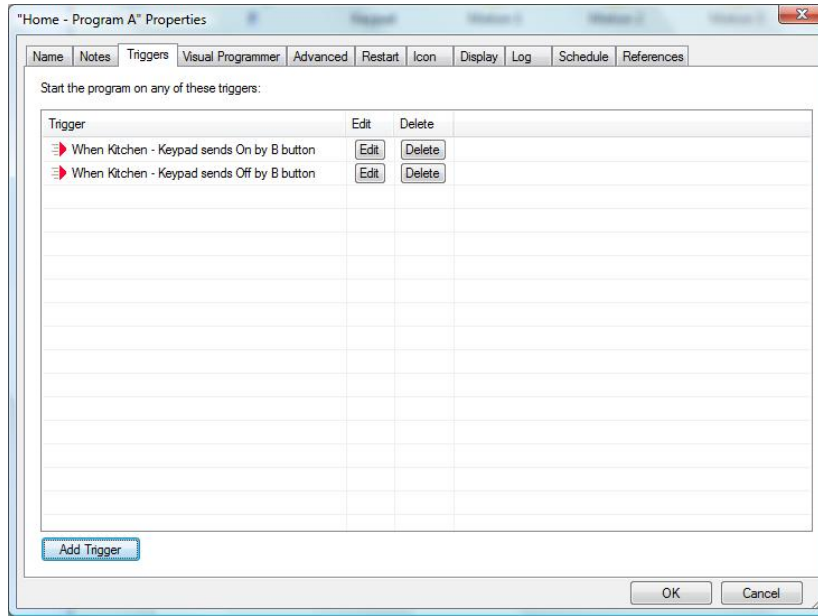
Simplest case: Suppose that program A takes a single object parameter.

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.

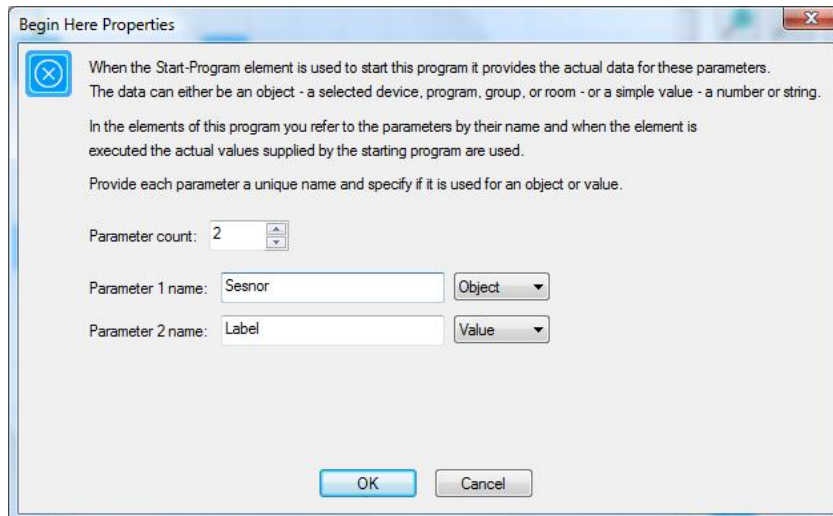
This is of course only good for triggers that have an associated object. The X10, Insteon, and UPB triggers have an associated object. A weather trigger for example doesn't.

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.

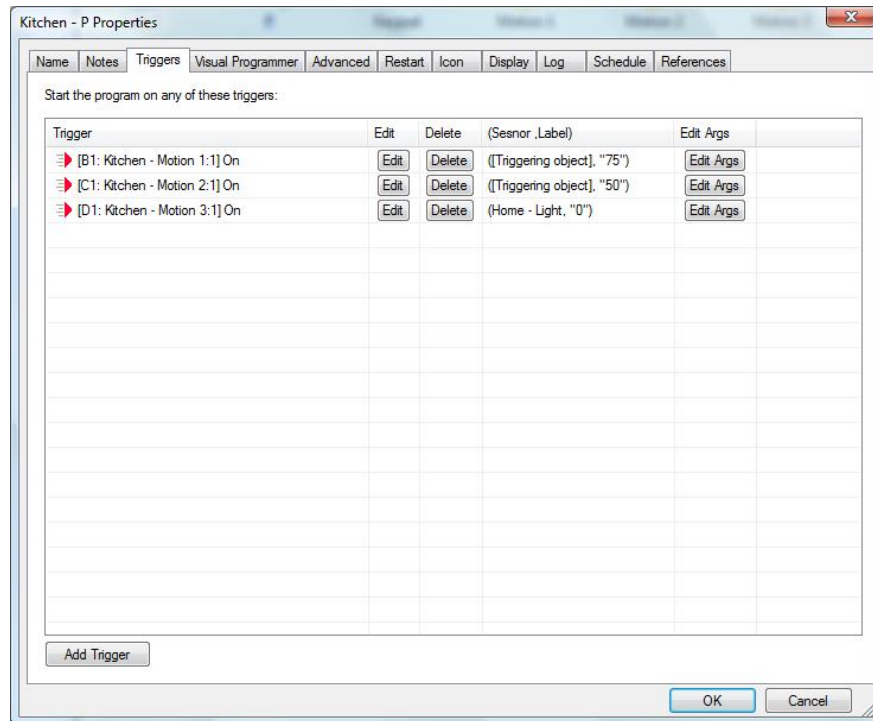
To implement this, the trigger tab in an object's properties has been changed. Here is what the trigger tab looks like now for programs without parameters.



Now suppose that the program takes parameters. For example:

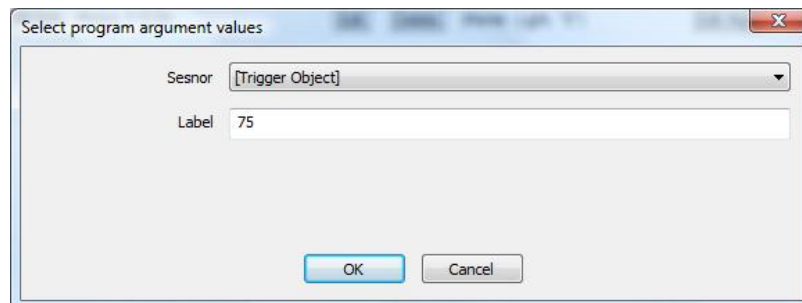


Now when you are on the triggers tab for this program, this is what you would see:



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.

---

## Setting properties for program elements

Now that you know how programs start and how to draw them using the program canvas, let's look at the elements of those programs, and see what they can do. You control what an element does by setting its properties, using the properties dialog box specific to that element.

**Hint:** Most programs will not use all the different types of elements that HCA provides. The sample file provided when you installed HCA contains several simple and several complex programs. A good way to learn about the Visual Programmer is to take a look at what these programs do.

**web tip:** Refer to the technical notes section on the web site for more information on the MyHome sample file and its programs.

There are two ways that you can open the properties dialog box for an element. The first way is automatic: as you add an element to the program canvas, its dialog box opens so that you can set the properties for the element.

The second way is to use the right mouse button:

- Click an element and choose Properties from the popup menu or double click on the element. The dialog box for the element opens so that you can enter or change its properties.

---

## Elements

This section lists the various elements used in programs and provides details about how you can set and use their properties. The properties dialog boxes for the elements are very similar and very easy to use. They have different components, many of which will be familiar to you.

Note all elements are fully explained here. Some are very specific to some automation hardware. These are fully explained in the appendix for the hardware they apply to.

---

## Start Here

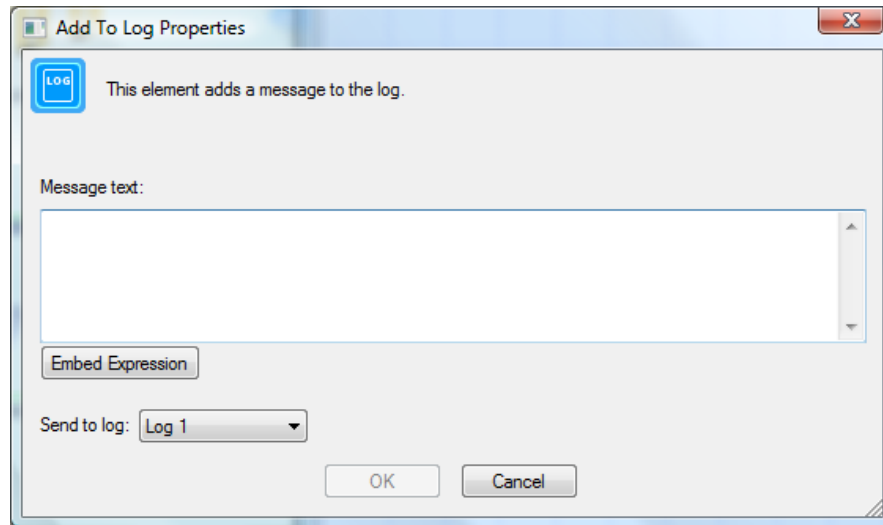
This element provides a marker for where a program starts running. It has no properties.

---

## Add to log

This element adds an entry to the log. To see that entry, open a log viewer from the Control category.

You might use this element if you are attempting to determine why a program is not doing what you want it to do, or if you want to make a note whenever the program is started.

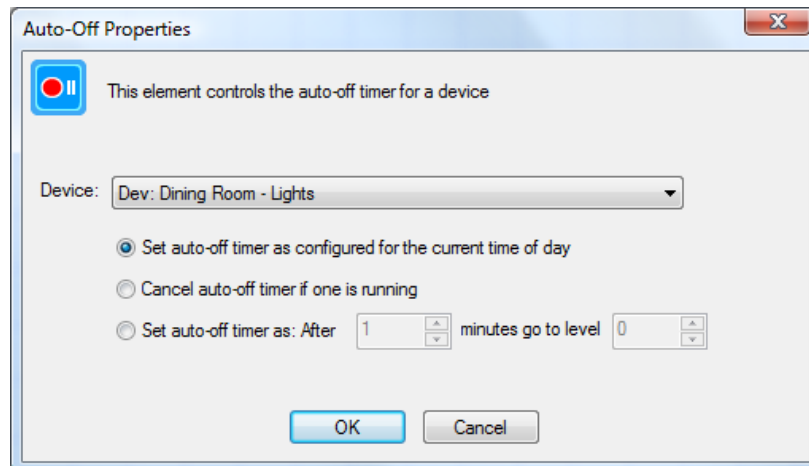


**Hint:** You can embed HCA expressions in the text between `%%`. See the chapter on expressions and the expression builder. For example, The time is `%%_now()%%`

---

## Auto Off

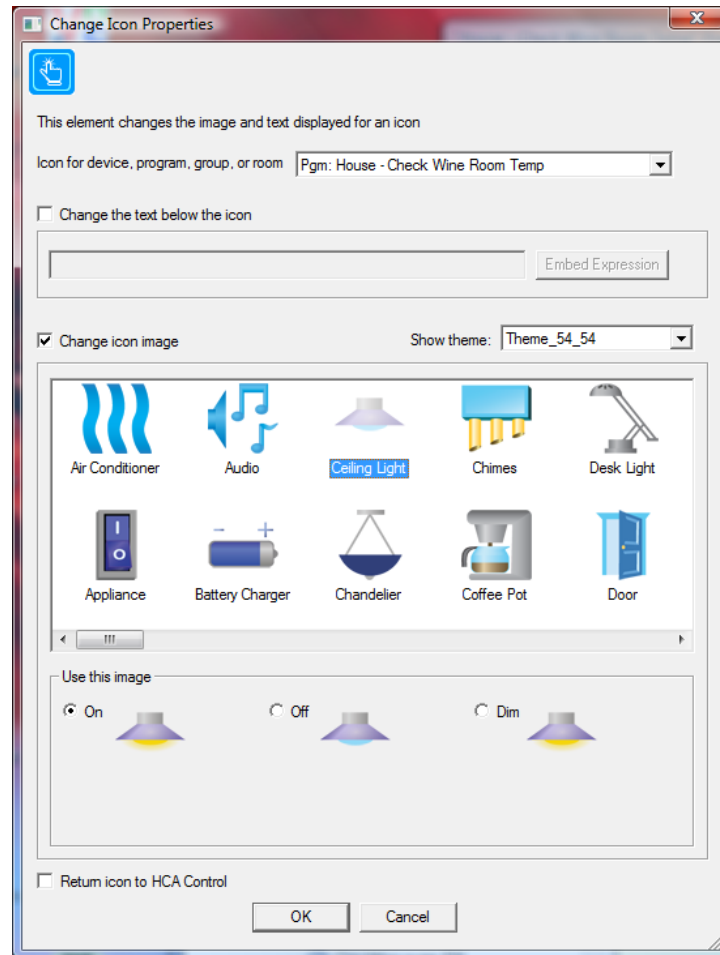
This element configures the auto off settings for a device. You may want to change the auto off settings for a device, for example, different times of the day.





## Change icon

The change icon element is one of the more complex elements. It allows you change the icon for any device, program, or group and/or the text below that icon.



With this dialog box, you can either select which icon to use and its representation (on, off, dim), the text below the icon, or both the icon and the text.

1. Select the device, program, or group to change.
2. Select the icon you want to use
3. Click the button for the icon representation that you want—On, Off, or Dim.
4. Click OK when done.

This is a good element to use when you have special needs for a program. For example, a program could modify a door sensor device icon and the text below to show if a garage door is open or closed.

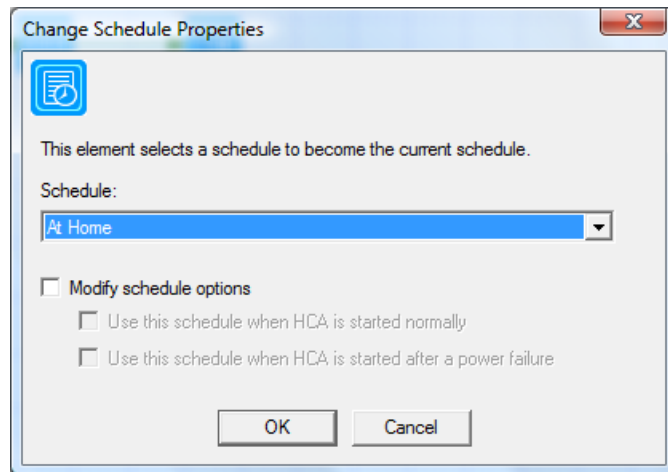
**Hint:** You can embed HCA expressions in the text. See the chapter on expressions and the expression builder.

You can draw these additional icons yourself with a paint program and add them to an HCA icon theme. Refer to the Icon Theme chapter for more information.

---

## Change schedule

This element changes the current schedule. With this element, you can cause HCA to stop monitoring one schedule and start monitoring another.

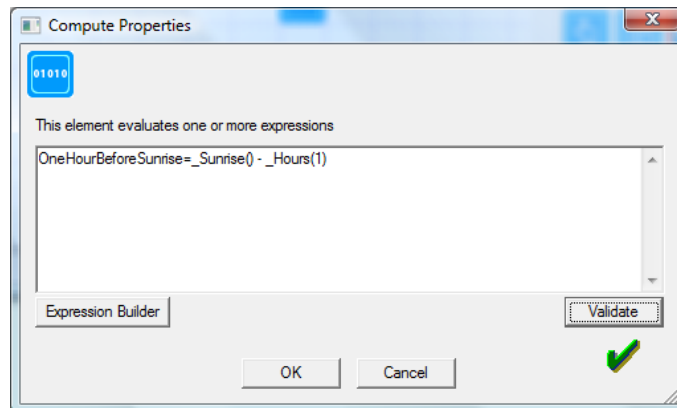


From the Schedule box, select the schedule you want to be the current schedule.

---

## Compute and Compute Test

These two elements assign values to flags and test those values. The Compute element properties are:



The Compute Test element is similar to the Compute element except that the expression entered is evaluated and if true, execution follows the Yes path, otherwise the No path.

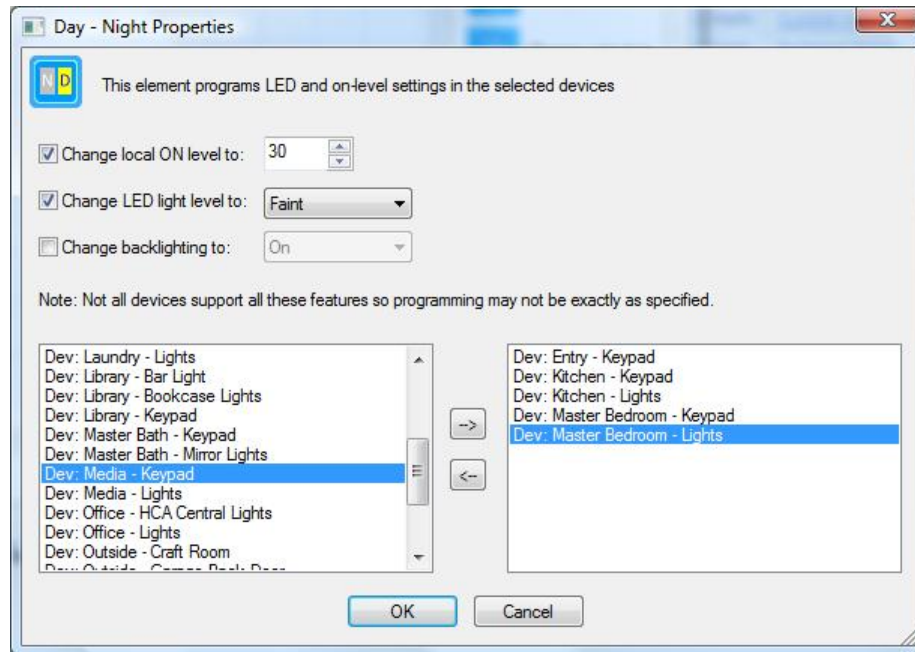
These two elements are used by more sophisticated programs and are not necessary for most programs.

What you place in the edit control is complex and powerful. It is described in the chapter on expressions and the Expression Builder.

---

## Day Night

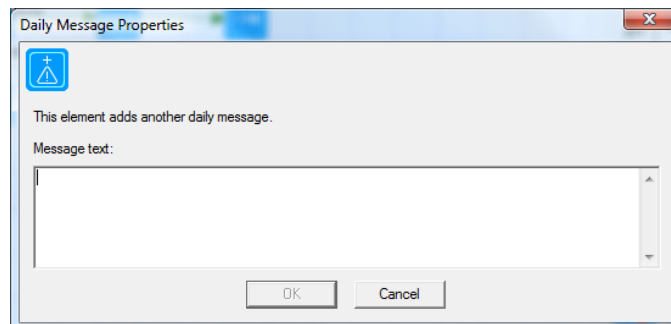
The *Day Night* element reprograms devices that have adjustable LEDs – keypad buttons or indicator LEDs – to a different level, and/or changes the local on-level for switches. Not all devices support these features. By creating programs that run at night you can “dim down” annoying lighting and then increase it during the day.




---

## Daily Message

The *Daily message* element creates another daily message that shows in a text display.

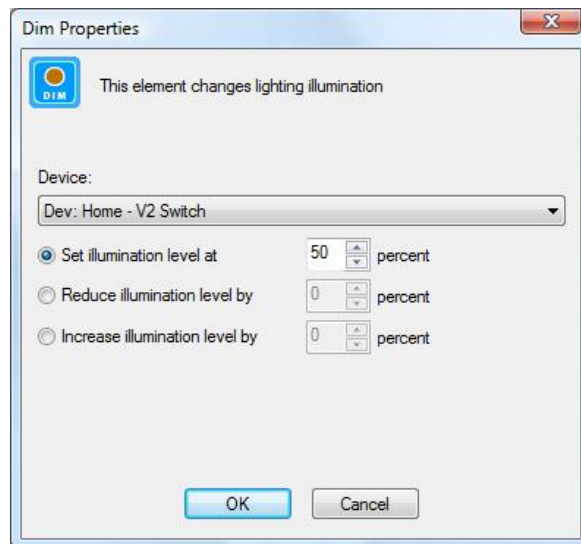


Like the Add To Log element you can embed expressions between %% that are evaluated before the message displays.

---

## Dim

This operation dims a device or group. It is a little more complex than the On and Off elements.



First select the device you want to dim then set the illumination level.

There are three ways to dim a device (or group):

- The first way is to set the illumination level—100% is full bright and 0% is full dim.
- The second way reduces the illumination level by a percentage.
- The third way increases the illumination level by a percentage.

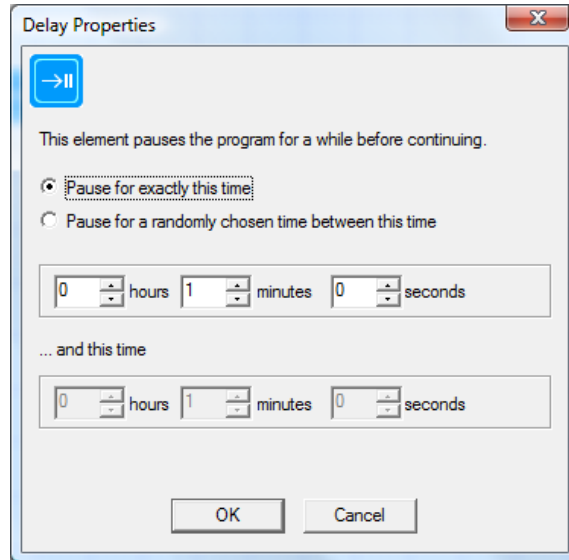
Using a percentage works, regardless of how many dim levels the device supports.

The percentage you select converts to a number of dim levels using this formula:

$$\text{Dim levels} = \frac{(\text{supported dim levels}) \times (\text{percentage you entered})}{100}$$

## Delay

This element causes a program that is running to pause for a while. You set the length of the pause using the element properties.

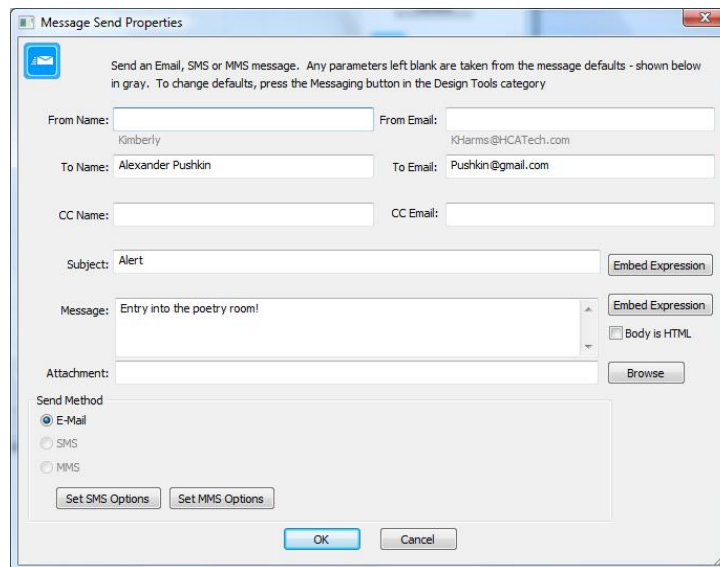


Use the hours, minutes, and seconds controls to set the hours, minutes, and seconds of delay. If you choose the second option, then HCA computes a different delay each time the element is executed.

**Hint:** Even if you choose an exact delay time, your program may not continue precisely after the delay time you set. But it should continue within a few seconds of the time you specified. Note that this is the only place in HCA where you enter a time to the second.

## Email / SMS

The send email element is used to send email, SMS, or MMS messages.



This dialog captures the parameters of the message. Any fields you have defined in the message defaults are shown in gray below the field. If you don't enter anything into these fields, then the defaults are used when the message is sent.

From Name:  From Email:

Before you use this element you have to configure messaging. Press the *Messaging Setup* button from the *Design* category.

VP Message Send defaults

SMTP Server

Server Name:  Port:   Connection is secure using SSL / TLS

Authentication Type:  Username:  Password:   Create email send log file

Wireless Options

Message Defaults

From Name:  From Email:

To Name:  To Email:

CC Name:  CC Email:

Subject:

Message:

In addition to the email server parameters you can also see that the remainder of the dialog is very similar to the *Email* element properties dialog. When the *Email* element executes any parameter not specified in its properties is taken from the messaging defaults. In this way you can enter the “from” information – and perhaps the “to” as well - so it doesn't have to be duplicated in each *Email* element added to a program.

**Hint:** To enter multiple recipients separate them with ‘;’ in both the “To name” and “To email” edit boxes.

---

## Exit

When a running program executes this element, the program ends. The *Exit* element has no properties.

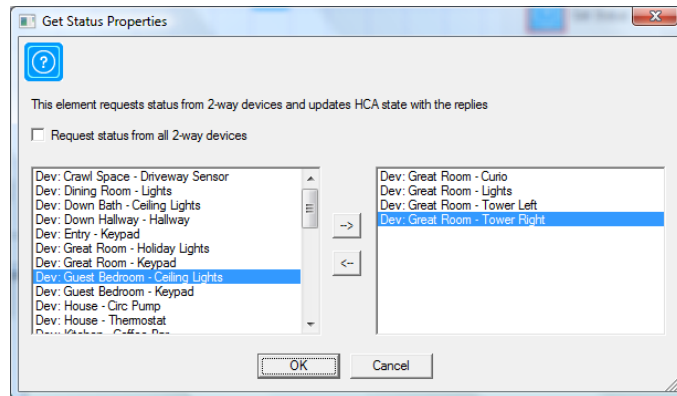
You may want to use this element in conjunction with the *Test* and *Repeat* elements (described later).

Each path of a program does not need to end in an exit element.

---

## Get Status

The *Get Status* element initiates a poll of one or more devices that respond to status requests.



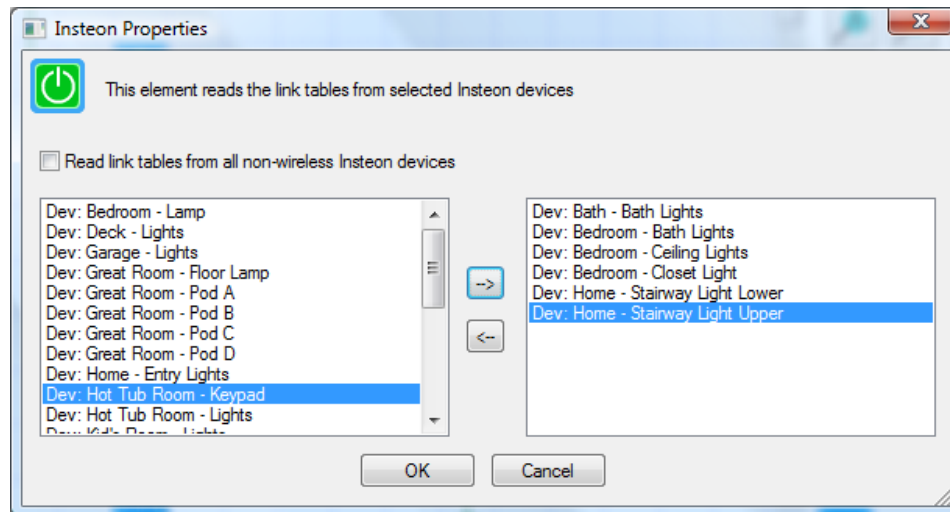
You can either select the devices you want to poll or you can just say to get status from every device that responds to status requests.

One important point about this element is that these status requests are not performed when the *Get Status* element is executed. The status requests are queued for later transmission. There is no way to predict when they will be sent.

---

## Insteon

The Insteon element reads the linking tables from one or more selected devices and saves that state within HCA. As described in the Insteon Appendix, these linking tables enable HCA to keep track of the state of your devices and also to correctly respond to messages from them.



This element is typically used in a program that you schedule to run periodically as a way of keeping your HCA design up to date with your hardware.

---

## Make flag No/Make flag Yes/Not flag

HCA provides flags as a very simple way for programs to keep track of things. Unlike other objects in HCA, flags are not created by using a wizard. Flags are created when you first use them. Flags are very important for some aspects of programs, and flags have a few simple rules:

- You can have any number of flags you want in any program.
- Flags can take on many values but these elements only work with Yes and No values.
- Each flag has its own name.

Flags are most useful with the *Test* element since it can check the value of a flag and do different things based upon the outcome of that test.

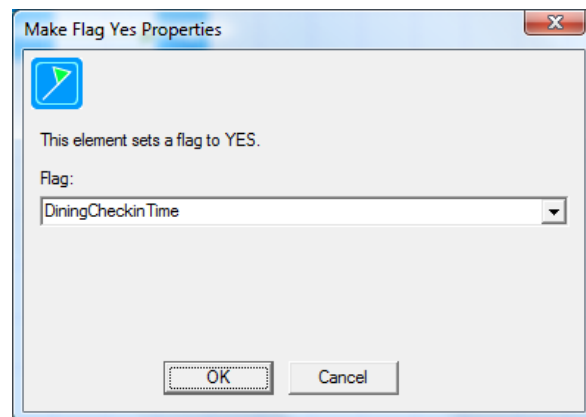
There are three elements that work with flags:

**Make flag No**—Makes the flag contain a No value

**Make flag Yes**—Makes the flag contain a Yes value

**Not flag**—This reverses the value of the flag. If the flag contains a Yes before the *Not* element is executed, it contains a No after. If the flag contains a No before the *Not* element is executed, the *Not* changes it to a Yes.

These three elements have similar properties.



When setting the properties, select a flag you already have created, or type the name for a new one.

Once you create a new flag it appears in the Flags inventory.

The properties for a flag are its name, its current value (Yes or No), and the value that HCA should assign to the flag when HCA first loads your design. To inspect the flag's properties open the *Flags Viewer* from the *Control* category.


---

## Multi

This element allows you to perform what the On, Off, and Dim elements do on a set of devices. Used in this way rather than a series of the other elements may make constructing programs simpler.



Mega Properties X

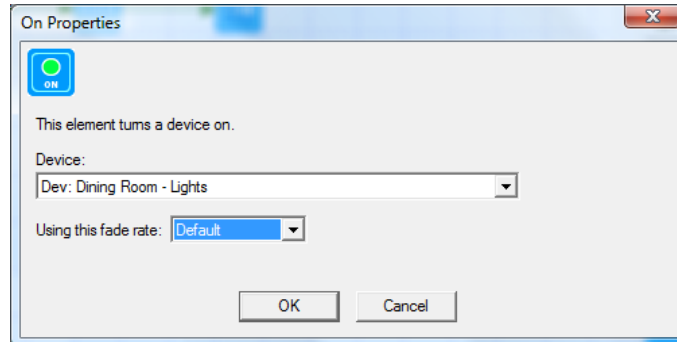
 This element changes the state of one or more devices in the order designated

Device	Action	Level
Great Room - Lights	On	0
Great Room - Tower Left	Off	0
Great Room - Tower Right	Percent	50
Unused	Do Nothing	0
Unused	Do Nothing	0
Unused	Do Nothing	0
Unused	Do Nothing	0
Unused	Do Nothing	0

---

## On

This element sends an ON command to a device or group, or starts another program with an ON command.



Select from the dropdown the device or program to control with an ON.

Some kinds of devices show additional parameters for you to select in the ON element properties.

You can also use this element to start another program running. If you do, that program runs concurrently with the current program. The current program continues to the next element as soon as the other program starts.

---

## Off

This element sends an OFF command to a device or group, or starts another program with an OFF command. Its properties dialog box looks very similar to the On properties.

From the list, select the device or group you want to turn off.

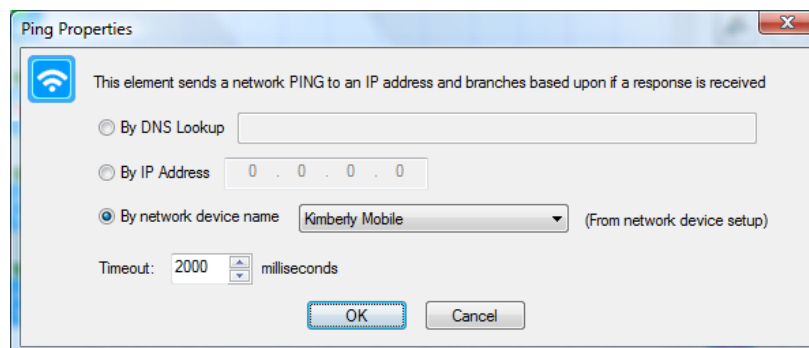
Some kinds of devices will show additional parameters for you to select in the OFF element properties.

You can also use this element to start another program running. If you do, that program runs concurrently with the current program. The current program continues to the next element as soon as the other program starts.

---

## PING

The Ping element is part of the "Network Devices" component and uses several of the configuration settings defined there. To configure those settings, use the *Network devices* button in the *Design* ribbon category.



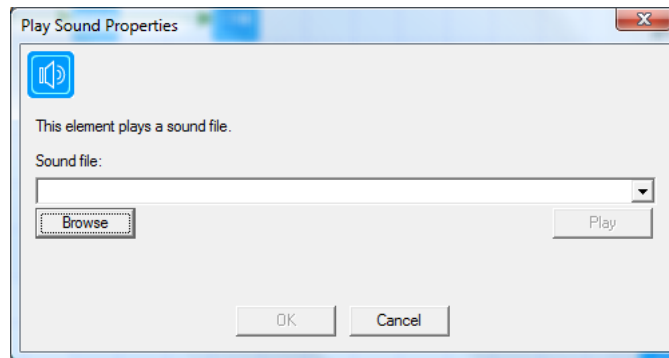
A device is specified by entering a DNS lookup name, an IP address, or selected from one of the network devices configured in the Network Devices dialog. When the element is executed, a network PING is sent to the device and if it responds in the time allotted then, like the Test element, the "yes" path is taken from the element otherwise the "no" path is taken.

---

## Play sound

The *play sound* element plays a WAV file using whatever sound system your computer contains.

**Hint:** In client-server mode, each client has configuration to say if it plays sound or not. The WAV file must be present on each client if they are to play the file.



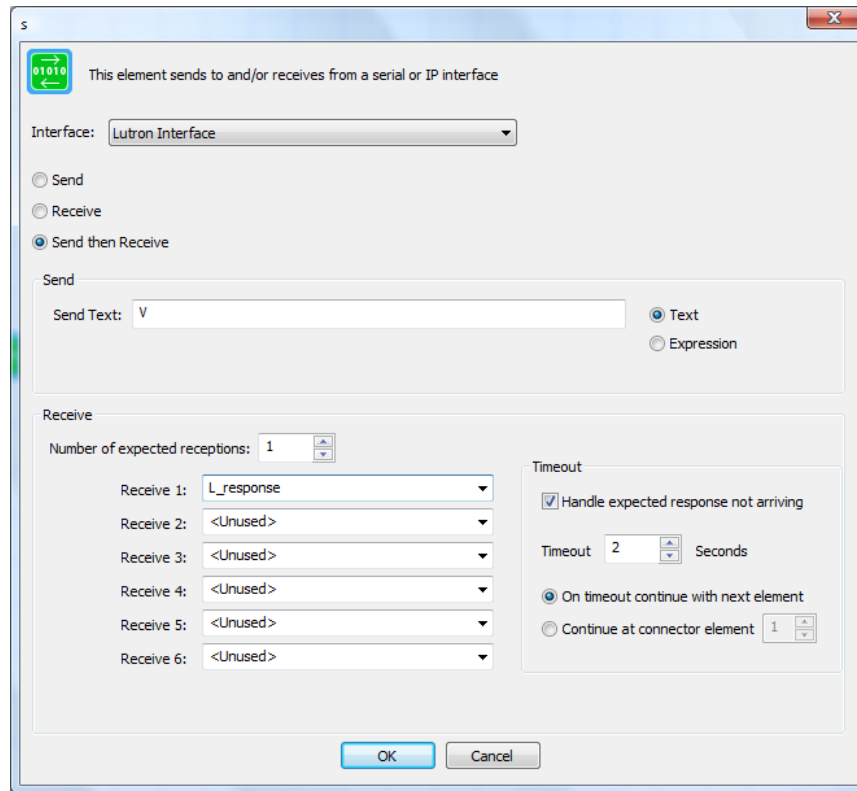
If you create your design file (the .HCA file) on one computer, and then move it to another computer, check your programs for *play sound* elements. If you have *play sound* elements, you must make sure that you move the WAV files and place them in the same directories as they were on the machine you are moving from.

**Hint:** Use the Design Inspector (from the HCA menu) to verify that all the paths to WAV files in your programs do indeed locate their WAV files.

---

## Port IO

The Port IO element works with the generic serial and generic network interfaces configured in HCA Options on the hardware tab. This element can send, receive, or send and then receive from the port.



Select from the interface dropdown the name of the interface to use, then choose the action option for send, received, or send then receive.

If sending, the text to be sent is entered. This can be the actual text or the result of an expression evaluation if the *Expression* option is used.

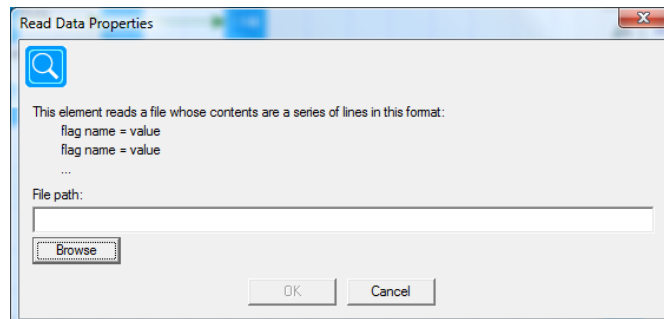
If receiving, enter the number of expected responses and then, for each response, select the flag that reception is assigned to, or the name of a new flag that the response is assigned to.

As with any communication, you can handle the case of what happens if an expected reception doesn't arrive. The *Timeout* group contains the options for this. Enter the max number of seconds to wait for the expected receptions. If that timeout limit is reached then execution can either continue at the next element or at an element pointed to be the numbered connector.

---

## Read Data

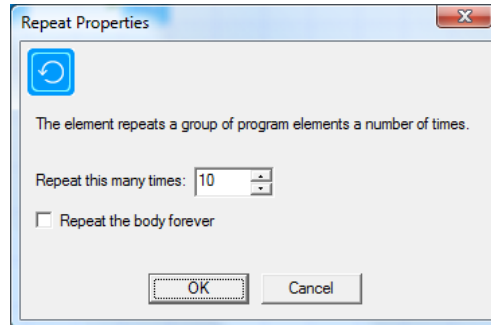
The *Read Data* element is a way to get data, perhaps produced by another Windows program, into HCA. The file it reads consists of a number of lines where each line is of the form `flagName = expression`. In effect, each line is treated as if it was the text in a *Compute* element.



---

## Repeat

This element allows a program to repeatedly execute one or more elements.



Set the number of times that you want the elements to repeat or tick the box to have the elements repeated continually.

If you choose to repeat continuously, the program never stops until the program is stopped (by selecting Stop from the popup menu, another program using the Stop element, or by HCA itself terminating (maybe due to a power failure).

If in the body of the repeat – those elements that are repeated – if an exit element is executed this can also terminate the repeat. This is configurable in *HCA Options*.

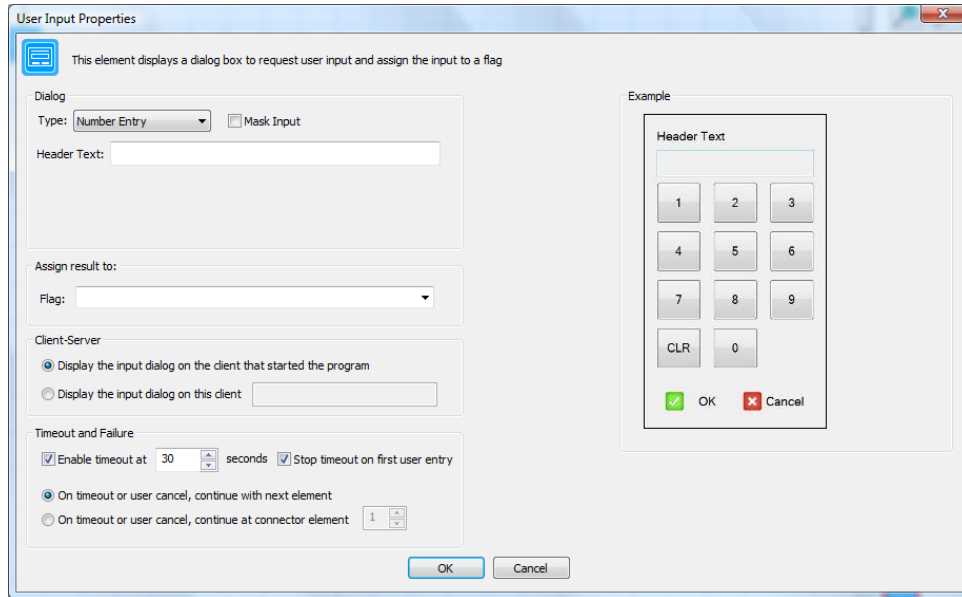
There is more information on how to construct programs using the Repeat element later on in this chapter.

---

## Request Input

The *Request Input* element presents a dialog box to the user and waits for input. Here are several different dialog types possible. For example, using the *Request Input* dialog a program can create a user interface for modifying a schedule's time, or request a numeric entry code for a control system.

The element properties dialog shows an example of the general shape and content of the displayed dialog.



In the above screen image the numeric entry dialog is shown. The other dialog types are:

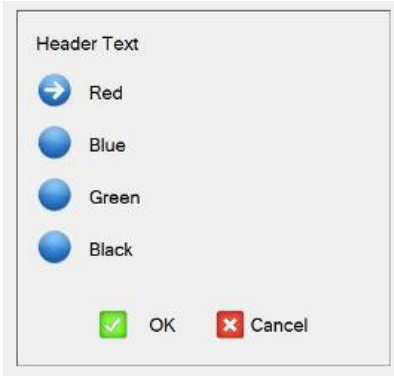
Text Entry:



Time Entry:



Choice:



Security Console:



Each different dialog type has different configuration where the various text and button labels can be modified for your application.

The other parts of these element properties are common for all dialog types. These are:

#### Assign Results To

Select the name of a flag or enter a new flag name to assign the input – what the user entered – to.

#### Client Server

If HCA is operating in client-server mode, the dialog appears on either a designated client – using the client name which is set in *HCA Options* on each client – or on the client that started the program that contained the *User Input* element.

#### Timeout

The user input dialog can be limited in the length of time that it appears on the screen. If a timeout occurs then the next element executed in the program can be the next element or an element pointed to by the numbered connector element.

---

## Resume and Suspend

The suspend element is very similar to the suspend menu selection for devices, programs, and groups. In the properties of this element the device, program, or group is chosen and the type of suspension is selected. A item can be suspended from:

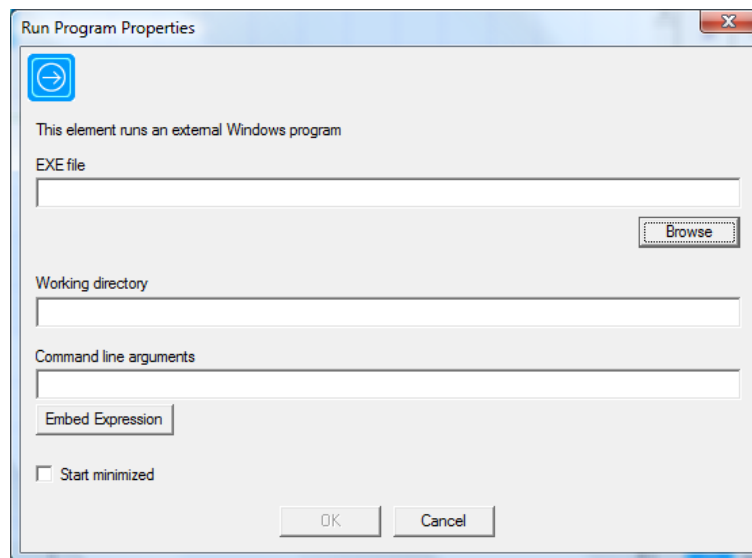
- Schedule control. Any schedule entries for this item don't happen while the item is suspended.
- Program control. Any On, Off, or Dim elements of any program don't control the item while it is suspended.
- Remote control. Any command received for a program or group doesn't start the program or control the group.

The Resume element is used to undo the effects of a Suspend.

---

## Run

When you use the Run element you can start another Windows program. Don't confuse that with starting a HCA program. A Windows program is started from the Windows Start button or from a command line.



The properties for the Run element are:

- The path to the executable file. Typically these end in .EXE
- The directory that the program is started in. The working directory is the directory where a program will first find any files it opens.
- The command line to pass to the program.
- An option to start the program in the minimized state.

This element is very similar to starting a program from the Run command you get from the Start button.

**Hint:** You can embed HCA expressions in the text. See the chapter on expressions and the expression builder.

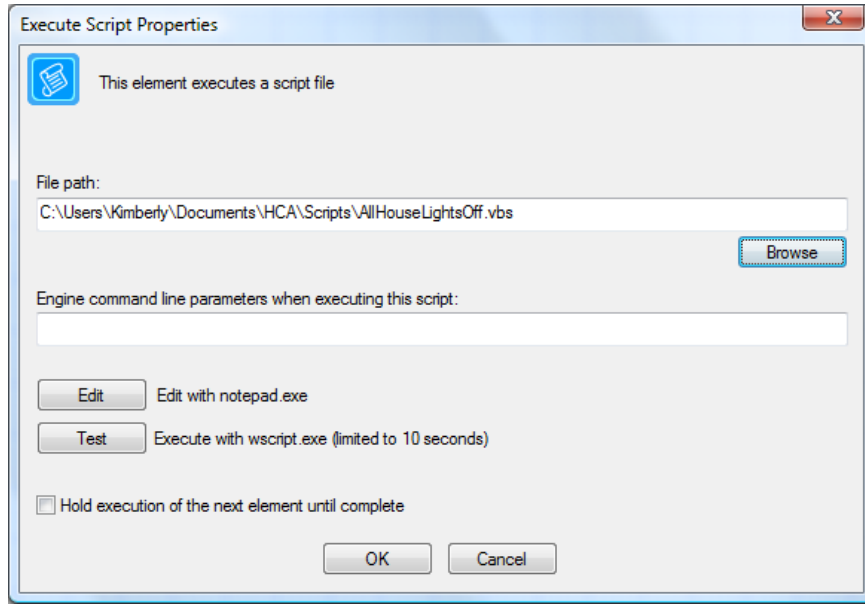
---

## Script

The run script element starts a text-based script using the currently selected script engine.

This is a large topic and is fully described in the Script chapter.

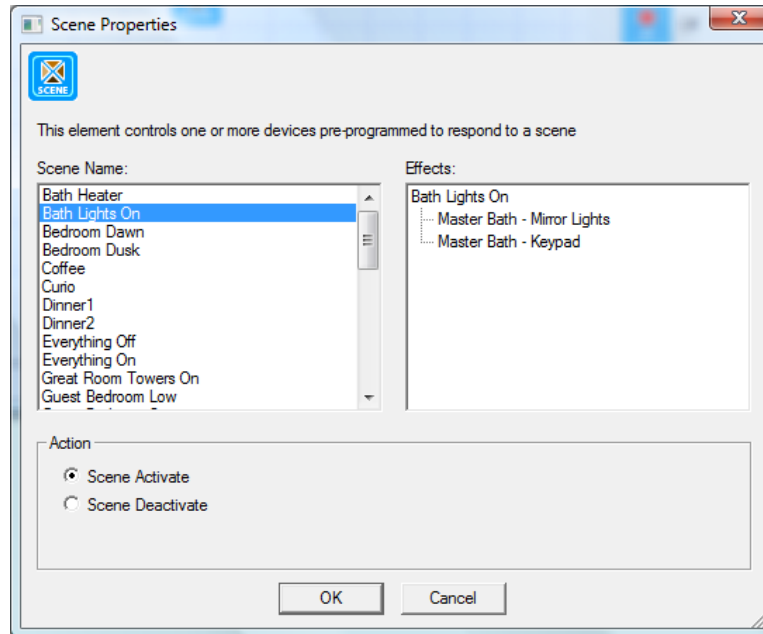




---

## Scene

The scene element is used with devices that can be programmed to react to commands to activate and deactivate scenes.

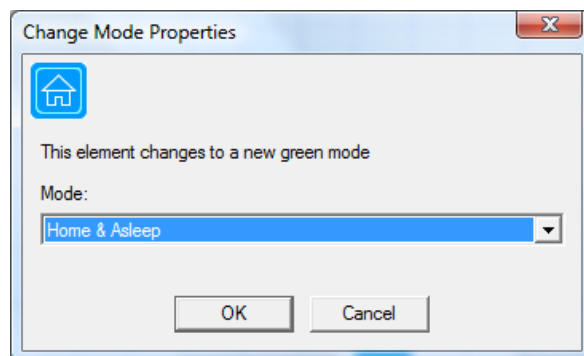


Scene names are listed in the left column and the effects of that scene in the right column. Depending upon the type of scene – Insteon, UPB, or XP – various options may display at the bottom of the dialog.

---

## Set Mode

The Set Mode element changes the current home mode tone of the home modes configured in the Home Properties.




---

## Show Display

This element causes the display pane to show the display selected.

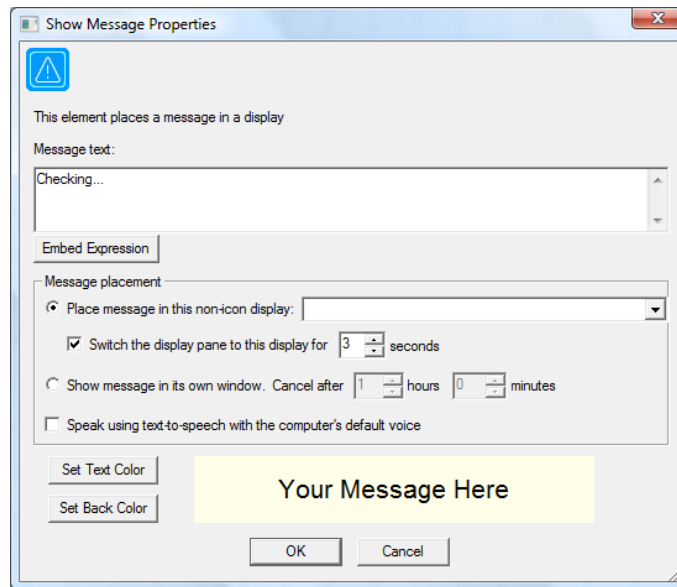
---

## Show Message

The *Show message* element creates a message that is either displayed in the HCA display pane or in its own window. If the message is shown in the display pane it persists for a few seconds before being removed.

If the message is displayed in its own window, that window persists until you close it or it expires. You can set the expiration time in the HCA properties on the display tab.

The properties for the Show message element allow you to enter the text to display, the destination of the message (display pane or own window) and the color for the text and background.



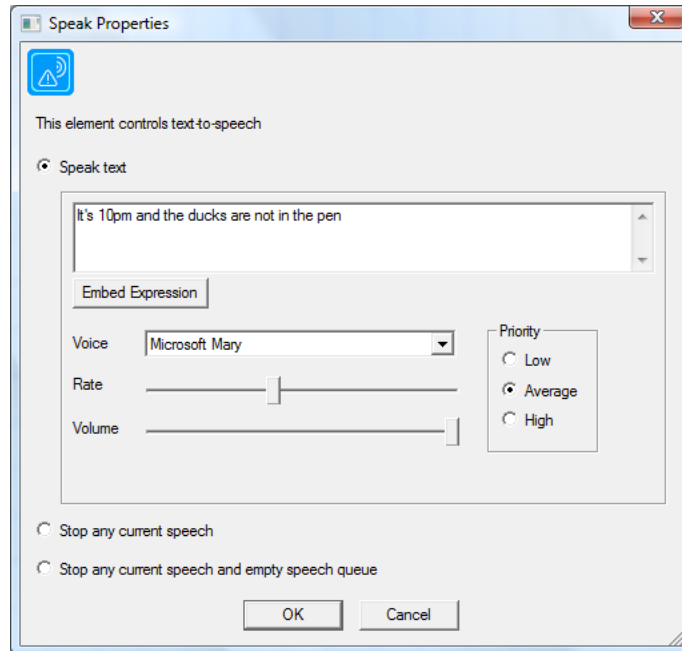
The text can contain expressions embedded in `%%`. These are evaluated before the final text is assembled and displayed.

---

## Speak

The *Speak* program element works with a Text-to-speech engine installed on your computer to convert a piece of text into spoken words played through the computer sound system. HCA doesn't contain a text-to-speech engine and you must acquire one and install it before HCA can use it. Check the Windows Control Panel Speech applet for information and installation on text-to-speech.

The properties for the Speech element are:



There are three options:

1. Speak some text. This text can contain embedded expressions that HCA evaluates and replaces with the result.
2. Stop any current speech.
3. Stop any current speech and empty the speech queue.

It is important to remember that the Speech element adds the message to the text-to-speech engine queue and moves on. As each item in the queue is completed, then the next item in the queue is spoken. The last two options give you the ability to cancel the current text and to empty the speech queue.

---

## Start Program

The *Start program* element starts another program running. (Refer to the description of the *On* and *Off* elements). Like the *On* or *Off* elements, the *Start program* element starts another program, but it does **not** continue until the started program completes.

1. Select the program that you want this element to start.
2. Select the type of command that starts the program. This is only important if the program being started tests for the command that starts it. (Refer to the description of the *Test* element.).

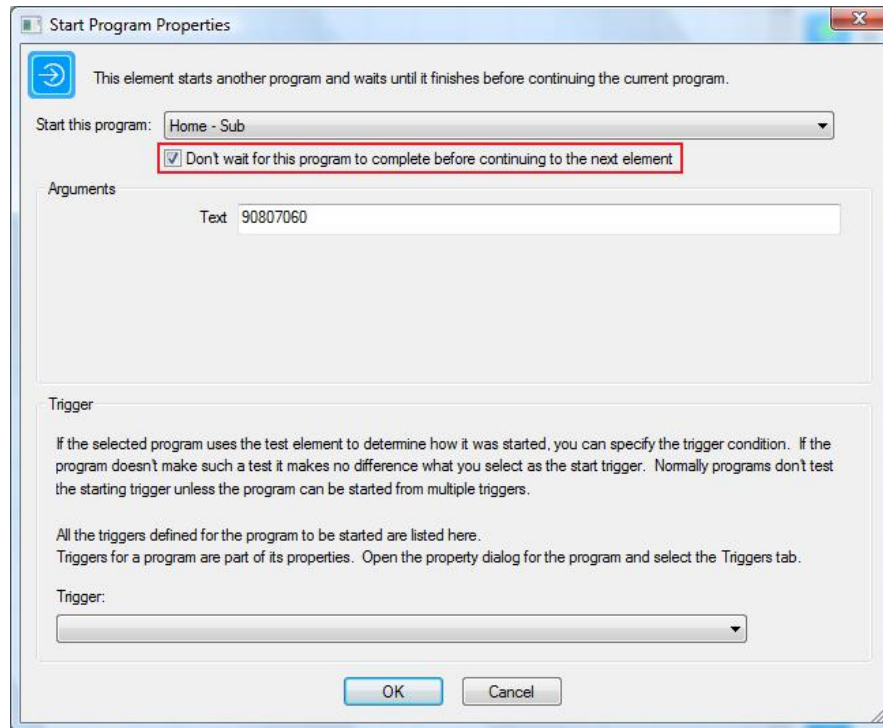
One program can cause another program to start in one of two ways: using this element or using the ON or OFF elements.

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 parameterized programs – described later in this chapter - you also got to supply arguments for those parameters. Neither can 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.

There is an additional property of the Start-Program element that allows you to overcome these limitations.



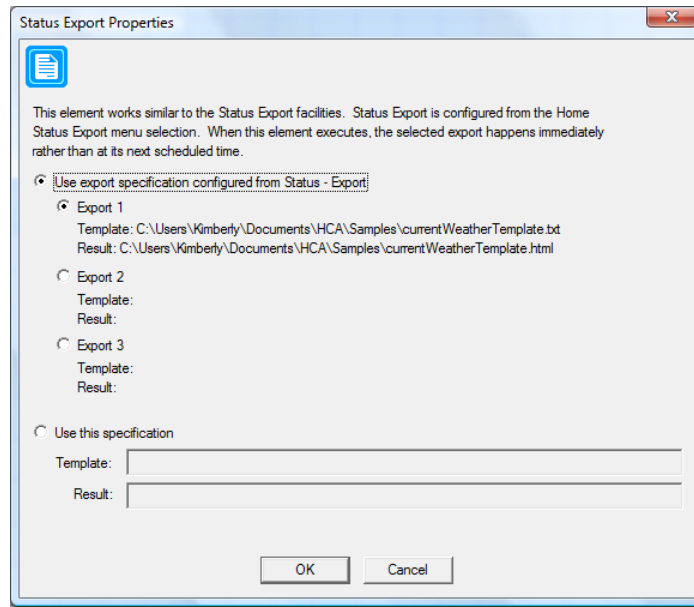
If this option is enabled then the program is started as a co-routine as described above.

The major advantage over the ON and OFF elements is that the started program 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.

---

## Status Export

The Status Export element starts a status export using one of the three configured status exports. The properties of this element are:



The action of the Status Export and how it is configured is described in the *Status Export* chapter.

---

## Stop Program

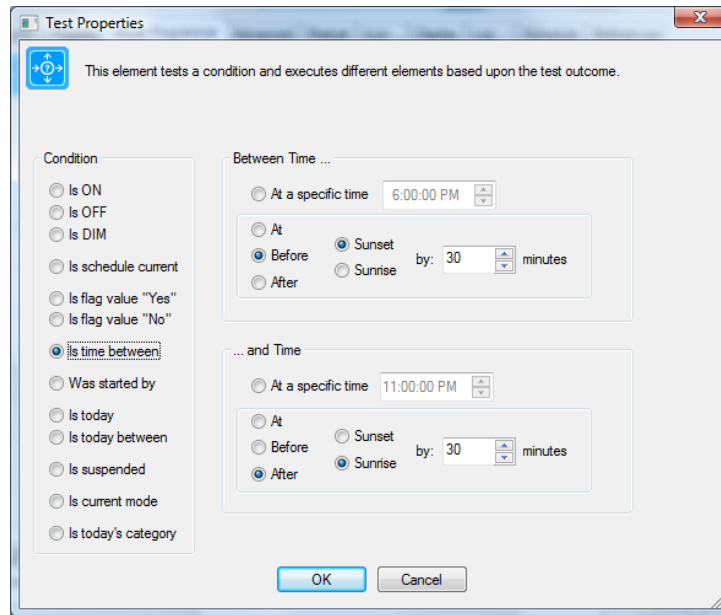
This element stops a running program. If the selected program is not running when this element is executed, it has no effect. If multiple copies of the program are running (see the section on advanced program properties), all copies of the program are stopped.

**Hint:** You can't stop the program that contains the stop element. To do that use an *Exit* element instead.

---

## Test

The *test* element is the most complex element HCA contains. Without the test element, each program would start with the “Start here” element and continue from element to element following the connecting lines until the last element has been executed. With the test element, your program can execute different elements based upon conditions that are present when the program runs.



This element dialog box has two parts. First click the button to set the condition you want to test for. Second, select the object that you want to test.

Here is a list of conditions that this element can be used to test:

Test choice	Tests for
<b>Is On</b>	Is a device/group on? Is a program running?
<b>Is Off</b>	Is a device/group off? Is a program not running?
<b>Is Dim</b>	Is a device/group dim?
<b>Is Schedule current</b>	Is a schedule the current schedule?
<b>Is Flag Value Yes</b>	Is the flag value Yes?
<b>Is Flag Value No</b>	Is the flag value No?
<b>Is Time between</b>	Is the current time between two selected times?
<b>Was Started by</b>	What trigger caused this program to start?
<b>Is Today</b>	Is today a specific day of the week, day of the month, or date?
<b>Is Today Between</b>	Is today between two dates?
<b>Is Suspended</b>	Is the device, program, group, controller or schedule entry suspended?
<b>Is Current Mode</b>	What is the current home mode?
<b>Is Today's category</b>	Is today the selected category as defined by the HCA calendar?

Some conditions are fairly obvious, others are not. Following are more details on each condition.

### ON / OFF / DIM

This type of test checks the state of a device, group, or program in your design to see if it's on, off, or dim. One very important point to note is that this test is based upon what state HCA thinks the device is in. For example, if HCA sends an On command to a lamp it records it as On. If you subsequently use a wall switch to turn the light off, and that switch doesn't automatically report its status then HCA may not be informed that the light is off.

When testing for dim you can test to see if the illumination level is less than, greater than, or in a range of levels. If the device supports stored scenes, you can test to see if the device is currently set to some specified scene illumination.

For programs the ON and OFF tests see if the program is currently running or not.

### Schedule is Current

This test checks to see if the selected schedule is the current schedule.

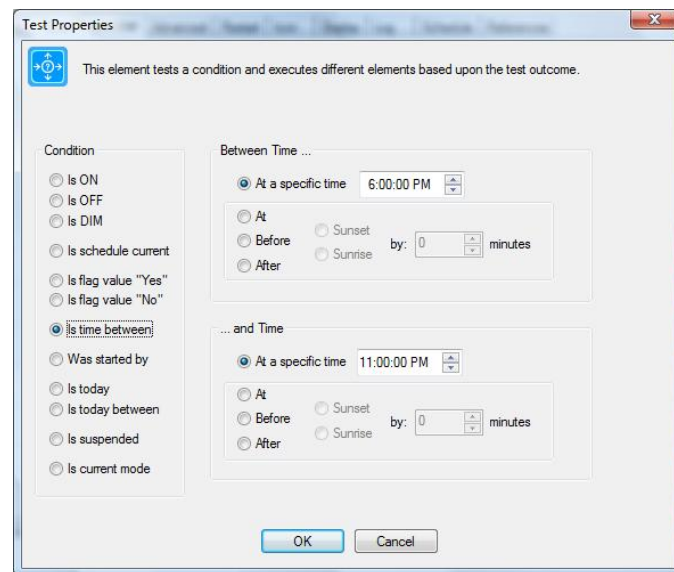
### Flag value is "Yes"/Flag value is "No"

This type of test checks the current value of a flag. Since flags can contain Yes and No values, you can test for either of those conditions. Choose the flag being tested from the dropdown or type in the name of a new flag.

This element only tests flags for simple yes and no values. To test for other values use the *Compute Test* element. If you do use the *Test* element and the flag doesn't contain a simple Yes or No value, the value is converted into a Yes or No value. See the chapter on expressions for more details on this.

### Time Between

The *Time Between* test allows your program to perform different actions depending upon the time of day when the *Test* element is executed. This condition is a little more complex, and allows you to specify several different settings.



You specify the two time points for this test in a manner very similar to the way that you specified time in the Schedule Entry wizard.

In the *Between Time* area, set the **start point** for this test:



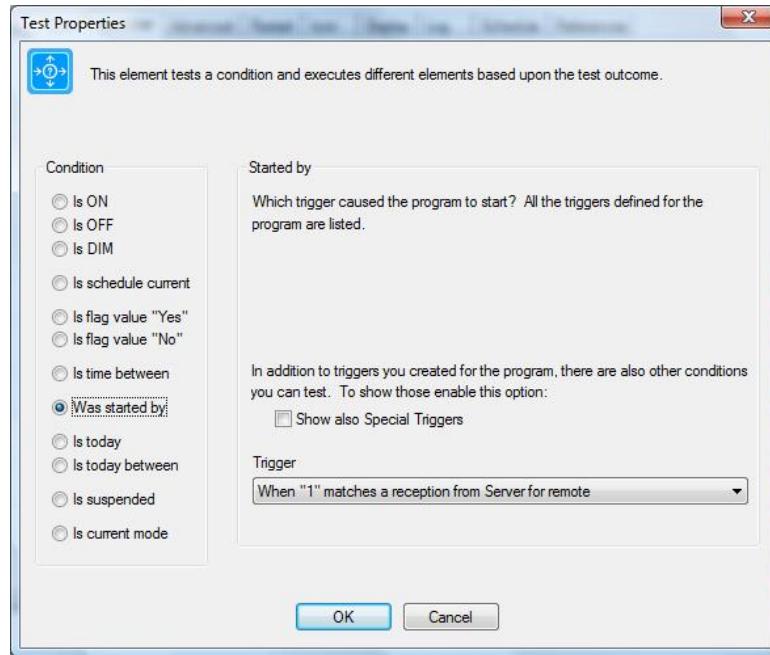
1. Click the button of the time you want to use, and specify either the specific start time (like 6:00 a.m. or 5:45 p.m.), or a number of minutes before or after sunrise or sunset.

In the *...and Time* area, set the **end point** for this test:

2. Click the button of the time you want to use, and specify either the specific end time, or a number of minutes before or after sunrise or sunset.

### Started by

This type of test is to check how the program was started. Programs can be started by many different types of triggers. These are specified on the Triggers tab of the program properties.



There are two options besides the trigger that can be useful. The first option tells HCA that when testing the starting trigger to compare the trigger command but not the X10 address. This is, of course, only useful if the starting trigger is an X10 command. For example, you could test for an ON command regardless of what address the command came from.

The Special Trigger option provides some additional options in the trigger list you can test for. These are:

- Started by any trigger
- Started by a schedule
- Started by user action. That is, by the user using the User Interface to start the program.
- Started by an ON or OFF program element
- Started by a StartProgram element
- Started by group membership

These tests don't compare specific triggers but rather the action that started the program. In this way you could do different things if the program was started, for example, from a schedule and when it starts because you selected the program icon in the user interface and selected ON from the popup menu.

### Is Today

This type of test allows you to see if today (the current date when the program is running) is a specific day of the week or month. The same date options used in schedule entries are used here.

#### Is Today Between

This type of test checks to see that the current date is between two dates without checking the year – only the day and month. In the properties of this test both dates are selected.

#### Is Suspended

This test type checks to see if the selected device, program, group, controller, or schedule entry is suspended. In order to suspend a schedule entry it must have a name. Suspend is described in the user guide chapter on schedules and schedule entries.

#### Is Mode

Is the current home mode the selected mode? The home modes are described in the "Home modes" user guide chapter.

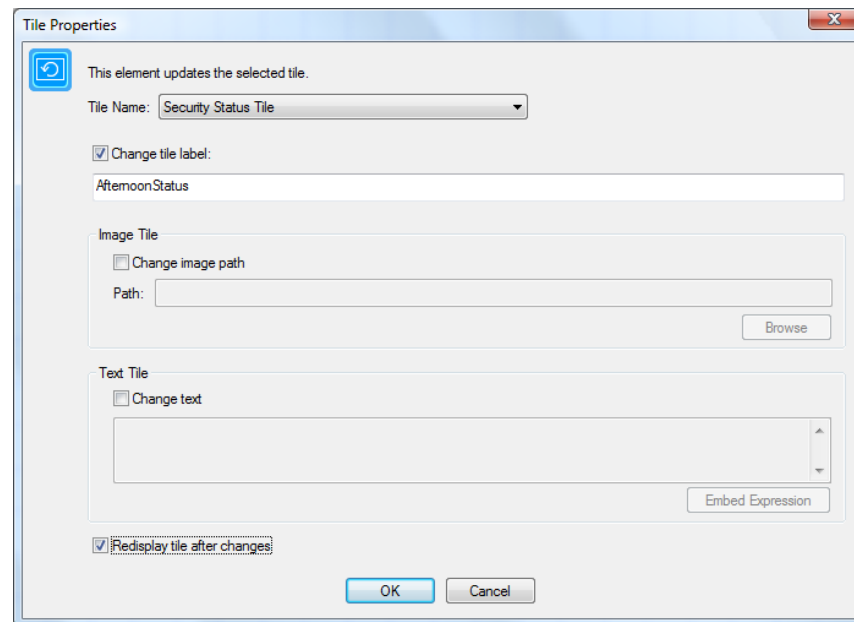
#### Is Today's category

Is today the selected category? Each day of the year can be assigned to a category using the HCA Calendar. The Calendar is described in the "Calendar" user guide chapter.

---

## Update Tile

The *Update Tile* element causes a selected tile on a tiled display to refresh with new information.



Select the name of the tile to update and what should be updated. For all types of tiles the tile label – displayed at the lower right of the tile – can be updated. For a image tile the image can be changed and for a text tile the text can be changed.

As in other elements – Add to Log, Show Message – you can embed expressions in the text between %%. Before the tile is updated the expressions are evaluated and the final text shows in the tile or the tile label.

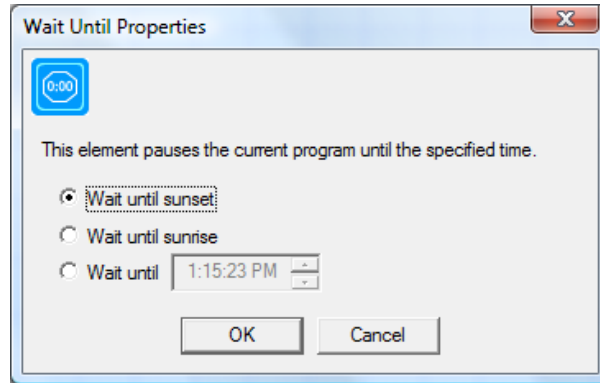
Tiles can also be updated based upon a update timer that can be set in each tile. This is an alternative way to cause the update immediately upon this element executing.

In client-server the update is broadcast to all clients.

---

## Wait Until

The Wait Until element is like the delay element in that it causes the program to pause for a while. But unlike the delay element, it does not wait a specified *interval* of time (like 5 or 10 minutes), it waits until a specific *selected* time (like sunrise, or 5:45 p.m.).



There are three different types of *Wait* elements. Click the button corresponding to the type of wait that you want.

1. You can set the program to wait until sunset.
2. You can set the program to wait until sunrise.
3. You can wait until a specific time like 4 p.m., or 6 p.m. If the program starts after the wait time that you enter, the program waits until the selected time on the next day.

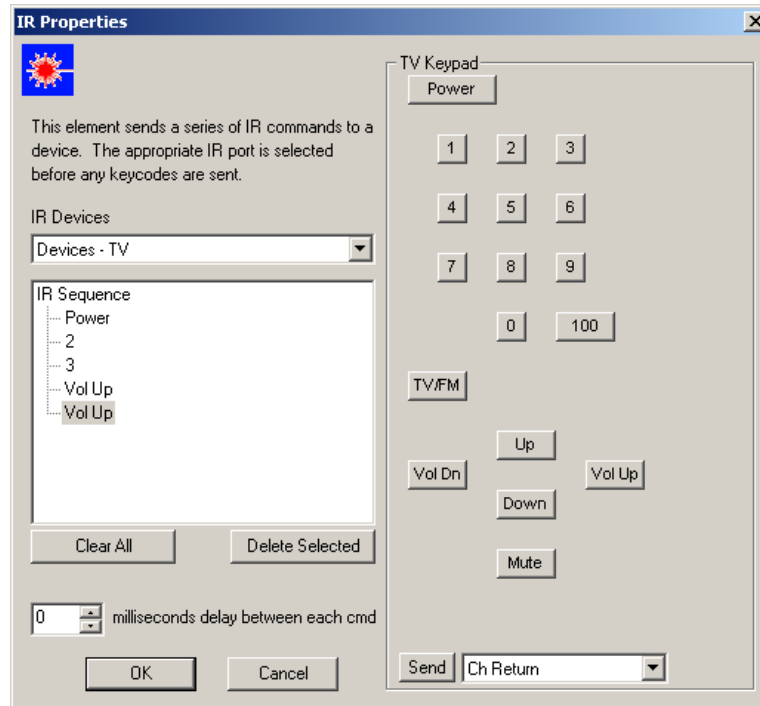
**Hint:** If you find yourself using the Wait element then you may want to re-think your design. It is almost always better to not use the Wait element but rather break your program into more than one part and schedule them to start at the times you want.

## Specific Hardware Elements

These elements are for specific hardware that you may or may not have.

### IR

The IR element is used to send sequences of IR commands to a device. This element can only be used if you have an interface that supports sending IR.



This dialog contains a number of parts. In the *IR Devices* list are all the devices you have setup that use IR. In the right side of the dialog is shown the keypad associated with the device. As explained in the appendix on IR interfaces, each IR device can have its own keypad.

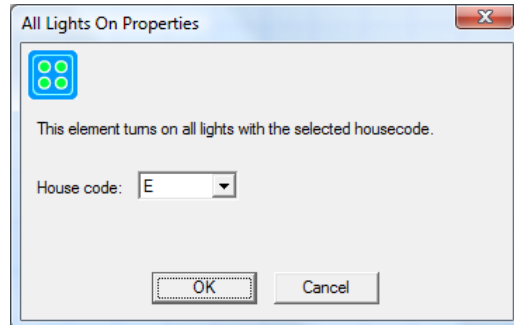
To enter an IR sequence, push the buttons for the signals that you want to send. They are displayed in the IR Sequence tree. New keycodes are entered after the currently selected item. To delete the keycodes previously entered, select then and press the Delete button.

Depending upon the receiving device it may be necessary to space out the IR commands. Enter in a time (in milliseconds) to delay between each key sent.

---

### X10 All lights on / X10 All lights off / X10 All units off

These three operations have the same properties. What each does is obvious. Their properties dialog boxes are the same.



---

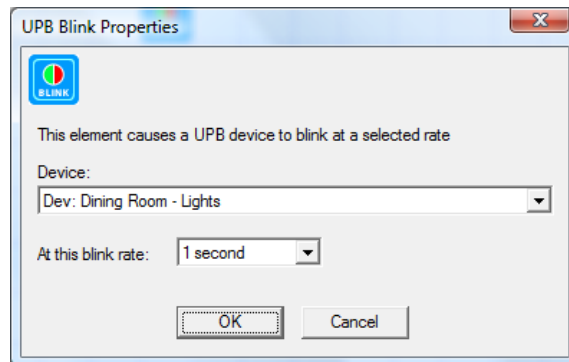
### Thermostat and Thermostat Test

These elements apply to thermostats and are described in the appendix on Thermostats.

---

### UPB Blink

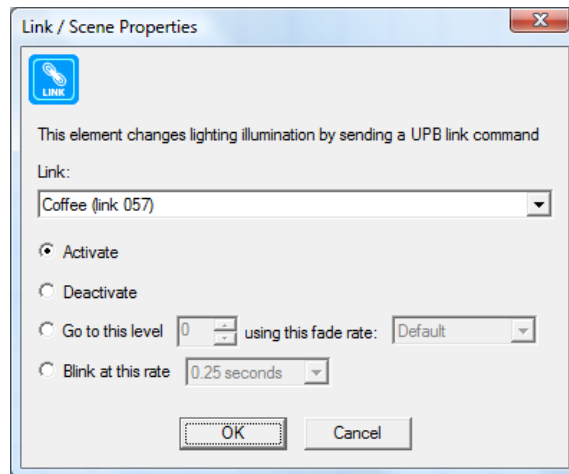
The *UPB Blink* element sends the UPB Blink command to a device.



---

## UPB Link

The UPB Link commands send a UPB link command



When you Activate a link, any device with that link in its receive components table responds to the preset level and rate stored in the device configuration.

When you Deactivate a link, any device with that link in its receive components table goes off. Doesn't matter what the level in the Receive Components table is, but it does use the rate when going off.

When you dim a link you provide a level and rate. All the device does is say "Is that link in my receive components table?" and if so it responds to the dim as the command specifies.

---

## Weather Test

The weather test element is used to read and test data from a weather station. It is described in the appendix on weather stations.

---

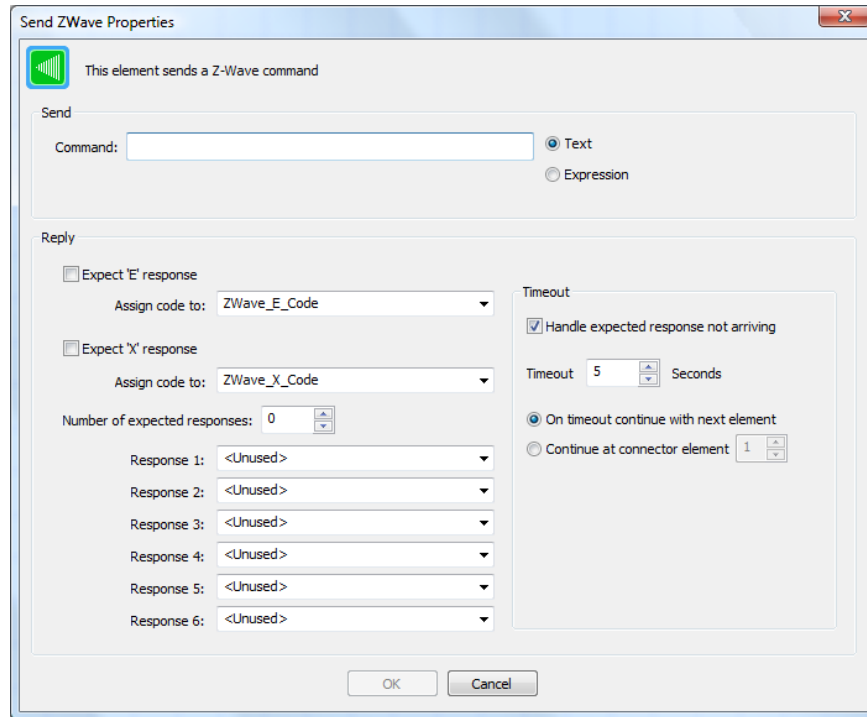
## Send X10

The SendX10 element is used to send a stream of X10 commands. Unlike the ON, Off, Dim elements, which send commands to control devices, the SendX10 element can send an arbitrary sequence of X10 commands. This would probably be used in only special circumstances.

---

## Send ZWave

The *Send ZWave* element is very similar to the Port I/O element except that it handles the special reports generated by the Zwave interface as well as the send text and receive text.



The only difference in the properties of this element and the Port I/O element is the handling of the ‘E’ and ‘X’ responses. To use this element you must understand the syntax and expected responses of the Zwave protocol.

**Hint:** There is a Zwave technical note available on the support web site that might help get you started with this.

---

## Legacy Elements

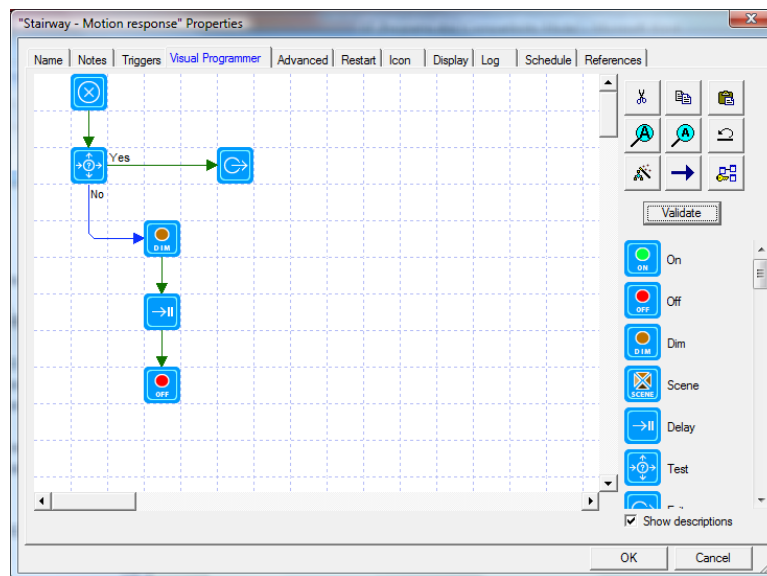
There are additional Visual Programmer elements for Legacy hardware support. These are detailed in the User Guide appendixes for those interface types.

## Constructing programs

Now that you know about all the elements used in constructing programs, and the general methods of using the Visual Programmer, you are ready to learn how to draw *Test* and *Repeat* elements.

### Test

Here is a sample program that uses a *Test* element.



Notice that from the *Test* element there are two connecting lines drawn: one goes to the *Dim* element, the other to the *Exit* element. The connecting line to the *Exit* element is labeled “Yes,” the other “No.” When you draw the *Test* element and draw connecting lines from it to other elements, you need to specify which connecting line to follow if the test succeeds or fails.

1. Right click one of the elements directly connected to the test element.  
The popup menu contains two additional options: “Do when Test succeeds,” and “Do when Test fails.”
2. Pick one of these options, and the connecting line from the *Test* element to the selected element is labeled appropriately.
3. Since you have labeled one path, HCA now labels the other.

If you later decide that you have the paths incorrectly labeled, you can re-label them by using the same method.

In this example, assume that the test element has its properties set so that when it executed it does: Test flag “Web is on” for Yes.

- If the flag “Web is on” has the value *Yes*, then with the paths labeled as in the above graphic, the next element executed after the *Test* element is the *On* element.
- If the flag “Web is on” has the value *No*, then the next element executed after the *Test* element is the *Off* element.

The *Exit* element comes in handy if you have nothing to do in one path from the test. Since there is nothing else for the program to do, it stops running. This is what the *Exit* element is designed for.

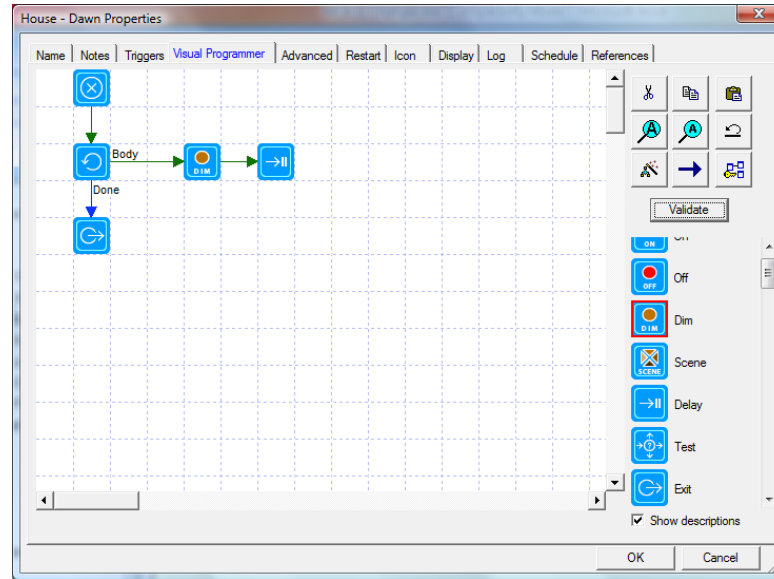


A program ends when it has executed an element that has no connecting line leading to another element, or when an *Exit* element is executed.

**Hint:** All the test elements work in this way, that is, they have two paths that lead out from the element. If the test succeeds, the next element executed is on the Yes path, otherwise on the No path. The Weather Test, Thermostat Test, MM input test, and others, work like this.

## Repeat

The *Repeat* element is another complex element. Here is a sample program that uses the *Repeat* element:



Notice that there are two connecting lines drawn from the *Repeat* element. The one labeled “Body” specifies which path to use for the elements to repeat. In this example, the Dim and Delay elements are to be executed repeatedly.

Like the *Test* element, you place the labels on the connecting lines by using the popup menu from either of the elements connected to the *Repeat* element.

In this example we also needed to use an *Exit* element, since there was nothing we wanted to do when the repeat was finished.

**Hint:** Note that in the above picture that the elements that are repeated are not connected back to the Repeat element. To do so would cause your program not to function as you expect it.

The Exit element has a special use when used in conjunction with the Repeat element. The repeat element can do one of two things. It can:

- Terminate the sequence of elements that make up the body of the repeat and continue the repeat with the next iteration
- Terminate the repeat completely. That is, the next element executed will be the first element in the *Done* path.

Which action the Exit element takes depends upon the setting in the HCA Properties Visual Programs tab.

---

## The Validate button

After you have added all the elements you need in your program and connected them with connecting lines, use the Validate button to perform a simple check. The validate operation can't guarantee that your program does what you want, but it does check for the most obvious errors. It will check to see if:

- All elements are connected to each other
- *Test* and *Repeat* elements have two paths from them and are both labeled
- All elements have their properties set
- Other tests to see if your program is correctly constructed

If you don't use the Validate button, the validate operation is automatically done when you change to another tab in the property dialog or attempt to close the dialog with the OK button.

The first thing that the Validate operation looks for is to make sure that all elements are connected together. That is, there are no elements which can't be reached from the Begin Here element by some path. Normally this is a good test. But you can instruct the program validation to not check for this. There is an option on the HCA Properties Visual Programs tab to set the behavior you want. It can be useful when you are developing programs to cut sections of the program off but not to remove them from the programming canvas.

---

## Troubleshooting: Getting programs to do what you want

Drawing programs is the easy part. HCA makes it easy to draw your program, and check it. Getting the programs to do what you want is a bit trickier. This section discusses some ways that you can check how your programs are working. **The next chapter discusses the Debugger tool that can really help a lot.**

When you create a program, think very carefully about what you want the program to do, and, most importantly, the sequence for the events. You may have a program that turns on a couple of lights. Maybe the order in which you control these lights is not important, but then maybe it is. You need to consider the intent and expected outcome to properly sequence your elements. Once you have a clear idea of what you want to do, you can begin drawing your program.

You may want to start with a simple program and add to it as you refine what it needs to do. You may want to keep a version of your base program that works all right before you improve it. Just make a copy of it, and paste the copy into your design with a new name, and work on that one.

When designing your program, you may want to watch for the use of the Delay, Wait, and Repeat elements. These are very useful in the proper circumstances, but they may also be used inappropriately. Remember that a program can be scheduled to start at any time you want—just like scheduling a light to go on at a given time. If you find that you want a program to do a few things, wait, and then do some more things, you may really need two programs. And you may want to schedule them both.

The Delay element is very useful in circumstances where its clear you want to wait, not until some future time, but just for a short pause. For example, turn on some lights, wait a minute then turn them off.

Let's assume you have written your program and it contains some *Test* elements. You schedule the program to start at a certain time and you notice, after that time has passed, that things aren't quite right in your home. How can you tell what the program did, and why it didn't work correctly? You can answer this question by using the Log.

The log is covered in the Tools chapter but the general idea is simple: The log contains an entry for each element in the program as it is executed. Using this you can quickly get a "picture" of what the program did.

If you want to see only when the program starts and stops, enable the “Log when programs start and stop”.

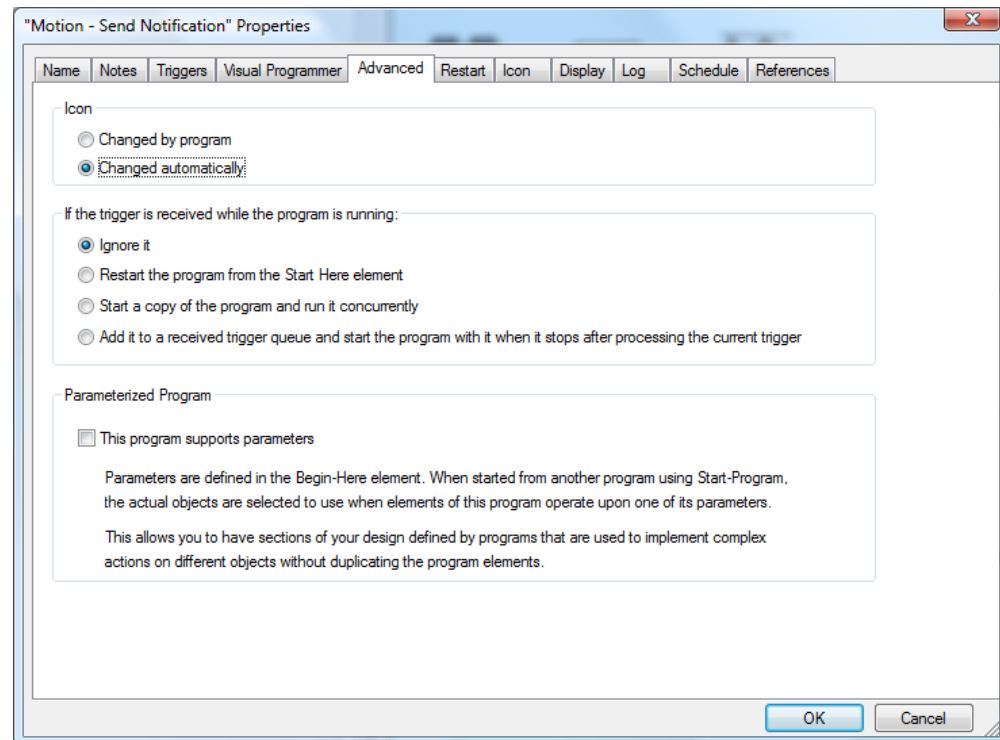
When log entries are made for the *Test* and *Repeat* elements, a note is made of which path is taken from those elements.

You can also use the *Log* element to add your own messages to s log.

---

## Program properties Advanced tab

The Advanced tab of the program Properties dialog box has two options



### Icon Option

The first of these options, changing the icon, is discussed earlier in this chapter (see Change icon).

### Trigger options

The second option determines what happens if the housecode and unitcode that starts a program is received and the program is already running.

There are four possible choices for this option:

- The trigger can be ignored
- The program can be restarted from the *Start Here* element.
- Another copy of the program can be started and will run concurrently with all other running copies of the program.
- The trigger can be queued if the program is currently executing then when it completes the program is started with the trigger at the start of the queue.

Which option is best depends upon your specific needs.

Supports Parameters

As described in the section above on parameters, this is the option that enabled is the program can be started with parameters. Until this option is enabled the Begin-Here element can't be configured to define what the parameters are.

## Chapter 11

# Visual Program Debugger

In the previous chapter on programs a section titled *Getting programs to do what you want* discussed using the log to trace how programs execute. That is a useful technique but it would be much better if you could execute each element of a program, one at a time, and follow the progress of the program that way.

HCA contains a tool to do just that called the *Program Debugger*.

If you are familiar with any program development environment that has a debugger then the facilities the HCA debugger provides will be familiar.

With the HCA debugger you can:

- Execute one element at a time
- Set breakpoints. A breakpoint set on an element causes the program to stop before that element executes and turns control over to the debugger
- Set the element about to be executed so you can change what the program does next
- Examine flag values and modify them

There are two different ways to interact with the debugger. A program can be started by the debugger so you have control right from the start. Or, in what is known as “just in time debugging”, the program can start in the normal course of events by some trigger or schedule and then when it encounters an element with a breakpoint, control is turned over to the debugger.

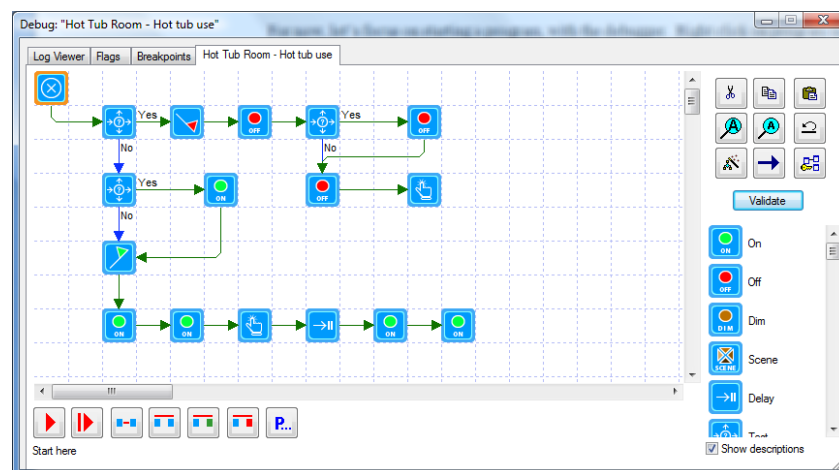
The program debugger operates in either stand-alone or client-server mode.

---

### Using the Debugger: The basics

Before describing in detail all the features of the Debugger, let’s just jump into using it so you can see how it works.

For now, let’s focus on starting a program with the debugger. Right click on program in the design pane, or on the icon for a program in the display pane and select *Debug* from the popup menu. The debugger opens.



The first thing to note about the debugger is that it looks a lot like the Visual Programmer. The debugger dialog has the same programming canvas, the same tool panel, and the same list of elements.

There are two differences you can immediately see: the seven buttons at the bottom and the fact that the Start Here element has an orange border.

The element with the orange border is what is called the *current element* and is the element that has not yet executed but when the debugger is told to proceed, it is the element first executed. At the bottom of the dialog the text image of the current element is shown. Unlike the Visual Programmer this text always matches the current element unless you place the mouse over another element and then it shows that element until you move the mouse off.

## The debugger window

The next thing to note about the debugger window is that it is a separate window from the HCA application window. You can minimize it separately from the HCA window. While it is open you can still perform other actions with HCA: View and modify the properties for devices, groups, and programs, look at schedules, etc.

There is one main limitation with what you can do while the debugger is open and that is discussed in a later section.

## Debugger Buttons

The buttons at the bottom of the debugger dialog control the actions of the debugger. They are from left to right:

**Run until breakpoint.** Runs the program as normal from the current element until the program either completes or the program encounters an element with a breakpoint. This is called the **Run** button.

**Restart from Begin-Here.** Changes the current element to be the Begin-Here element.

**Execute current.** As it says, executes the current element and then determines a new current element. If the element executed is a *Test* element then the next element is based upon the test outcome. This is also called the **Step** button.

**Don't execute current.** In this case the current element is not executed and the next element is determined as usual. This and the next two buttons are collectively known as the **Skip Over** buttons since they skip execution of an element.

**Don't execute and take the YES path.** Same as the *don't execute current* button except when the current element is a *Test* element. In that case the first element in the *Yes* path becomes the current element.

**Don't execute and take the NO path.** Same as the *don't execute current* button except when the current element is a *Test* element. In that case the first element in the *No* path becomes the current element.

**Open program.** Opens a dialog where you can select a program so you can set breakpoints in it. This can be useful if the program you are debugging uses the *Start-Program* element to execute another program and you want to set a breakpoint in that program before the *Start-Program* element starts it.

## Debugger Tabs

At the top of the debugger dialog are a number of tabs. These are:

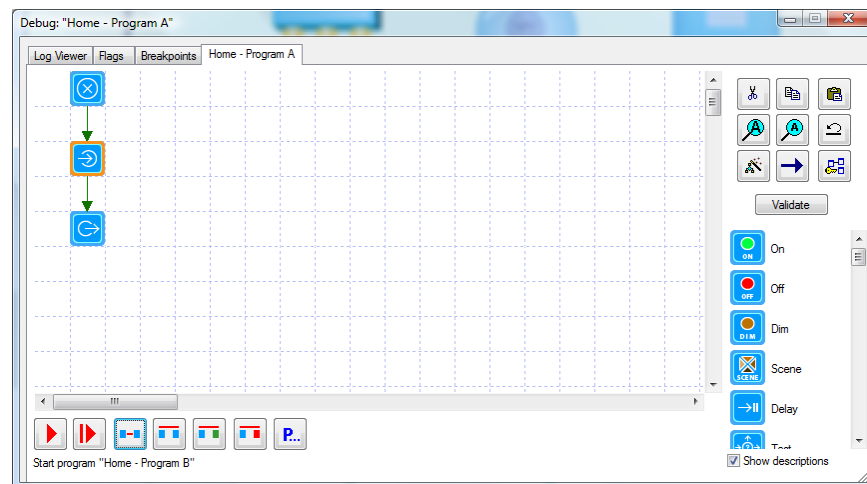
**Log Viewer.** A similar log viewer that opens from the ribbon *Control* category. Having it as a tab in the debugger may make managing multiple windows simpler.

**Flag Viewer.** A simplified viewer for flags with the ability to change their current value.

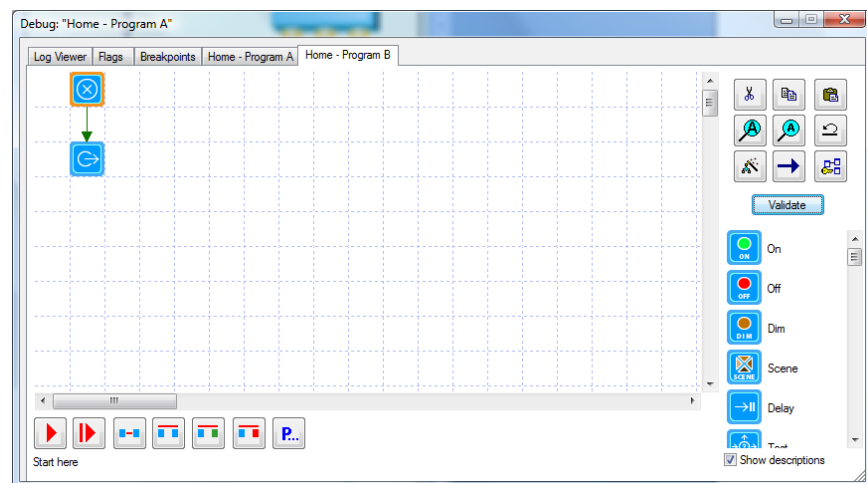
**Breakpoints.** A list of all breakpoints in all programs and on what elements they are set. From this list the breakpoints can be disabled and enabled.

The next tab is the program being debugged. Programs, as described in the previous chapter can start another program using the *Start-Program* element. In a traditional programming language it would be like a *subroutine* invocation.

If you use the *Start-Program* element then the program that is started appears in an additional tab and the current element is show in that tab. For example, suppose program A uses the *Start-Program* element to start program B. If program A is started in the debugger then the debugger window looks like this:



Press the *Step* button and the current element becomes the *Start-Program* element. Press the *Step* button again and the *Start-Program* element executes and a tab for program B appears.



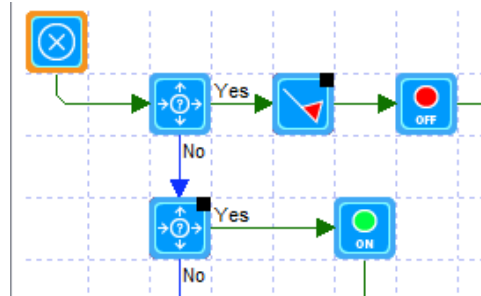
Note that the current element for program B is the *Begin-Here* element. If you switch back to the *Program A* tab, the current element is still at the *Start-Program* element since it has not yet completed.

Again, in a traditional programming language, the tabs at the top of the debugger window represent the execution *stack* with the program at the bottom of the stack showing in the left-most tab and the program at the top of the stack showing in the right-most tab.

If you continue to use the *Step* button, Program B finishes and the tab for it disappears.

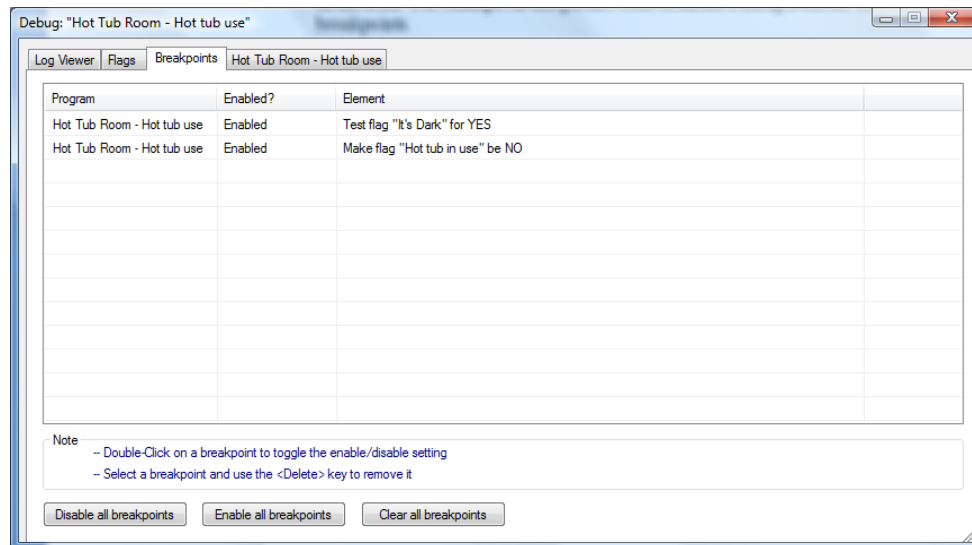
## Breakpoints

When a program is about to be executed and a breakpoint has been set on that element then execution stops. You can tell an element that has a breakpoint by a black square in the upper right of the icon. For example in this picture both elements leading from the Test element have breakpoints.



To set a breakpoint on an element, right-click on it and choose *Set Breakpoint* from the popup menu. If the element already has a breakpoint the menu selection is *Clear Breakpoint* which removes the breakpoint.

If you want to see a list of all breakpoints, select the *Breakpoints* tab.



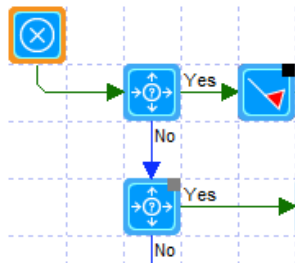


In addition to being set on an element, a breakpoint can be enabled or disabled. When you first add a breakpoint it is automatically enabled. If you no longer want execution to stop at that element you can disable or remove the breakpoint.

Why disable a breakpoint but not remove it? Sometimes during debugging you may want to temporarily disable a breakpoint rather than remove it. By doing that you are saved the trouble of opening the program again and reestablishing the breakpoint. Also, enable and disable can be simply done from the breakpoint list.

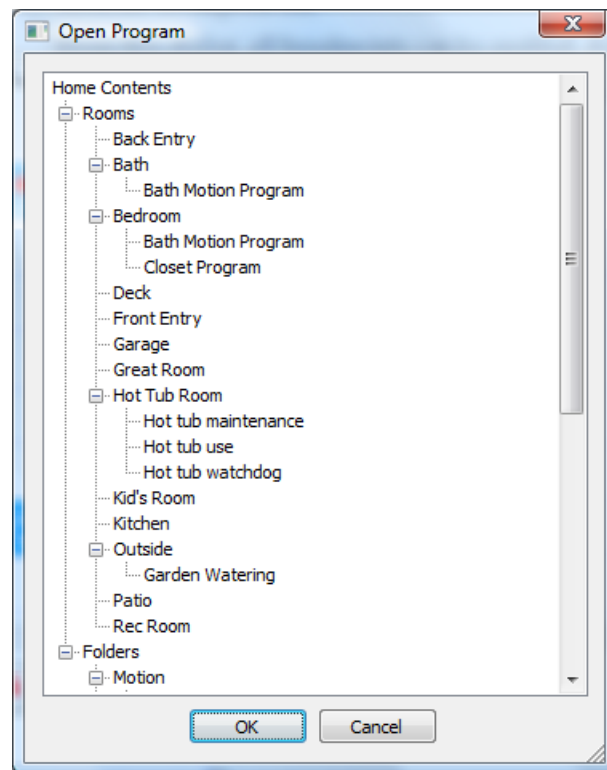
In addition to working with a single breakpoint, all breakpoints can be enabled, disabled, or cleared using the buttons at the bottom on the *Breakpoints* tab.

When a breakpoint is disabled the black square is shown in a lighter color.



The *No* path from the test element has a disabled breakpoint. The *Yes* path element has an enabled breakpoint.

If you want to set a breakpoint in a program that is not the program being debugged, press the *Programs* button – the right-most button with the “P” below the programming canvas. A list of all programs in the design opens.



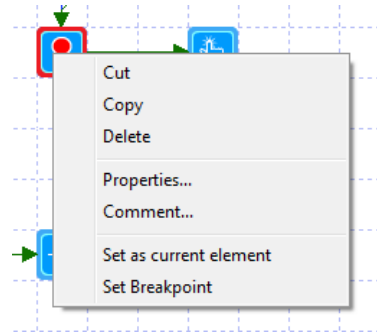
Choose a program and press OK. That program's properties dialog opens and you can locate and set a breakpoint on one or more elements. As before, right-click on the element and choose *Set Breakpoint* from the popup menu.

---

### Changing the current element

Sometime during debugging you want to change the current element. This could be because you want to move to a different point in the program to test the elements there or to skip over elements that you don't want to test.

To change the current element, right-click on the element you want and select *Set as Current Element* from the popup menu.

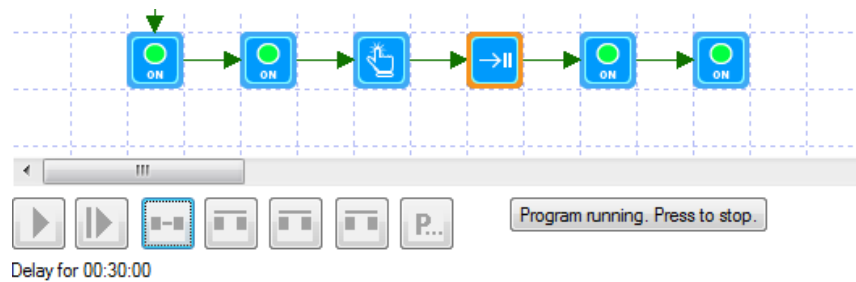



---

### Fast and Slow elements

Elements when executed in the debugger are classified into *slow* and *fast* elements. For example, an ON element is fast since it quickly queues for transmission an ON message to the device. A Delay element is a slow element since it can take minutes or hours to complete depending upon its settings.

When a fast element executes, the next element is selected usually so quickly that you probably will not notice a slight pause. But when a slow element executes then the debugger changes to show an extra button.



Note that the buttons to control the debugger have been disabled and an additional button has appeared. This button when pressed aborts the execution of the current element – in this example a *Delay* element. Once you stopped execution of the current element you can move to the next element by using the *Skip* button or by changing the current element as described above.

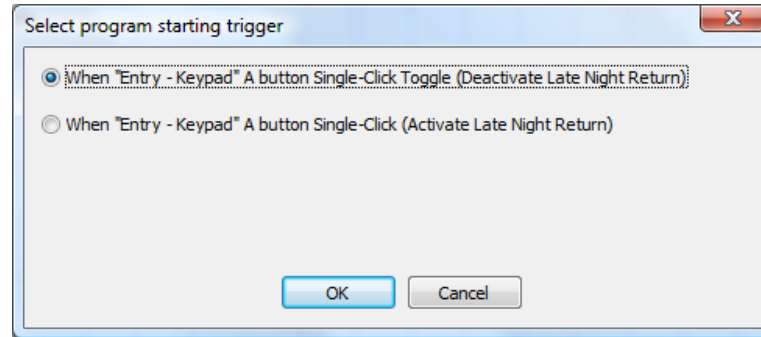
If you stop execution of the current element and then press the *Step* or *Run* button that element starts executing again.

---

## Setting the starting trigger

As described in the previous chapter, programs are started by triggers and a program can have many different triggers to start it.

If a program with multiple triggers is run in the debugger, when the *Run* or *Step* button is pressed when *Begin-Here* is the current element then a popup appears to let you select the starting trigger.



This program has two triggers. If there were other triggers then additional options or a dropdown would be shown so you can select the trigger to use.

Selecting a starting trigger is important if the program tests to see which trigger started it. Using the debugger in this way you can quickly test all the various triggers even those that don't often occur.

---

## Modifying programs

When in the debugger you can modify the program being debugged. You are free to delete elements, add elements, change the connecting lines between elements, and modify the element properties.

If you do make a change then when execution continues – you press the *Step* or *Run* buttons – a popup asks you to save the modified program.

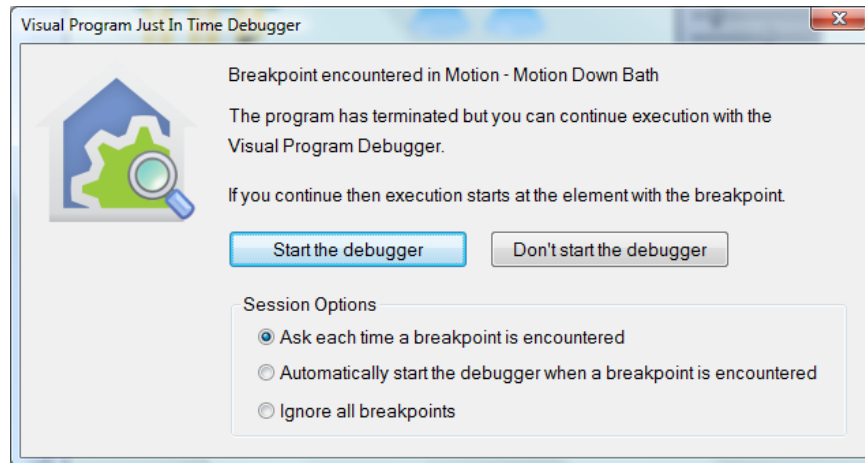
---

## Just in time debugging

In all the above discussion the debugger was started by selecting *Debug* from a right-click popup menu on a program name or icon. In addition to this method you can instead use what is called the *Just In Time* debugger.

The *Just In Time* debugger is the same debugger as described above but rather than starting the program in the debugger, you instead use the debugger to set breakpoints in one or more programs then close the debugger. Those breakpoints remain set. As your design is run by HCA, programs start and stop, schedules are monitored, etc. If a program encounters an element with one of the breakpoints you have previously set, then the debugger starts and you can debug from that point forward.

When a breakpoint is encountered in this manner then a popup dialog appears:

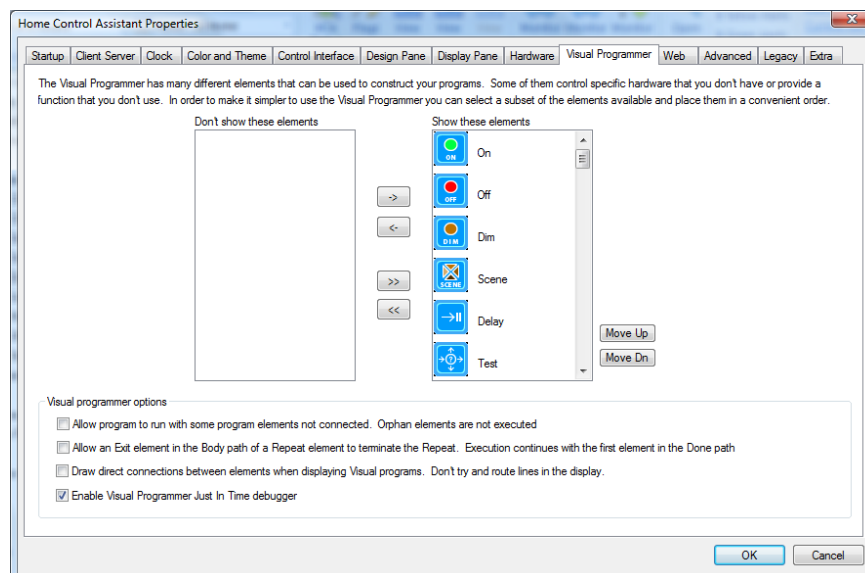


If you press the *Start the debugger* button then the debugger opens and the current element – the one with the breakpoint that has not yet been executed – is designated with the orange border showing it is the current element.

If you don't choose to start the debugger then the program is stopped - it doesn't continue.

The *Session Options* settings in the popup lets you configure the action of this popup if subsequent breakpoints are encountered. These stay in effect until HCA is restarted.

Before you can use *Just In Time* debugging it must be enabled in the HCA options on the *Visual Programmer* page. It is the last option in the *programmer options*.




---

## Debugging Limitation

There is one major limitation of program debugging: Only one program can be debugged at a time. Normally this is not a problem but if you have several programs with breakpoints set then you should be aware of this limitation.

Suppose you have two programs – A and B – and both have breakpoints set. Program A is triggered and the debugger starts. Assume that program B is also triggered.

Because of the limitation of being able to debug only one program at a time, while program A is being debugged the breakpoint in program B is ignored and program B just continues as if it had no breakpoints.



## Chapter 12

# Expressions

This chapter describes the expression services in HCA. Expressions are used like in a traditional programming language to change the value of a variable – in HCA called flags.

This chapter covers these topics:

- Introduction to HCA expressions
- The Visual Programmer Compute and Compute Test elements
- The expression builder
- Managing flags
- Important uses of flags besides the Visual Programmer
- Error handling
- Expression syntax and built-in functions

In most cases the simpler flag values – Yes and No – and the three Visual Programmer elements – Make flag yes, Make flag No, and Not flag are sufficient for applications. The Compute and Compute test elements are used for more sophisticated programming.

---

### Introduction to expressions

As described in the chapter on the Visual Programmer, HCA flags are usually used with simple Yes and No values. But in addition to those you can create and manipulate variables that can store text, numeric, Boolean, or date-time values.

To maintain compatibility with previous HCA versions, these variables are still called "flags".

Each flag can contain data of any type. HCA converts the data to the type it needs for the operator being evaluated. For example, consider these expressions:

```
a = 10
b = 20
c = "The result is" + (a + b)
d = #01-01-2001#
e = a - "8"
```

After these expressions are evaluated:

```
a is a number with value 10
b is a number with value 20
c is a string with value "The result is 30"
d is a date
e is a number with value 2
```

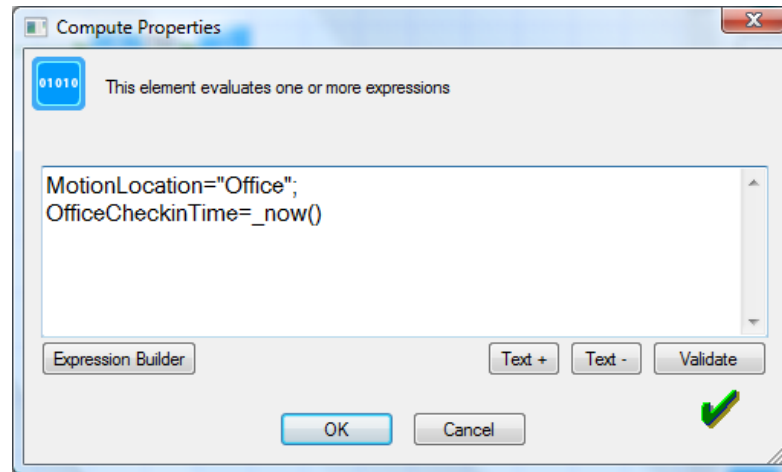
If you understand, or can learn about, how expressions in traditional programming languages like Visual Basic work, you will understand HCA expressions.

---

## Compute and Compute test visual programmer elements

To use these expressions two visual programmer elements are available: Compute and Compute Test.

The properties of the Compute element are:



In the Compute element is placed a series of expressions in the form:

<flag name> = <expression> ;

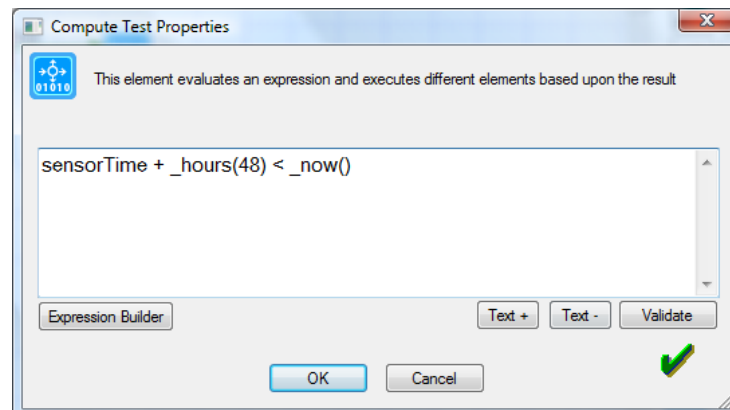
<flag name> = <expression> ;

...

<flag name> = <expression>

When the compute element is executed, the expressions are evaluated and the computed values are assigned to the named flags.

The Compute Test element contains a single expression that is evaluated to determine a yes or no value. If the value is "yes" the path marked "yes" in the program is taken from the Compute Test element, and likewise for "no".



In both these elements the Validate button is used to check that the expression you have entered is correct – it matches the syntax that HCA expects.



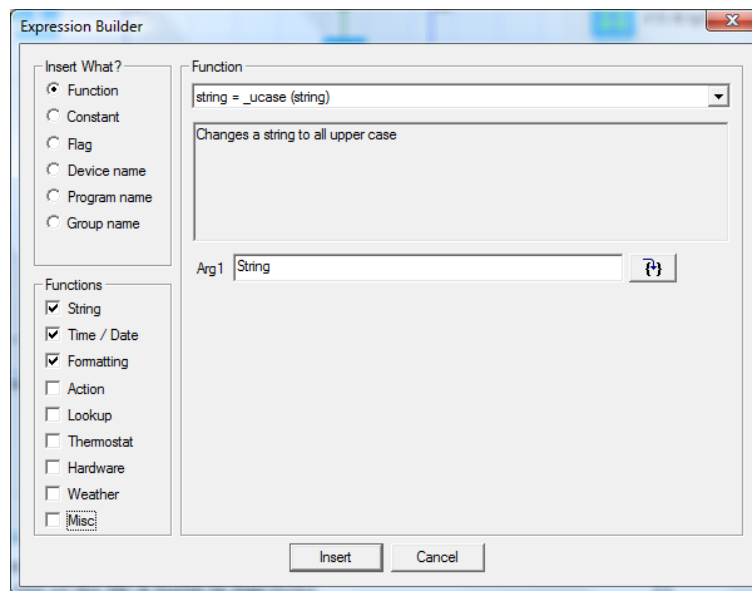
A lot of work went into the Visual Programmer to allow HCA users to create programs without all the baggage of existing programming languages – careful syntax, programming terms and concepts. These two elements take a step back from that and leave you in the realm of the programmer. If you have never used, for example, Visual Basic, or all this sounds Greek to you, stick with simple yes and no flags managed with the visual programmer elements for them. You can do many wonderful things with them alone.

---

## Expression builder

To help you create expressions, rather than always having to refer to this documentation, HCA contains a tool called the Expression Builder.

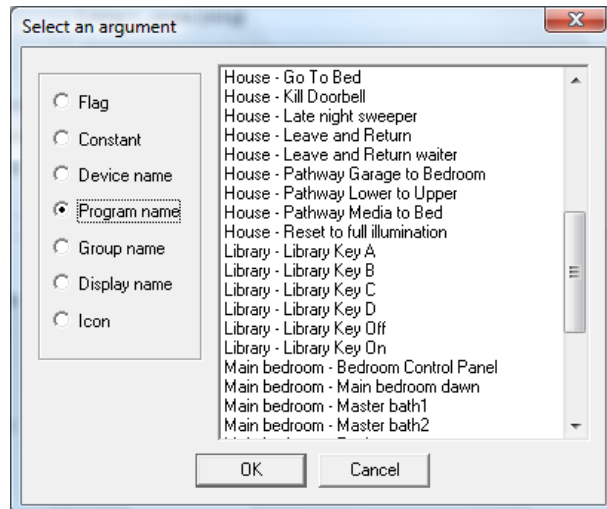
On dialogs where you enter an expression, a button labeled *Expression Builder* helps you compose your expression. Pressing this button opens this dialog:



The upper left box specifies what sort of item you are inserting. The most common case is one of the HCA build-in functions.

The box below that lets you limit the number of choices of the possible functions you have to choose from.

The third section of the dialog changes depending upon what you are inserting. In the picture above, a function is being inserted. Choose the name of the function in the dropdown and two things display: a short explanation of what it does, and the parameters to the function. In this example, the `_right` function takes two arguments. You can simply type in the two arguments, or to get more assistance, press the button next to the argument. This opens this dialog:



This dialog lets you insert common things that you may want to work with. Things like the names of the objects in your design, flags, and constants.

When you close the Expression Builder, the constructed expression is inserted into the text of the element properties at the cursor. Or it replaces the current selection if there is one.

---

## Managing Flags

An important point about flags is that they get created when expressions are evaluated. There is no other way than that to create a flag. That is, there is no "new flag" wizard. When a program is executed any new flags that are used in its expressions appear in the flags list in the flags inventory dialog.

The flags inventory dialog is described in the chapter on design tools.

---

## Other uses for expressions

In addition to using expressions in the Compute and Compute Test visual programmer elements, you can place expressions in other elements. Just enclose the expression in %'s. When the element is executed, the expression is evaluated and the result in text form replaces the % enclosed section. For example, to show the value of an expression, use this text in the ShowMessage element:

The value of beta is %beta1 + beta2%

If your string wants to display a percent sign, use two in the string:

Inside humidity is %humInside%%

These placeholders - % enclosed sections - can be used in these elements:

- Show Message in the text
- Run Program element in the command line
- Speak
- Change Icon
- Add to Log

In these elements, an Embed Expression button appears. This lets you build an expression then encloses it in %'s when it places the expression into the element's properties.

---

## Error Handling

Because these elements happen at a more complex layer of HCA than most elements, errors can happen that could not be detected in the Visual Programmer. If errors occur while executing these elements, the errors are logged, the Compute or Compute Test element is abandoned, and execution continues with the next element. In the case of the Compute Test element execution follows the Yes path. These errors show up with a red "P" marker and can be filtered as an Error.

Some of the possible errors are:

- Naming a device, thermostat or a magic module as an argument to a function and no device, thermostat, or magic module with that name in your design.
- Divide by zero.
- Using a weather function but no weather provider available.
- Trying to construct a date-time with something out of range. Like a month of 13.

---

## Expression Syntax

HCA expressions are very similar to expressions in any programming language like Visual Basic or VBScript. The usual operators are available:

Comparison operators

< > <= >= <> ==

(**Note** that the operator that checks for equality is 2 equal signs not one)

Arithmetic operators

+ - \* / mod

- (unary minus) + (unary plus)

Logical operators

and or not eqv imp xor

Binary operators

Binary or is a Vertical bar |

Binary and is an Ampersand &

Binary not is a circumflex ^

Date and Time constants enclosed in #'s as: #1/15/2001 07:19 AM#

Boolean constants: Yes No

String constants can be enclosed in single or double quotes

If you are using a flag in an expression and the name of that flag has a blank in it, enclose the flag name in square brackets. For example [My Flag]

In creating expressions there are a number of functions that HCA provides. Some of these are very general and can be found in almost any programming language, and others are specific to HCA.

All functions provided by HCA begin with the underscore character. As long as none of your flag names begin with an underscore, if in subsequent versions of HCA new functions are added, none of your flag names will conflict with any new function names.

**Hint:** Don't start any of your flag names with the underscore character.

---

## String functions

The string functions are identical to the Visual Basic functions of the same name.

**string = \_ucase (string)**

example: `_ucase("web")`

result: "WEB"

**string = \_lcase (string)**

example: `_lcase("WEB")`

result: "web"

**string = \_left (string, n)**

example: `_left("web", 2)`

result: "we"

**string = \_right (string, n)**

example: `_right("web, 2")`

result: "eb"

**string = \_mid (string, n, m)**

example: `_mid("webber", 2, 3)`

result: "ebb"

**number = \_len (string)**

example: `_len("web")`

result: 3

**string = \_LTrim (string)**

example: `_LTrim(" example")`

result: "example"

**string = \_RTrim (string)**

example: `_RTrim("example ")`

result: "example"

**string = \_Trim (string)**

example: `_Trim(" example ")`

result: "example"

**string = \_chr (number)**

example: `_chr(65)`

result: "A"

**number = \_asc (string)**

example: `_asc("A")`

result: 65

**number = \_InStr (string, string)**

example: `_InStr("webber", "bb")`

result: 3

**string = \_TextPiece (string, number, string)**

example: `_TextPiece("apple, banana, grape", 2, ",")`

result: "banana"

**string = \_TextReplace (string, number, string)**

example: `_TextPiece("Wind speed 10 mph", "mph", "miles per hour")`  
 result: "Wind speed 10 miles per hour"

**number = \_HexToDec (string)**

example: `__HexToDec("1A7")`  
 result: 423

**number = \_DecToHex (number, number of digits)**

example: `__DecToHex(423, 4)`  
 result: 01A7

**Bool = \_Match (string, pattern string)**

Performs a regular expression match between the supplied string and pattern. If the expression matches the pattern then the function returns Yes.

example: `__Match("00773", "00.*")`  
 result: Yes

---

**General use functions**

These functions are generally useful and many are similar to Visual Basic functions.

**Bool = \_IsDate (any)**

Returns YES if the argument is a datetime or an expression that evaluates to a dateTime.

**Bool = \_IsText (any)**

Returns YES if the argument is a string or an expression that evaluates to a string.

**Bool = \_IsBool (any)**

Returns YES if the argument is a yes/no value or an expression that evaluates to a Yes/No value

**Bool = \_IsNumber (any)**

Returns YES if the argument is a number or an expression that evaluates to a number

**number = \_Max (number, number, ...)**

Returns the largest number from any of the arguments provided. You can have up to 10 arguments.

**number = \_Min (number, number, ...)**

Returns the smallest number from any of the arguments provided. You can have up to 10 arguments.

**number = \_Abs (number)**

Returns the absolute value of the number. That is, always a positive number.

**number = \_int (number)**

Returns the number as an integer.

**bool = \_IsOdd (number)**

Returns yes if the number is odd

**bool = \_IsEven (number)**

Returns yes if the number is even

**number = \_Round (number)**

Rounds the number to the nearest integer

**any = \_Choose (number, any, any, ...)**

Returns as its result the Nth argument. The 1<sup>st</sup> argument chooses which argument to be returned. The *any* arguments can be of any types and there can be up to 10 of them. For example: `_Choose (3, "Jan", "Feb", "Mar", "Apr", "May")`

Result: "Mar"

**any = \_iif (bool, any1, any2)**

Returns any2 if the 1<sup>st</sup> argument is NO. Otherwise returns any1.

**number = \_rand (number, number)**

Returns a random number chosen between the two numbers supplied.

**bool = \_PlaySound (string, number)**

Plays a Sound file using the computers sound system. The 1<sup>st</sup> argument is a path to the sound file. The second argument is as follows:

- 1: The sound file starts playing and HCA moves to the next element
- 2: The sound file starts playing and HCA moves to the next element. When the sound file finishes, it starts playing again.
- 3: The sound file starts playing and HCA waits until it is complete before moving to the next element

If you used option #2, at a later time you can stop the sound file playing by using the `PlaySound` function again with "" for the path.

## Time and date functions

For these examples, assume that the current time is 02:12:45 pm and the current date is Friday 14-September-2001

**number = \_hour (dateTime)**

example: `_hour(_now())`

result: 14

**number = \_minute (dateTime)**

example: `_minute(_now())`

result: 12

**number = \_second (dateTime)**

example: `_second(_now())`

result: 45

**dateTime = \_time (hour, minute, second)**

example: `_time(14, 12, 45)`

result: A time of 02:12:45 pm

**number = \_day (dateTime)**

example: `_day(_now())`

result: 14

**number = \_month (dateTime)**

example: `_month(_now())`

result: 9

**number = \_year (dateTime)**

example: `_year(_now())`  
 result: 2001

**dateTime = \_date (year, month, day)**

example: `_date(2001, 9, 14)`  
 result: A date of 14-September-2001

**dateTime = \_datetime (year, month, day, hour, minute, second)**

example: `_datetime(2001, 9, 14, 14, 12, 45)`  
 result: 14-September-2001 02:12:45 pm

**number = \_totalHours (dateTime)**

Returns the number of hours represented by the date-time.  
 example: `_totalHours(_time(7,30,50))`  
 result: 7

**number = \_totalMinutes (dateTime)**

Returns the number of minutes represented by the date-time.  
 example: `_totalMinutes(_time(7,30,50))`  
 result: 450

**number = \_totalSeconds (dateTime)**

Returns the number of hours represented by the date-time.  
 example: `_totalSeconds(_time(7,30,50))`  
 result: 27050

**number = \_dayOfWeek (dateTime)**

Returns the day of the week as a number from 1 to 7, where 1 is Sunday  
 example: `_dayOfWeek(_now())`  
 result: 6

**number = \_dayOfYear (dateTime)**

Returns the day of the year as a number from 1 to 366, where 1 is January 1st  
 example: `_dayOfYear(_now())`  
 result: 257

**dateTime = \_now ()**

Returns the current date-time

**dateTime = \_sunrise ()**

Returns the time of sunrise for today

**dateTime = \_sunset ()**

Returns the time of sunset for today

**dateTimeSpan = \_Days (number)**

Returns a time span of the given number of days. See below for some date time span examples.

**dateTimeSpan = \_Hours (number)**

Returns a time span of the given number of hours

**dateTimeSpan = \_Minutes ()**

Returns a time span of the given number of minutes

**dateTimeSpan = \_Seconds()**

Returns a time span of the given number of seconds

**dateTimeSpan = \_TimeSpan (days, hours, minutes, seconds)**

Returns a time span of days, hours, minutes, seconds

**string = \_weekday (number)**

Returns a string of the three letter abbreviation of the weekday.  
 The argument is the number of days to go back. So if today is Monday then

- `_weekday(0)` is "Mon"
- `_weekday(1)` is "Sun"
- `_weekday(2)` is "Sat"

**string = \_weekdayName (number, bool)**

Returns a string of the weekday. If the 2<sup>nd</sup> parameter is 0, the three letter abbreviation is used.

- `_weekdayName(1, 0)` is "Sun"
- `_weekdayName(1, 1)` is "Sunday"
- `_weekdayName(7, 0)` is "Sat"

**string = \_monthName (number, bool)**

Returns a string of the month. If the 2<sup>nd</sup> parameter is 0, the three letter abbreviation is used.

- `_monthName(1, 0)` is "Jan"
- `_monthName(1, 1)` is "January"
- `_monthName(12, 0)` is "Dec"

**string = \_FormatTime (dateTime, pattern)**

Returns a string of the date-time formatted according to the pattern. The pattern is a string made up of these replacements:

Pattern marker	Meaning
\$a	Abbreviated weekday name
\$A	Full weekday name
\$b	Abbreviated month name
\$B	Full month name
\$c	Date and time appropriate for locale
\$d	Day of month as number (01-31)
\$H	Hour in 24-hour format (00-23)
\$I	Hour in 12-hour format (01-12)
\$j	Day of year as a number (001-366)
\$m	Month as a number (01-12)
\$M	Minutes as a number (00-59)
\$p	Current locale's AM/PM indicator for 12-hour clock
\$S	Second as a number (00-59)
\$U	Week of year as a number, with Sunday as the first day of the week (00-51)
\$w	Weekday as a number (0-6). Sunday is 0.
\$W	Week of year as number with Monday as the first day of the week (00-51)
\$x	Date representation appropriate for locale
\$X	Time representation appropriate for locale
\$y	Year without century as a number (00-99)
\$Y	Year with century as number
\$z or \$Z	Time-zone name or abbreviation. Blank if not known



\$\$	Dollar sign
------	-------------

**There are four major uses of the time functions in the Compute element. These are:**

- Determine how long something took. This is done by:
 

```
t = _now()
... do something...
timeItTook = _now() - t
```
- Add or subtract from the current time to generate a date-time in the past or future:
 

```
TwentyFourHoursAgo = _now() - _days(1)
SixAndAHalfHoursAgo = _now() - _timeSpan(0, 6, 30, 0)
```
- Compose a date-time from its component parts:
 

```
t = _DateTime(2003, 3, 15, 9, 8, 3)
```
- Format a date-time to a string:
 

```
s = _FormatTime(_now(), "$d-$b-$y $H:$M")
```

 This would show as "15-Mar-03 09:08"

---

## Numeric formatting functions

### **string = \_FormatNum (number, # of decimal places)**

Converts the number to a string with the given number of digits after the decimal point.

`_FormatNumber(1.6764, 1)` evaluates to "1.6"

### **string = \_FormatInt (number, # digits, leading zeros?)**

Converts the number to a string with no fractional part. If the third parameter is supplied and if it evaluates to Yes, the string contains leading zeros.

`_FormatInt(100.5,4)` evaluates to " 100"

`_FormatInt(100.5,4,1)` evaluates to "0100"

### **string = \_FormatPattern(number, string pattern)**

This function is intended for users familiar with programming languages. The pattern uses the same pattern characters as the C language `printf` function. Refer to C runtime documentation or books on the programming language for full info. This documentation is not included here.

---

**Device / Program / Group / Schedule functions**
**string = \_status ("name")**

Returns text that contains the status of the device, program, group, or controller. The text is designed to be displayed to the user rather than operated on by subsequent Compute expressions. What is returned depends upon the type of object. For example, a device returns “ON”, “OFF”, or “Dim xx%” where a program would show “Running” or “Not running”. In no case is the device queried to find out its true status – the status returned is based upon HCA’s internal state.

**bool = \_isOn ("name", send status?)**

Attempts to look up the supplied name as a device, program, group, or controller. If it is On, or is running if a program, then return Yes, otherwise No. If the second parameter is not given, or evaluates to No, the status of the device is determined from the internal HCA state. If the parameter is supplied and evaluates to Yes, and if the device supports status requests, a status request is sent to the device and the response back determines the result.

**bool = \_isOff ("name", send status?)**

Attempts to look up the supplied name as a device, program, group, or controller. If it is Off, or if a program the program is not running, then return Yes, otherwise No. If the second parameter is not given, or evaluates to No, the status of the device is determined from the internal HCA state. If the parameter is supplied and evaluates to Yes, and if the device supports status requests, a status request is sent to the device and the response back determines the result

**bool = \_isDim ("name", sendStatus?)**

Attempts to look up the supplied name as a device, group, or controller. If it is Dim then return Yes, otherwise No. If the second parameter is not given, or evaluates to No, the status of the device is determined from the internal HCA state. If the parameter is supplied and evaluates to Yes, and if the device supports status requests, a status request is sent to the device and the response back determines the result

**Additional note on \_IsOn, \_IsOff, and \_IsDim functions:**

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.

**bool = `_isRunning ("programName")`**

Attempts to look up the supplied name as a program. If it is running then return Yes, otherwise No.

**bool = `_isDim ("name", sendStatus?)`**

Attempts to look up the supplied name as a device, group, or controller. If it is Dim then return Yes, otherwise No. If the second parameter is not given, or evaluates to No, the status of the device is determined from the internal HCA state. If the parameter is supplied and evaluates to Yes, and if the device supports status requests, a status request is sent to the device and the response back determines the result

**number = `_dimLevel ("name", sendStatus?)`**

Attempts to look up the supplied name as a device, group, or controller. Returns the illumination level. If it is off, then zero is returned. If On then the maximum illumination level is returned. If the second parameter is not given, or evaluates to No, the status of the device is determined from the internal HCA state. If the parameter is supplied and evaluates to Yes, and if the device supports status requests, a status request is sent to the device and the response back determines the result

**number = `_dimPercent ("name", sendStatus?)`**

Attempts to look up the supplied name as a device, group, or controller. Returns the illumination level as a percentage of 100%, where 0 is Off and 100 is full bright. If the second parameter is not given, or evaluates to No, the status of the device is determined from the internal HCA state. If the parameter is supplied and evaluates to Yes, and if the device supports status requests, a status request is sent to the device and the response back determines the result

**bool = `_isCurrentSchedule ("scheduleName");`**

Returns Yes if the current schedule matches the name supplied.

**string = `_currentSchedule ()`**

Returns a string of the name of the current schedule. If no schedule is current returns an empty string.

**number = `_on ("name")`**

Turns the named device or group on. Returns the current dim level before the On command is sent.

**number = `_off ("name")`**

Turns the named device or group off. Returns the current dim level before the Off command is sent.

**number = `_dimToLevel ("name", level)`**

Adjusts the illumination level of the named device to the supplied level. Returns the current dim level before any commands are sent.

**number = \_dimToPercent ("name", percent)**

Adjusts the illumination level of the named device to the supplied percent. Returns the current dim level before any commands are sent.

**string = \_CurrentScene ("name")**

If the named device or controller supports stored scenes, returns the name of the scene set at the device. This function does not query the device so the result is based upon HCA internal state.

**string = \_SetToScene ("name", "scene name")**

Changes the current scene in a switch or controller to the scene named. The current scene in the device before the new scene is established is returned.

**string = \_ComposeScene ("name", level)**

Returns the name of the scene for the given Compose command (preset dim level) in the named device or controller. This only applies to Lightolier CI and CP devices in Compose mode.

**Number = \_GroupMemberCount ("group name")**

Returns the number of members in a group. Used in conjunction with the GroupMemberName function to get the name for each group member.

**String = \_GroupMemberName ("group name", number)**

Returns the name of the Nth member of the named group. To get the first member name the 2<sup>nd</sup> parameter is a 1.

**String = \_Trigger ()**

Returns a string which represents the trigger that started the program.

NOTE: If not used in the context of a program, for example in an expression used in a Status Export it returns ""

**NOTE: The function is no longer supported and should not be used in new designs**

The first character of the string is the kind of trigger. These are:

X = X10  
M = Magic Module  
W = Weather  
F = Flag  
S = Special  
E = Expression  
I = Wireless

M, W, F, S, and E triggers are not defined in HCA 5

**The X triggers are: XHUCCMDXLV**

X: Always start with 'X'

H: One character House Code 'A' .. 'P'

UC: Two characters of unit code '01' .. '16'

CMDX: 4 characters of command. Right padded with blanks to get 4 characters

"ON "

"OFF "

"DIM "

"BRT "

"AUF "

"ALO "

"ALF "

"HRQ "

"HRPL"

"PDIM"

"SON "

"SOFF"

"SREQ"

LV: 2 characters of preset dim level '00' .. '31'

If any section doesn't apply it is left blank. So a status request trigger to housecode G would look like (blanks shown as '-' for clarity)

XG--SREQ--

**The I triggers are: IIDXMMMMM**

I: Always start with 'I'

IDX = the id in the range 1-255

MMMMM: 5 characters of message. Right padded with blanks to get 5 characters

AAMIN

AAMAX

AHMIN

AHMAX

PANIC

ALERT

NORML

LTON

LTOFF

DSARM

**String = \_X10NameOf (string)**

This returns the name (text) of the device or controller with a supplied HC/UC.

The argument string must be formatted as a valid house and unit code. For example "A1" or "A16" or "G5", etc.

This first tries to find a device with that X10 address. If that fails controllers are searched for a match.

NOTE: For multi-unit devices / controllers only the base address is considered. If there are more than one device / controller with the HC/UC, one is found. No guarantees which one will be found.

**String = \_X10AddressOf (string)**

This returns the primary address of the named device or controller.

The returned string is formatted as "A1" or "A16" or "G5", etc.

**String = \_WirelessName (number)**

This returns the name (text) of a wireless component with that hardware id number.

**Number = \_IconChange (“name”, “icon name”, “display name”)**

The first argument must be the name of a device, program, group or room. The second argument is the name of an icon from the icon gallery. The third argument is the name of a display.

This operation changes the displayed icon for an object to the named icon on the named display.

If the display name argument is not given then any displays the object appear on are affected. If the icon name argument isn't given then the icon is restored to the one chosen in the object properties icon tab.

**Number = \_IconChangeEx (“name”, "code", “text”, “number”)**

The first argument must be the name of a device, program, group or room. The second argument is code:

- 0 = Return control of the object's icon back to HCA
- 1 = Change the object's icon to the named icon (3<sup>rd</sup> argument) with specified representation (4<sup>th</sup> argument: 0 = ON, 1 = OFF, 2 = DIM)
- 2 = Change th object's icon level to be the supplied text given by argument 3.

Unlike the ChangeIcon function, this function changes the icon for the named object where it appears and not just on a single display. This function always returns zero.

**string = \_ChangeSchedule (“name”, code, dateTime, dateTime)**

Modifies the named schedule entry. The code is as follows:

- 0 = Change on time
- 1 = Change off time
- 2 = Change on and off time

The first dateTime argument is the time used for code 0 and 1. The 4<sup>th</sup> argument is only needed for code 2

**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.

---

## Thermostat

These elements are useful only when a bi-directional thermostat device has been added to your design. See the thermostat appendix for information on this.

**NOTE:** The next 7 functions are still available but should not be used in new designs. The `_GetThermostat` and `_SetThermostat` functions should be used instead.

**number = `_temperature ("thermostat name")`**

Returns the current temperature read from the named thermostat. The thermostat must have two-way capabilities. The returned value is in the units selected in the properties for the thermostat.

**number = `_setPoint ("thermostat name" )`**

Returns the current setpoint read from the named thermostat. The thermostat must have two-way capabilities. The returned value is in the units selected in the properties for the thermostat.

**string = `_mode ("thermostat name")`**

Returns the current mode read from the named thermostat. The thermostat must have two-way capabilities. The returned value is a string of "Heat", "Cool", "Auto", or "Off"

**bool = `_isFanOn ("thermostat name")`**

Returns Yes or No if the fan is On or Off when the named thermostat is queried.

**bool = `_isEconomy ("thermostat name")`**

Returns Yes or No if the thermostat is in economy or setback mode when the named thermostat is queried.

**bool = `_isAuxHeat ("thermostat name")`**

Returns Yes or No if the thermostat is in aux heat mode when the thermostat is queried.. Not supported by all thermostats..

**number = `_humidity ("thermostat name")`**

Returns the current humidity read from the named thermostat. The thermostat must have two-way capabilities. Not supported by all thermostats.

**number = `_TempDecode (Unitcode, level)`**

Uses the standard X10 RCS temperature decode table (also used by the SmartHome TempLinc), to change a unit code and preset dim level into a temperature. For example:

`_TempDecode(15, 2) = 70`

The 1<sup>st</sup> argument is the unit code given as a number from 11 to 16 corresponding to UC 11 to UC 16. The 2<sup>nd</sup> argument is the preset dim level between 0 and 31, inclusive.

**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 uses 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.

**Bool = `_SetThermostat` (“thermostat name”, code, value, code , value, ...)**

You can supply up to 11 arguments. The 1<sup>st</sup> 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.



---

## Weather

The weather functions are useful only if you have set up a weather provider using the weather provider setup dialog. See the weather appendix for information on this.

The `_Weather` function allows you to access all data in a weather observation. For observation data that have units – temperatures for example - the values returned are in the units selected in the weather provider setup dialog. The `_Weather` function format is:

**value = `_weather` (name)**

The "name" is the name of data from the weather observation. Since the names change as more weather providers are supported and as those providers change, the documentation of those names is on the HCA web site in the weather technical note.

**Hint:** The support web site is at [www.HCATech.com](http://www.HCATech.com). Look for the technical notes link on the main page.

In addition to the `_weather` function, other functions compute their results by examining data in the historical weather log files. If you don't have a log file setup in the weather setup dialog none of these functions work.

The "minutes, hours, days" arguments tell how far back in time to go to determine a high, low, or average.

The units used are the same as the analogous current weather item. That is, all temperatures are returned in the same units.

When using these functions it is not necessary to supply all three arguments. For example, you can determine the high outside temperature in the last 90 minutes from this expression

```
temp = _weatherHigh(OutsideTemp, 90)
```

The functions are:

**Value = `_WeatherHigh`(name, minutes, hours, days)**

**Value = `_WeatherLow`(name, minutes, hours, days)**

**Value = `_WeatherAvg`(name, minutes, hours, days)**

In addition to these functions, there are two other sets of weather functions. These also compute their result by looking at historical weather data. But in this case the argument is not a length of time but rather the number of days to go back in time.

For example if today is Monday:

**`_weatherDayHigh`(Barometer, 0)** is the barometer high for Monday

**`_weatherDayHigh`(Barometer, 1)** is the barometer high for Sunday

These functions are:

**Value = `_WeatherDayHigh`(name, day)**

**Value = `_WeatherDayLow`(name, day)**

**Value = `_WeatherDayAvg`(name, day)**

The final set of functions is like the Day function except the parameter is an hour. For example is it is 3:15pm:

**`_weatherHourHigh`(Barometer, 0)** is the high barometer from 3pm to 4pm

**`_weatherHourHigh`(Barometer, 1)** is the high barometer from 2pm to 3pm

**string = \_BarometerUnits()**

Returns a string of the current barometer units. For example "in" or "mm"

**string = \_humidityUnits()**

Returns a string of the current humidity units. Always "%"

**string = \_tempUnits()**

Returns a string of the current temperature units. For example  
"° F" or "° C"

**string = \_windSpeedUnits()**

Returns a string of the current wind speed units. For example "m/s" or "knots"

**string = \_windDirUnits()**

Returns a string of the current wind direction units. Always "°"

**string = \_wunderground([1 to 10 strings])**

Retrieves data from the Weather Underground internet site. It must be configured as your weather provider for this function to operate correctly. This function is somewhat complex and is described in the Weather user guide appendix.

**string = solarRadiationUnits()**

Returns a string of the current solar radiation units. Always  
"W/sq m"

**string = \_UVUnits()**

Returns a string of the current UV units. Always "UV index"

**string = \_soilUnits()**

Returns a string of the current soil units. Always "c"

**number = \_tempConvert (number, fromUnits, toUnits)**

Converts a temperature between F and C. The first parameter is the temperature. The 2<sup>nd</sup> parameter is the current units and the 3<sup>rd</sup> parameter is the units wanted. The encodings of the units parameters are:

F = 0

C = 1

**number = \_barometerConvert (number, fromUnits, toUnits)**

Converts a barometer measurement between the four supported units. The first parameter is the barometric reading. The 2<sup>nd</sup> parameter is the current units and the 3<sup>rd</sup> parameter is the units wanted. The encodings of the units parameters are:

Inches = 0

Millimeters = 1

Millbars = 2

Hecto Pascals = 3

**number = \_\_windSpeedConvert (number, fromUnits, toUnits)**

Converts a barometric reading between the four supported rates. The first parameter is the wind speed. The 2<sup>nd</sup> parameter is the current units and the 3<sup>rd</sup> parameter is the units wanted. The encodings of the units parameters are:

Miles per hour = 0  
 Knots = 1  
 Kilometers per hour = 2  
 Meters per second = 3

**number = \_rainConvert (number, fromUnits, toUnits)**

Converts a rain amount between the two supported units. The first parameter is the rain amount. The 2<sup>nd</sup> parameter is the current units and the 3<sup>rd</sup> parameter is the units wanted. The encodings of the units parameters are:

Inches = 0  
 Millimeters = 1

**string = \_windDirection (number)**

Changes a wind direction in degrees into a string of the form, N, NNE, NE, ENE, E, etc.

**string = \_wunderground(...)**

Gets information from Weather underground. See the weather underground technical note for use.

## File Operations

This category of functions comprises a set of functions that operate on disk based files. HCA allows a maximum of 4 files to be open at one time.

**Bool = \_FileExists(string)**

Determines if the supplied file path exists. Returns true or false.

**number = \_FileOpen(string, number)**

Opens the file at the supplied path and returns a "handle" to the opened file that can be used in the Read, Write, and Close functions. The second argument is a code:

Code	Use
0	Open file for reading
1	Open file for writing. If the file already exists delete it's contents.
2	Open file for writing. If the file already exists, append new data to the end of the file

**number = \_FileClose( number)**

Close the file. The argument supplies the handle returned by the FileOpen function.

**number = \_FileWriteString(number, string)**

Writes the string to the file. The first argument is the handle of the file returned by FileOpen. The return value is the length of the string written.

**string = \_FileReadString(number)**

Reads a string file from a file and returns it. The first argument supplies the handle of the file returned by FileOpen.

## Miscellaneous

This category of functions comprise a set of generally useful things that don't fit into any other category.

**number = \_problemLevel ()**

Returns the alert level as displayed by the red-yellow-green status indicator on the status bar. Red is 2, yellow is 1 and green is zero.

**number = \_SetProblemLevel (number)**

Changes the alert level as displayed by the red-yellow-green status indicator on the status bar to the value given. The existing level is returned.

**number = \_delay (time1, time2)**

Delays execution of the program for a given time. If one argument is given the delay is the total number of seconds in time1. If two arguments are given a random time is chosen between the two given times. The return value is the number of seconds delayed.

**number = \_DelayShort (milliseconds)**

Delays execution of the program for a given time. The time is in milliseconds. The return value is the number of milliseconds delayed.

**number = \_SetHomeMode(number)**

Changes the home mode to the mode supplied. The home mode is given as in index into the modes you select in the *Control* ribbon category *Current Home Mode* dropdown. The first mode in the list is zero, the second is 1, etc. The return value is the mode that was changed from using the same numbering.

**number = \_HomeMode()**

Returns the current home mode. Uses the same numbering as in SetHomeMode described above.

**Number = \_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. Always returns 0.

**string = \_DesignTitle ()**

Returns the design name as set in Home Properties.

**string = \_DesignTitle ()**

Returns the name of the design as set in Home Properties

**string = \_DesignSave (number)**

Saves the current design. The single argument is optional. If not supplied the design is saved unconditionally. If supplied and is zero, the design is saved unconditionally. If supplied and set to 1, the design is saved only if marked as modified.

**string = \_CurrentWattage (string)**

Returns the wattage currently used by the named object. That name could be a device name or room name. To get the current wattage of the whole home, don't supply any argument.

**number = \_CurrentWattage (string, number)**

Sets wattage of the named object to the value. That name could be a device name or room name. Always returns 0.

**dateTime = \_LastReceptionTime (string)**

Returns the time of the last reception from the named device. If the supplied string names a room then the time returned is the last reception for any device in the room./

**number = \_InsteonBeep (string, number)**

Sends a command to an Insteon device that could cause it to "beep". Not all Insteon devices support this command.

**number = \_RGB (number, number, number)**

Returns a number that is the RGB value of the 3 color components.

**number = \_\_InsteonBeep (string, number)**

Sends a command to an Insteon device that could cause it to "beep". Not all Insteon devices support this command.

**Bool = \_TileUpdate (string, number, any, number)**

Update the tile named by the first argument. The second argument is a code and that determines the use of the next two arguments. Returns Yes if the operation worked.

Code	Use	Arg3	Arg4
0	Change label	Label text	Not used
1	Change colors	Background color	Text color
2	Change image path	Image path	Not used
3	Change text	Text	Not used
4	Refresh	Not used	Not used



## Chapter 13 Scripts

In previous chapters, Home Modes, the Visual Programmer and the Visual Scheduler were covered in detail. These three tools allow for the creation of sophisticated automation solutions.

However if you are looking for more complexity in programming and decision making, HCA provides for the inclusion of text based scripts that are used as an adjunct to Visual Programs.

What scripting languages are supported? HCA supports any script engine that can use a type library. This means that common ones like VBScript and JavaScript are supported as well as the less common ones like WebBuilder.

What can a script do and how does it do it? HCA provides a rich object model with many methods available for each object. This is all documented in the HCA Object Model appendix to this user guide.

This chapter focuses on how you configure HCA to work with text based scripts, the Visual Programmer Script element, and some thoughts on how scripts are best used.

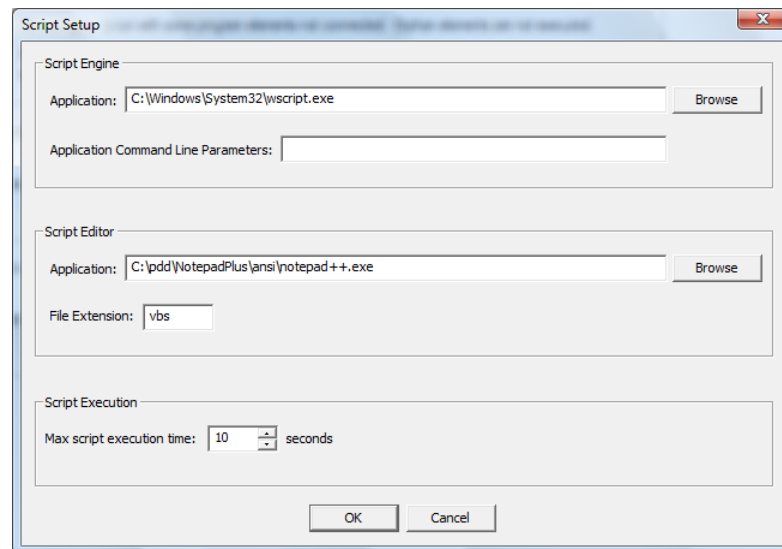
This is a complex topic! Writing scripts is a true programmer activity and, unlike Visual programs, doesn't have a friendly environment to work in. Included in the HCA installation are script examples you may want to look at prior to starting any of your own.

**NOTE: This feature is only available in HCA Plus and does not function in HCA Standard or HCA Limited.**

---

### Script setup

Before you can begin using scripts you first must configure HCA for what script engine and editor to use. To do press the *Scripting* button in the ribbon *Tools* category.



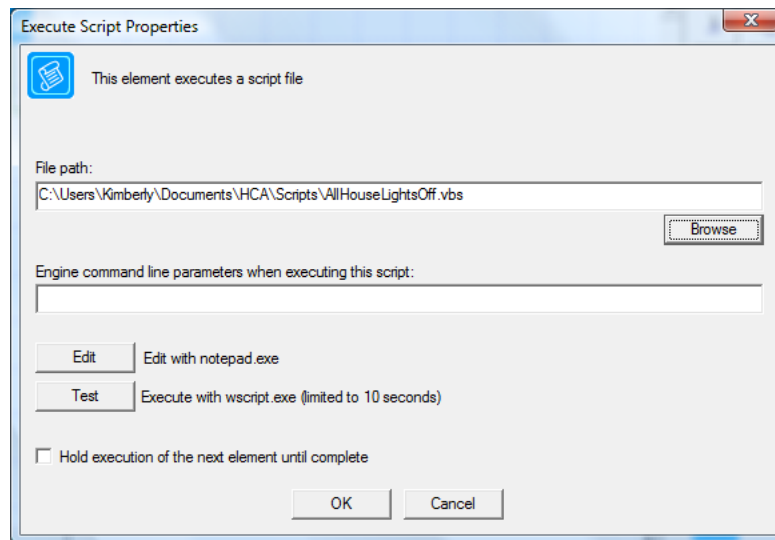
In this dialog are these parameters:

- Script Engine.  
This is the application that will process the script. In the above screen image VBScript is being used.
- Command line parameters to the script engine.  
This will vary between engines
- Script editor.  
In the Script Visual Programmer element is a button that you can use to start an editor to open your script. This may make it simpler to make quick changes as you modify your script to achieve the function desired. In the above screen image the text editor Notepad++ is being used.
- The File extension for your scripts.  
This makes locating your scripts in the Visual Programmer Script element simpler when you use the browse button.
- Script execution time.  
Each script is executed in its own windows process and HCA watches for it to terminate. Due to programming errors, your script might not complete its task in a timely manner – or it may never complete its task. HCA watches for it to complete within the allotted time and if it does not then HCA terminates the process.

---

## The script element

To use a script in a Visual Program, add the Script element.



In this dialog is specified the path to the script file. Use the browse button to locate it. Script files can reside anywhere on your disk but it is recommended that you keep your scripts in the HCA area of your documents in the sub-folder Scripts.

There are two buttons on this dialog:



- **Edit.**  
This starts whatever text editor you have previously configured in Script Setup. The name of the application is shown here for your reference.
- **Test.**  
This starts the script engine to execute your script. Whatever action you have developed your script to do will then happen. The script engine and the max execution time is what was previously configured in Script Setup. The name of the engine and max execution time is shown here for your reference.

Also in this dialog is a checkbox to say if the Visual Program execution continues with the next element immediately or waits until the script is complete.

---

## Using scripts

Why use a script when Visual Programs have so many different elements that you can use? There may be times where it simply is convenient to use a script rather than a visual program just because of the number of elements you would need or your need to be kept up to date as your design changes.

A good example of this is shown in a sample script included with the HCA installation. This is the script `AllHouseLightsOn.vbs`

```
Option Explicit

Dim HCA
Set HCA = GetObject(,"HCA.Object")

    Dim count
    Dim i
    Dim name

    count = HCA.device.count()

    For i = 0 To count - 1

        name = HCA.device.Name(i)

        if (HCA.device.SupportsDim(name)) then
            HCA.device.On (name)
        end if

    Next

Set HCA = Nothing
```

This simple script turns on all the lights in your home. You may have one light or one hundred it doesn't matter. If you did have a hundred lights it would be a very large Visual Program and you would have to update it each time you added or removed a light. With this script it is much simpler.

Another reason is to perform some kind of test that HCA can't do. For example, this script determines if today is the 3<sup>rd</sup> Monday of the month.

```
Option Explicit

' Sets the rcScript variable in HCA to one if
' today is the 3rd Monday of the month
' Otherwise sets it to zero

Dim HCA
Set HCA = GetObject(,"HCA.Object")
```

```

Dim date
date = Now()

Dim y, m, d
y=Year(date)
m=Month(date)
d=Day(date)

dim ith
ith=3

dim i
i = 1

do while (ith > 0)

    date=DateSerial(y,m,i)

    if (Weekday(date) = vbMonday) then
        ith=ith-1
        if (ith = 0) then exit do
    end if

    i = i + 1

loop

Dim rc
if (i = d) then
    rc = 1
else
    rc = 0
end if

HCA.Flag.Set "rcScript", rc
Set HCA = Nothing

```

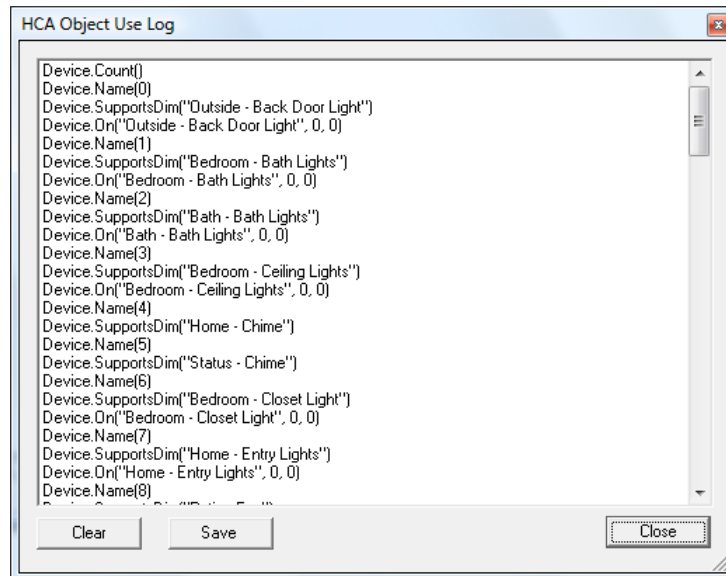
This is a test that HCA can't perform with the Visual Programmer Test element.

---

## Helpful Tool

Included in HCA is a helpful tool in working with scripts. The feature is normally hidden a bit as it wouldn't make sense to HCA users not working with scripts. To expose this tool, open the application and press the *HCA Options* button. Select the *Extra* tab and enter the code OBJUSE. Once you have done that, then open the Object Use Log with the *Remote Access Viewer* button in the ribbon *Control* category.

When open, the viewer shows all the calls to any HCA object and method as well as its parameters and if the method generates an error it shows that also.



It's not really a "log" as such it is more of a Viewer as things only "log" when the viewer is open.

The viewer can be cleared with the *Clear* button and the contents saved to a text file with the *Save* button.

Instead of using the Extra feature code OBJUSE you can use the code OBJERR. Instead of showing all the calls to the object method, only those methods that result in an error are shown.



## Chapter 14

# Displays

A display is a way to view devices, programs, and groups in arrangements that make sense to you. HCA uses the term *Display* rather than floor, room, or area because there are different types of displays that don't show areas in your home.

Displays have five main uses:

- To display icons for your devices, programs, and groups
- To display text messages
- To display power track graphs
- To display HTML
- To display tiles each of which contain one of the above 4 types.

In the first case, the display can have a background image and that image can be in the form of a drawing interchange format (DXF) file, or picture (BMP, GIF or JPEG) file..

Displays can also be organized as a series of tiles with each tile showing a different display type. For example, you could construct a display with a graph tile, an HTML tile, and two icon tiles.

Everything that appears in the Display Pane – right pane in the HCA main window - has a display object associated with it. At different times, you may see the Display Pane contain a text message, an arrangement of icons, or a HTML file. And for each of these there is a display object - one for the text display, another for the icon display, and a third for the HTML display.

Sections in this chapter include:

- Types of displays
- Creating a display
- Text displays
- HTML displays
- Icon Display
  - Icons for displays
- Power Track Displays
- Tiled Displays
- About DXF files
  - Using DXF files in HCA
  - DXF files and layers
- Picture files
- Placing icons on a display
  - With a picture or no background
  - With a DXF background
- Working with icons on a display

## Types of Displays

In HCA there are two types of displays: Displays you create and displays automatically created for the contents of a room or folder.

Each time you create a room or folder HCA automatically creates a display for it. What you can do with that display is the same as you can do with a display you create. You can add a background to it and position icons on it as best fits you needs.

So why create you own displays in addition to the folder displays that HCA creates? The answer is simple, to create a better grouping of icons from more than one room or folder. For example, in your design each room could contain all those device, programs, and groups for that room. You could then create a display that shows all the icons on a floor and another that shows all the icons in the whole home.

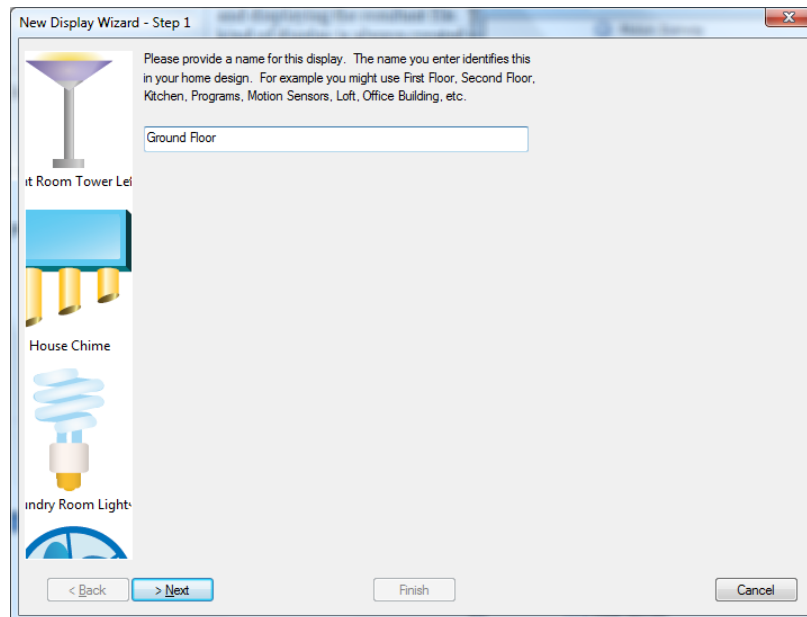
As described above, there are different types of displays in HCA and each has a different use. It is important to understand the utility and limitations of each type before you decide which kind you want.

Display type	Major features
Icon Display	Shows one or more icons for devices, programs, and groups. You can, but don't have to, use a background for the display. Backgrounds are DXF files or picture images. The display that HCA creates for a folder or room is always an Icon display.
Graph display	Shows a power track graph
Text display	Show text messages created by the Visual Programmer <i>Show-Message</i> element. No icons can be placed on this display.
HTML display	Shows a rendering of an HTML file. This file can be, but doesn't have to be, produced by HCA reading a template file, substituting data for placeholders, and displaying the resultant file
Tiled Display	Shows tiles – sub windows – each containing a different type of display. There are text tiles, icon tiles, html tiles, etc.

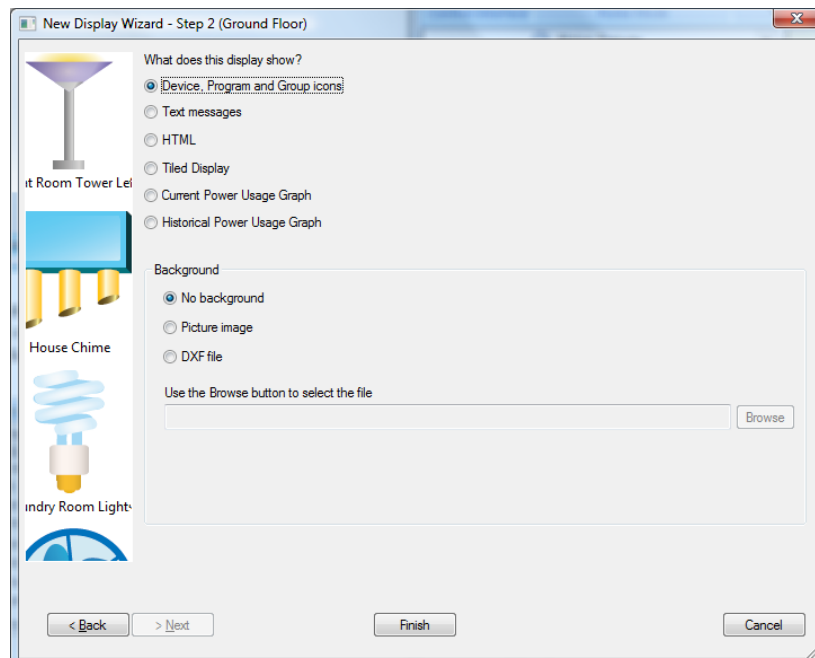
A design can contain as many displays as you need.

## Creating a Display

To create a display, the New Display Wizard is used. In the ribbon *Design* category, press the *New Display* button. This opens the New Display Wizard. Enter a name for the display. Display names must be unique within the design.



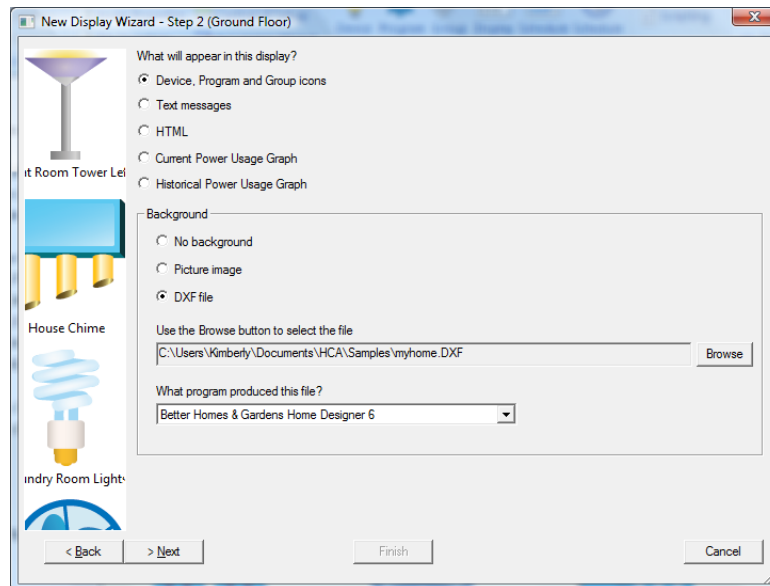
On the second step of the wizard, indicate the type of display. Each display type may have additional configuration. For example, when creating an icon display there are background options.



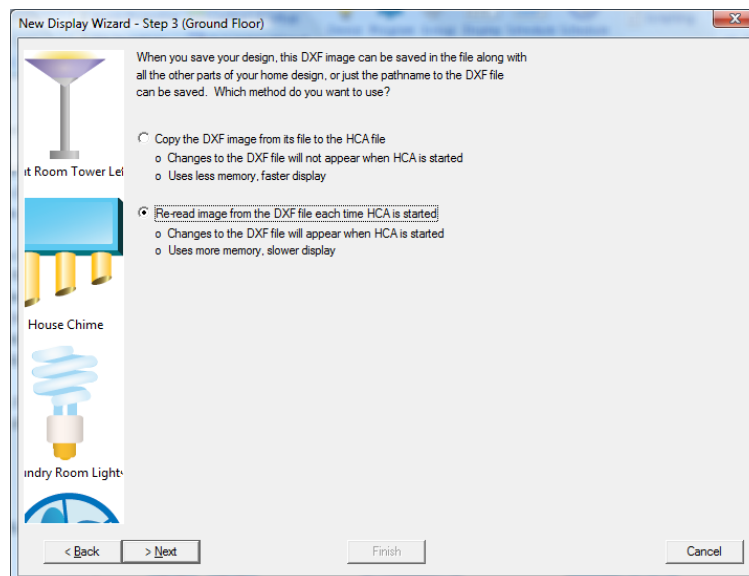
If you don't want a background for this display, simply select *No Background* and press *Finish*.

If you have a picture background, select *Picture File* and using the *browse* button locate the image file. Then press *Finish* as there are no other steps.

If you have a DXF file, browse to locate the file that you want. Select it and be sure to indicate which drawing program was used to create the file. If the program that created your DXF file is not listed, select “Other.”



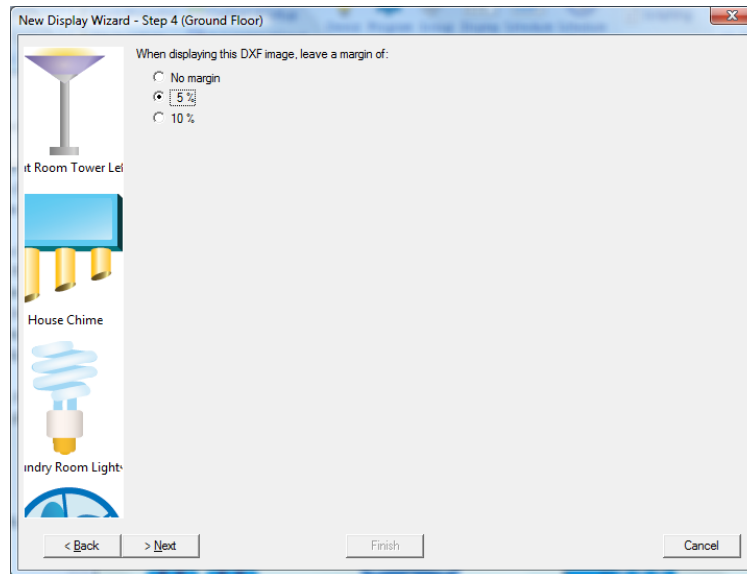
Additional steps of the New Display wizard depend upon the type of display and the options. For an Icon display with a DXF background, the next step indicates whether you want to import (copy) the floor plans from the DXF file to the HCA file, or link to the DXF file.



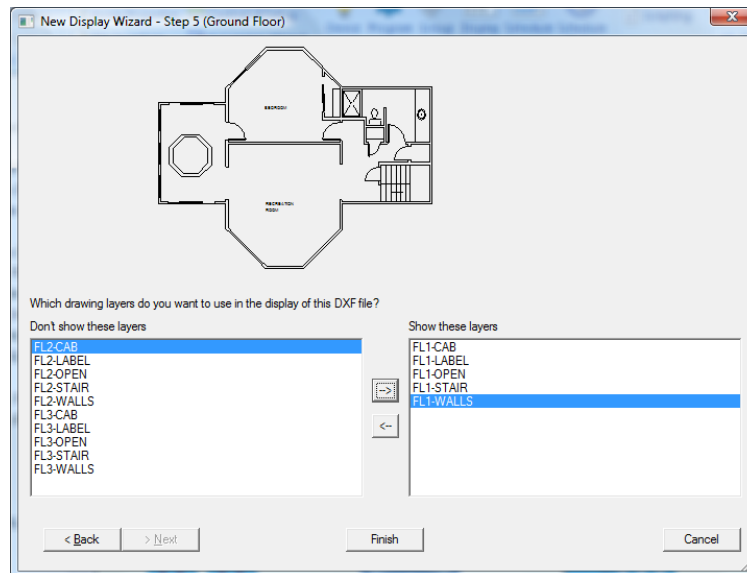
**Hint:** A good starting place is to initially link the files. When you are sure your drawings are complete and correct, you can import the floor plan (using the display properties dialog box).



When working with an icon display and you selected a DXF background, the next step is for DXF display sizing options.



Again, with an icon display with a DXF background, the final step is where the layers from the DXF file are selected.



**Note:** Not all drawing programs support layers but it should be possible to choose some sort of layer that makes the floor plan appear.

When you are done, click *Finish*. The wizard closes and your display appears in the display pane.

---

## Text Displays

Text displays are the simplest kind of display in HCA. The only property for a text display is the display name.

The text for a text display comes from many different places.

- Power Failure outage and restore messages (Restart chapter)
- Show Message elements (Programs chapter)

The messages need to be short because HCA shows them on only a few lines in a large font.

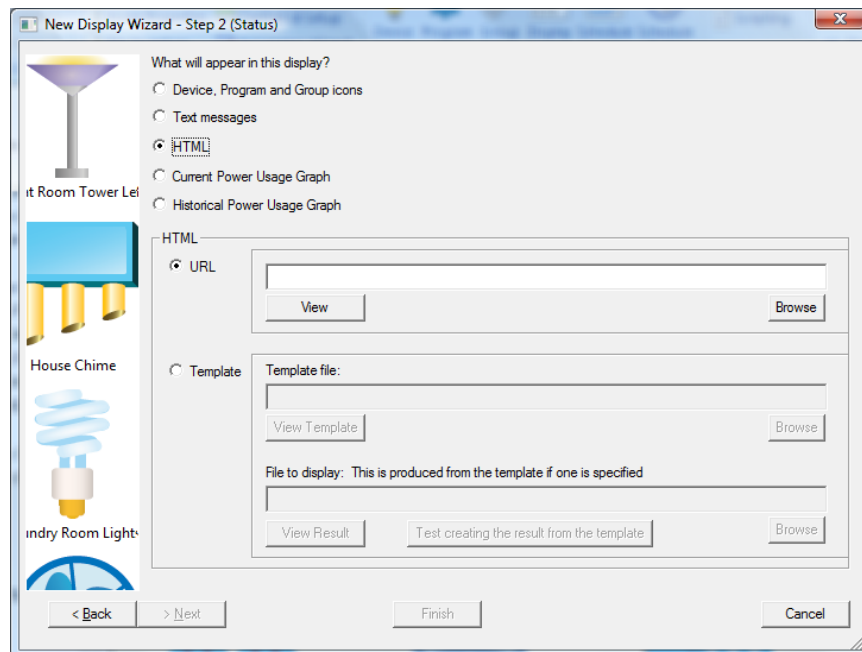
## HTML Displays

An HTML display uses the features of Windows to render a HTML file, with all its directives, into an image within the HCA display pane or within a tile on it.

This is done in one of two ways. The simplest way is to provide a path to the HTML file to display. For example, a page that displays your current weather.

The other way is more complex but can be more useful. Two paths are provided, the first to a template file and the second to a resultant file. Before this HTML display appears in the Display Pane, HCA takes the template file and copies it to the resultant file. During this copy, any placeholders in it are replaced by the result of evaluating them as HCA expressions. This is identical to the process used by the Status Export feature (see the chapter on status export).

For HTML displays, the second step of the Display Wizard appears as:



The upper *Browse* button locates the template file. There is no rule for the file type for template files but .HTX is often used. As a convenience the *View* button opens the template file using Notepad.

The lower section is similar and is where you enter the path to the file produced from the template file. The *View Result* button attempts to open the file using its file type. For example if you created an HTML file, then a browser is used to show the file.

To help you get the paths and the template file correct use the *Test* button to create the result file from the template file. Then you can use the *View Result* button to see what the result looks like.

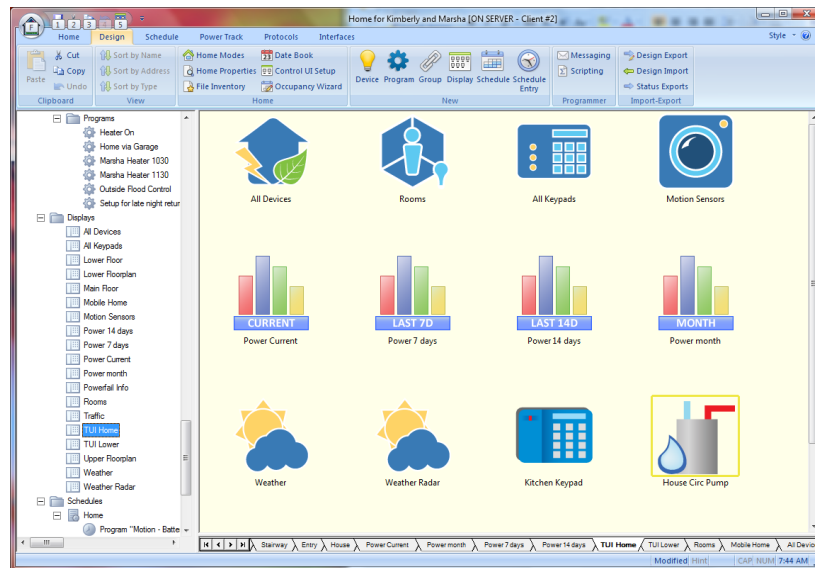
You will need to have a browser installed on your machine in order for HCA to use its HTML display capabilities. HCA doesn't do any HTML interpretation but passes the job entirely to other software. If the rendering is not as expected, the fault doesn't lie within HCA.

## Icon Displays

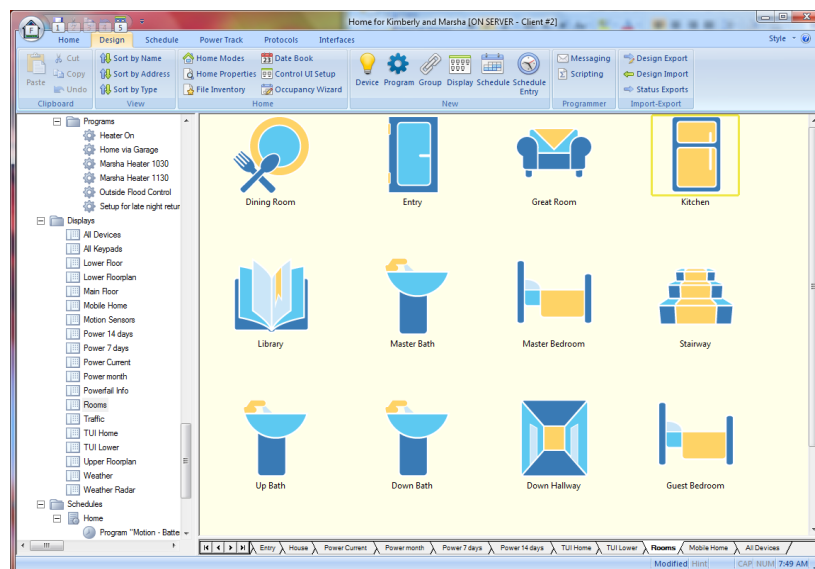
Icon displays show the icons for your devices, programs, and groups. These icons show their state by changing their image or by a colored box around the icon. If a box is used then the color shows the dim level with yellow being ON and darker shades for lower dim levels.

The major feature of icon displays is that you can arrange the icons into whatever order makes sense to you. Also you can have a background that makes the display look like your home, a floor, or room. This is done using a picture file – bitmap image – or an architectural drawing file – a DXF file. The next sections discuss DXF files and pictures files.

Displays can also contain icons for other displays. This feature lets you create your own Control User Interface pages. This display contains a mix of icons for devices, and other displays.

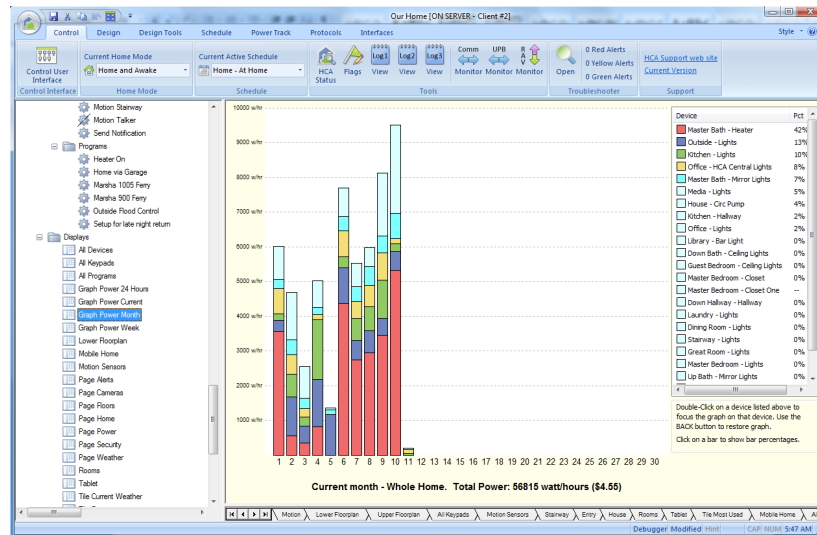


To open a display from its icon, double-click on it. A double-click on the *Rooms* icon in the above example changes the display pane to show:



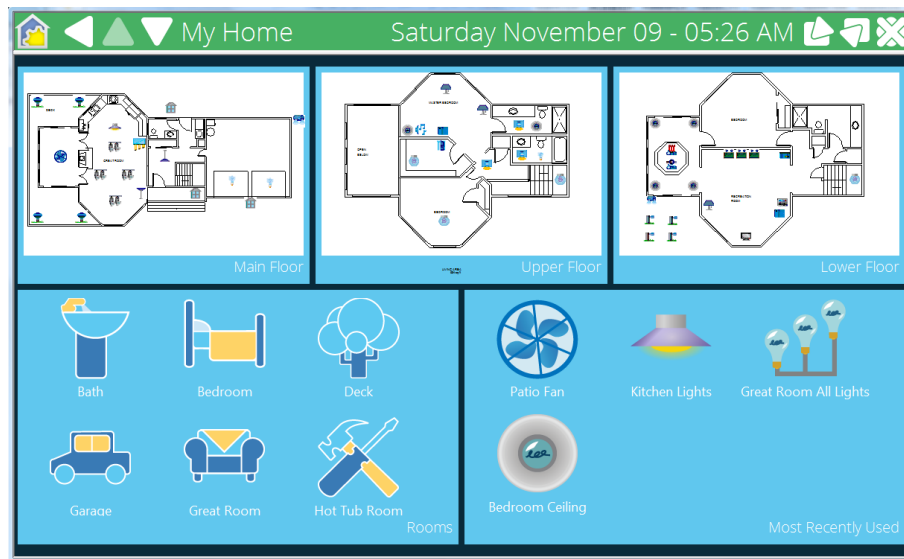
## Power Track Displays

In addition to the HTML, Text, and Icon displays described above, a display can also show a Power Track graph. This is described in the *Power Track* chapter.

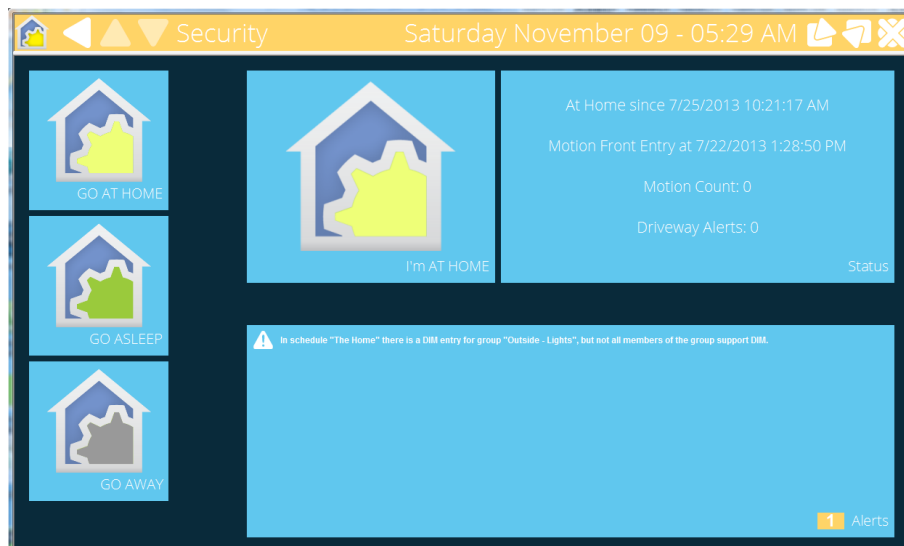


## Tiled Displays

Each of the display types described above is of a single type – the display only shows icons or only shows an HTML file. It is possible to create tiled displays to show more than one type of display each in a sub-window or *tile* of a display. For example this is a tiled display that has five tiles each showing icons. Three tiles have DXF backgrounds.



In this display, there are six tiles: Four are image tiles, one is an alert tile, and one is a text tile.



While the screen images above showing tiled displays were made when using the *Control User Interface*, tiled displays also appear in a similar manner in the *Development User Interface*.

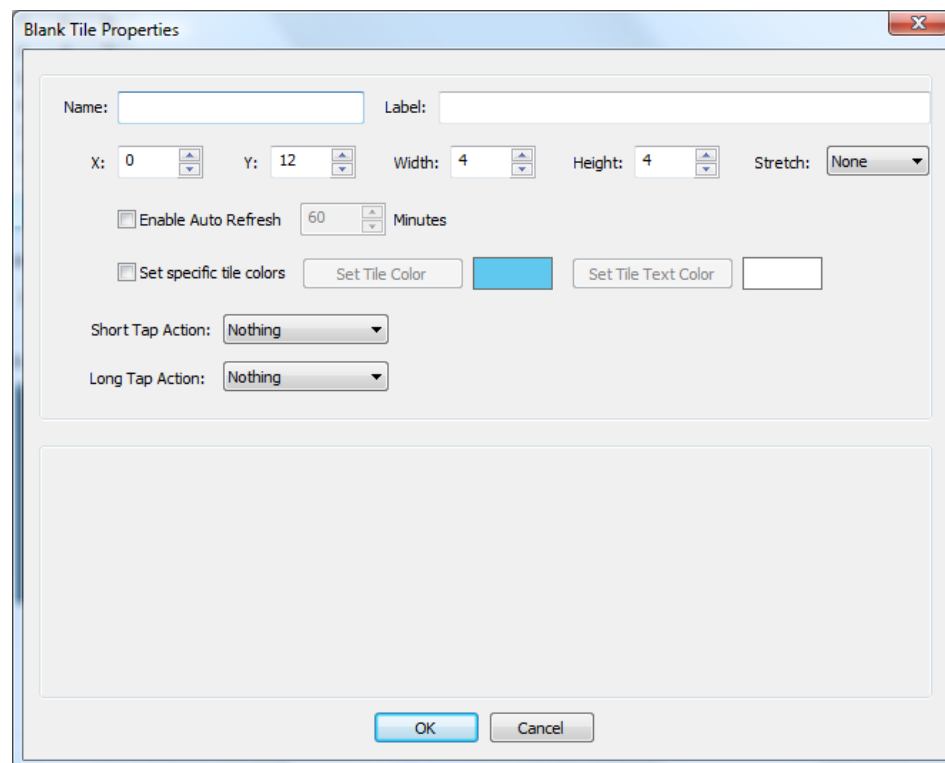
## Tile Types

In a tiled display there are nine different tile types you can use. These are:

- Alerts
- Blank
- Graph
- HTML
- Icons
- Image
- Power Meter
- Text
- HCA status

Some of these tiles types get their properties from another display and other tile types are configured separately.

All tiles, regardless of type, have a common set of properties. The common properties for a tile are configured in the tile properties dialog.



The tile properties as shown in the above image are:

### Name

Each tile has a name which is used in programs to update the tile. This is discussed later in this chapter. Tile names should be unique.

### Label

A tile has a label which appears in the lower right of the tile. This label can be blank.

Position

A tile has a position – in units described later – that determine its location and size.

Refresh

A tile can be periodically refreshed after a selected time. This may be useful, for example, with an image tile where the image is generated by something external to HCA like a camera.

Color

The color of the tile background and text – if there is any text – can be specified on a tile-by-tile basis or the tile can use the color scheme set for the display.

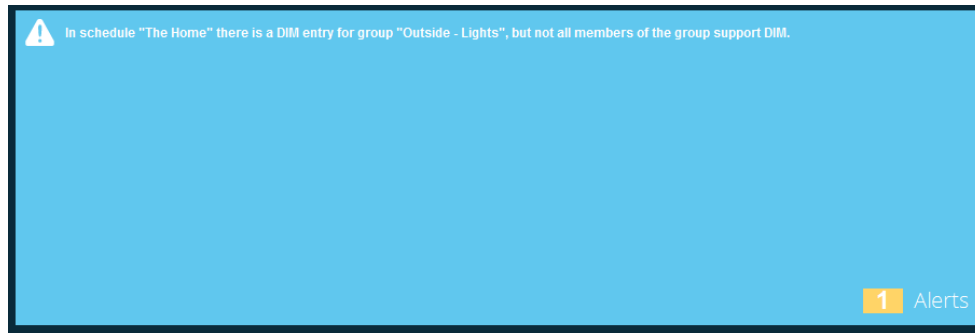
Actions

Like icons, a short tap – click – or a long tap – double click – on a tile can initiate an action. There are three possible actions: Do nothing, make a designated display the current display, or start a program.

---

## Alerts Tile

The *Alerts* tile shows the messages that appear in the Troubleshooter *Alerts viewer*. Depending upon the size of the tile only the most severe and most recent alerts are shown. Regardless of the number of alerts shown in the tile, the total number of alerts you would see if you opened the troubleshooter is shown in a highlighted box at the lower right. The color of this box is the current alert level color – green, yellow, or red.



Besides the common tile properties, there are no additional properties for the Alert tile.

---

## Blank Tile

The *Blank* tile displays nothing but an empty box in the color set in the tiled display properties. It can be used for filler in a display. There are no additional properties for a blank tile.

---

## Status Tile

The Status tile shows the HCA status similar to what the HCA Status dialog shows. There are no configuration settings for it.

Time: Friday, March 4, 2016 2:27 PM  
 Date: 64th day of 2016 - Vernal Equinox in 15 days  
 Sun Time: 6:09 AM - 5:46 PM  
 Up Time: 0 days 0 hours 1 minute

---

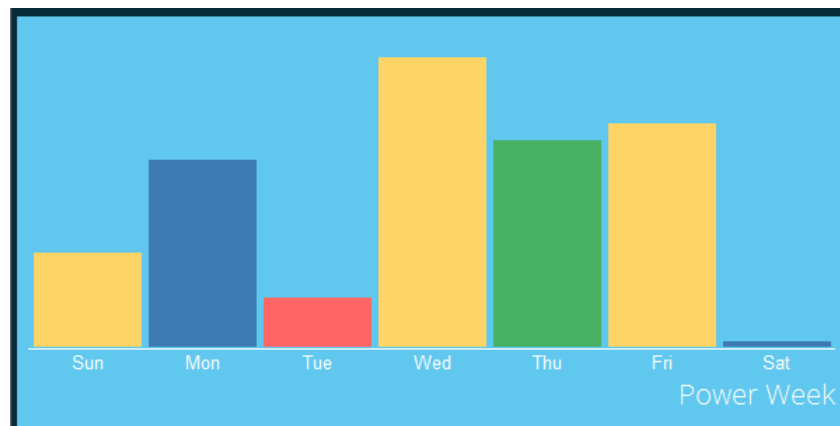
Current Schedule: The Home

5:01 PM	On Rec Room - Picture Gallery Lights
5:21 PM	On Outside - Lights
5:34 PM	On Great Room - Floor Lamp
5:40 PM	On Deck - Lights
5:44 PM	On 65% Rec Room - Lamp
5:46 PM	On Rec Room - Fish tank
6:31 PM	Off Patio - Fan
8:00 PM	Off Hot Tub Room - Hot tub maintenance
10:31 PM	Off Rec Room - Picture Gallery Lights
10:34 PM	Off Rec Room - Lamp
10:56 PM	Off Great Room - Floor Lamp
10:57 PM	Off Rec Room - Fish tank

HCA Status

## Graph Tile

A *Graph* tile displays a power track graph. Rather than specify all the graph settings in the tile properties, this tile references the graph display.



Since a graph tile is typically smaller than the graph display it is based upon, the graph tile omits details seen in a graph display – the scale and the graph metrics.

The only property in addition to the common tile properties for a graph tile is the choice of the graph display.

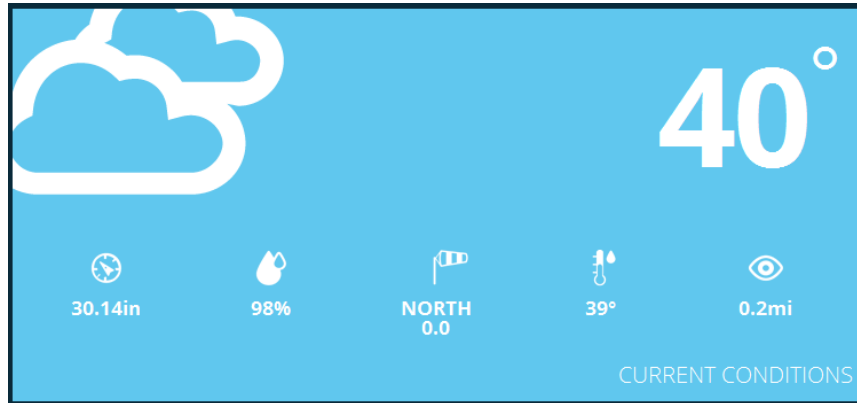
Graph Display:



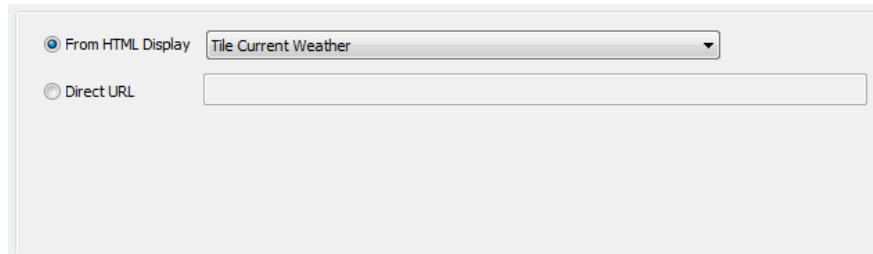
---

## HTML Tile

A *HTML* tile is similar to a HTML display in that it uses a browser component to display HTML that comes either from a disk file or from some web server on the network.



The properties for a HTML tile, in addition to the common tile properties are:

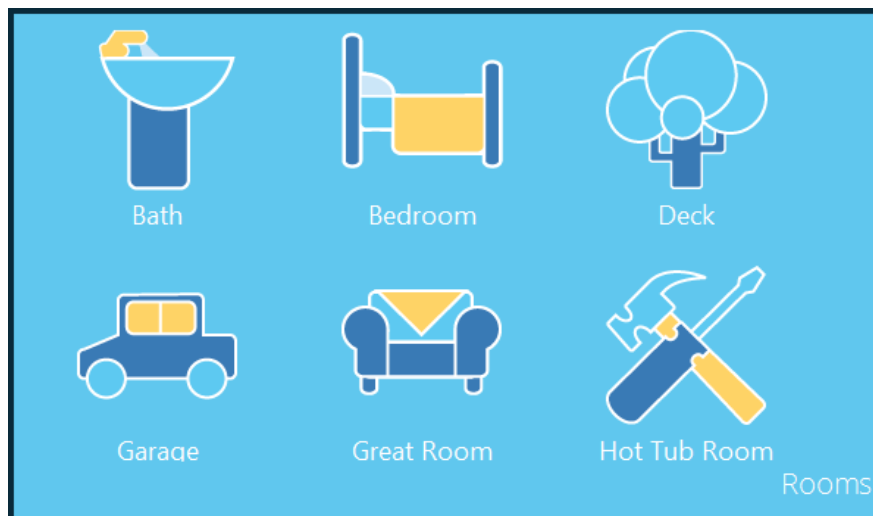


The source of the HTML can come either from a direct URL – a network reference or a file reference – or from a HTML display. When using a HTML display in a tile, the HTML file can be produced by reading a template file and producing a resultant file. This is specified in the HTML display.

---

## Icon Tile

The *Icon* tile is very similar to an icon display.

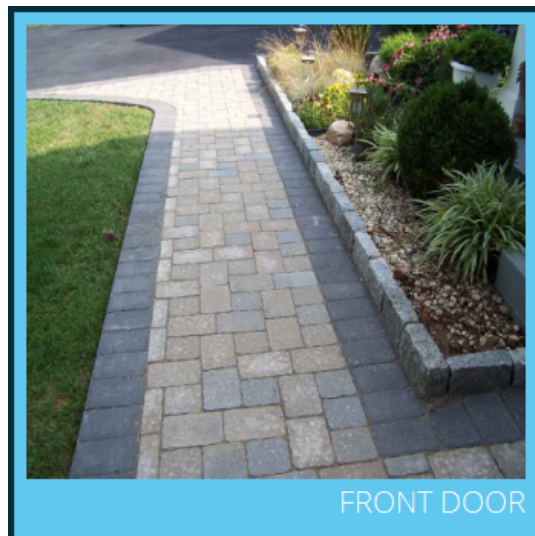


An icon tile can either display the icons from an icon display or automatically create a set of the most used icons in your design. This choice is made in the tile properties.

In addition to the source of the icons the icon spacing can also be modified. Again, because tiles generally have a smaller size than a display, you may want to “pack” the icons a bit tighter than in the display.

## Image Tile

An *Image* tile displays a picture from a selected image file.



The picture can be resized to fit in the tile or displayed in its original size – and possibly cropped to the size of the tile.

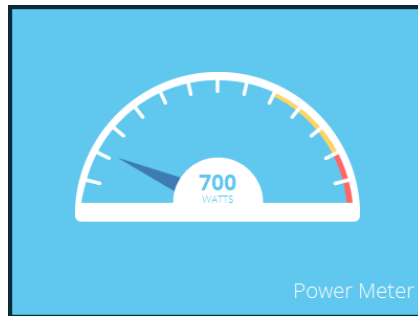
The properties for an image tile are:

**Note:** If you are using client-server, the client requests the file from the server when the tile is displayed or refreshed. The path given must be one that can be located by the server. See the section on HTML displays above that discusses relative and absolute paths.

---

## Power Meter Tile

The *Power Meter* tile shows an analog power meter that displays the total power usage of all devices in HCA based upon their current state.



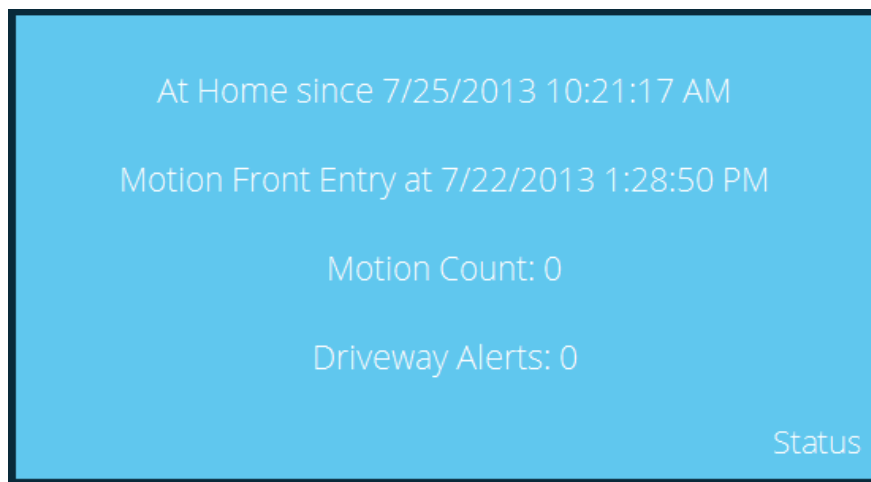
The Power Meter tile properties are:

You can scale the meter so that its maximum value – when the needle is all the way to the right – is based upon the total wattage of all your devices or on a value you select.

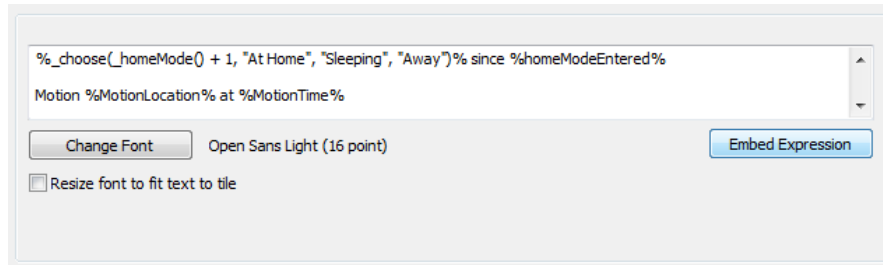
---

## Text Tile

A *Text* tile displays text information.



The text shown is determined from the tile properties.



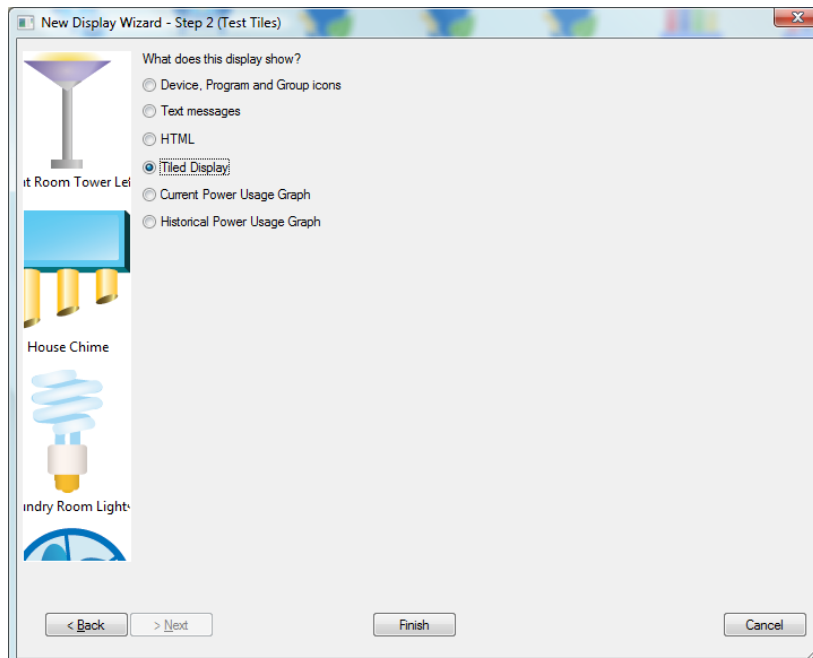
In a manner similar to several Visual Program elements – Add To Log , Show Message, etc – and in the Status Export feature, you can embed HCA expressions within the text by placing them inside of%%. These expressions are evaluated and the final text determined before it displays.

In addition to the text you can also select the font and point size used for the text.

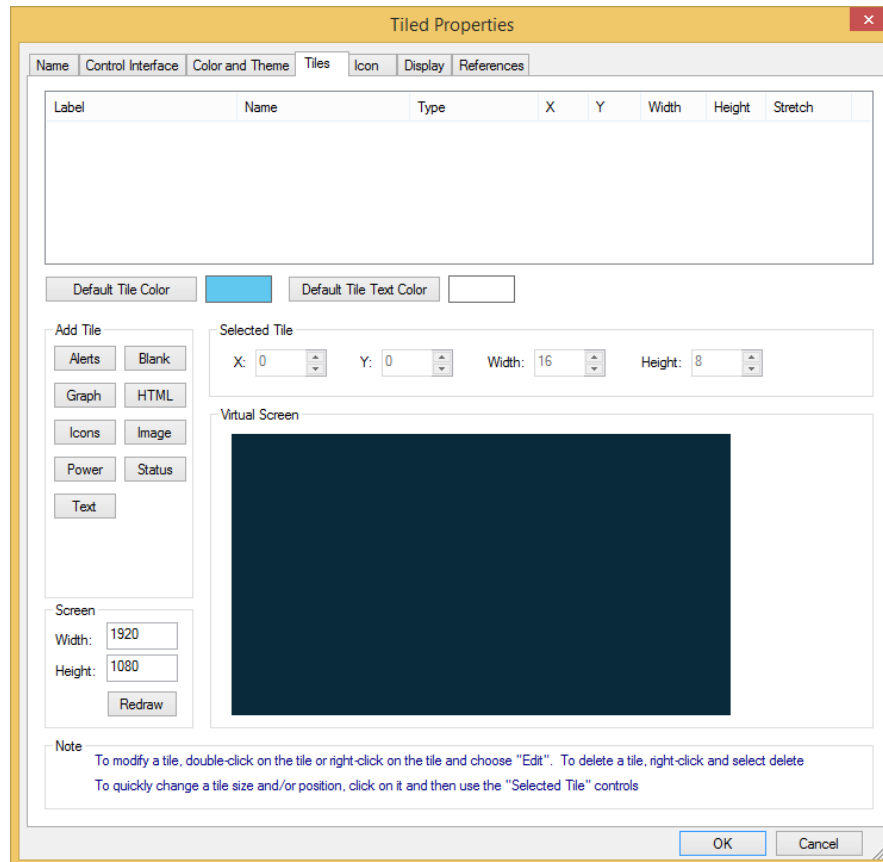
---

## Creating a Tiled display

A Tiles Display is created similar to other displays. At the second step of the New Display Wizard choose the Tiled Display option.



At this point the tiles and their layout are not yet specified. When the wizard completes, the display is added to the design. Open this new display's properties as usual and choose the *Tiles* tab.



This tab can seem overwhelming but don't panic! It contains tools for adding tiles, an image of what the display looks like with all the tiles, facilities for adding and positioning tiles, and a way to set the default tile colors.

---

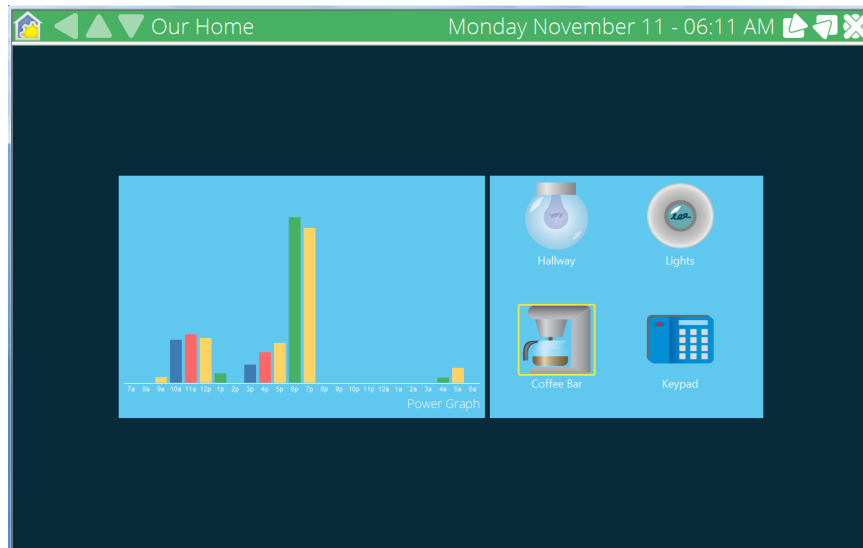
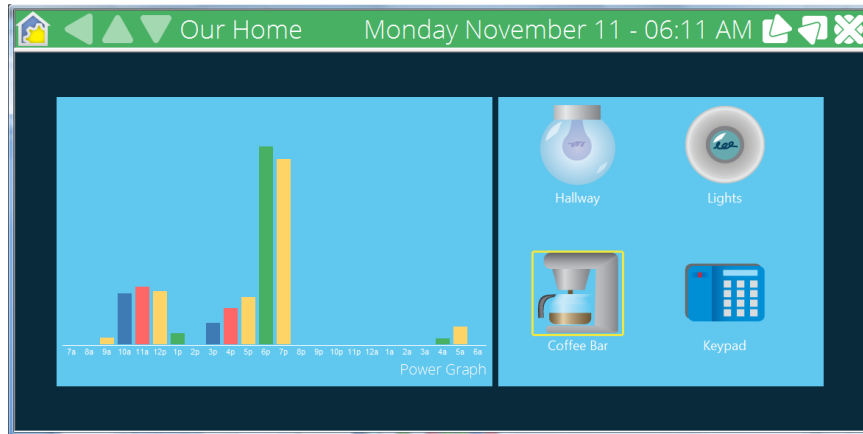
## Tile coordinates

The positions and sizes seen in the tile editor are in units determined by the spacing used for the large icon theme. As described in the user guide *Icon Themes* chapter, a theme is composed of a set of icons all of the same size spaced on a display according to the theme specification.

One tile unit – used for tile size and position - is  $\frac{1}{4}$  of the large icon theme cell spacing. For example, the default spacing for the large theme is 200 pixels by 200 pixels. In this case a tile sized as 2 by 4 is sized as 400 by 800 pixels.

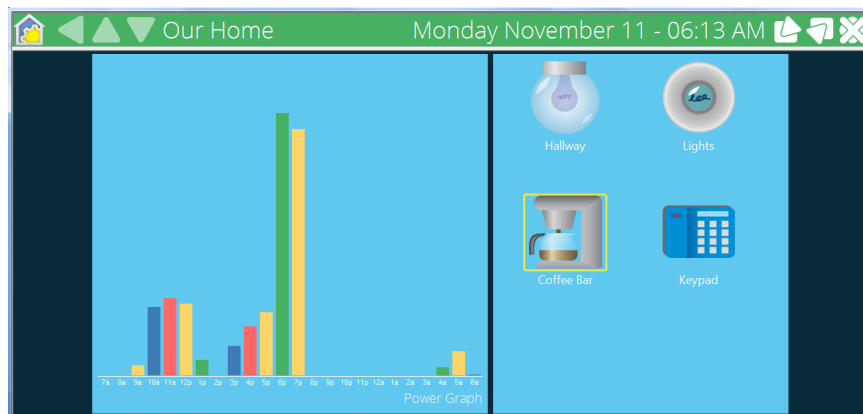
Why not use pixels? When using HCA in client-server mode a tiled display may appear on several different screen resolutions. By basing the tile size and position on the large theme you can modify the large theme settings – local to each client – to adjust how a tiled display appears.

When a Tiled Display is shown in the HCA Window right pane the tiles center themselves in the display. For example, in the two screen images below, the size and position of the tiles in the display were not changed – only the HCA application window was made larger.



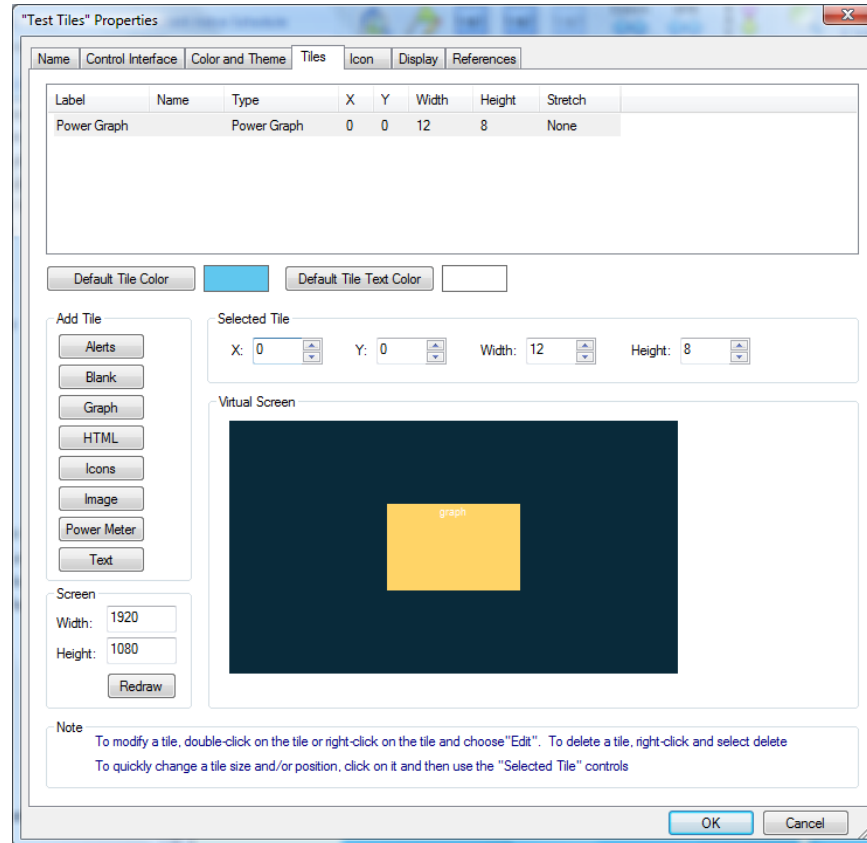
Tiles can also stretch to fill the width or height of the display. Of course this only works well when you stretch a tile horizontally if there are no tiles to the right of it and stretch vertically when there are no tiles below it. A tile can also stretch in both directions which allows a single tile to fill the whole display.

The two tiles in the above picture were modified to stretched vertically and now display as:



## Adding a tile

A tile is added by pressing one of the tile type buttons on the left side of the dialog. This opens the properties dialog for that type tile. A initial size and position is chosen so that the tile doesn't overlap an existing tile if possible. It is best not to be too concerned about the position or size at this point. Modify any tile specific properties and close the dialog with OK.



The tile is added to the list at the top of the dialog and a small representation of it displays in the virtual screen. Small text in the tile image displays the tile type. This virtual screen lets you see quickly what the display will look like and how the tiles are positioned and their sizes in relation to each other.

As described above, tiles center in the screen so even though the tile position is (0, 0), it is positioned in the center of the screen as it did in this example.

You can set the size of the virtual screen using the width and height in the screen box. The initial values are the screen resolution of the monitor attached to the computer you are using. You may want to adjust the screen size if you are designing a tiled display that will be shown on a client machine with a different screen resolution.

## Modifying a tile

The properties of a tile can be modified by either right-clicking on the tile in the virtual screen and selecting *Edit* from the popup menu or double-clicking on the tile. If because of the tile position or due to an overlapping tile you can't see the tile you can double-click on the row for the tile in the list at the top of the dialog.

---

## Repositioning a tile

There are two ways to set the size and position of a tile after it was created. You can open its properties and change the position and size or you can click on the tile in the virtual screen to select it – the selected tile shows in yellow – and then use the position and size controls – the ones labeled *X*, *Y*, *Width*, and *Height* above the virtual screen.

It's best to use the up and down arrows associated with the four edit boxes. When the size or position is changed in this way, the virtual screen updates immediately to show the change. You can quickly resize and position tiles in this way.

---

## Deleting a tile

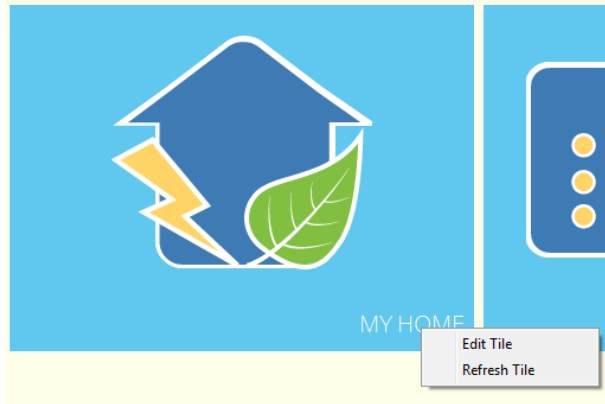
A tile can be deleted by right-clicking on the tile in the virtual screen and selecting *Delete* from the popup menu. The tile is then removed from the display.

---

## Working with tiles in the Development UI

All the screen images in this chapter come from the *Control User Interface*. But when in the *Development UI* there is a shortcut to modifying a tile.

Right click on the tile and select *Edit* from the popup menu.



**Note:** When working with a HTML tile you must click on the tile label – or at least on the bottom part of the tile where the label is normally seen if the tile has no label.

---

## Updating Tiles

Tiles can be updated by using the auto refresh mechanism configured in the tile's properties. It also can be updated by the *Update Tile* Visual Program element. This is described in the chapter on *Programs*.



---

## DXF files as backgrounds

If you have one, you can have HCA use a DXF file as the background for a display. There are a couple of things you might want to know about DXF files before you start.

---

### Linking or importing

When you get to step 2 of the New Display wizard, HCA gives you a choice of the kind of display. If you choose an Icon display you get the additional choice of the background. If you chose a DXF background, then you must select the DXF file and the program that produced it.

At step 3 of the New Display wizard you must make an important choice. You can either copy (import) the floor plan from the DXF file into your design file, or you can link the DXF file to your design file so that HCA reads in each time you open your design file.

Each option has its benefits and drawbacks, and you'll probably want to decide in advance which to use:

#### **Characteristics of linking:**

- Changes to the DXF file show up each time you load your design file
- This uses more memory, and has a slower display.

#### **Characteristics of importing:**

- Once you save your design file, all your information is now in one file.
- You can save the HCA file, and need move only one file to another location or PC.
- The saved file uses less memory, and provides a faster display.
- If you change your DXF file, those changes don't automatically appear when you load your design file - you will have to re-import the file.

**Note:** If you import your file, you will have fewer options when you subsequently open the display's properties dialog.

**Hint:** If you're happy with the drawing you have, and don't plan to change it, then import the DXF file to HCA.

Regardless of whether you use a linked or imported file, the icons for any devices, programs, etc, on the display are available the next time you open your file. You may also use a linked file for a while, and then decide to import it. All the changes to the display and the icon placements remain in place.

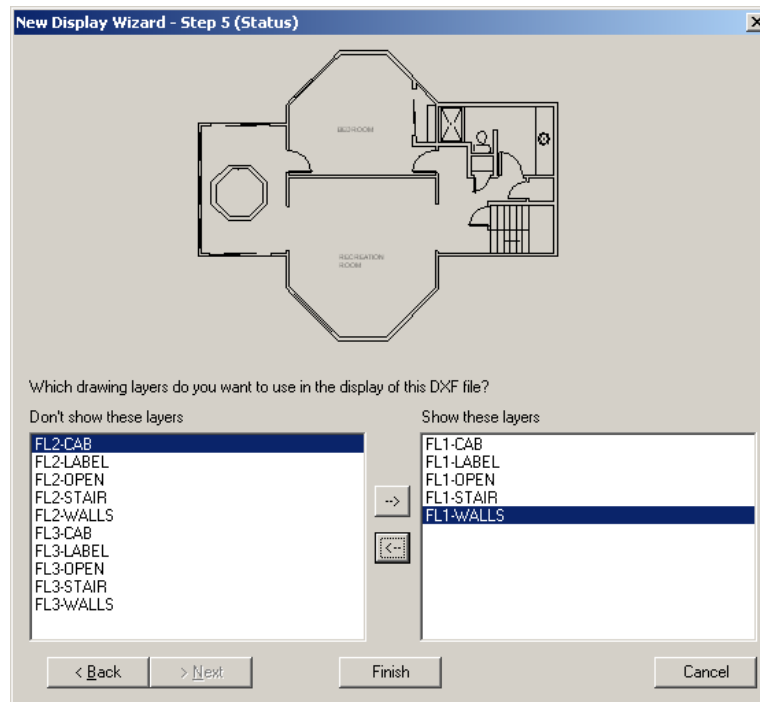
---

### DXF files and layers

Standard DXF files are provided in layers, but not all programs layer the same and not all drawing programs work identically. Some drawing programs don't even support layers.

In step 5 of the New Display wizard, HCA asks you which layers you want to include in the floor plan image. There are several layers to choose from. Of course, you cannot choose to display layers that your drawing program doesn't provide. And you don't have to choose to display all the layers that are available.

One way that HCA helps you is in step 5 of the New Display Wizard. As you select layers for display the preview in the dialog updates to show you how the DXF file appears with the layers you selected. You can add and remove layers and see immediately the effect.



If the software program that you are using does not support layers in the same way as in this example there will probably be some – maybe only a single – layer choice.

---

## Picture files as backgrounds

HCA supports several formats for picture files that can be used for display backgrounds. The supported formats are:

- Windows Bitmap files. These usually have a BMP file type
- Graphics Interface Format files. These usually have a GIF file type
- JPEG files. These usually have a JPG, JIF, or JPEG file type

HCA always reads these files from this disk each time HCA displays the display. Unlike DXF files, there is no method where these can be imported into your design file.

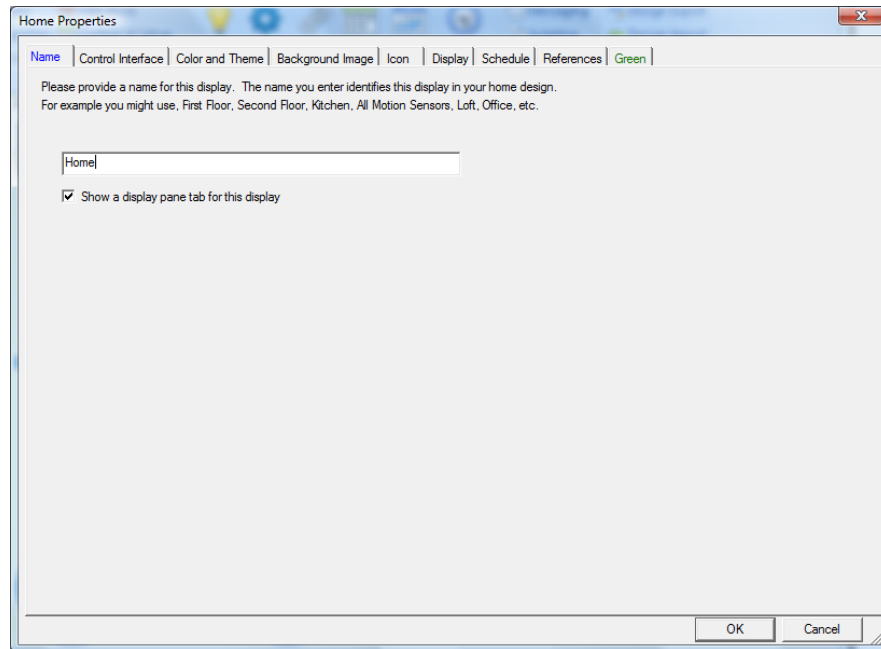
Since HCA reads them from the file for display each time, if you have some external Windows program produce them periodically, your display in HCA will change over time as long as the files are created with the same name.

---

## Modifying a Display

Like other objects in HCA, you can always modify their properties after they are created. For a display select the name of the display, or the folder name if you are modifying the display created for a folder, in the display pane then right-click and select properties.

Another method is to right-click in the display pane when not on an icon and select Properties from the popup menu.

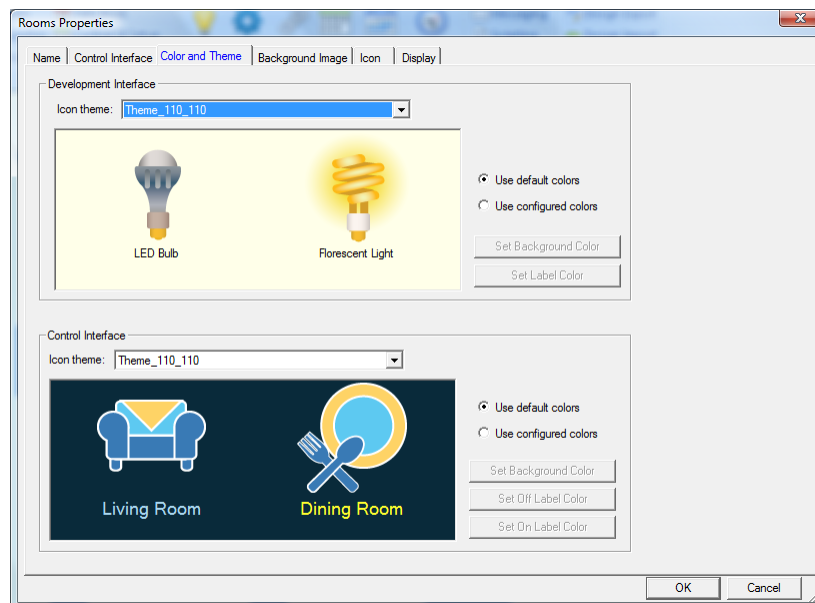


Using the properties dialog for a display you change the display name, add or remove a background, or choose options like having a tab for this display in the tab bar.

The number of tabs in the display properties may be different than shown above. The display for a *room* contains tabs for schedule, references, and Green - described in the *Rooms* chapter – but the properties for a *Text* display does not contain many of these.

An *Icon Display* shows the icons using a specific theme. There are different themes to choose from that use different icon sizes. Icon themes are more fully described in the *Icon Themes* chapter.

One tab that is common to icon displays for rooms and folders is the *Color and Theme* tab.



On this tab you select the themes and colors used for this display. Colors and themes can be specific to a display or the display can use the default color and theme as set in *HCA Options*.

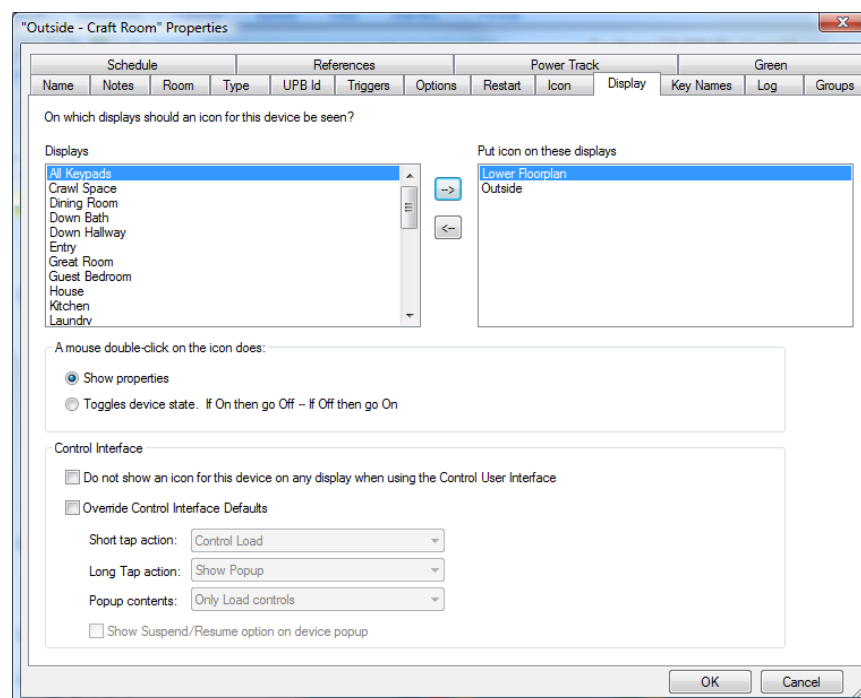
## Placing icons on the display

If you created an Icon Display, you're ready to start placing icons on your display for various devices, programs, and groups. In order to do this, you need to create them first. Refer to the device, program, and group chapters.

**Hint:** Not every device, program, or group has to have an icon on the display—decide if you want it there.

Once you have created them, just grab the name in the design pane, and drag and drop it on a display. Remember that only Icon Displays can show icons - text and HTML displays can't.

You can also open the properties of a device, program, group, room, folder, or display, and on the *Display* tab add or remove an icon to one or more displays.



## Tab Bar

Across the bottom of the display pane is a tab bar that facilitates quick access to displays.

To switch the display pane to a different display using the tab bar just click on the tab for the display. If there are more tabs that can fit, use the arrow buttons at the left end of the tab bar.

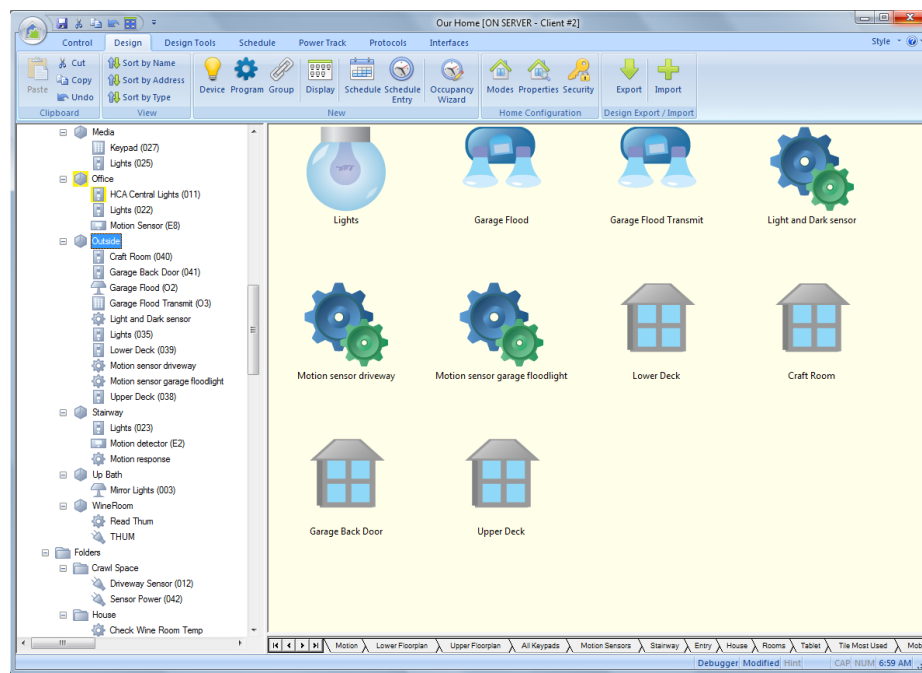
You can reorder the tabs by clicking on a tab and dragging it to a new location. The order of the tabs is saved in the design file.

Not all displays must have a tab. There is an option on the display's properties to have a tab for the display or not.

## Displays with Icons

Once an *Icon Display* is created you can show it in the display pane by clicking on the display name in the design pane or clicking on the tab for it in the tab bar.

Without a DXF background, the Display pane looks like the Windows Explorer using the Large Icon view. Each icon is shown with a label below it—the name of the device, program, or group. The icons are arranged in a grid pattern, but, as with the Windows Explorer, you can arrange the icons in various ways. On the ribbon Design category there are a number of buttons to sort the icons in various ways.



The buttons in the ribbon for this are:

**Sort by Name**—places the icons in alphabetic order.

**Sort by Address**—orders the icons by primary address.

**Sort by Type**—places the icons with similar devices next to each other.

In addition to the ribbon, if you right-click on the display background when not on an icon, then the popup menu contains additional options.

**New**—Submenu with choices for Device, Program, and Group. This starts the appropriate wizard setup to place an icon for the new object on this display.

**Line Up Icons**—use this to tidy things up after you've been dragging icons around; it lines them up to the nearest grid position.

**Auto Arrange**—this is a toggle. When checked, it is on, and when you uncheck it, you turn it off. This option is helpful when you are re-sizing the HCA window or display pane. When Auto-Arrange is enabled it eliminates, if possible, the need for any scroll bars. When Auto -Arrange is disabled, and you resize the HCA window, scroll bars may be necessary and are displayed.

**Hint:** If you disable Auto Arrange, and create your own ordering and layout by dragging icons around, HCA preserves this arrangement when you save and reload your design. If you are familiar with the Windows Explorer, HCA uses the same methods for dragging and dropping icons to create your own order. When Auto Arrange mode is off, you can create any layout you want. The icons don't have to be in a simple rectangular grid.

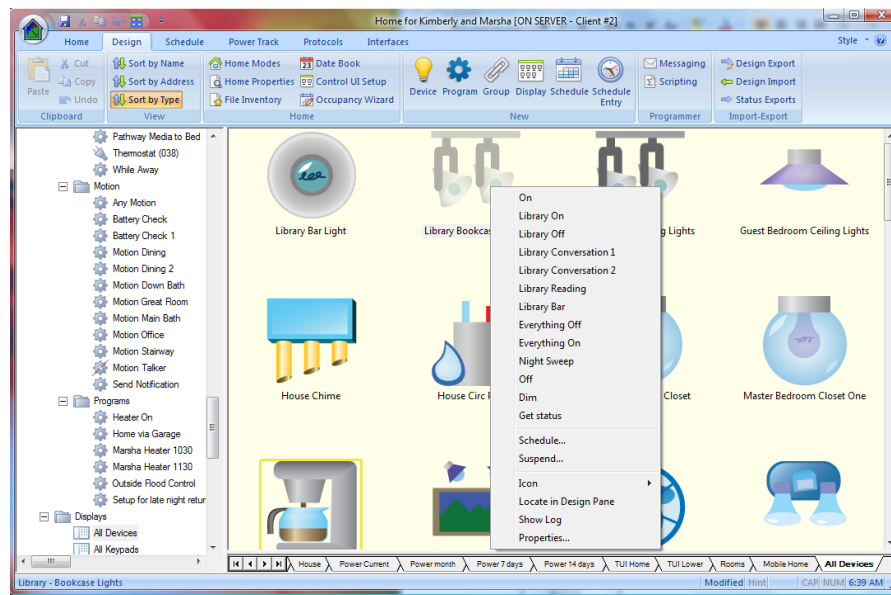
**Lock Icons**—if the display is locked then you can't move or rearrange your icons. This can be useful once you have everything the way you want and you want to prevent unintended movement of the icons.

**Note:** With a DXF background, each icon is represented as a picture scaled appropriately to the DXF image. Each icon is not labeled but can be dragged around to place it in a position you decide is appropriate. Because you are expected to place icons as appropriate when using a DXF background, the icon positioning menu choices described above are not available.

**Properties**—displays the properties dialog box for the display.

If you are working with a picture background there are additional menu choices to display labels below the icons or not to display them.

You can add as many icons as necessary to a display, and arrange or rearrange them as necessary. If you right click on an icon, a popup menu appears containing options appropriate to that type of icon. For example,



In this case the popup menu is for a device. The *On*, *Off*, *Dim*, *Delete*, and *Properties* commands are obvious. The *Suspend* command is described in the chapter on schedules.

If the device supports stored scenes, any scenes you have created for the device are listed in the popup menu between the *On* and *Off* commands. In this example there are quite a few of these.

The *Schedule* command starts the Schedule Entry Wizard for this device.

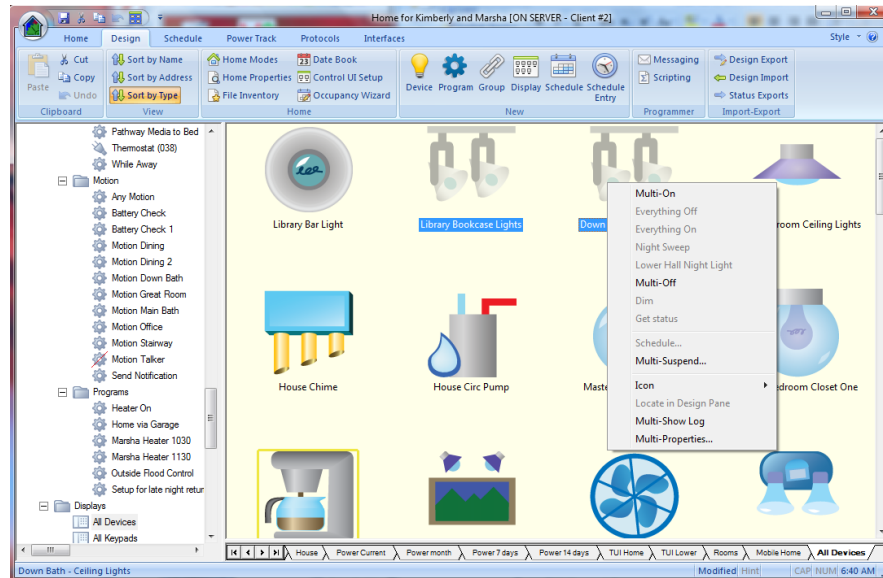
The *Icon* sub-menu is used to create, remove, and change the icons for this device. This is very useful if you want to place icons for lights on a display and in your home there are multiple lights that are controlled from one switch. In this case you create one device to control them, but you may want to see multiple icons on your display to match reality.

The *Locate in Design Pane* menu choice causes HCA to select in the design pane the device, program, or group for the icon. This can be useful when you have a large design and you don't remember what folder it is in.

*Show Log* is a quick way to check the log for any entries about the device, program, or group you selected Show Log on. The Log is described in the *Troubleshooter* chapter.

## Multi-Select Operations

If you have more than one icon selected in the display pane, the choice in the popup menu changes.



The only options available for a multi-selection are *On*, *Off*, *Suspend*, *Show Log*, and *Properties*.

*Multi-On* and *Multi-Off* control each of the selected items On or Off. *Multi-Suspend* suspends or resumes each selected item.

*Multi-Show Log* is like Show Log in that the log viewer is opened and it is automatically filtered to show only log entries for any of the selected items.

*Multi-Properties* opens the property dialog for the icon you right-clicked on. The edit of this object creates a change log and applies that log to all the other members of the selection where it makes sense.

For example, suppose you are editing an Insteon device and change some options, those changes are not replicated into a UPB device that was part of the selection. But a change in the logging options on the Insteon device would be replicated to the UPB device.

It is important to remember that only those properties you change matter. *Multi-select Properties* does not make all the items exactly the same!

**Note:** Multi-select is not supported on DXF displays.

## HTML Folder Sync

These are a few additional points to consider when using HTML displays in your design.

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 file, result file, 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 be in c:\users\kimberly\HCA\Tile Sun Moon

The HTML display is configured as:

Template and result file

Template file:

Result file produced from the template:

*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.*

If you set things up as these guidelines specify then the client “HTML Folder Sync” operation moves to the client computer all the auxiliary files – style sheets, images file, etc. – that your HTML display uses.



## Chapter 15

# Insteon Visual Scene Editor (VSE)

The Visual Scene Editor (VSE) is a way to program Insteon devices so that they can initiate or respond to a *scene*. A *Scene* is a collection of devices that respond together when the scene is activated or deactivated. Devices in a scene control themselves to a specific level at a specified rate according to how that scene has been programmed. The advantage of a scene is that scenes are activated and deactivated with one command regardless of how many devices participate in the scene. This can make a lot of devices respond simultaneously after only a single command is sent.

Inside each device is an area of memory called the *linking table*. This table contains one or more entries for each scene and how that device participates in the scene. It's the VSE's job to update the linking tables in your devices as scenes are added, deleted, and modified.

The VSE is a simple, yet powerful way to create and visualize a scene. At a glance you can see which devices are transmitters and which devices are receivers. Using a simple drag and drop mechanism you can add or remove devices from the scene.

It is important to note that you can use HCA to program scenes that HCA will never activate. For example, suppose you want a KeypadLinc button to control a lamp module. You can set this scene up as described in the KeypadLinc and LampLinc documentation – with all those “press and hold” and “press the set button” directions. But by using the VSE you can create and program that scene without getting out of your chair. In this case what the VSE writes to the device's linking tables is exactly the same as you would have created with the manual method. This can make programming a whole house full of Insteon devices a simpler and not quite as arduous a task.

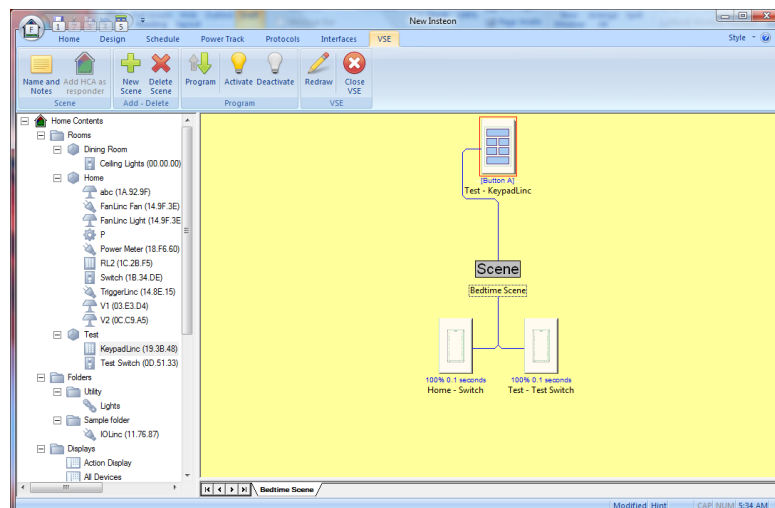
The VSE is only used with Insteon devices. It can't be used with X10, UPB, or IR devices.

---

### Using the VSE: The basics

Before describing all the details on starting the VSE and creating scenes, let's just jump into using the VSE so you can see in general how it works. All the details on creating scenes and starting the VSE are covered in detail in later sections.

For now, let's focus on one start method. Right click on an Insteon device and select VSE from the popup menu. The display pane of HCA is used for the VSE.



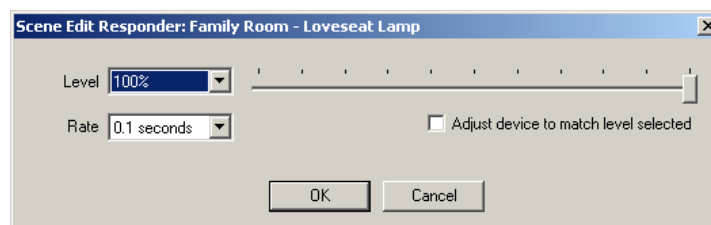
Note these elements of the VSE display:

- When the Visual Scene Editor is open, an extra category appears in the ribbon with commands used to control the VSE.
- What displays above the gray “scene” box is the initiator of the scene. In this example, a KeypadLinc.
- What displays below the “scene” box are responders to the scene.
- The name of the scene – either one that you entered or one that HCA made up – displays below the “scene” box and on the tab at the bottom of the display pane. In this example the scene name is “Bedtime Scene”.
- The lines drawn from all the responders to the scene name and from the scene name to the scene controller are there to help you visualize that all these devices are connected together.
- You may need to drag the HCA window larger to see the entire scene. When you change the HCA window size you can ask that the VSE display be redrawn. Press the Redraw button in the ribbon.
- Below each responder is text showing the level the device is controlled to and the rate at which that control happens. For example, the “Home - Switch” goes on to 100% with a ramp rate of .1 seconds. This text displays in blue if the device has been previously programmed and red if the programming has not yet been made.
- You can right-click on a device and a popup menu displays.
- There could be one or more tabs at the bottom of the display pane. Depending upon how you started the VSE, you may see one or more than one tab. More on this in a later section.

---

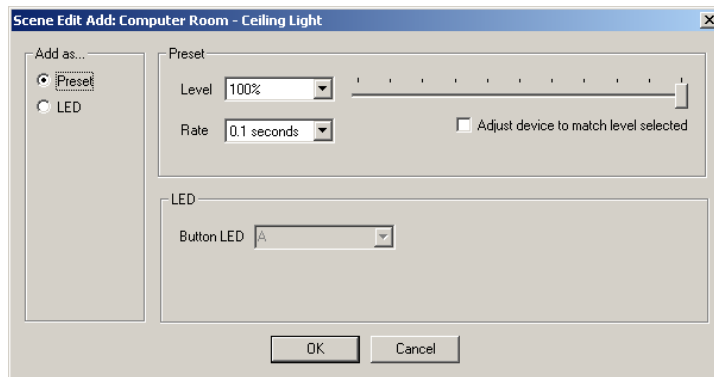
### Adding responders to a scene

Using the VSE is very similar to many other HCA tools. To add a new device to a scene all you need do is drag it from the design pane and drop it into the scene. At that point in time a popup dialog allows you to set the properties of that device in the scene. For example, dragging a LampLinc or SwitchLinc displays this dialog:



In the dialog you specify the level and the rate. If you want to see what that level looks like, you can check the “Adjust device to match level selected” box. This causes HCA to send dim/bright commands to the device as you change the level. You may not want to do this if you are building a scene late at night and you don’t want to disturb family members.

If you drag and drop a KeypadLinc this dialog appears:



A scene can control the KeypadLinc’s load or just control one of the button LEDs. This dialog lets you select the one you want and its properties.

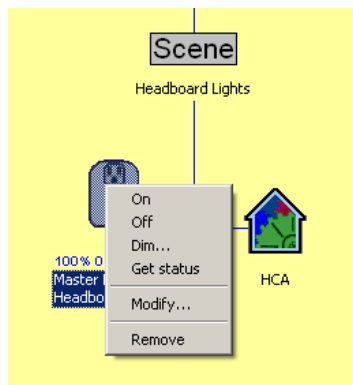
If you want to add HCA as a scene responder press the *Add HCA as Responder* button in the ribbon.

The VSE will not let you drag and drop a device into a scene that the VSE doesn’t support, nor will it let you add the same device more than once unless it has more than one sub-device.

---

## Modifying a device’s scene properties

When you added a device to a scene using drag-and-drop, the popup dialog let you set the device’s properties. You can change them any time by a right-click on the device.



The first four choices in this popup menu are just provided for your convenience and perform the same actions on the device as they do when you right-click on the device in other places in HCA.

The *Modify* option opens a similar dialog as when you added the device. You can change the level and ramp rate, or if a KeypadLinc LED, which indicator is used.

---

## Removing responders from a scene

To remove a component from a scene, right-click on it as described in the last section and select “Remove”. You can’t undo this action but you could abandon the scene changes if you discover you really didn’t want to remove that scene component.

---

## Changing the scene name

To change the scene name, double-click on the scene name and a dialog appears that lets you name the scene. You can also press the *Name and Notes* button in the ribbon to get to the same dialog.

Note: If a scene has HCA as the controller, you must give the scene a name.

---

## Testing a scene

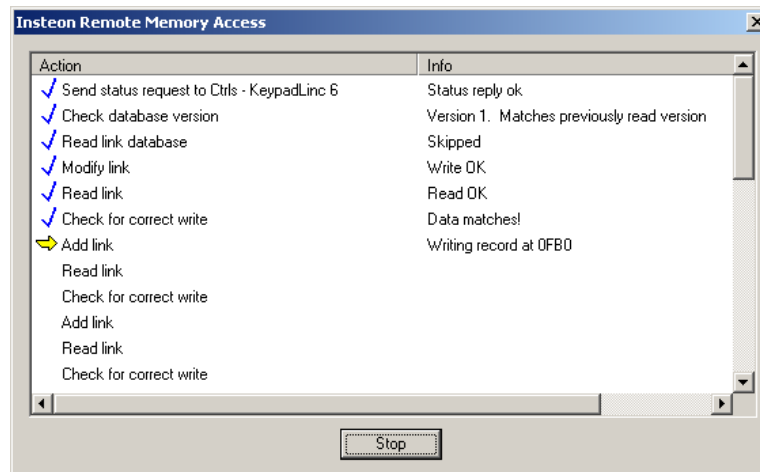
To test a scene, press the *Activate* or *Deactivate* buttons in the ribbon. Activate controls all the scene devices to their level at the specified rates. Deactivate turns all the scene devices off.

There are some limitations of the Activate option and these will be described later.

---

## Programming a scene

Programming a scene is very simple. Select *Program* from the ribbon. HCA uses the same dialog in many places when remote communication is being performed with Insteon devices. This checklist dialog shows each operation to be done and proceeds down the checklist operating on each device in turn.



Actions that complete without error are checked off when done. The current action is shown with the yellow arrow, and operations that fail show with a red X.

Depending upon the complexity of the scene, there may be a few actions or many.

When the programming is complete the *Stop* button changes to *Close*.

---

## Closing the VSE

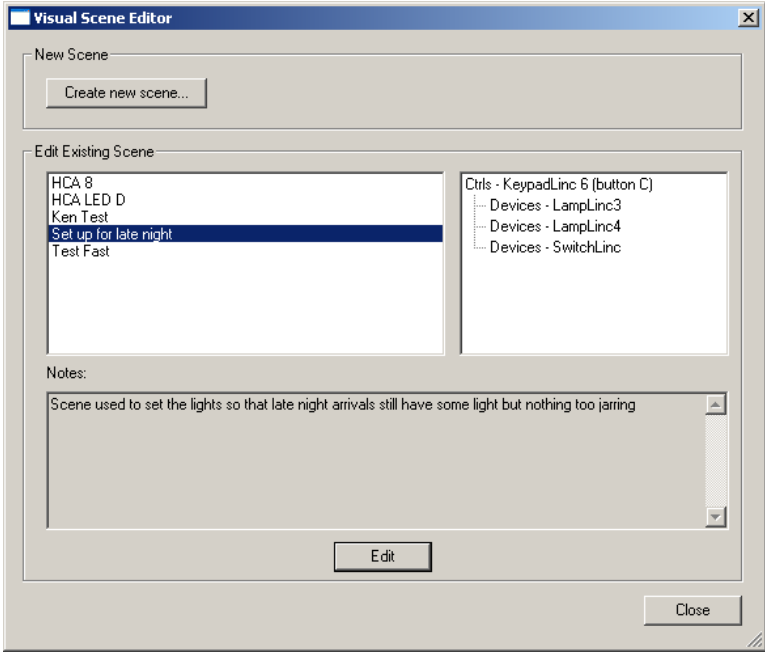
To close the VSE, press the *Close* button in the ribbon. If there are changes not yet programmed, a warning appears and you can either cancel out those changes or remain in the VSE.

## Using the VSE: Starting, stopping, and creating scenes

The VSE was designed to facilitate creating and editing scenes in two different manners. You can focus on a scene or you can focus on a device within the scenes – possibly multiple scenes - it participates in.

### Working in a “by scene” manner

To start the VSE select from the ribbon press the *Visual Scene Editor* button in the *Insteon* panel of the *Protocols* category. This dialog appears:

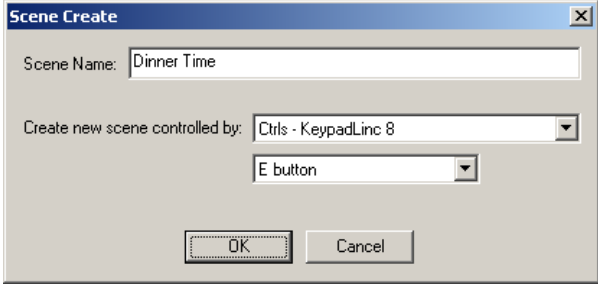


The left-hand list shows all the scenes you have explicitly given names to. The right-hand window displays, in a shorthand manner – what devices participate in the scene. Below are any notes you have saved for this scene.

Hint: In even a small Insteon installation you can wind up with many scenes. Some of these are important ones that you will spend time creating and modifying. Others are simple “press this button and control that device” type of scenes. Give names to the scenes that you expect to be working with over time but it is not necessary to name every scene.

In this dialog, to edit a scene all you need do is to select its name and press the edit button.

To create a new scene, press the Create button. This dialog appears:



Enter a scene name, and what device controls the scene. The controller can be any device that has a button or rocker. Also you can create HCA controlled scenes by choosing HCA in the *controlled by* list. In the list for the device you select are all the transmit components available and if they are currently being used by a scene or not. In this example, the KeypadLinc E button will be the scene controller.

When you close the create dialog, the VSE opens with the new scene displayed. The VSE helpfully adds HCA to all new scenes as a responder but you can remove that if not needed.

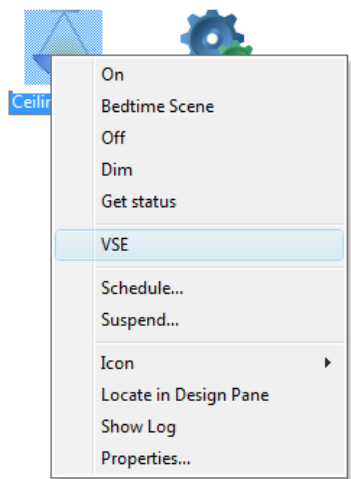
When you start the VSE in this manner – by selecting from the *menu Insteon – Visual Scene Editor*, you are focusing on a specific scene. The VSE shows only one tab and that tab shows the scene.

---

### Working in a “by device” manner

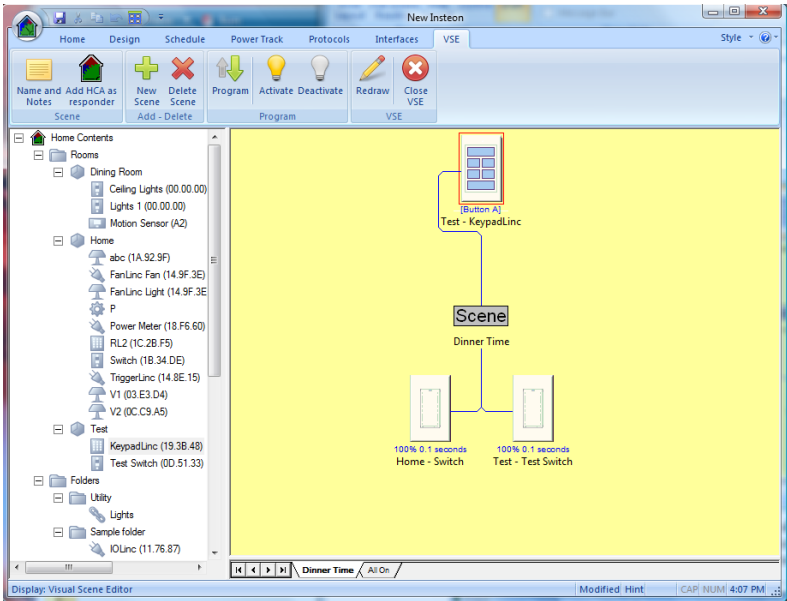
The other way to work with the VSE is to focus on a specific device and the scenes that device participates in. In this manner, the VSE shows a tab for each scene that the device is in.

To start the VSE in this way, right-click on a device in the design or display pane and select VSE from the popup menu.

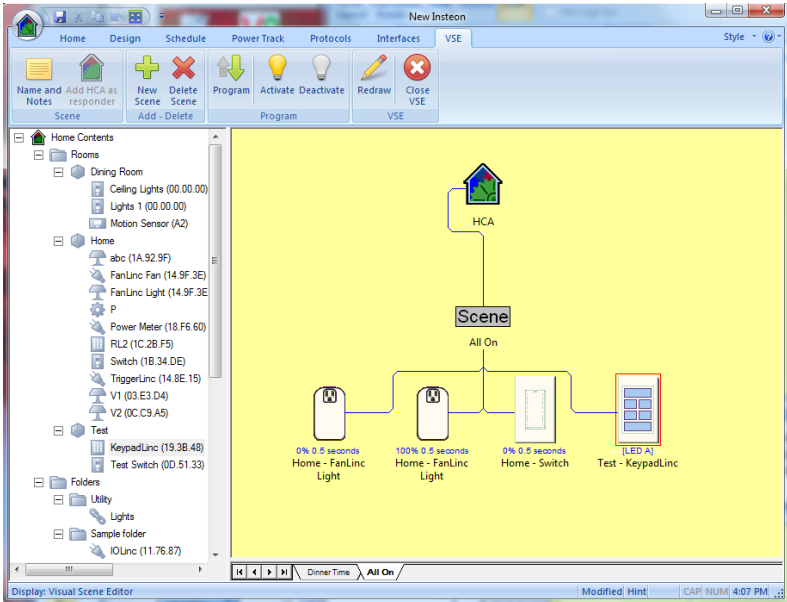




This opens the VSE with a tab for each scene.



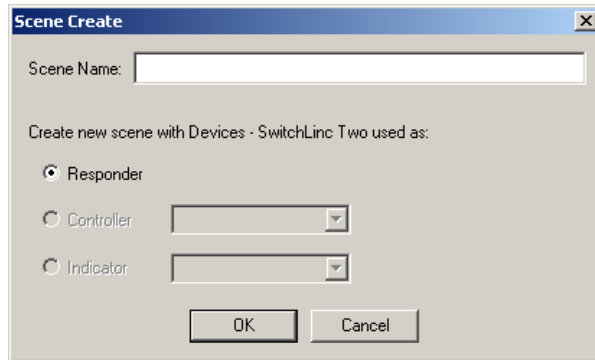
Note the red box drawn around the device. This helps show you the device you are focusing on – the device you right-clicked on and chose VSE from the popup menu. In this scene the device is used as a controller. Here is another tab:



In this scene the device is used as a receiver.

You can click through the tabs and see each scene that the device participates in.

To create a new scene, press the New Scene button in the ribbon. This dialog appears:



Note that since you are focusing on a specific device you are not asked to choose a device, only to enter a scene name and choose how the device will participate in the new scene – as a controller or responder.

---

## Deleting scenes

Regardless of how you start the VSE, if you decide that you have a scene you no longer need it is simple to have HCA remove it. Press the ribbon *Delete Scene* button. The same dialog that appears when programming a scene is used to remove the programming from all devices in that scene.

---

## Scenes initiated by HCA and by other devices

Insteon scenes have a number of features that are best understood in order to get full use out of them.

The key fact about Insteon scenes is that each scene has a controller – the device that initiates it – and only that controller can initiate the scene. Let’s try an example.

Suppose you have two KeypadLinc and three LampLinc. Also suppose that you create a scene so that when you press the B button on the first KeypadLinc the three LampLinc go to 50% in 2 seconds. Let’s call that Scene One.

Suppose you want the second KeypadLinc B button to do exactly the same thing with the three LampLinc. But there is no simple way to do that in the Insteon world. There is no command that the second KeypadLinc can send that will cause the LampLinc to respond. The second KeypadLinc can’t “pretend” to be the first KeypadLinc and have the LampLinc respond.

You can create a new scene – let’s call that Scene Two – that also is initiated by the second KeypadLinc B button and controls the LampLinc in the same way. So even though both scenes have the same effect on their responders, Scene One is not the same as Scene Two and that is important to remember.

**Note for UPB users:** If you are familiar with the Universal Powerline Bus (UPB) linking model this probably all sounds confusing. In the UPB world once the receivers are programmed, any controller can send the link number and the responders will respond. This isn’t the way that Insteon works. In the Insteon world the responders only respond when the link is sent by the

controller they were programmed for.

Is this just an obscure point? How does it affect you?

From the example above, suppose you want HCA to initiate Scene One. You want HCA to do the same thing that pressing the KeypadLinc button does. Can HCA do that? No it can't. Just like the second KeypadLinc as described above can't initiate the scene, neither can HCA. And just like the above example, you would have to create a new scene with HCA as the controller and the LampLincs as the responder. Now we have three scenes that have the same effects – all three LampLincs go to 50% in 2 seconds - but all three scenes are not the same scene.

In many ways this is annoying as you have to create a lot of “duplicate” scenes. You may have many controllers that all want to do the exact same thing and you have to program each of them to do that. Good thing you have the VSE!

This also affects the ability in the VSE to test a scene. If you are creating a scene that is not initiated by HCA, when you use the Activate and Deactivate VSE right-click menu picks HCA can't control the scene with a simple scene command. It has to simulate the scene by sending each responder commands to adjust them to what would have happened to them had the scene controller initiated the scene. HCA will have varying degrees of success in doing this.

---

## VSE, the linking tabs, and the multi-way wizard

There are several other methods in HCA that create scenes. One of these is the Multi-Way Wizard – described in the Insteon appendix – and the linking tabs on the device property dialog.

Like the VSE these tools update the linking table stored in the devices. So their effects can be seen in the VSE after they have performed what they do. That is, if you link HCA to a KeypadLinc button using the property tab, next time you open the VSE on that KeypadLinc you will see a scene where the KeypadLinc is the controller and HCA the responder.

In this way the VSE is the “tool above them all” in that it can do whatever the linking tabs and the Multi-Way wizard do. But those tools may be more convenient to use as they were created to handle specific common tasks so don't forget them!

---

## Using scenes

Using the VSE you can create scenes. But what to do with them? If they are scenes controlled by one of your Insteon devices – a KeypadLinc button or SwitchLinc paddle for example – you use them by pressing the button or tapping the paddle.

If the scene is one that HCA initiates, you can use the Visual Programmer Scene element to cause HCA to send the commands to initiate the scene. You can also use the Visual Scheduler to schedule the scene activation or deactivation as part of your schedule.

For more information, see the User Guide chapters on Schedules and Programs.



## Chapter 16

# Power Track

Modern devices are mostly two-way - they can both send and receive. Most importantly then can often be configured to transmit their status when locally controlled.

Using that 2-way feature HCA contains *Power Track* which is a method for keeping track of what time a device is turned on and when it is turned off. This information is recorded in a file for each device.

HCA can analyze these files and produce both current power usage graphs – how much power are your devices using at this moment – and how much power have my devices used over a selected time period, for example, the last seven days, or the last 48 hours.

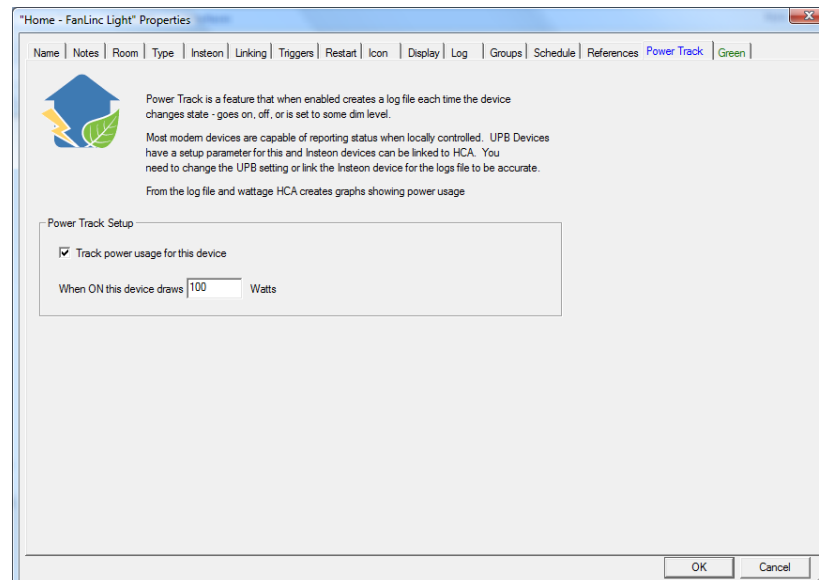
This chapter describes:

- How to enable Power Track on a device
- The current power graph
- Historical Power Graphs
- Displays and Graphs
- Support for the Insteon Power Meter device

---

## Enabling Power Tracking

Before you can use the Power Track features you must enable power track for a device. This is accomplished on the Power Track tab of a device's properties dialog.



All you need do is to tick the checkbox and supply how much current the device draws when at full power. For lighting this is the bulb wattage. For appliances this is usually listed on a sticker someplace on the device.

Once this is done, HCA creates a power log file for the device in the Power Track folder in your HCA documents area. The filename is created from the name of the device – replacing any invalid filename characters with ‘\_’s and adding month suffix. Each month a new file is created. These files are small so there shouldn’t be a disk space issue.

Once these files exist, then each time HCA controls the device – from a schedule, a program, or when you interact with the user interface - an entry is made in the file.

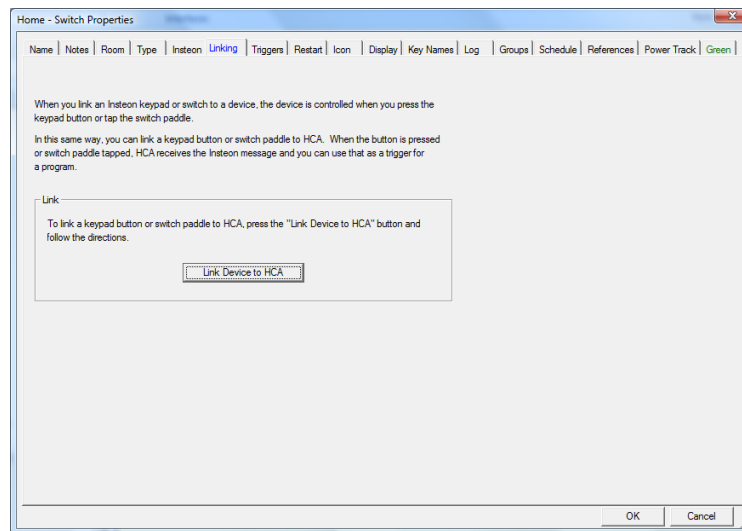
But this is only half the situation: If you locally control the device you want HCA to receive notice so that the power track file can be kept up to date.

---

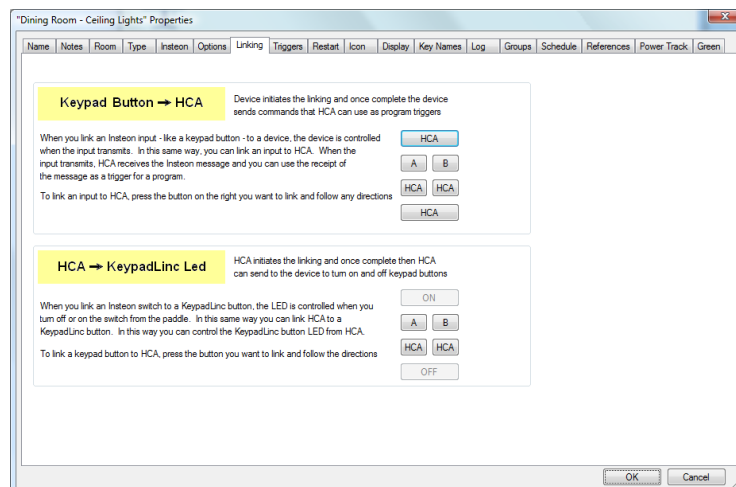
## Insteon Devices

For Insteon devices you need to link the device to HCA. Just like when you link two devices together so that when one device is controlled it sends to the other device – for example, tap a switch paddle and a module is controlled also – you must link HCA to the device as well.

This is accomplished on the linking tab of the device’s properties dialog. In the case of a switch, there is only a single button:



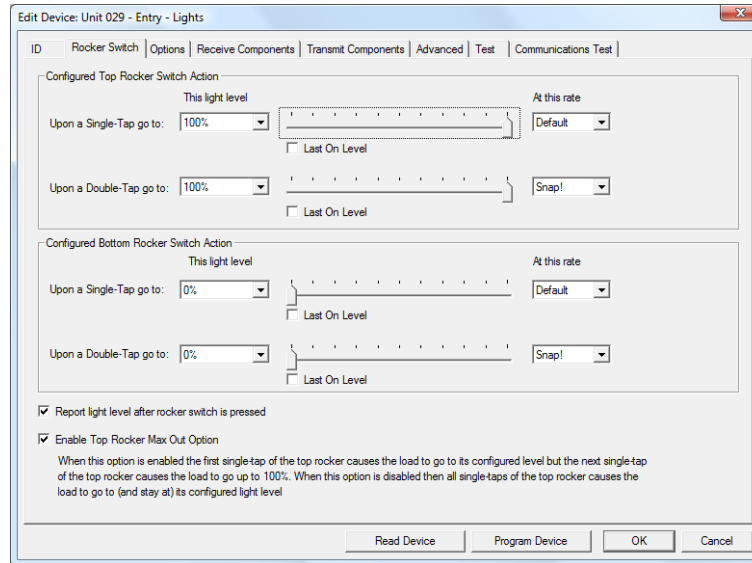
In the case of a KeypadLinc, you should link the “On” button to HCA.



More on linking is in the *Insteon* Appendix.

## UPB Devices

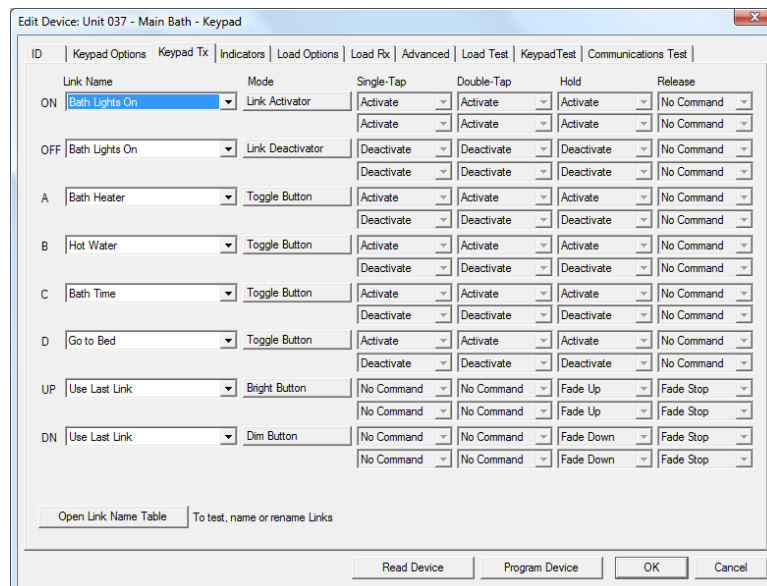
UPB Devices use a simpler mechanism to enable them to report status when locally controlled. In the UPB Configuration program – UPStart - the properties of each device type that controls a load contains an option to report status when locally controlled. This option is on one of the device property tabs. For example, here is a switch:



Note the 2<sup>nd</sup> checkbox from the bottom.

You don't have to use that option if the device is set to transmit a link from the switch paddle or keypad button. Once you have imported your UPB Network into HCA, the transmission of the link from the device will keep HCA up to date on the device status.

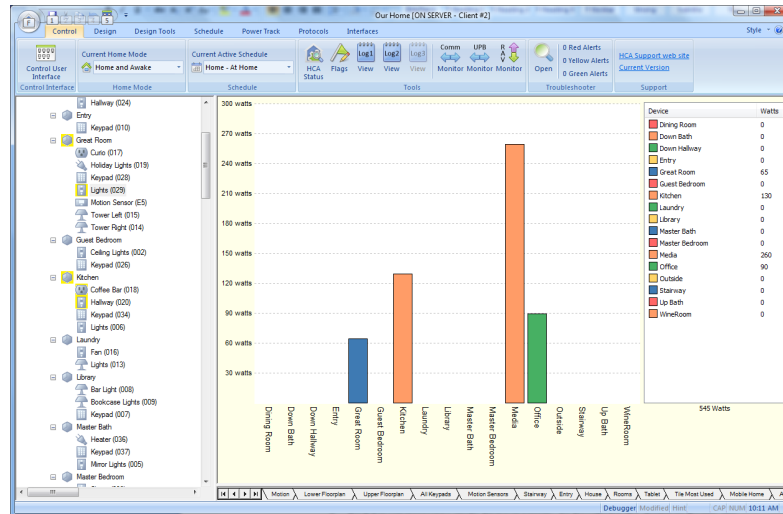
For example, here is a KPLD configuration.



When HCA sees the “Bath Lights On” link being transmitted it “knows” that the load has been turned on and at what level.

## Current Power Graph

The current power graph appears in the display pane. For example:



Each device or room shows as a bar representing the current power usage in watts. The list on the right shows the data in tabular form.

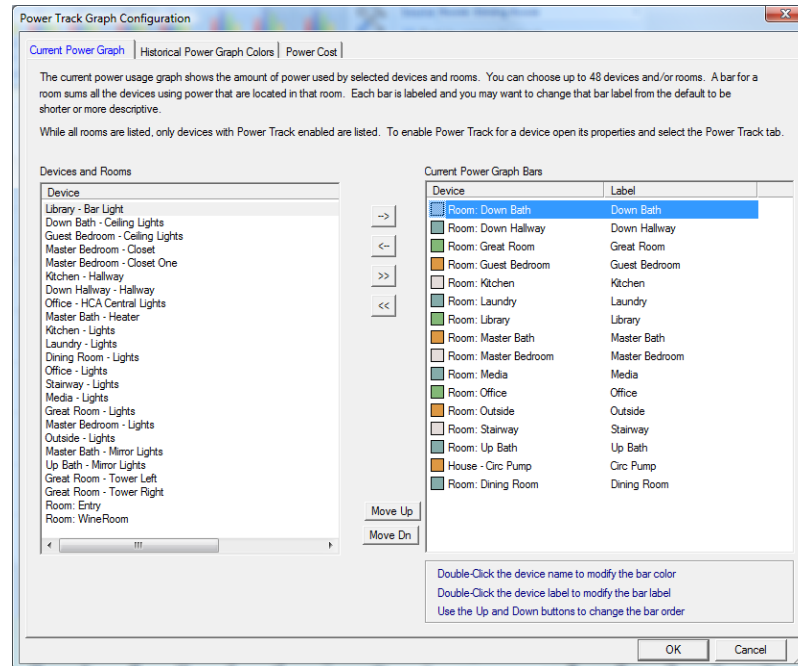
There are several important items to note about the current power graph.

- Before you can display the graph it must be configured. This is described in the next section.
- The graph is limited to 48 bars
- Instead of showing the current wattage, you can have the graph show the current cost of the power. In the ribbon is a checkbox that toggles between showing percentage and cost. The cost per watt hour is defined in the setup and is described in the next section.
- Each bar can represent a single device or a room. As described in the chapter on Rooms, any device that contributes to the state of a room includes its power usage to the bar for the room.



## Configuring the current power graph

To configure the current power graph, press the *Graph Setup* button from the ribbon *Power Track* category.



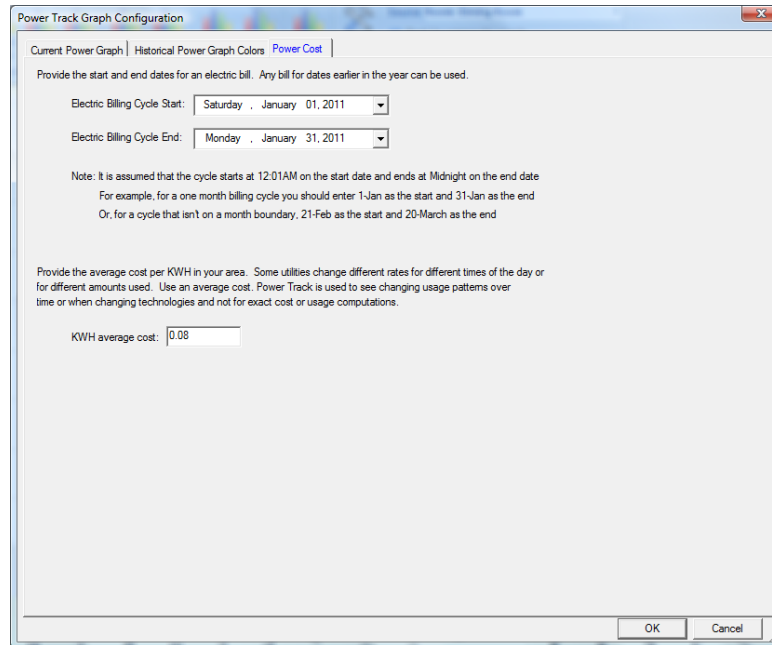
This dialog lets you select what bars appear in the Power Track graph, how they are labeled, what color is used for the bar, and the order of the bars.

To select which bars to appear in the graph, use the arrow buttons to move the device or room name from the left list to the right list. A bar is shown for each item in the right list. To change the bar order, use the *Move Up* and *Move Dn* buttons.

There isn't much space in the graph for the bar labels – show vertically below each bar. For long room or device names, you may want to choose a different name so it fits better. To assign a label different from the device or room name, double-click on the label text in the left list and a dialog opens that allows you to change the label text.

To change the color assigned to the bar, double click on the device name in the first column and a color selection dialog opens.

The last item to configure is the cost per Kilowatt Hour (KwH). This is how the graph shows cost information. Press the *Graph Setup* button from the ribbon *Power Track* category and select the Power Cost tab.



The power cost is at the bottom on the dialog. You can usually get this information from a power company bill. In some locations the power cost varies with time or day or usage over the billing cycle. In this case just enter an approximate average value. For these purposes this is sufficiently accurate.

While you are in this dialog you may want to enter the start and end dates of your billing cycle. Again, this can be determined from your last billing statement. These dates are used by the Historical Power Graphs which are described in the next section.

---

## Historical Power Graphs

While the current power graph is useful it really only reports on what you probably already know – what’s on right now. Historical Power Graphs are where the real action is!

By plotting the power usage for one or more devices or rooms over a period of time you can easily see what are the biggest power users and then see if their power usage can be reduced.

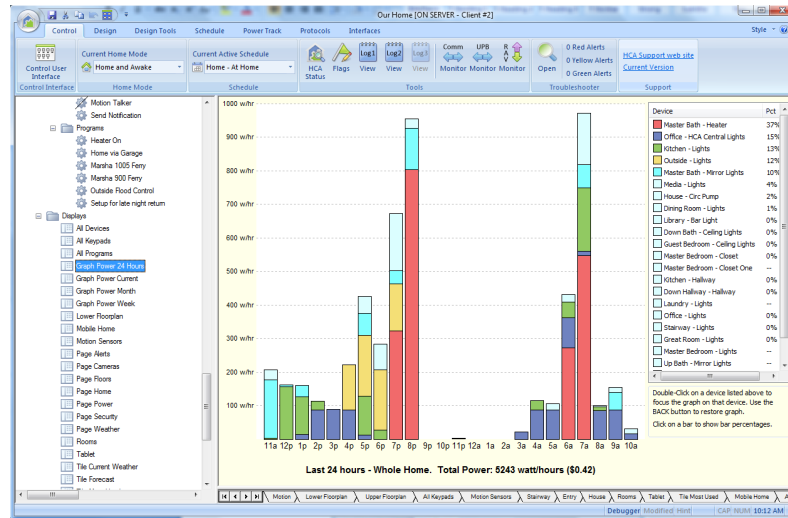
Unlike the current power graph, no setup is needed other than enabling Power Track on one or more devices.

A graph can be constructed for these time periods:

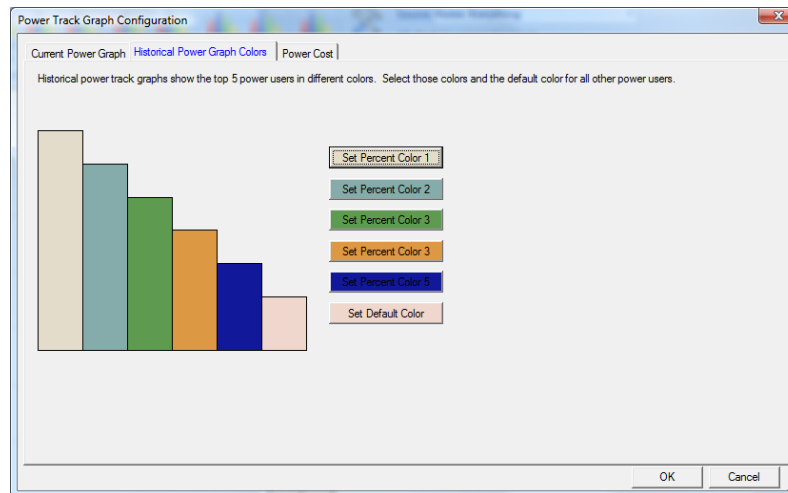
- Last 12 Hours
- Last 24 Hours
- Last 48 Hours
- Current Week – Sunday to Saturday
- Previous Week – Sunday to Saturday
- Last 7 Days
- Last 14 Days
- Last 30 Days

- Current Month
- Previous Month
- Current Billing
- Previous Billing
- Current Year

To show any of these graphs, press the button for that graph in the ribbon. This is an example for a graph over the last 24 hours.



Unlike the current power graph where colors are assigned to individual devices or rooms, here colors used are to show the largest power users. These colors are selected from the Graph Setup dialog on the *Historical Power Graph Colors* tab.



You can assign any colors that appeal to you. Each button on the right of the dialog opens a color selection dialog.

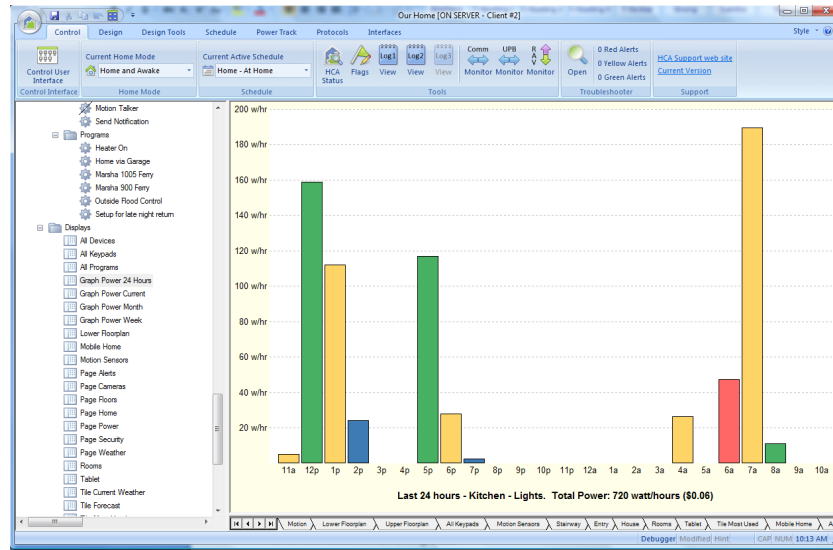
When working with Historical Power Graph there are a number of features you may want to take advantage of. These are described in the next sections.

## Focus the graph on a single device or room

While a composite graph – showing data from more than one device or room – gives you an overall picture of power usage over the selected time period – it may be advantageous to focus in on a single device or room for the selected graphing period.

To do this either change the graph “source” dropdown in the ribbon or double click on a device in the tabular list. The graph is redrawn to show just that device or room.

Look again at the graph shown above. By double-clicking on “Kitchen Lights” the graph is redrawn to show power usage for just that device.



To return to the composite graph, use the *Back* button in the ribbon.

## Show a breakdown for a single bar

The tabular list on the right side of the graph shows the breakdown – by cost or percentage – for the whole graph. To show the breakdown for a single bar, all you need to do is to click on the bar. The label below the bar turns red to show it is selected and the tabular list shows data for only that bar.

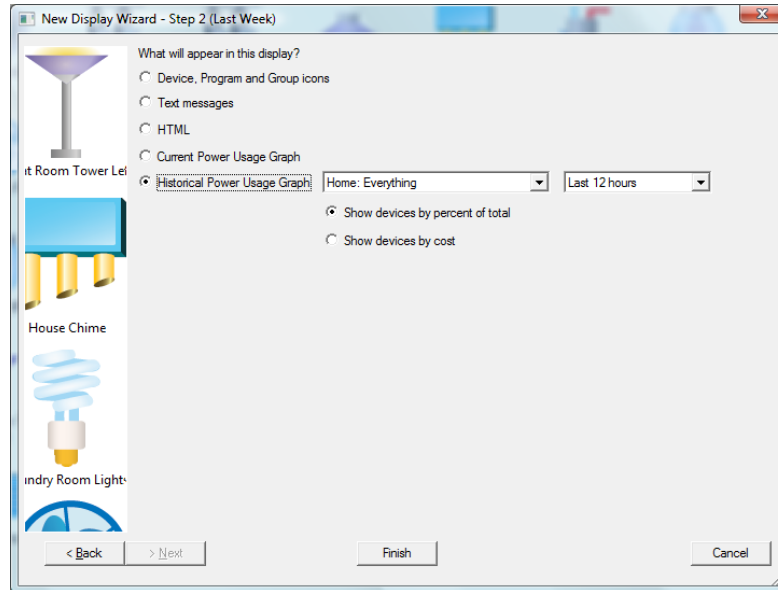
To return the tabular list for the whole graph, click someplace on the graph background.

---

## Displays and Graphs

In addition to using the graphing buttons in the ribbon, you can also create displays that show power track graphs. This can be advantageous when working with the Control Interface as you can then have icons on a page that open a graph.

To create a graph display, use the new Display Wizard. At step 2 of the wizard choose either the current power graph – in which case there are no options – or the historical power graph:



Select the graph source – room, device, or “Everything”, and one of the time period choices.

Once you have that graph display defined, selecting it in the design pane or clicking on the icon for it in the display pane opens that graph.

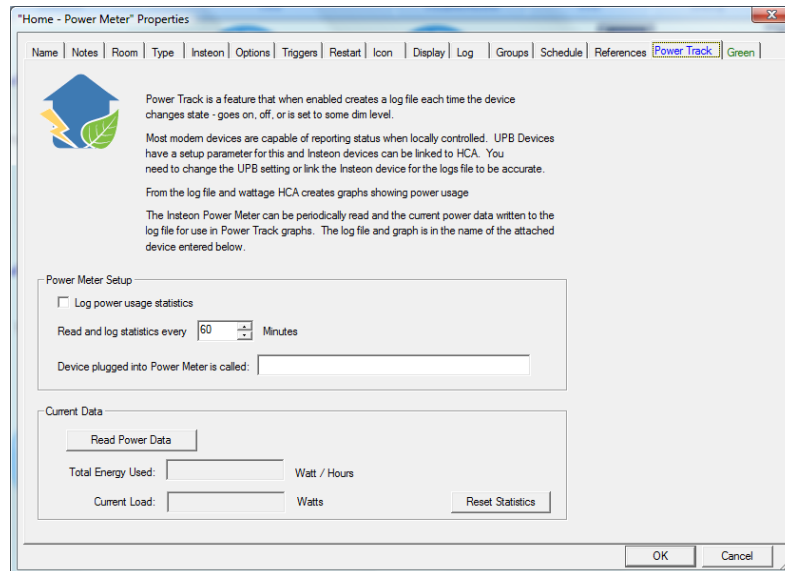
---

## Insteon Power Meter

The Insteon Power Meter is also supported for Power Track. This device provides an actual measurement of a device’s power usage over time. Using it you can get power usage data for devices that are not part of your automation solution – like televisions, refrigerators, etc.

The device is supported by periodically reading the data from it and saving it in a power track file.

After you add the device using any of the methods for adding devices, open its properties and select the Power Track page.



For the Power Meter, the tab appears a bit differently than other devices. Here you specify:

- How often to poll the device to read out the accumulated power
- The name of the device the meter is connected to. This name is used for the filename of the power track file and also used in graphs. This way if you want to relocate the power meter to a different appliance you can just change the name of the appliance it is attached to and you need not rename the Power Meter device in HCA. And because the power track file uses the appliance name, you don't lose the data from the previous usage of the meter nor does the old appliance's data get mixed in with the new one.

You can also use this tab of the dialog to read the device and see the current load and accumulated power.

Here is an important point about the power meter that is best shown by an example:

Suppose the meter records an accumulated power of 100 watt/hours at 10am. Suppose the meter is read again at noon – a poll of every 120 minutes. At that time the accumulated power is 500 watt/hours

You can't tell if the device was off – accumulating no power – for example, from 10 to 11:15 and then turned on and the 400 watt/hours were accumulated in the next 45 minutes.

Because of this, when the graph is constructed for this device, HCA assumes that the power is accumulated evenly over the time period – that is, 200 watt/hours from 10 to 11, and 200 watt/hours from 11 to noon.

## Chapter 17

### HCA Keypads

Unlike physical keypads that you may have installed in your home, the "HCA Keypad" isn't a physical keypad but rather a way to create a user interface for controlling devices and programs.

---

#### The Keypad Idea

The idea behind HCA keypads 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.

---

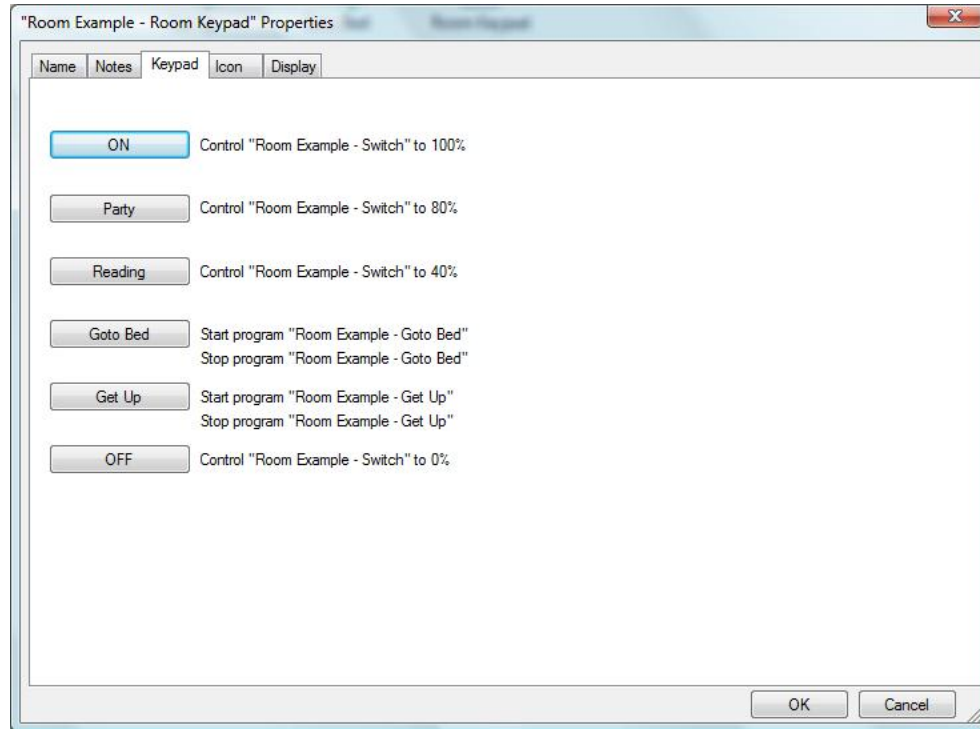
#### Adding and configuring a keypad

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 next to the button.



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.

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:

- Keypad Room Name
- Keypad Device Name
- Button label
- 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.

## Examples

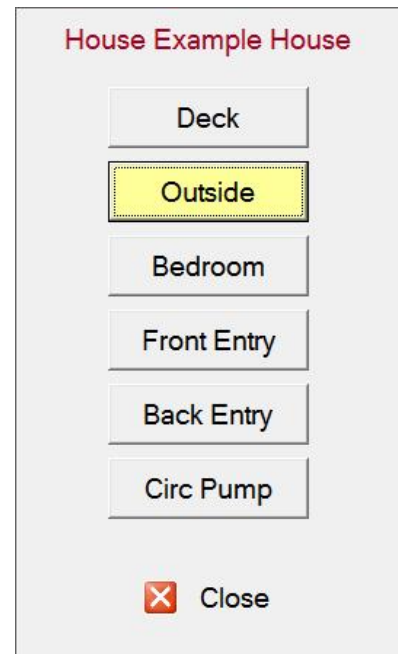
Here are four examples that cover some of the common ways of using keypads.

### 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.

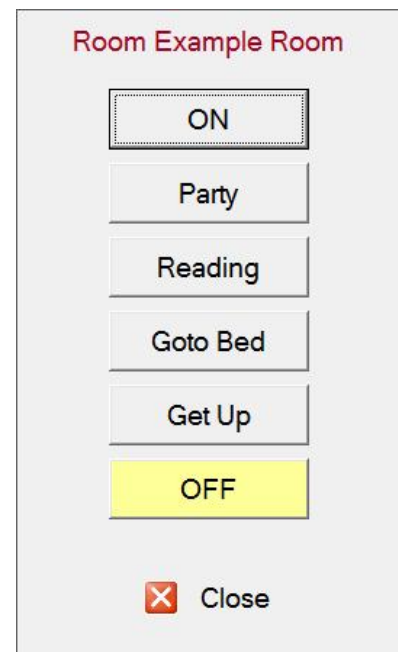
In this example some of the buttons control a single device and other buttons control a whole room using the HCA "room" operation.

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

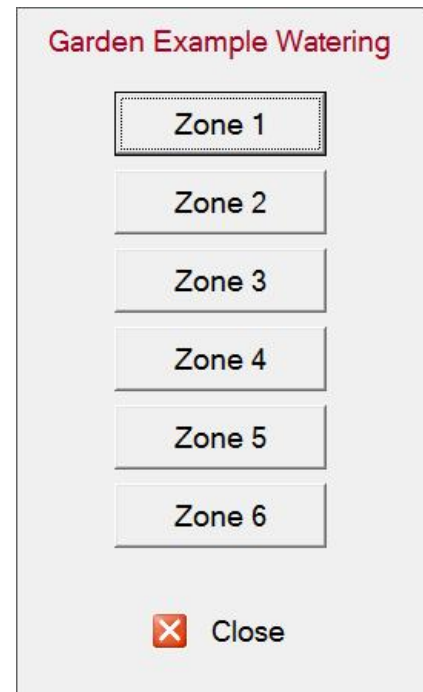
In this keypad each button controls the same set of devices. At any given time the room reflects one scene or another so only one button is ever on at a time. The keypad button that is ON shows what scene is the current scene.



Special Device Control Keypad

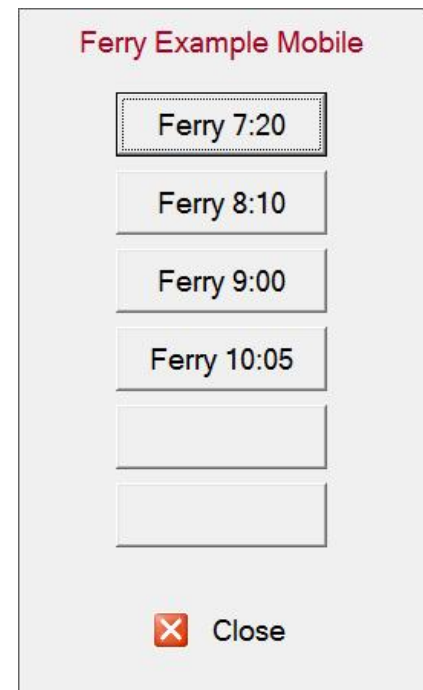
The keypad created for the Irrigation device is an example of this. Each button is tied to one part of the same device. In this case the keypad controls 6 zones of the irrigation controller.

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 craft a user interface for non-technical users that gives them a limited set of actions that they can EASILY use.**



## Chapter 18

# Network Devices

HCA contains a tool that lets you determine how HCA interacts with network devices via a network "Ping". A Ping is defined as:

*Query another computer on a network to determine whether there is a connection to it.*

What this tool does is to periodically ping one or more network devices and based upon if the respond or not and the rules you set carry out different actions.

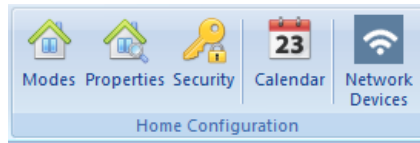
When we say "network devices" we are mostly talking about mobile phones and tablets that are communicating on your internal network using WiFi.

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?

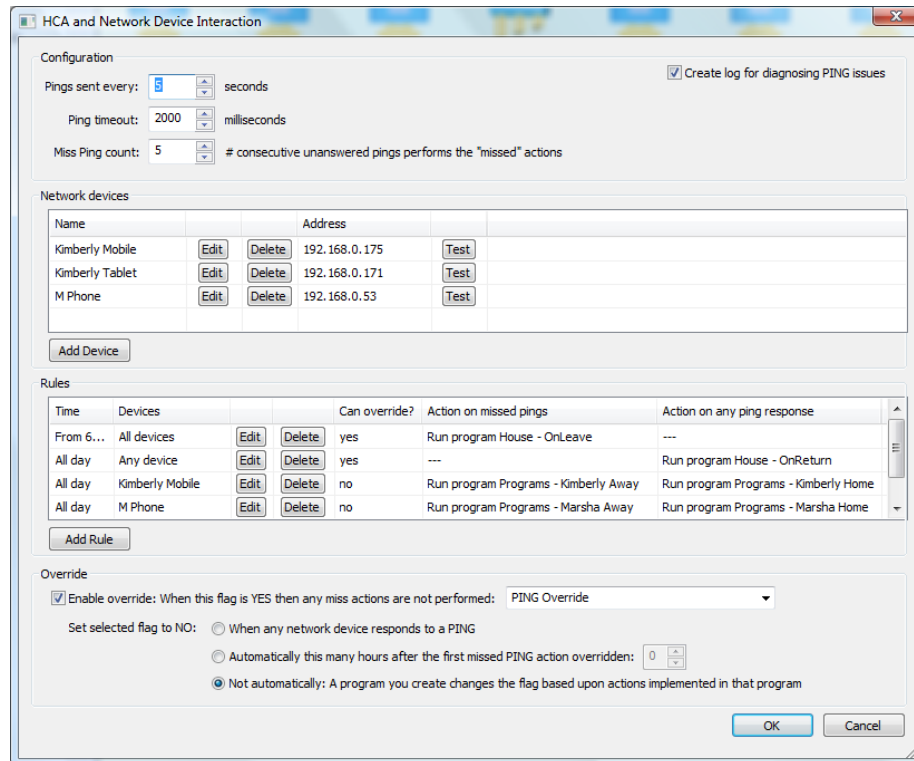
---

### Configuration

In the Design category in the Home Configuration panel is a Button named “Network Devices”



When that button is pressed, this dialog opens.



In the first part of the dialog you specify how often the devices are pinged and how long they have to respond before they are considered unavailable. An important configuration parameter is the "miss count". Sometimes devices don't respond to a ping because the device is busy, or temporarily out of range. It is best to set the miss count to greater than 1. However if a device does go out of range for real then it will take 'n' pings, each timing out at the time you have set before the "miss" rules are evaluated.

For example, in the above dialog the count is set to 5 and the timeout is set to 2 seconds. That means that in the best case it will be 10 seconds after a network device leaves the range of the network before the miss rules will be evaluated.

To add a device, press the Add Device button. A dialog opens where you can enter in the IP address of the device and give it a name for easier reference. This dialog also contains a "test" button to verify that you have the IP address correct.

You can input up to 32 network devices and specify how often they are pinged.

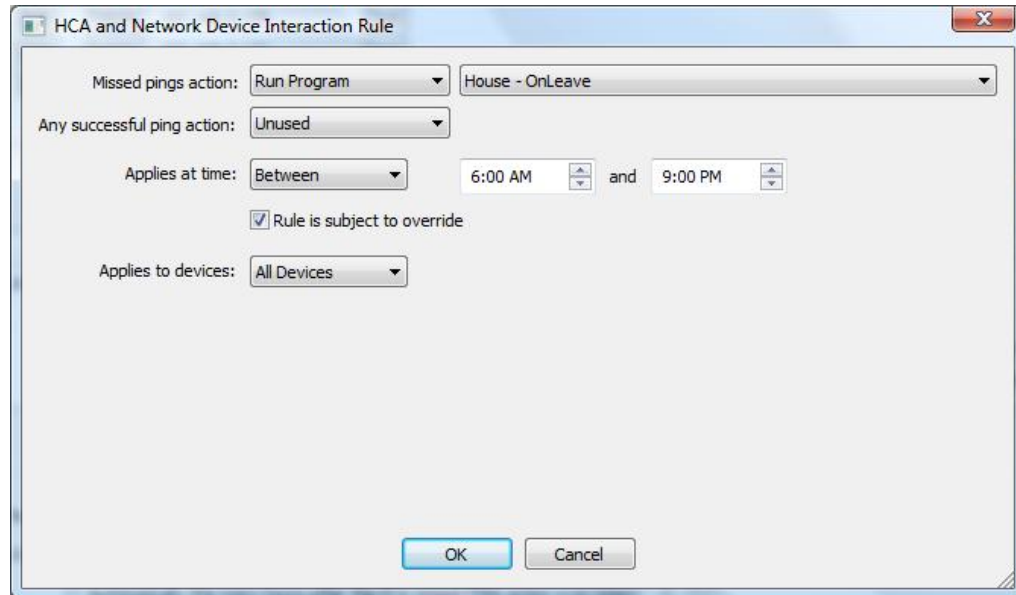
---

## Rules

Also specified in the configuration 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). The action of 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.

**Tip:** See the programs chapter for information on Parameterized programs.

---

## Override

There is 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 use the override mechanism? Suppose you are using this tool to change the home mode when all the occupants – each with their mobile phone/tablet – are away. But also suppose that while “everyone” is leaving the baby sitter is home with your 3 year old who – remarkably doesn’t have their own mobile phone. You don’t want the home to go into “away” mode. In this case you can enable the override – perhaps via a HCA keypad – before leaving home.

---

## Logging

There is a logging facility that you can enable which creates “ping.log” in the HCA documents area Logs folder. This lets you track how it works and get a sense on how your devices respond and the rules are evaluated.

## Chapter 19

# Troubleshooter

Once you have been using the Home Control Assistant for a while, you will probably want to look at the Troubleshooter. The Troubleshooter is a set of several tools you can use to make sure that your home design is working the way you want it to.

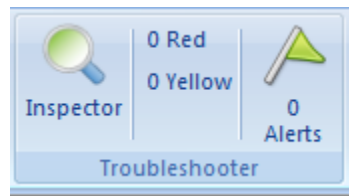
---

### Overview of the Troubleshooter

The Troubleshooter consists of three tools:

- Inspector
- Alert Manager
- Log Viewer

The composite of how things are working appears in the ribbon *Home* category *Troubleshooter* panel and tells you several things at a glance.



There are two parts to look at. The right and middle report what the Inspector has found and the left side reports what the Alert manager reports. To open the Inspector click on the left side. To open the Alert manager click on the right side.

Before getting into each of these tools it is important to first understand the differences between what the Inspector reports and what the Alert Manger reports.

The Inspector reports on potential problems that can determined from a static analysis of your design. The Alerts Manger looks for problems that occur over time while the system is running. It looks for dynamic problems.

---

### The Inspector

The Inspector is a tool that locates and describes possible problems in your home design. It serves as a check on your design, and can point out potential difficulties before they create a problem in controlling your home.

As you work with objects in your home design, HCA observes errors or anomalies in many different places, and records them as they are encountered. The Inspector pulls all these error reports together into one place. Not everything it describes is necessarily a problem, but you should look into each item it notices.

Some of the problems (there are many more) the Inspector checks for are:

- Is there more than one device with the same primary address?
- Has a schedule been selected as the normal start schedule?

- Does any schedule contain schedule entries tied to specific calendar dates that are now gone by?
- Are all of the programs in the design ready to start?
- For any programs that reference external files, are the paths to the files correct and the files exist?
- Are there any devices, programs, or groups that have been suspended?
- Are there conflicting schedule entries for a group and any of its members?
- Are operations with devices consistent with their properties? For example if you are using a DIM operation does the device properties show it supports DIM?

If any of the conditions are found, then you have a possible problem in your design, and the Inspector notes it for you.

The Inspector periodically examines your home design. Each time you make a change to your design the Inspector checks for possible problems.

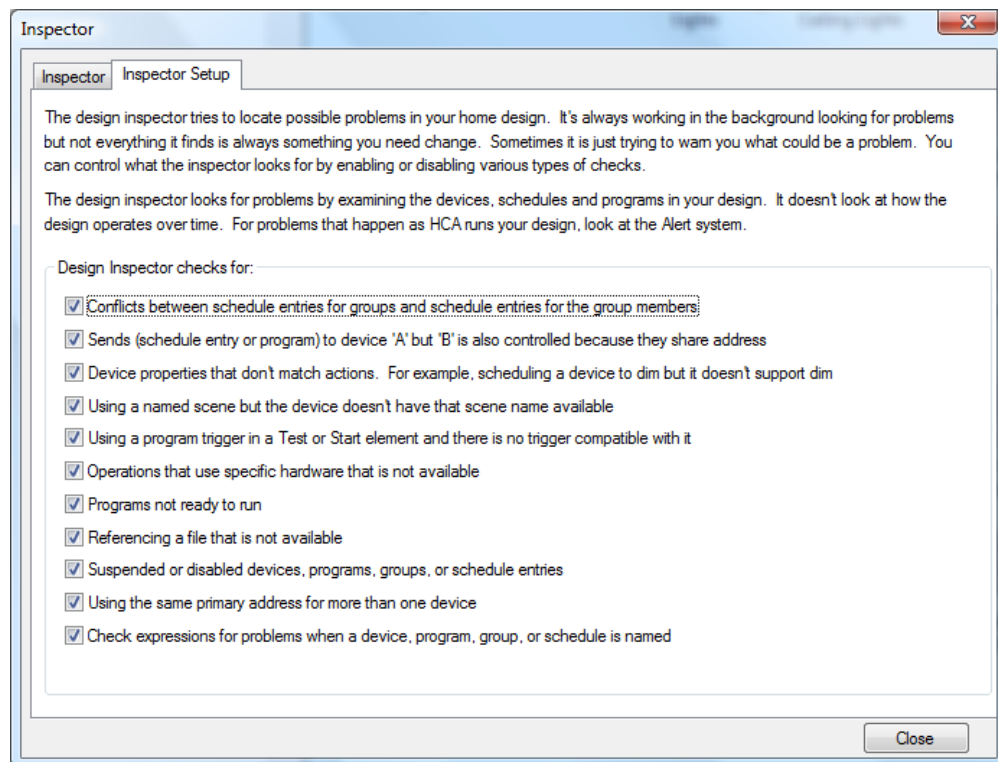
Depending upon the severity of the problems found, the inspector icon in the ribbon changes.

The Inspector classifies each issue found as a red or yellow issue. As you would expect, red issues are more serious. In the ribbon the current count of red and yellow issues are displayed.

---

## Inspector Configuration

The inspector can check for many conditions. There are so many possible conditions that it may be useful for you to select only some of them. The Inspector Setup tab appears as:



After reading the text on this dialog, select the conditions you want the inspector to check for.

If you uncheck one or more of the options, those checks are not done so no messages for those kinds of checks appear on the Inspector tab. If you do check one of the options, messages may appear but you can *check them off* if they are not problems. This is described in the next section.

## Inspector Checklist

The Inspector report is the 1<sup>st</sup> tab of the inspector.



Associated with each message in the checklist is a box that appears to the left of the message number. Some conditions the Inspector reports upon are problems and should be corrected, but others may not be. Only you know which is which.

If a specific message reports on something that is not a problem in your design, use the left mouse button to click the checkbox associated with the message. In doing this, you are telling the Inspector that this message is something that truly is not a problem. For example, in the above screen image, issue number 8 reports that a program is suspended has been checked off, because I made a conscious decision to suspend that program—hence it is truly not a problem

If the Inspector has reported on a condition that is really a problem don't check it off - just close the Troubleshooter and resolve the problem. Once you have made the correction, the Inspector will no longer report the problem.

**Hint:** If the message displays in blue text you can click on it and the inspector closes and the appropriate dialog where you can make changes to resolve the issue opens.

The two options on the bottom of the Inspector tab are used to control what you see in the checklist. Conditions that you check off don't disappear; they are still reported each time the Inspector checks your design. To see just those messages that are not checked off, select the *Show only what's not checked off* option.

Once a message is checked off, the Inspector no longer uses that message in determining if the status bar lights should show a yellow or red condition. In this way, once you fix all problems or check off those messages that are OK, the status light will show green. If the Inspector finds anything new, the light will change to yellow or red.

## Alert Manager

The Inspector as described above looks for problems in your design by analyzing your design at any minute in time.. There are a second set of things that can go wrong as HCA operates your design. These are just a few examples:

- A device is polled for it's status but it doesn't reply
- An attempt is made to take a weather observation but it fails.
- One part of a UPB message being received is missing
- A message is received from a device that doesn't match anything in your design.

As you can see from this list, all of these items can't be checked for by the inspector as they only happen over time. What happens when these sorts of problems occur is what the Alert Manager is for.

Each of these conditions is called an Alert and each can be configured to how you want HCA to react.

The Alert Manager dialog is opened by clicking on the Alerts side of the troubleshooter panel in the ribbon. It consists of two tabs: Summary and Details.

### Alert manager summary tab

The summary tab show each possible alert, if and when it occurred, and tools for configuring.

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	Disabled	0		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	Disabled	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

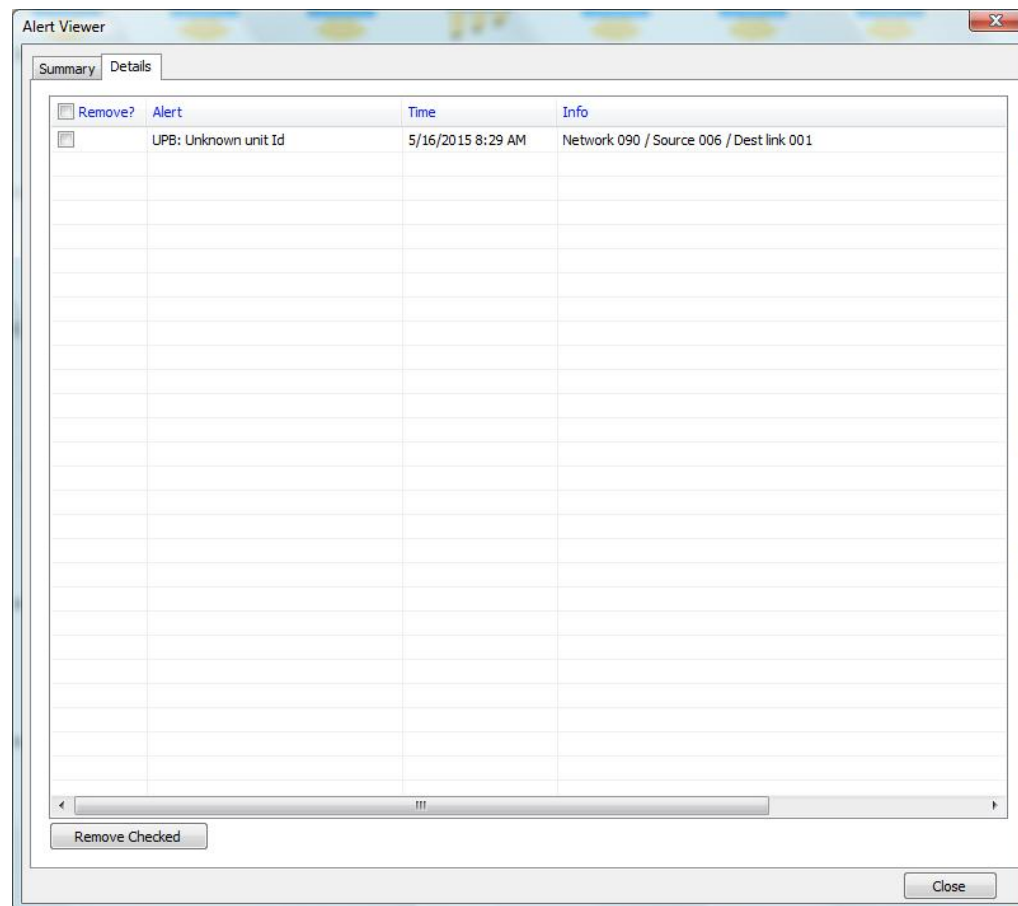
The Summary tab lets you view and clear the count for each alert or clear them all using the "Clear All" button. It also shows the alert level – green, yellow or red - for each alert determined from its count. You can also re-configure the alert from this view. Press the "Configure" button or double 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 manager details tab

The details tab shows information about each alert that happened, when it happens, and any details about it.



You can also re-configure the alert from this view. Double click the line to open the alert configuration dialog.

---

## Configuring alerts

Each alert can be configured in several ways:

- How often it logs the alert in the alert details log. The possible choices are "never" , "each time", "once a day", "start after x and stop after y" a simpler "log after x" and "once"
- How its occurrence changes the green-yellow-red level shown in the HCA ribbon

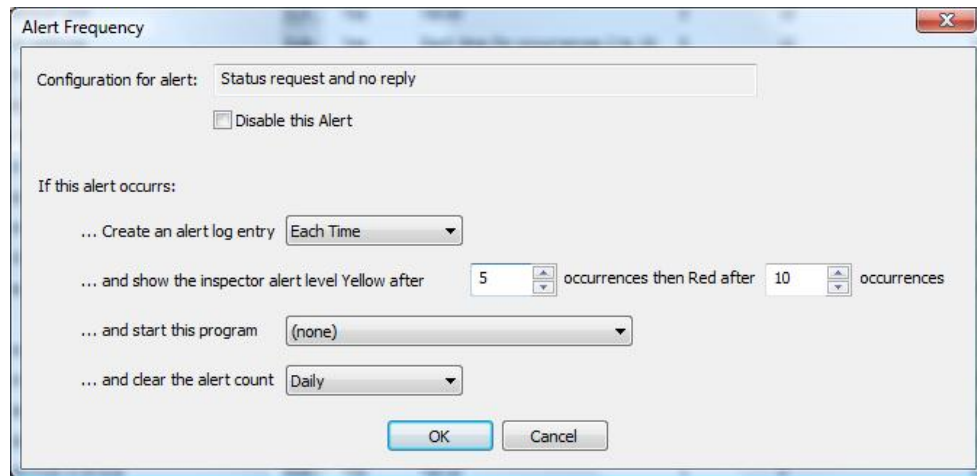


- Does HCA start a program when it happens? This program could, for example, send an email or SMS message to you reporting what has happened.
- When to clear the alert count.

The next major change was to separate logging from counting. So now you can "never log" but still count occurrences. This can be useful for example if you really don't care about specific devices failures just how well the network is working in general

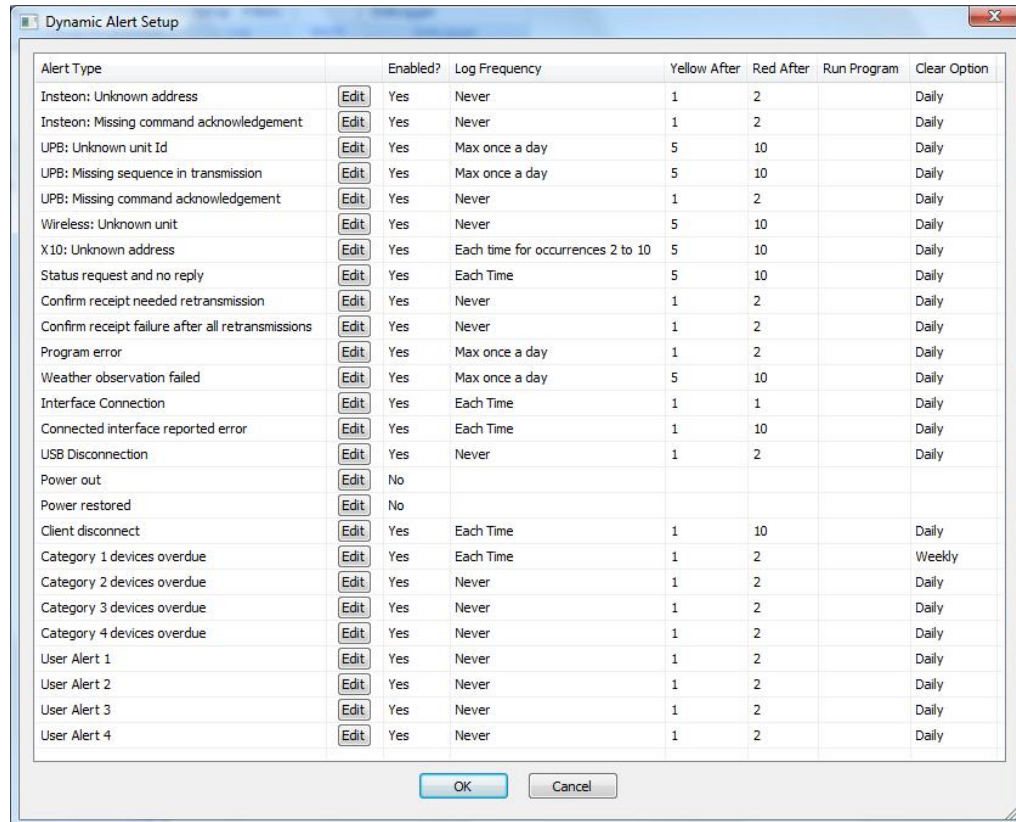
The alert setup has been changed to have daily, weekly, monthly clear options.

Added to the log options of "never" , "each time", "once a day", "start after x and stop after y" a simpler "log after x" and "once"



In addition to the alerts that the manager watches for HCA also has four "user" alerts that can be triggered by programs using "\_AlertAdd" function in a Compute element.

From the viewer you can configure each alert. There is also an additional way to configure and view the configuration of all alerts at one time. Select from the ribbon *Tools* category *Alerts Setup* button.



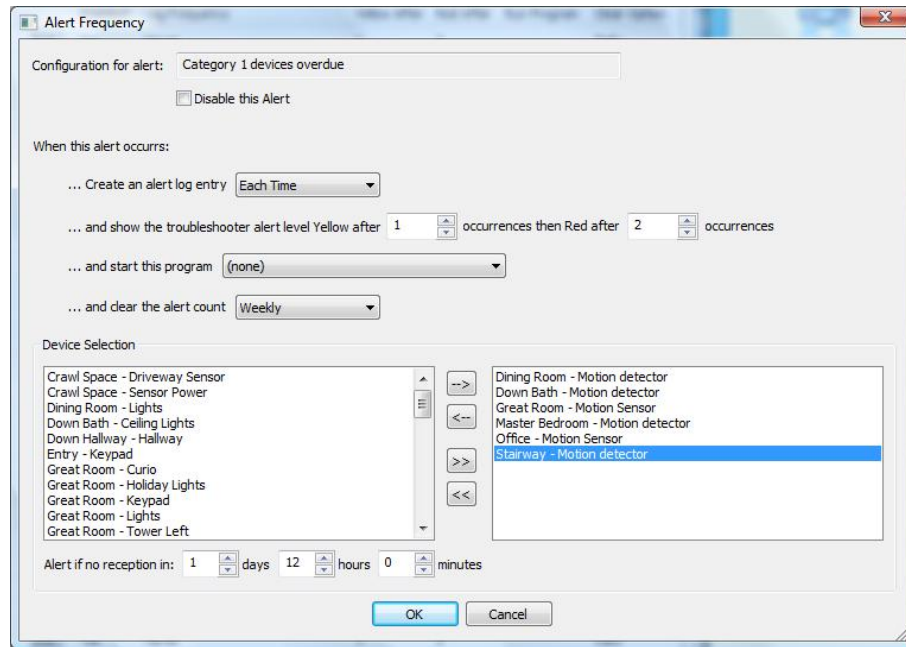
Like the Alert Viewer this shows all the alerts and information about their configuration. Press the *Edit* button to configure an alert.

## Device Overdue alerts

One of the most useful alerts and one that takes some additional configuration is the 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.

This alert can be very useful when working with battery powered devices to make sure that they don't "go dark" and are not heard from.

There are 4 separate alerts that you can use. Why four? The idea is that some devices you expect to be heard from daily, others weekly, and still others are a different time schedule. By sorting your devices into these different categories and configuring separate alerts for each category you can express the different "check in" times.



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

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.

- If program P makes an error it will log since it is the 1<sup>st</sup> for “P” today.
- If program P makes another error it will not log since it is the 2<sup>nd</sup> today
- 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.

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.

## Alert colors

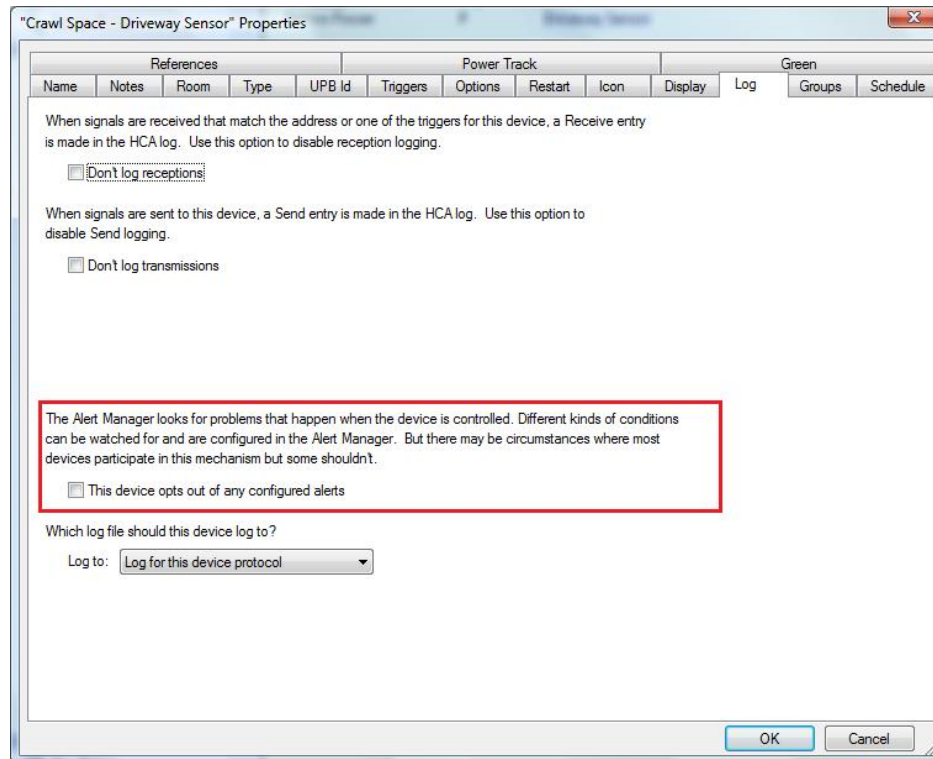
An alert that appears in the details log doesn't have - in itself - a red/yellow/green level. This is different than the static inspector which classifies all things it finds as red or yellow so each item listed in the inspector report does have a level for it.

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.

## Alert manager final issues

There are two issues related to your automation interfaces. Interface capabilities problems are reported as an inspector issue – for example needing a UPB interface but not having once configured. Issues with interfaces configured but not connected are now reported to as an Alert issue.

There is 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.




---

## The Log

Logging is a vitally important first step for troubleshooting because it helps you keep track of what's going on. You can also use the log to help diagnose odd behavior in your home.

The log keeps track of:

- when commands are sent and received
- which interface, if applicable, was associated with the message
- when errors are encountered
- start and stop times
- actions that programs take

As you can see, the log is a file that contains an entry for just about every action that HCA takes.

You can configure HCA to use a single log file, or you can configure up to three log files each assigned to log different actions.

There are two parts to each log. These are:

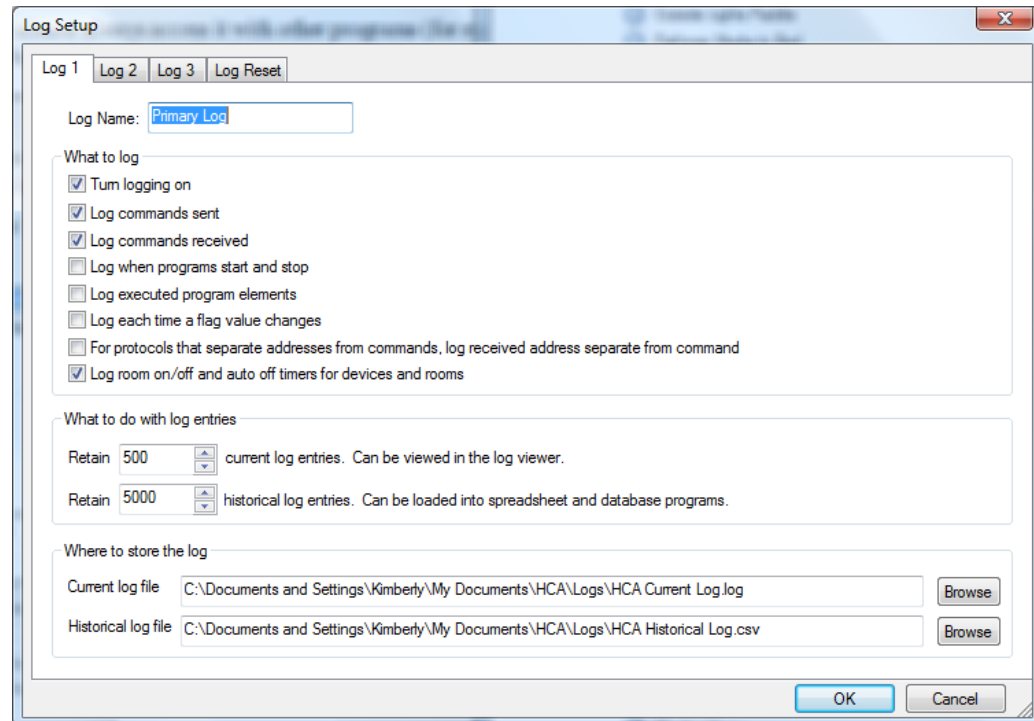
- The current log file. The contents of this file are stored in a proprietary format and its contents are viewed using the Log Viewer. HCA always has the current log file open.
- The historical log file is stored in a comma separated format and can be loaded into spreadsheet and/or database programs. HCA opens the historical log file only at

midnight so you can almost always access it with other programs (for example, Microsoft Excel or Microsoft Access) in order to view its contents.

## Log Setup

Before discussing how to view the contents of the log, the first steps are to establish where the log files reside, what information is placed in them, and how big the files can become.

These choices are made in the Log setup dialog. There are three tabs for log setup – one for each log.



Each log is given a default name “Log 1”, “Log 2”, “Log 3” but that isn’t very descriptive. It is a good idea if possible to name the logs somehow associated with their intended use. In the above example the log names are “Primary Log”, “Program Log”, and “Port Interface Log”. There are advantages in naming the logs with something descriptive. Each device and program and interface can specify where they log. The choices are made by the log name. Also the troubleshooter log viewer tab names use the log names you enter.

The next option in this tab, **Turn logging on**, controls whether or not logging takes place at all. Until this option is checked, nothing is placed in the log, regardless of the other settings in this tab.

### Where to store the log

Once you’ve decided that you want to log HCA activities, you need to provide the path of where the log files are to be placed. Use the Browse button to locate and specify the directory to use and the file name for each file.

### What to do with log entries

Associated with both log files are separate *retain* values. The operation of these two files and their *retain* values is as follows.

- At midnight of each day the number of entries contained in the current log file is compared to its retain setting. If there are fewer log entries than the setting, nothing more is done.

- If there are more entries than the retain setting, the oldest log entries are moved from the current log file to the end of the historical log file. The number of entries moved is the number necessary to bring the entry count of the current log file down to the retain setting. If there is no historical log file, the excess log entries are simply deleted.

Once the current log file has been pared down to its retain value, the same operation is performed on the historical log to reduce it, if necessary, to the number of entries in its retain value. Any excess entries removed are simply deleted.

HCA can operate without an historical log file. Without an historical log file, only those entries in the current log will be retained and only up to the limit specified in the “retain” value.

### **What to log**

The other options in this dialog allow you to determine what activities are entered in the log. If an option is not checked, that type of entry is not included in the log.

### **Log commands sent**

Check this box, and every time an On, Off, or Dim command is sent to your interface hardware, HCA enters it into the log.

### **Log commands received**

Check this box, and every time an command is received from your interface hardware, HCA enters it into the log.

### **Log when programs start and stop**

Check this box if you want to log whenever a program starts or stops. Unlike the "Log executed program elements" option, this option only adds log entries when a program starts running and when it stops. No other executed elements are logged.

### **Log executed program elements**

Check this box if you want to log program elements. This means that for every program that is started, as each element of the program is executed, HCA makes an entry in the log. You may find this useful for verifying new or modified programs perform as you expect them to.

### **Log each time a flag value changes**

Check this box if you want to log whenever one of your flags changes its value. Flags are described in the *Programs* chapter.

### **Log addresses received separate from commands**

Used only for older X10 commands. X10 commands are received in pairs with an address and a command. HCA can consolidate this into one entry in the log, for example “A1 ON”. If you would rather see a log entry for each reception, enable this option

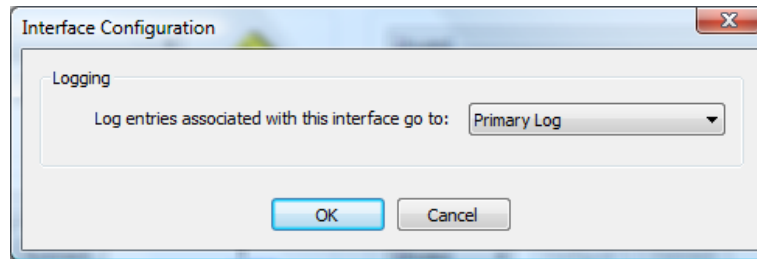
### **Log room on/off and auto off timers for devices and rooms**

The log can show when rooms are determined to be on and off and the action of auto off configuration for devices.

**Note:** When configuring each log the paths to the log files must be different from each other!

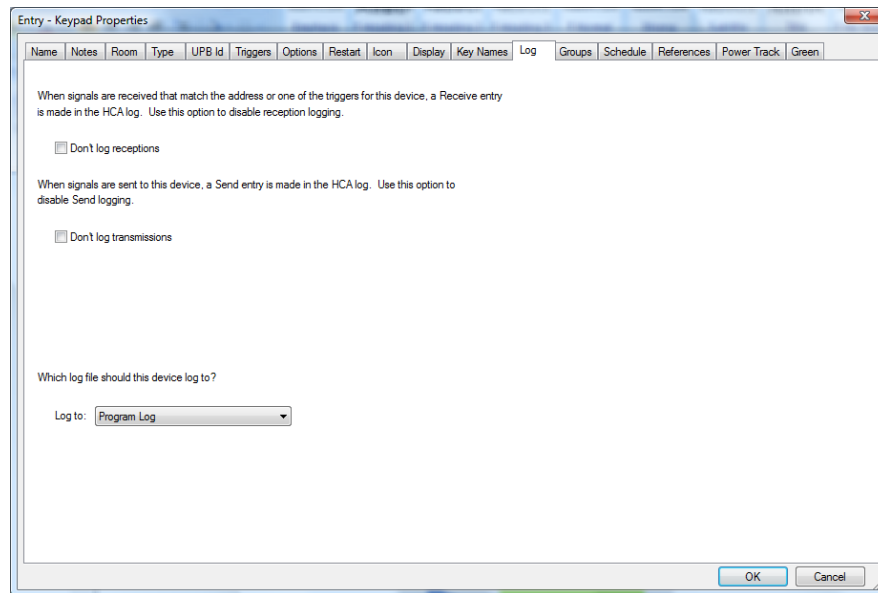
In the configuration for each interface – *HCA Options Hardware tab* – are logging options.

For example,



Using this, each interface can direct any log entries for it to a specific log.

In addition to this, any device or program can also direct log entries for it to their own log. This choice is made on the Log tab of a device, or program properties dialog.



For devices, in addition to choosing a specific log, you can also direct the device to log to the same log that is used by the device's interface.

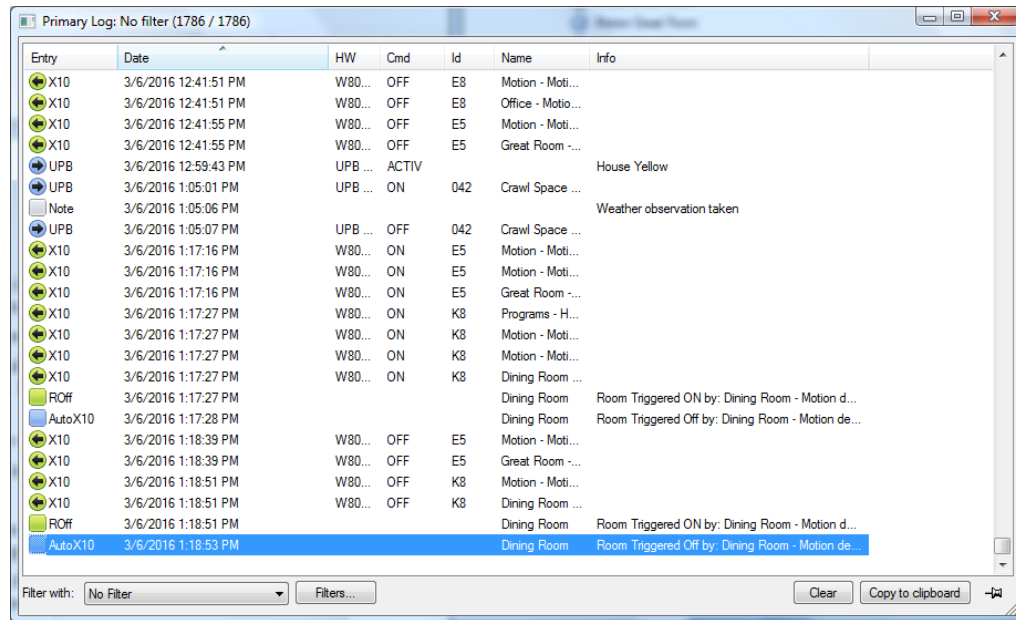
For example, if a UPB interface is configured to use Log2, and a UPB device is configured as *Log for this device protocol* then its entries go to Log2. If the UPB interface was subsequently changed to use Log3, then the device follows and now its entries go to Log3.

---

## Log Viewer

Each log has a viewer associated with it that can be opened from the log button in the ribbon Control category.





At first glance, the Log Viewer appears very complex but don't panic! The log viewer contains these items:

- A large display area lists each log entry—one per line. Each entry contains these items:
  - The type of entry
  - The date and time it was added to the log
  - The automation interface used if there is one. For an interface connected to a serial port, the port numbers is enclosed in ()'s
  - For Send and Receive entries, the type of command sent or received
  - The address or id or the transmitter or receiver. For X10, this is the house code (HC) and unit code (UC). For UPB, this is the unit id of the device, and for Insteon the address of the device.
  - The name of the device, program, group, or controller for this entry.
  - Any additional information or notes
- Along the bottom, an area labeled *Filter* provides options you can use to see just certain kinds of entries that the log contains.
- A *Clear* button. Click it to move the contents of the current log to the historical log and totally clear the current log file.
- A *Copy* button. Select a range of entries in the log then press the copy button. This places the log entries on the Windows clipboard. This is a very useful way to send sections of the log to Technical Support.
- A Push-pin that can stop or resume the log from updating while you have it open on the screen. When *pinned*, no new entries appear. When *un-pinned* any receptions, transmissions, programs, etc that are happening in the background log in real time as they occur. You may want to pin the log while you are looking though it, sorting, filtering, etc. When *un-pinned*, the log viewer jumps to the top or the bottom - depending upon the sort - when something in the background adds an entry while the log viewer is open.

There are different types of log entries that you may see. They appear in the Entry column. There are these general types

Entry type	Used for
Note	For general messages.
Send	For commands sent. There are many different sib-types of Send for the various protocols: X10, UPB, Insteon, Zwave, etc.
Receive (Rcvc)	For commands received. These are for commands whose source or target matches a device, group, or program in your design. Like Send there are different sub-types of Receive for the different protocols.
Program (Pgm)	For programs when they start, stop, and for each element while a program executes. There is also a sub-type used to show program errors.
Error	For problems.

For send and receive log entries, each interface type – Insteon, UPB, IR – has a different color and you can use the colors to quickly see where the messages are coming from if your automation solution integrates different protocols.

**Hint:** Remember that depending upon the configuration you can have a log record only a subset of these possibilities. For example, a log can be configured to save only entries about programs.

When HCA generates a log entry it checks each log configuration to see which log it should add it to. This is done in order and the 1<sup>st</sup> log that is configured for that type of entry receives it.

For example, if Log1 and Log2 – whatever their names – are both set to log program elements, Log1 will always receive those log entries since it is the first one checked.

There are several types of entries in the *Cmd* column. They show what type of command was sent or received. The Log Viewer shows these as:

Entry type	Used for
ON	On command
OFF	Off command
DIM	Dim command
BRT	Bright command
AUF	All Units Off command
ALO	All Lights On command
ALF	All Lights Off command
S-REQ	Status request
S-ON	Status On
S-OFF	Status Off
ADDR	Address
GOTO	UPB Goto command
FADE+	UPB Fade up command
FADE-	UPB Fade down command
FDST	UPB Fade stop command
ACTIV	UPB Activate command
DEACT	UPB Deactivate command
BLINK	UPB Blink command
IR	IR transmission

---

## Using the Log Viewer

In working with the Log Viewer, there are a few things you can do to make it easier to see and use the entries:

- You can resize each column (with the mouse pointer) by dragging the breaks between the column headers in either direction, to make the columns larger or smaller.
- Click a column header to sort that column. The first time you click the column it sorts in ascending order. If you click it again, it re-sorts into descending order. The sort column is shown with a gray background. The column header shows the sort direction with an upward pointing triangle (sort ascending) and a downward pointing triangle (sort descending).

---

## Log Filters

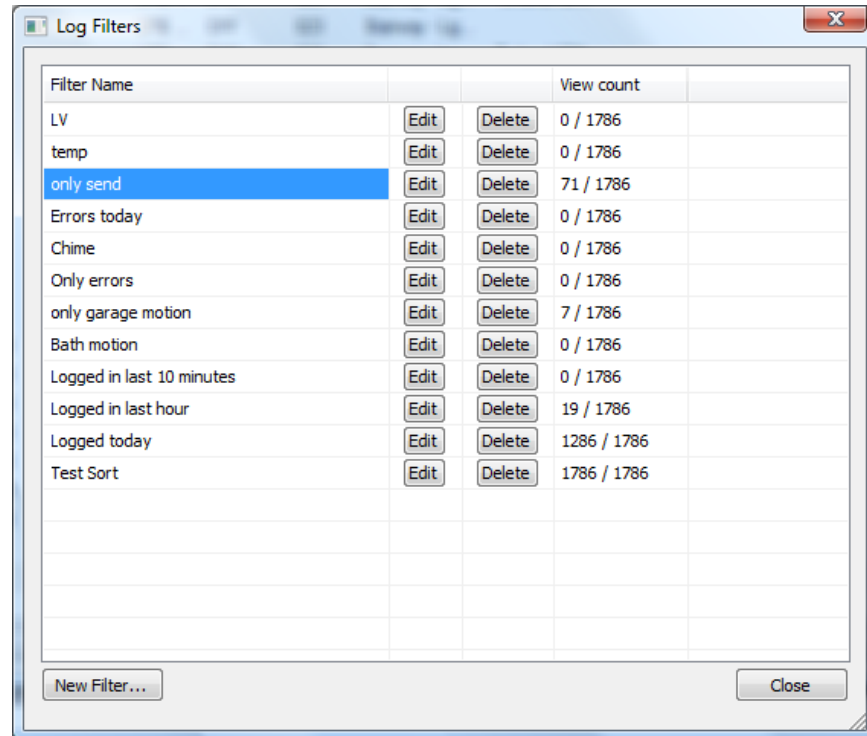
When viewing the log it is often difficult to focus on the specific data you want to see. The log filters allow you to select:

- Only a specific type of long entry such as Notes, Sends, Receives, etc.
- Only entries for selected devices, programs, groups, and controllers
- Only entries for a specific date, time, or range of dates and times.

In addition to selecting the type of entries to view, you can also specify the order these entries display in the log viewer.

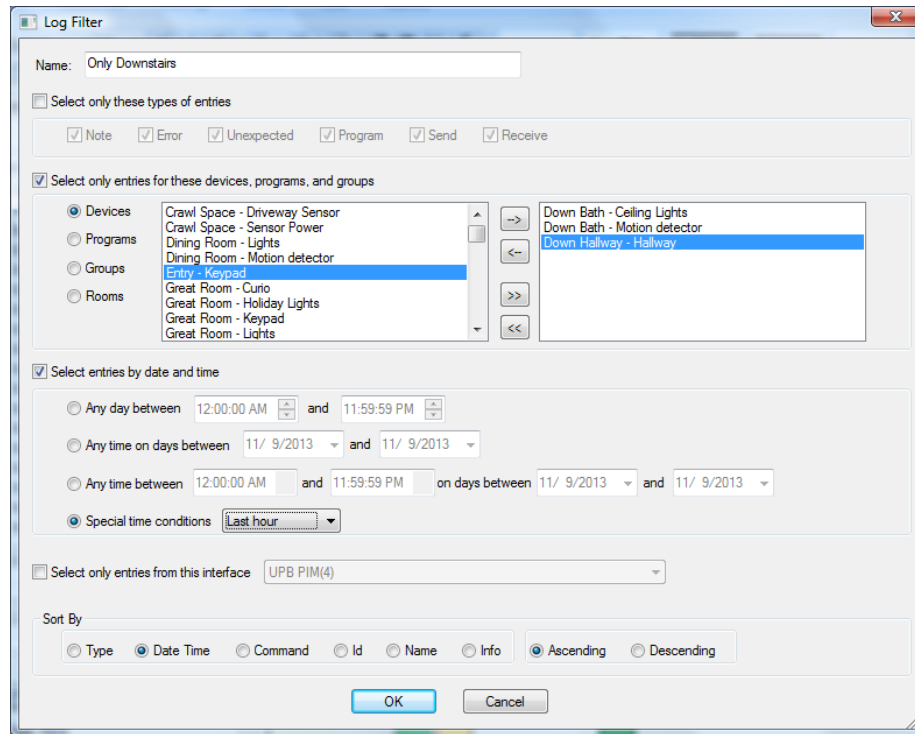
Each filter you create must be given a name. Once you have created a filter, it is saved with your design file.

To manage filters click on the Filters button at the bottom of the log viewer window. This opens the filter management window and in it list all the filters and their effects.



At a glance you can see which filters you have and how many entries would appear in the log viewer using each filter. From this dialog you can edit or delete the filter using those buttons. To edit a filter you can also double-click on the row for that filter.

To create a new filter, press the New Filter button. The filter creation dialog opens as:



The creation dialog for a filter is complex. It contains these sections.

#### Filter Name

Provide a descriptive name for the filter that you wish to create and save.

#### Select only these type of entries

If you want to limit the entries you see in the log display to only some kind of entries, check this option and then check the kind of entries you want to see.

#### Select entries for devices

If you want to limit the entries you see in the log display to only selected devices, programs, groups, and controllers, check this option. Then using the two lists move what you want to the right hand list. The option buttons on the left control what sort of things you see in the left hand list.

#### Select entries by date and time

To select entries created during a range of dates or times, check this option and then choose the selection type and the date/time range.

#### Select only entries from an interface

If you want to show only entries that are tagged with a specific interface.

#### Sort By

To control how the log entries are ordered in the log viewer when this filter is active, select the column to sort and in what direction.

In this example, a filter called *Downstairs only* is being created. Three devices are selected and only entries logged in the last hour are selected. When this filter is active the log viewer shows only entries for those three devices and only entries logged in the last hour.

**Note:** For an entry to pass the filter **all** the criterion must be met. Each log entry is checked and it must pass each selected criterion. In this example an entry must be made for one of the three devices and must be made in the last hour. There is no way to create a filter that passes entries that meet one set of criterion or another set of criterion.

---

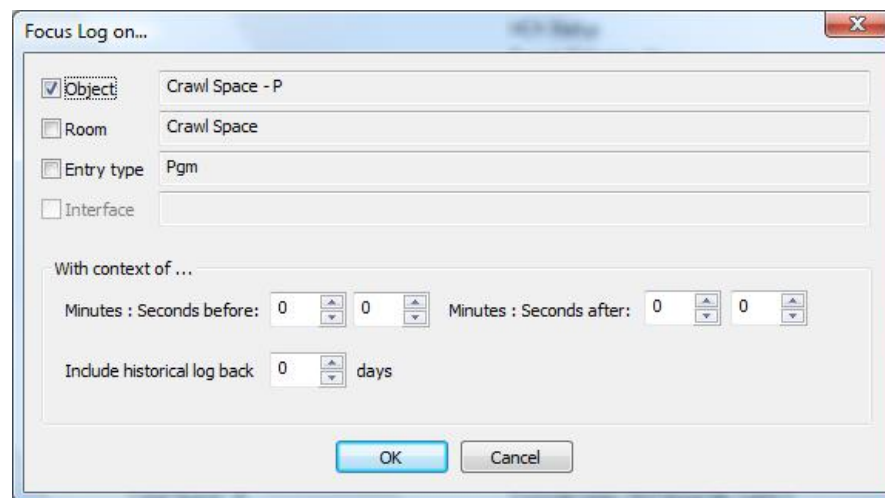
## Show Log

Device, Programs, Groups, and Controllers have a right-click popup menu pick called ShowLog. What this does is to create a filter for showing just that object (or objects if there is a multiple selection) and opens the log viewer with that filter. This allows you to quickly focus in on one object and see what the log contains for it. These special ShowLog filters are temporary and are deleted when the log viewer is closed.

---

## Log Focus

There is an additional tool in the log viewer to give you to method to locate useful information in tracking down problems. If you double-click on a line in the log viewer a “Focus” dialog opens.



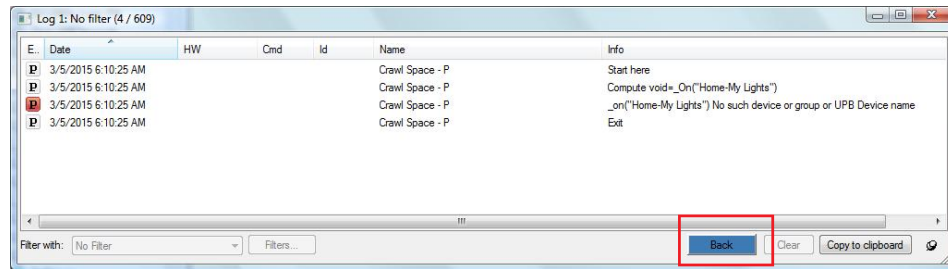
You can choose what you want to focus on. The log entry you double-clicked on pre-loads the details into this dialog and from the checkboxes you can specify what you want to focus on. Each box checked tells the focus to show only those log entries that have all of those properties.

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. In effect the settings in the top section of the dialog tell what you want to see and the settings in the bottom part of the dialog give you the log contents "around" what you want to see.

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 only 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.





## Chapter 20

### Restart

HCA is a program different than others that you may run on your computer. Unlike an email program or a word processor which you may start and exit many times a day, HCA is designed to be running almost all the time. Because of this when HCA is started and stopped, special things can happen.

Whenever HCA is started it attempts to pick up where it left off when it was shutdown. Of course, if it is starting for the very first time HCA has no prior state so it just begins. But from that point on when HCA is started, it is restarting from some prior shutdown.

This restart could happen after a power failure or after you selected Shutdown from the File menu

While restart after a normal shutdown will be explained, the most important use of this feature is to recover after a power failure.

Power failure detection and recovery is a very complex area. In general terms, the goal of power failure recovery is to restore the state of HCA and your home to what it would have been if the power had not failed.

Example: Suppose that HCA is active and the power fails at 7 p.m. When the power is restored at 8 p.m., HCA attempts to:

1. Make sure that lights that were on at 7 p.m. are turned on.
2. Make sure that any lights that should have come on between 7 p.m. and 8 p.m. also come on.
3. Make sure lights that were on at 7 p.m., and were scheduled to be off by 8 p.m. are not on.

Of course, more than just lights may be involved, and HCA checks and corrects status for **all** participating devices, programs, and groups. In fact, HCA will restart programs that were running when the power failed at the point they were interrupted. If the programs were executing delay or wait elements at that time, they are restarted and take into account the lost time due to the power failure.

This chapter discusses power failure and recovery, and includes:

- Types of power supply
  - No uninterruptible power supply
  - Using uninterruptible power supply
- Controlling restart
- Restart from a normal shutdown
- Some hints about power failure and recovery

---

### Types of power supply

HCA handles power failure recovery differently, depending on whether the computer is or is not connected to an Uninterruptible power supply (UPS). HCA has been designed to work with any UPS because it doesn't actually communicate with it.

---

## No uninterruptible power supply

If you are not using a UPS, and you enable power failure recovery, then while the power is on, HCA writes information to the disk about once a minute. This information includes the current time, state of all devices, and information on the state of any running programs.

When the power fails, the computer powers down and this stops HCA as well. When power returns, the computer restarts. Windows loads, and if HCA is in the Windows StartUp group, HCA restarts. If the HCA StartUp properties are set to reload the last design, HCA then performs three steps:

1. HCA determines that the power has failed by looking on the disk for the file that it has been writing its state to.
2. Once it finds that file, HCA should know the approximate time the power failed (to the nearest minute), and the state of all devices.
3. HCA looks at the current schedule and determines what would have happened during the time power was out, and restores all devices, programs, groups, and schedule entries.

UPB and Insteon Devices don't require HCA to keep track of their state. All these devices have their power restored to the state they were in when power was lost. HCA does have to restore them in accord with the current schedule like it does for all devices. This is described later in this chapter.

**Hint:** To add HCA to the Windows StartUp group, search for “StartUp” in Windows Help. Read and follow the directions in the topic, “Starting a program each time Windows starts.”

---

## Using an uninterruptible power supply

If your computer is attached to a UPS, nothing described above is done differently except that the state file may not be written until the house power has failed. Once the power returns, power failure recovery is as described above.

If, however, you have a UPS and the power fails but is not out long enough to drain the UPS batteries, HCA just needs to know the power has failed and when the power has returned. When power has been restored, HCA restores the state of your home.

None of this power failure recovery happens **unless**:

- The restart option is checked
- HCA is added to your Windows startup group
- The Startup option *Reopen the last design loaded* is checked

Additionally, you can independently set each device, program, and group to participate or not participate in power failure recovery. You do this in the device, program, and group properties, with the power failure recovery options.

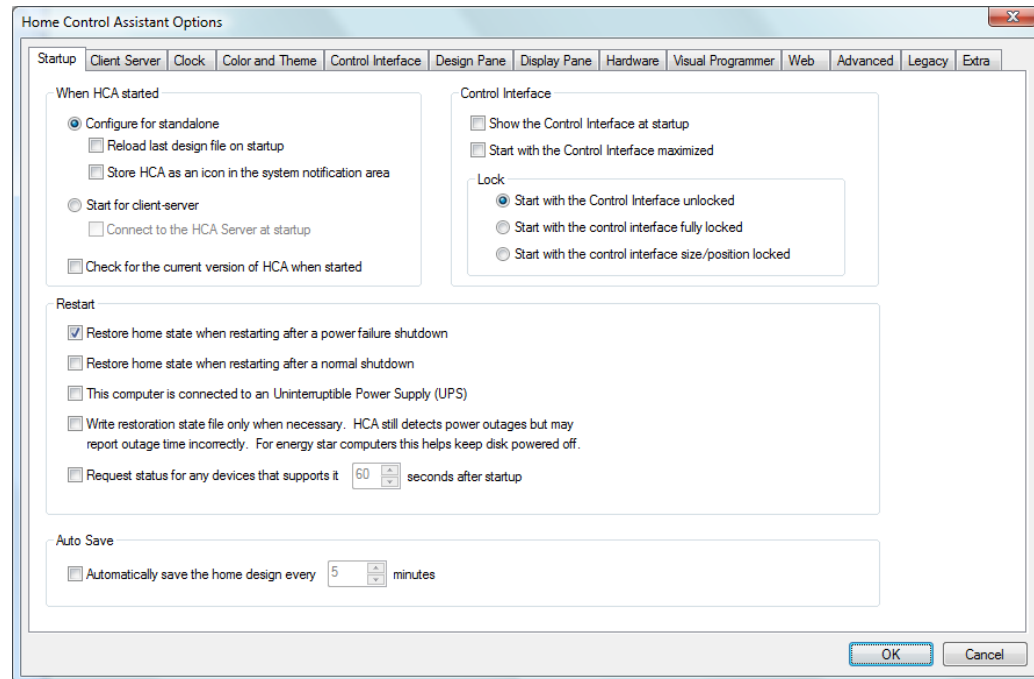
---

## HCA really doesn't know that the power has failed

The odd thing about all this “power out” talk is that HCA doesn't really know that the power has failed. What it does know is that one of its power line interfaces plugged into the wall is no longer communicating with the computer. If it stops communicating HCA reports this as a power failure and acts accordingly.

## Controlling restart

There are many different ways to control the actions of Restart. To start with, the whole feature can be enabled or disabled from the HCA Properties dialog on the Startup tab.



The first two options in the Restart group enable or disable the restart feature in the case of a normal or power failure restart.

The next option is for having a UPS as described above.

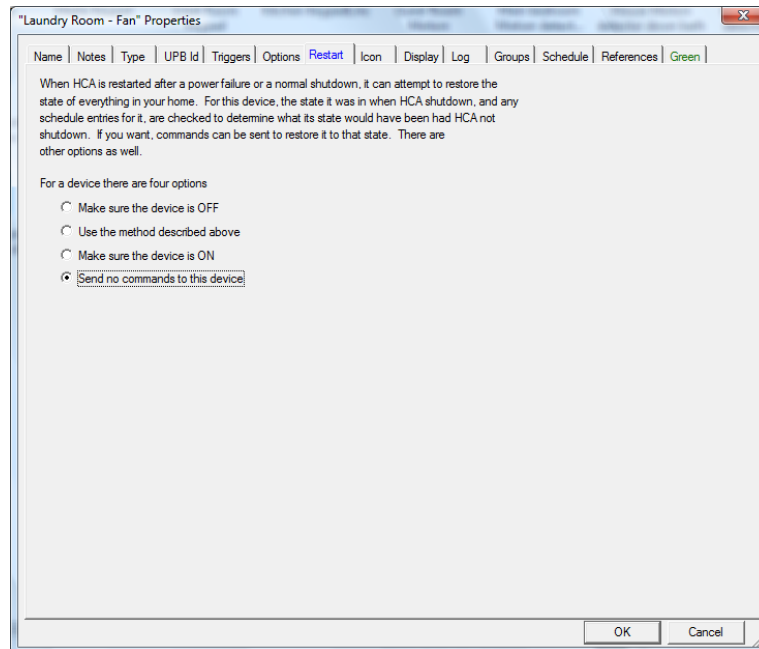
The next option in the section controls how often the state file is written. For computers that power off their hard drives after some period of inactivity, having HCA write the file once a minute defeats this. This option attempts to limit the writes to the state file to only when something has changed.

One important point you should be aware of if the state file write limit option is used. When HCA restarts after a power failure it shows the time and date of when the power failed and when it is restarted. If this option is used it may report an incorrect time for the power failure.

The final option of this section directs HCA that when started, to poll all devices that support status requests. This updates HCA's internal state to the device's state. It's best to wait a bit after startup before these polls happen so that they don't compete with any programs that run upon startup or HCA transmissions to restore state.

In addition to these options which turn on or off restart, you can also control restart on an individual device, program, or group basis.

For example, in the case of a device, these options are on the Restart tab of the device properties.



The restart options for the device allow you to control restart for just this device. Similar options are available from devices, programs and groups.

---

## Restart from a normal shutdown

Up until now all the discussion has focused on restart after a power failure. What is different when starting after a normal shutdown? Almost nothing. HCA still has a state file that is marked with the time of the shutdown. That state file contains the state of all devices, the current schedule, all the values of the flags, etc.

When HCA starts after a normal shutdown it proceeds as described above. That is, it catches up to where it would have been if it had not been shutdown.

Restart from a normal shutdown must be enabled as one of the options on the Startup tab as shown above.

---

## Some final information about power failure and recovery

Here is some additional information about restart

- **Restart is a very complex subject and this brief explanation is discussed in much more depth in a technical note on the support web site.**
- Power failure recovery is so complete that you may not notice that power has failed while you were out. Check the log—entries are made when power fails and when it is restored.
- If you are using an uninterruptible power supply, make sure that you power only the appropriate parts of the interface hardware from the UPS. Any interface that would plug into a wall outlet needs to be not plugged into the UPS.

- When HCA is restoring from a power failure, devices in your home are sent commands to control them. If your home contains devices where it doesn't make sense to set their level – like a device that makes noise – set their restart option to “send no commands to this device”.
- If you want to have a specific program run when a power outage is detected (of course, the computer must be powered by a UPS) or when power is restored, there are triggers for this. Add a trigger of the *Special Trigger* type and choose the trigger type you want.
- For restart to happen after a power failure your computer must start, log in, and start HCA. Use the web to find ways to automatically log in when the computer starts and how to automatically start a program when Windows starts.



## Chapter 21

### Calendar

The Calendar is a feature in HCA that lets you associate actions with different days of the year. This is done by assigning each day to one of eight different categories. For example, you may want all weekdays to be in one category, Saturday and Sunday to be in a different category, and all the days in August to be in a third category.

Each midnight HCA looks at how you have configured the calendar and if the category for the day just past is different than the category for the day just beginning then:

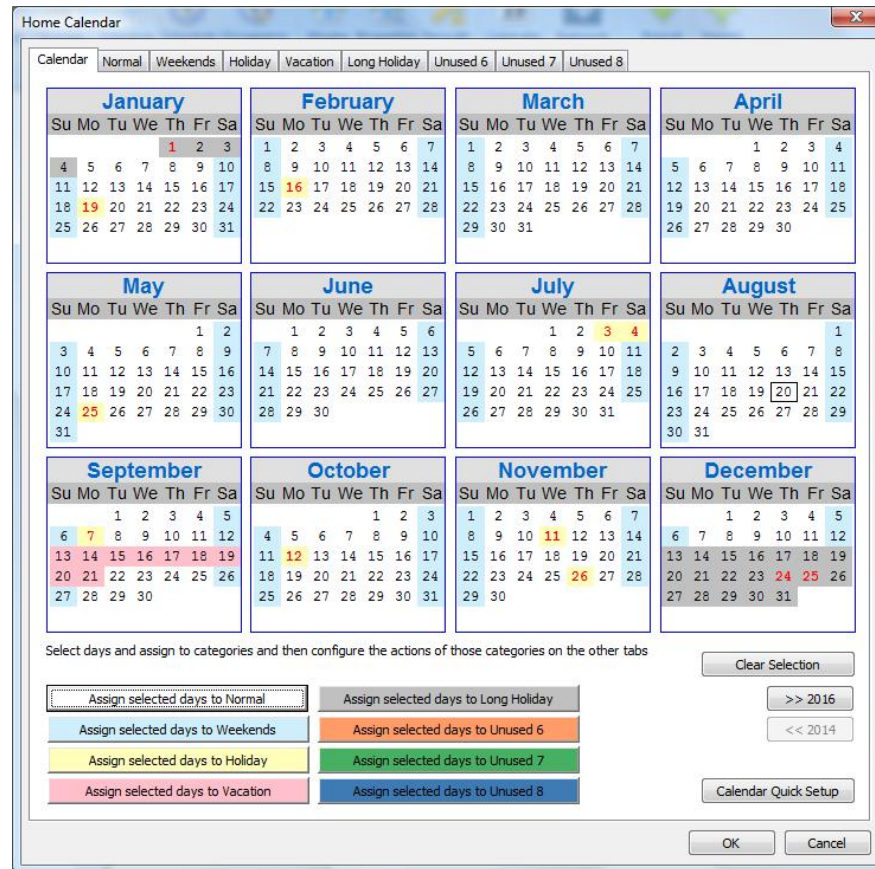
- A specified schedule can be made the current schedule
- A program can be executed when a category, based upon the calendar day, becomes the active category. That is, run the program upon entering the category
- A program can be executed when a category, based upon the calendar day, is about to no longer be the active category. That is, run a program just before leaving the category

*Why would you use the calendar?*

The calendar makes it simpler to do different things on select days of the year. Without the calendar you have to create programs that run everyday and then in the program test the day and see if that day is 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.

## Calendar display

To open the calendar, select the *Calendar* button in the *Design* ribbon category. The calendar opens:



The calendar covers 5 years and you can move from year to year by using the year buttons at the lower right of the dialog.

The calendar days are colored based upon the category that day belongs to. As an example, in the above calendar September is configured like this:

- Days 1-4 are in the Normal category
- Days 5-6 are in the Weekends category
- Day 7 is in the Holiday category
- Day 8-11 are in the Normal category
- Day 12 is in the Weekend category
- Days 13-21 are in the Vacation category
- Days 22-25 are in the Normal category
- Days 26-27 are in the Weekend category
- Days 28-30 are in the Normal category



Where do the names of the categories come from? Initially HCA assigns then simple names "Category 1", "Category 2", etc. When you configure the category then you can change its name to something that is useful in your design.

When you open the calendar most day numbers show in black text and some display in red text. The days in red are holidays as defined by the holiday file.

**Tip:** HCA ships with a holiday definition file that contains the observed dates for the US Federal holidays for the next 5 years. This file, called holidays.csv is in the HCA installation in the “Program” sub-folder. Its format is simple and can be edited if you want to add or change the shown holidays.

## Category Setup

To configure a category, click on the category tab in the top of the dialog. In the tab you can change the name of the category and determine the actions. In the example below, the "Special Weekends" schedule is made the current schedule on any days that are assigned to the "Weekends" category.

The screenshot shows the 'Home Calendar' dialog box with the 'Weekends' tab selected. The 'Category Name' is 'Weekends'. Below the name, there is a text box with the instruction: 'At midnight each day the calendar is checked to see what category the current day is in. Specify the actions that happen when the current day's category is chosen as category [Weekends]'. There are three checkboxes with associated dropdown menus:

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

In this next example, the program “Vacation Start” is run when the "Vacations" category becomes the active category. The program “Vacation End” is runs just before the active category is changed to a different category – not “Vacation”.

The screenshot shows the 'Home Calendar' dialog box with the 'Vacation' tab selected. The 'Category Name' is 'Vacation'. Below the name, there is a text box with the instruction: 'At midnight each day the calendar is checked to see what category the current day is in. Specify the actions that happen when the current day's category is chosen as category [Vacation]'. There are three checkboxes with associated dropdown menus:

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

## Pairing one or more days with a category

To assign one or more days to a category, click on the days in the calendar and those days are marked temporarily with a red background. In this example the last two weeks of July are selected.

May							June							July							August						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
					1	2	1	2	3	4	5	6			1	2	3	4							1		
3	4	5	6	7	8	9	7	8	9	10	11	12	13	5	6	7	8	9	10	11	2	3	4	5	6	7	8
10	11	12	13	14	15	16	14	15	16	17	18	19	20	12	13	14	15	16	17	18	9	10	11	12	13	14	15
17	18	19	20	21	22	23	21	22	23	24	25	26	27	19	20	21	22	23	24	25	16	17	18	19	20	21	22
24	25	26	27	28	29	30	28	29	30	26	27	28	29	30	31	23	24	25	26	27	28	29					
31																				30	31						

Then press one of the “Assign” buttons below the calendar display and the selected days are assigned to that category. You can see that has happened as the selected days are then displayed with the selected category background color.

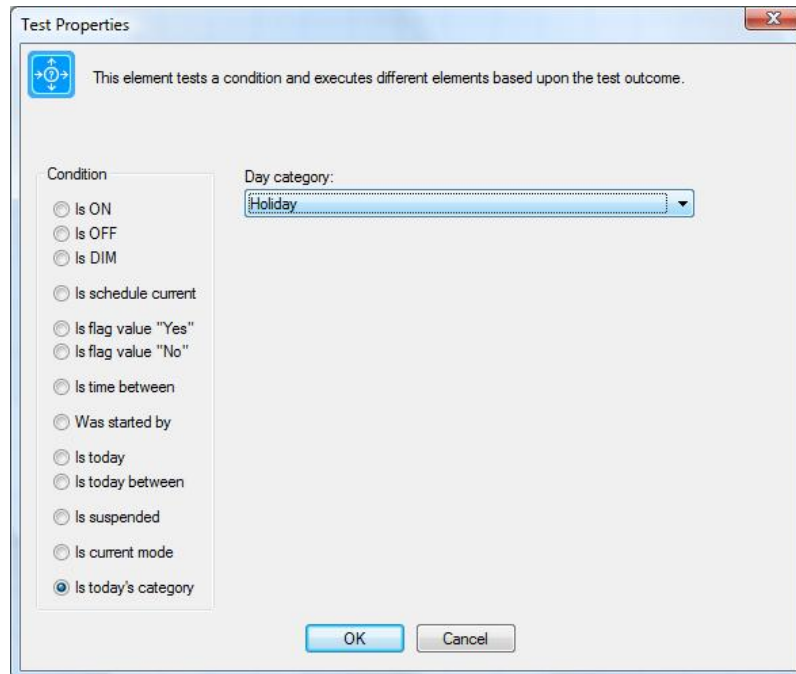
In this example, those two weeks that were selected in the image above were assigned to the “Vacation” schedule. As you can see in the next picture they now show in the “Vacation” schedule color – which in this example is gray.

May							June							July							August						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
					1	2	1	2	3	4	5	6			1	2	3	4							1		
3	4	5	6	7	8	9	7	8	9	10	11	12	13	5	6	7	8	9	10	11	2	3	4	5	6	7	8
10	11	12	13	14	15	16	14	15	16	17	18	19	20	12	13	14	15	16	17	18	9	10	11	12	13	14	15
17	18	19	20	21	22	23	21	22	23	24	25	26	27	19	20	21	22	23	24	25	16	17	18	19	20	21	22
24	25	26	27	28	29	30	28	29	30	26	27	28	29	30	31	23	24	25	26	27	28	29					
31																				30	31						

---

## Programs and the Calendar

If in a program you want to test to see if the current day is or isn't a selected category you can do this with the Test element.



When the program executes that contains this element, if the current day category is "Holiday" then the next element follows the "Yes" path, otherwise if the current day is a different category it follows the "No" path.



## Chapter 22

# Icon Themes

As you create devices, programs, and groups, you place icons for them on the displays shown in the display pane. HCA includes a varied selection of icons for the devices you may have in your home. However, HCA may not have anticipated all your icon needs. With Icon Themes, you can add additional icons that you have created.

You can draw icons using many popular graphics programs, including the good old Paint program that is part of your Windows installation. HCA can load icons from Windows bitmap files, those that end with a .BMP file extension, JPEGs, or PNG files that contain a transparency layers.

This chapter discusses the icon gallery and how to use it. Topics covered include:

- What are Icon themes
- Choosing a theme for a display and a default them for your design
- Adding icons to an existing theme
- Modifying theme icons
- Creating a new theme
- Important facts about icons

---

### What are icon themes

An icon theme is a set of icons that HCA gives a name to. When stored on the disk, a theme is a folder and the members of that theme are the bitmap files contained in that folder.

When HCA installs, in your documents area a folder called “Icon themes” is created as a sub-folder of “HCA”. In that folder are all the themes.

Each theme contains a number of icons. If you examine these folders you can see that they contain a number of image files and a file called Theme.txt. Here is theme.txt from the one theme:

```
[Metrics]
cxStandard=54
cyStandard=54
cxCell=108
cyCell=108
```

The cxStandard and cyStandard supplies the size, in pixels, of the width and height of each icon in this theme. The xcCell and cyCell supplies the spacing of icons on a HCA display showing icons that has the auto arrange option enabled.

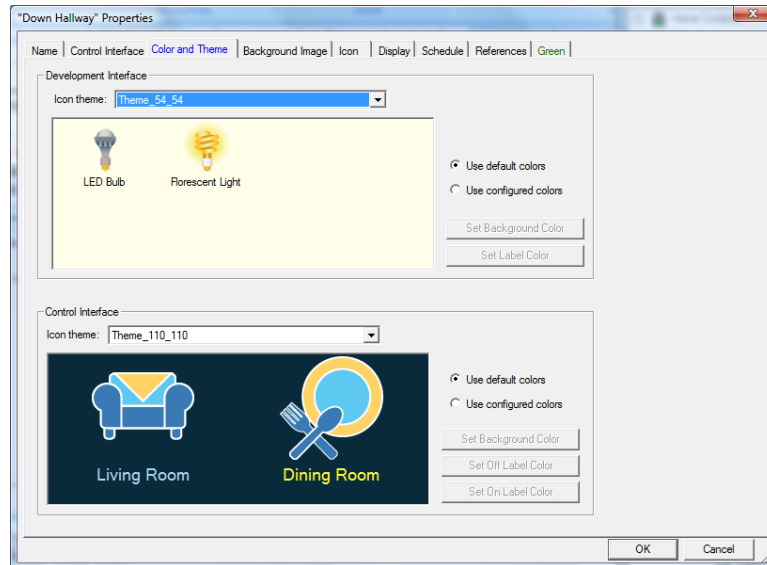
Each icon theme has a different theme.txt file and as such can be a set of icons of whatever size you need. The only requirement is that all icons in a theme are the same size.

Within the theme folder the filenames of the bitmap files are the names that you work with in HCA. For example, if there is an image file called “Table Lamp.png”, then in HCA where you select an icon for a device, program, or group, you will see the name “Table Lamp”.

If an icon has different representations based upon state – on, off, or dim – the image file is suffixed with “\_on“, “\_off”, or “\_dim

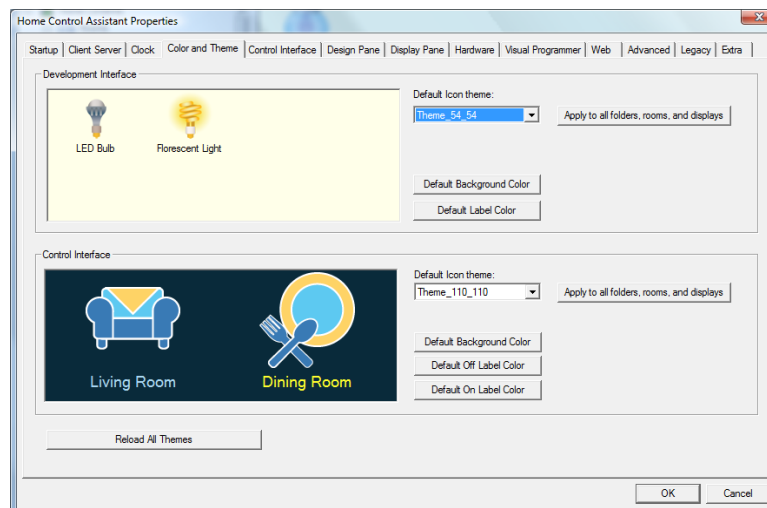
## Choosing a theme for a display and a default theme for your design

Each folder in HCA shows icons from a selected theme. To select the theme for a display, open the properties for the display and choose the Color and Theme tab.



All you need to do to change the theme for this display is to select a different theme name.

To change the default theme used by HCA for all new displays created, select from *HCA Options* the Color and Theme tab:



In addition to selecting the theme for all new displays, you can apply the theme to all existing displays using the *Apply to all folder, rooms, and displays* button.

---

## Adding icons to an existing theme

To add a new icon to a theme is very easy. Create the image files with appropriate names and then put them into the theme folder.

To have HCA detect the new icons, either restart HCA or press the “Reload All Themes” button on the HCA Options tab shown above.

---

## Modifying theme icons

To change icons in a theme just edit the image files with your changes.

To have HCA detect the changes, either restart HCA or press the “Reload All Themes” button on the HCA – Properties Display Pane tab.

---

## Creating a new theme

To add a new theme is also very easy. Follow these steps.

- Create the image files for your theme with the names you want.
- Create a new sub-folder in the Icon Themes folder with the name of the theme you want and place all your bitmap files in that folder.
- Create a Theme.txt file as described above.

To have HCA detect the changes, either restart HCA or press the “Reload All Themes” button on the HCA – Properties Display Pane tab.

---

## Important facts about creating icons

When you create your own icons for HCA, there are a few things you will want to remember, both about creating the icons for use in your HCA design.

- All icons in a theme must be the same size and that size must match what is specified in theme.txt. If an image file is read that has a smaller or larger size it is forced to that size – either cropping the bitmap or expanding it.
- If your display contains a DXF floor plan, as the HCA window is resized, the floor plan and any icons on that floor plan are also scaled. Because of this scaling, you should not draw icons containing very fine detail or text—these features don’t scale well, and the icons may not look the way you want them to in some sizes.
- When you select an icon during creation of a device, program, or group, HCA uses the default theme for the design to show the icon choices.
- Not all themes need contain the same set of icons. Because each display may have a different theme there may be times where an icon is looked for in a theme and not found. When this happens a substitute icon is used. The design inspector looks for this sort of them and makes notes on that.





## Chapter 23

# Control User Interface

The Control Interface – sometimes called the “Touch Screen Interface” even though it works well on non-touch screens - is an alternative user interface that can be used to control HCA. Instead of the standard HCA User Interface – sometimes called the Development User Interface - which looks like many other Windows programs with a ribbon at the top and status bar at the bottom, the Control User Interface looks completely different.

Aside from appearance, the major difference between the Control Interface and the Development Interface is that the Control Interface is only designed for viewing and controlling – but not modifying - devices in your home.

Using the Control Interface you can view your home, control devices, and start and stop programs. You can't add or delete devices, change schedules or programs, or modify any elements of your design.

If you don't have an actual Touch Screen you may be tempted to ignore this feature. Please don't! The Control Interface is a simple way for everyone in your home to work with HCA and not to “break” anything by inadvertently deleting devices or modifying schedules.

Also, using the Client-Server features of HCA, having a client configured to use the Control Interface is a very handy way to have a “public” machine in your home accessible to everyone.

---

### What does the Control UI look like?

Before getting to far into discussion of the Control Interface features, let's first look at it to get a general idea of its concepts.



Yes, believe it or not, this is produced by the same HCA.exe as you already use. To engage the Control Interface, press the *Control Interface* button in the ribbon *Home* category.

The display first shown when starting the control interface is called the “*Home*” page.

A few things you probably already noticed. There is no menu bar at the top or status bar at the bottom of the window. The standard Windows features for minimize, maximize, and exit are all there on the left right, they are just very large and look a bit differently.

The top pane of the Window - called the “status bar” - shows, from left to right:

- The HCA icon showing the home mode. When the gear within icon is mostly yellow then the mode is “Home & Awake”. When it is mostly green then the mode is “Home & Asleep”, and when it is mostly gray then the mode is “Away”.

You can click on this icon to get a popup where you can shutdown HCA, return to the development interface and perform a few other actions. These are covered later.

- A Back arrow: Using this will bring you back to the home page from any room page. When the arrow is gray it is inactive.
- Up and Down arrows. Due to the difficulty of using scroll bars on touch screens, they are not used to move the display up or down. The up and down arrows lets you scroll up and down if there are more icons than can fit in the window. The arrows are gray when inactive.
- The text sections, the name and time in the above example, are configurable.
- A minimize button. This minimizes to the Windows task bar.
- A maximize button. This *really* maximizes the display to cover the entire surface of the monitor including the task bar.
- The background color of the status bar is the alert state of HCA – like the red, yellow, or green flag shown in the ribbon troubleshooter panel.

You can move the control interface window by a left-click drag action on the status bar background.

## Configuring the Control Interface pages

The pages the Control Interface displays can be created in one of two ways:

- HCA creates a default page structure from the rooms in your design
- You can completely control the pages by creating displays and placing on those displays the icons for devices, programs, groups, and displays you want to appear.

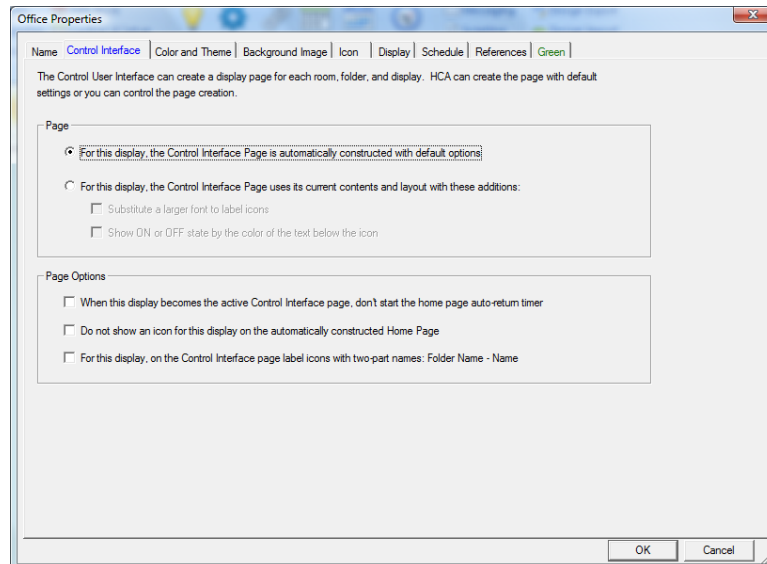
The section covers both approaches.

### HCA created pages

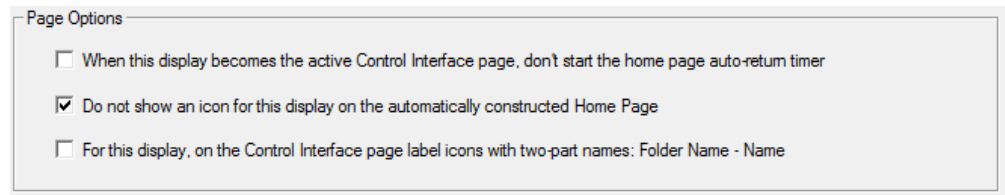
The Control Interface has the same concept of “rooms” and “folders” as used to organize your design. If HCA constructs the Control Interface home page, an icon is placed on the page for each room or folder in your design.

Also, you can create displays as described in the Displays chapter. These displays can also show as icons on the Control Interface Home page.

Rather than show icons for all rooms, folders, and displays, you can configure the Control Interface so that only selected rooms, folders, and display are shown as icons on the Home page. There are properties of each room and display that controls this.



This is the option that tells HCA if this folder/display automatically gets an icon on the home page when HCA builds the home page:



Other options selected in the *Page* section of this tab control the appearance of the page. There are two possibilities - if the constructed page should use all the default settings for colors and theme or not. If not, then you can select if the page uses a slightly larger font and also if it uses a different mechanism to show the state of devices.

---

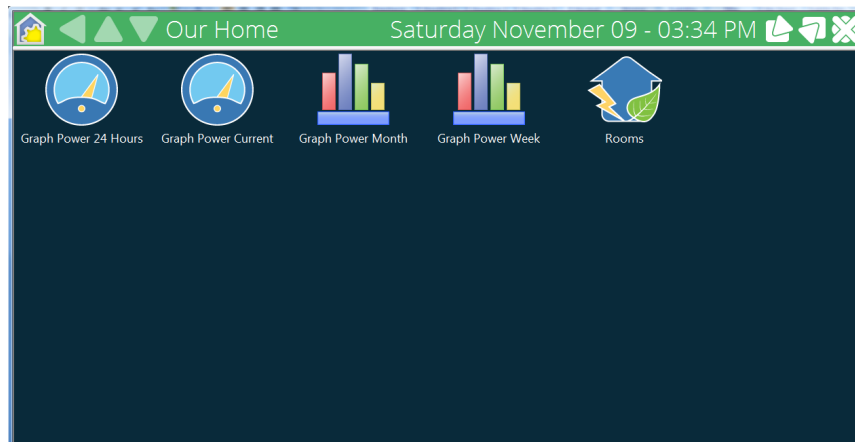
## Custom pages

The other option is to create a display and designate that display as the Control Interface home page. On that page you can place icons for those devices, programs, groups, displays, and folders you want to see and in the order you want them to appear in.

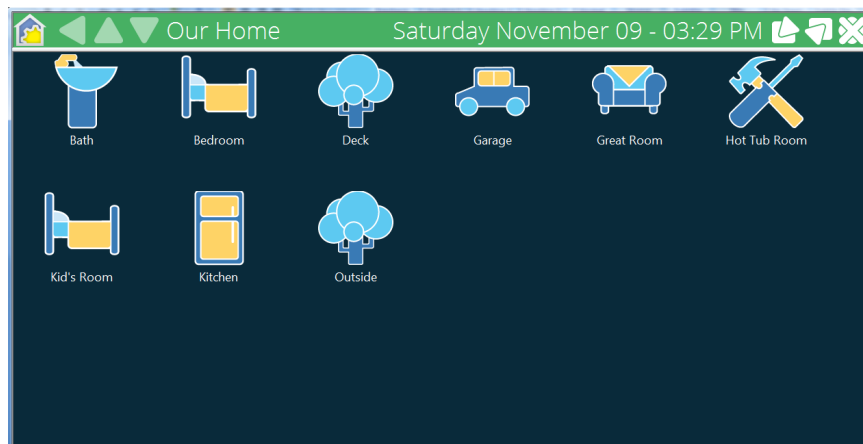
Using this option you can create pages for each level of the control interface structure you want.

The example used for the screen images in this chapter uses a custom constructed home page that contains icons for displays and for some frequently used devices. Some of those displays are to access Power Track graphs and other for HTML displays. One display – the icon labeled *Rooms* – opens to another page that contains an icon for each room in the home.

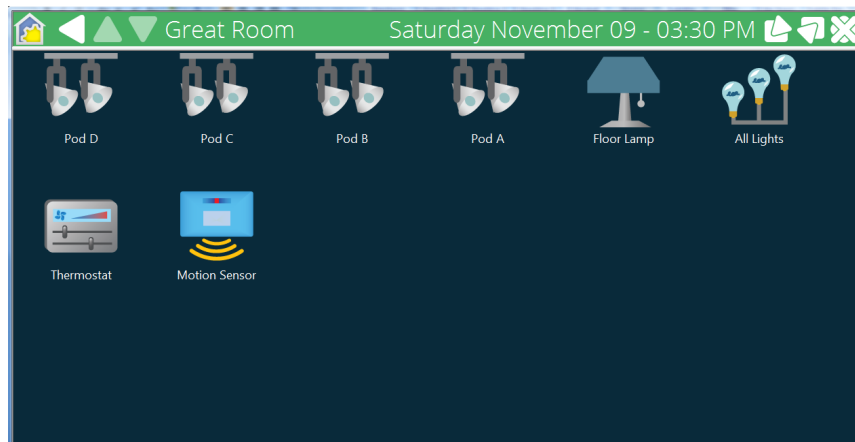
For example, this is the home page:



A short tap on the *Rooms* icon goes to this page:

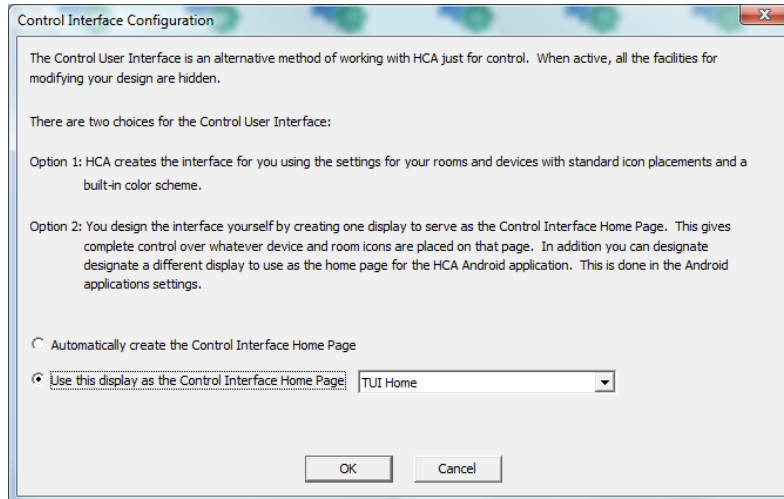


A short tap on the *Great Room* icon goes to this page:



The back button from this page goes back to the *Rooms* page and back from there to the Home page.

The last piece of this puzzle is how to designate what display to use as the Control Interface home page. Press the *Select Home Page* button in the ribbon *Design Tools* category.



Think of the Control Interface as a programmable user interface. It is worth spending some time to select which of your folders and displays become pages in the Control Interface, which devices, programs, and groups are shown on the pages, and what the popup control panel shows for each device. Like designing your automation solution, it makes sense to spend time crafting the user interface to work with it.

---

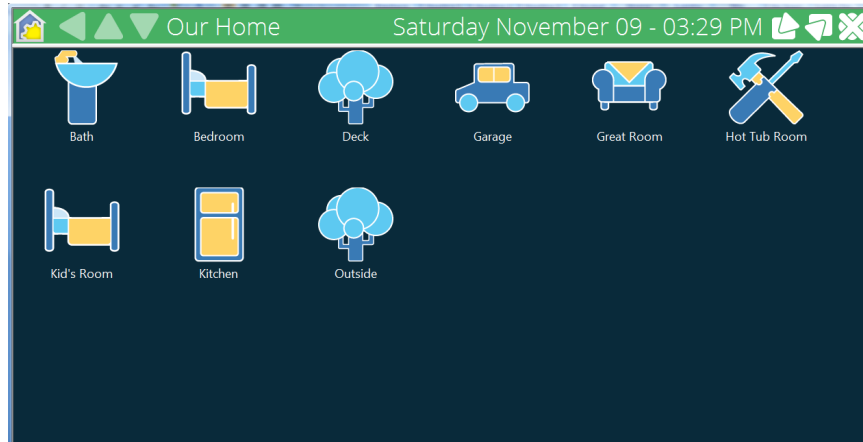
## Controlling devices

Unlike the Development UI, where you single-click a device to select it, double-click to act upon it, and right-click to open a context menu, the Control Interface is all single click – or “tap” in touch screen terminology. This “tap” is further divided into short taps and long taps.

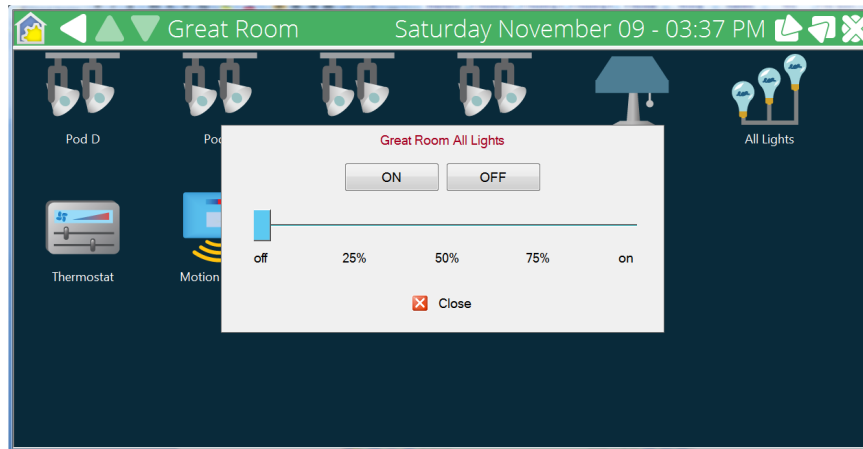
A **short tap** – like a mouse click – acts upon what you tapped on. If it is a room icon then the room is opened. The default action is that when you tap on a device it is turned on if it is off, and off if it is on – it toggles the state. The default action for a **long tap** opens up a control panel for the device.

A Short tap is a quick touch then release. A Long Tap is a press-down and hold until the popup appears then release. You can do these taps with a mouse. A short tap is just a left mouse click. A long tap is made by depressing the left mouse button and holding it until the popup appears then releasing. A long tap can also be made by double-clicking on an icon with a mouse.

For example, short tap on the Garage icon from the home page and a new page appears



If you long tap on the Library Bar Light icon then the control panel for this light appears:



From this control panel popup you can turn the device on, off, or change its dim level.

Long tap on a keypad icon and this popup appears:



This shows one of the next concepts in the Control Interface to explore. These types of popups are called “glass keypads” and are described in the next section.

---

## Glass Keypads

Glass keypads are a concept in the Control Interface that makes it possible to “push” a button on a keypad or “tap” the paddle of a switch. What this all means is best described by an example.

You probably have a number of keypads in your home that do various things. In this example a keypad in the bedroom has multiple functions. When you are actually in the bedroom you can walk up and poke one of the buttons. What that button does is really up to your design. It could control one or more devices directly using Insteon scenes or UPB links. Or it could trigger an HCA program.

Using the Control Interface you can *cause whatever that keypad does when you poke the button on the real keypad to happen by pushing the button on the glass keypad*. In this example, the *Go To Bed* button controls many devices in the home by a scene. When you tap on the *Go to bed* button in the glass keypad it will have the same effects as if you walked up to the actual keypad and poked the *Go To Bed* button.

**Hint:** Where did the text on the buttons come from? In the device properties dialog for the keypad is a tab where you can enter names for the transmit components. That text is used in the Control Interface glass keypads.

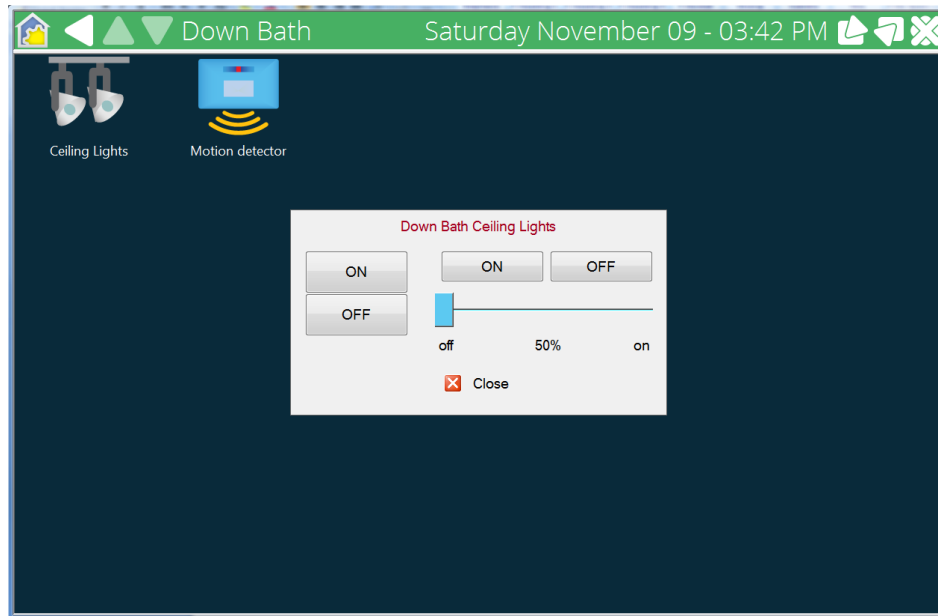
But “glass keypads” aren’t just for keypads! Anything that has a transmit function that can be invoked from a physical tap on it can have a glass keypad.

For example, switches have a single rocker but you may also want the Control Interface to show a glass keypad for it. The idea is that with UPB and Insteon it is not uncommon to have scenes programmed into a switch so that when you tap the paddle it not only controls the load but also controls other linked devices.



But what if you want to just control the load attached to the switch? What if the switch paddle sends a UPB or Insteon scene that controls other devices but all you want is to control just the load connected to the switch? From the Control Interface you can do that as well.

Take a look at this example:

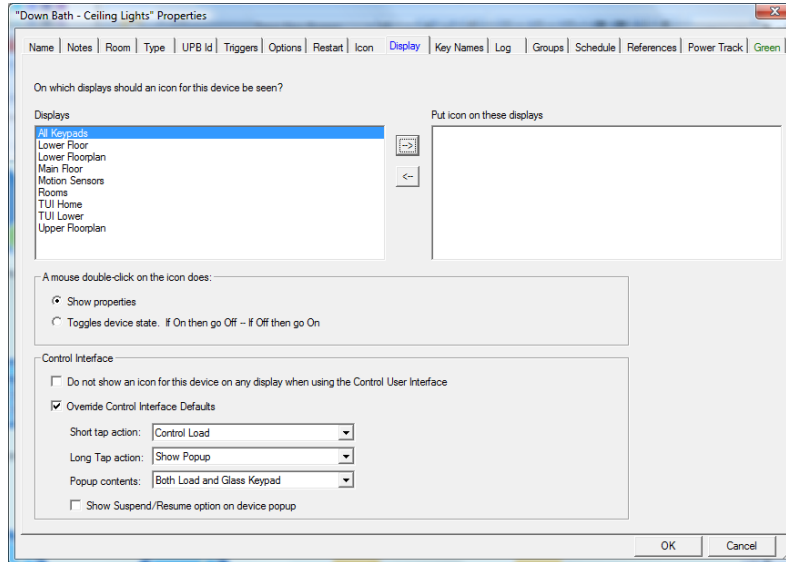


This example is from a long tap on the Down Bath Ceiling Light icon. The popup control panel has both the glass keypad – the bigger on/off buttons at the left - and also the on/off/dim controls for the load at the right. You can control just the load by using the controls on the right or you can “tap” the switch paddle using the controls on the left.

**Hint:** If the switch doesn’t transmit anything when you use the switch paddle – that is you don’t have a UPB or Insteon scene programmed into the switch – then using the left or right controls have the same effect.

But for some devices you may not want a glass keypad but only the load controls. Or sometimes you want to see only the glass keypad and no load controls. And finally, sometimes you want both the load controls and they glass keypad. The Control Interface is configurable for this in this manner.

Here are the properties of the Down Bath - Ceiling Lights device used in this example.

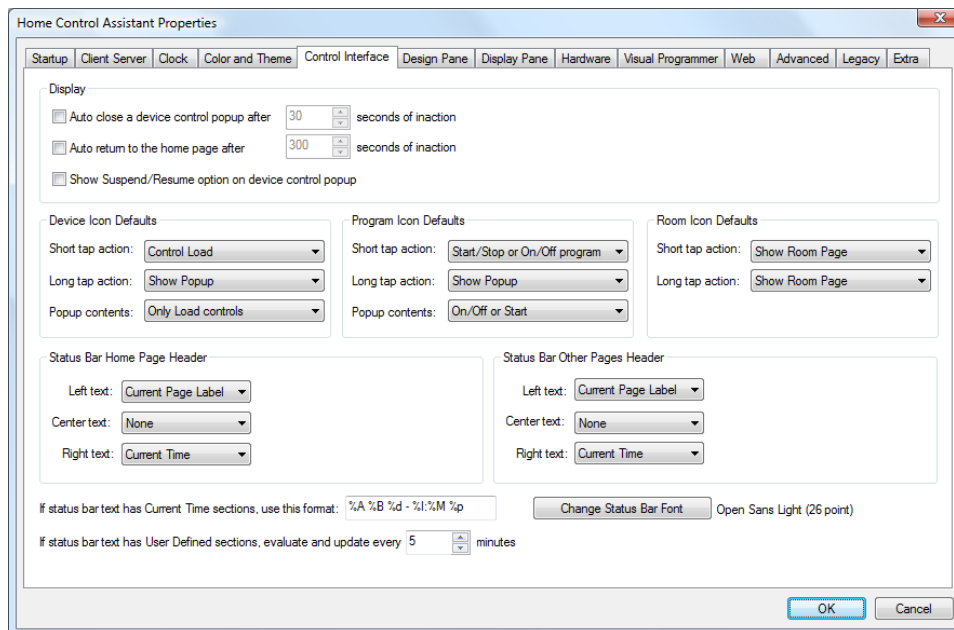


As you can see in this dialog you can configure all aspects of the Control Interface for this device – what happens on a short and long tap and what appears in the popup for the device.

Since configuring all your devices would be time consuming – even with multi-select edit – HCA has defaults you can set and then for those devices that you choose to not follow the defaults you can individually configure. These defaults are set in the HCA Options dialog on the Control Interface tab.

## HCA Options Control Interface Tab

When the *HCA Options* dialog is opened, choose the Control Interface tab.



As you can see from the above picture, this is where you select the default actions for short and long taps as well as the control panel popup contents. The settings can be different for devices and programs.

**Hint:** You can open this tab of HCA Options by pressing the *UI Options* button on the *Design Tools* category

Also on this page are additional configuration options. The Control Interface can automatically close an open popup after some set inactivity time as well as return to the home page from a room display after some period of inactivity.

The status bar at the top of the Control Interface by default shows the design name or room label at the left and the time at the right. Using the options in this tab you can change these defaults. You can also use custom text that is the result of evaluating HCA expressions. Finally, you can also control the time/date format and the status bar font used.

Another of the options is to show or not show a Suspend button on popups. The Suspend / Resume operation is very useful but can be confusing to those not well versed in HCA. Also you may want to prohibit users for overriding the schedule for a device by using Suspend. You can set the default for this option on the HCA – Properties tab but also control it individually for each device and program from the device properties dialog.

When available, the Suspend button shows this popup optimized for each use on a touch screen

Suspend for this time period

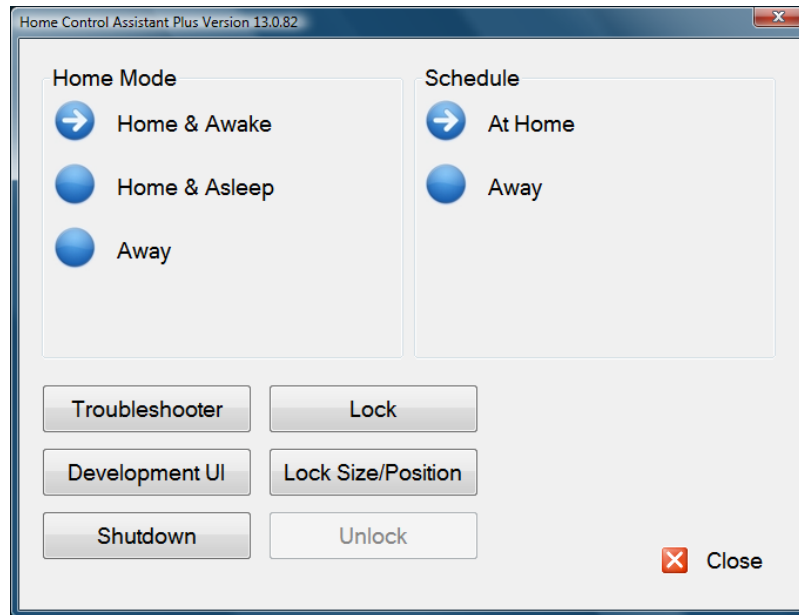
1 days   0 hours   0 minutes

Or check here to suspend until manually cleared

---

## Control Panel

As described above, you can tap on the HCA Icon at the left end of the Control Interface status bar. This popup appears:



The first two sections on this popup are for changing the schedule and home mode. The current schedule and mode are shown by the arrow inside the button.

The *Development UI* button returns to the Development User Interface.

The *Lock* button prevents the HCA Application Window from being moved or resized. Use the *Unlock* button to restore the window to movable and resizable.

The *Lock Size/Position* button is like the *Lock* button but leave the windows minimize button available.

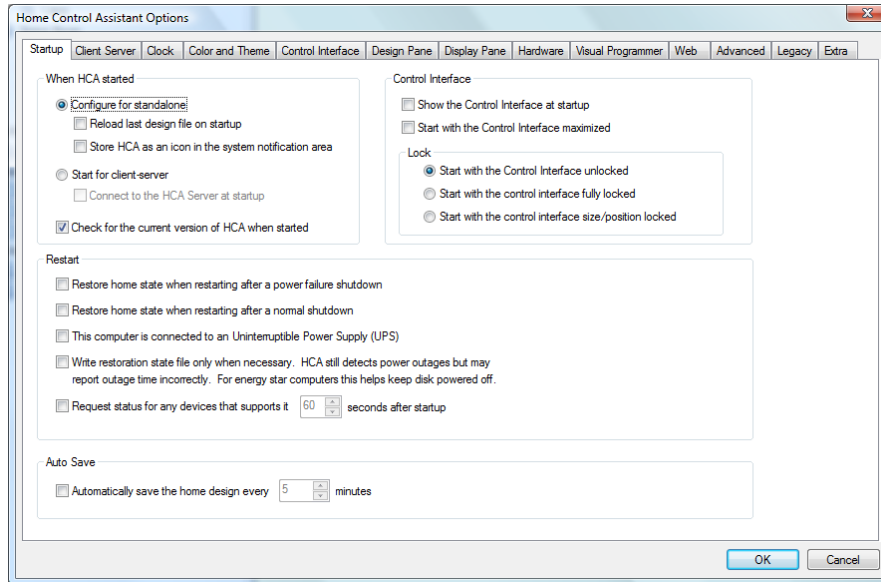
The function of the *Shutdown* and *Troubleshooter* buttons should be obvious.

---

## Configuring the Control Interface for dedicated machines

There are a number of facilities that, working together, make it possible to configure the Control Interface so that it creates a Kiosk style of operation. That is, it makes the computer appear that only HCA with the Control Interface is running.

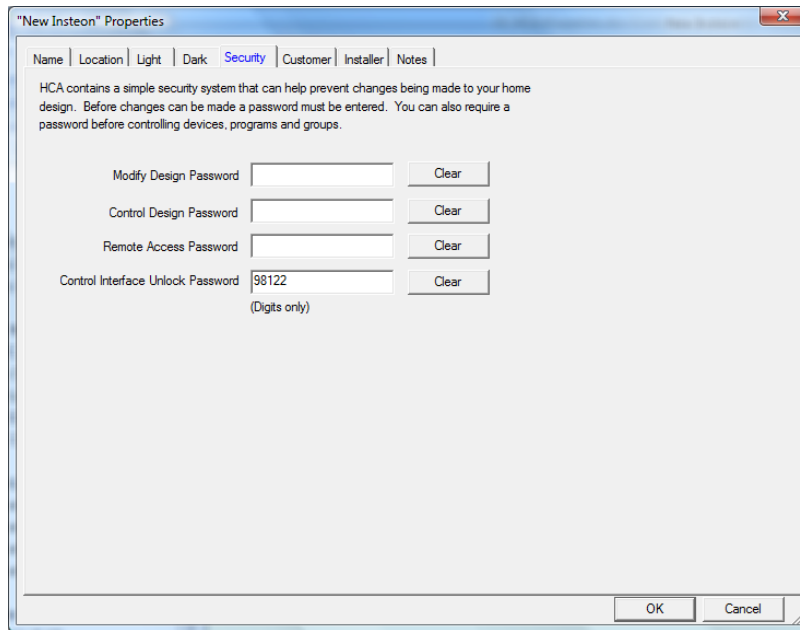
The first place to start is the HCA Options Startup tab.



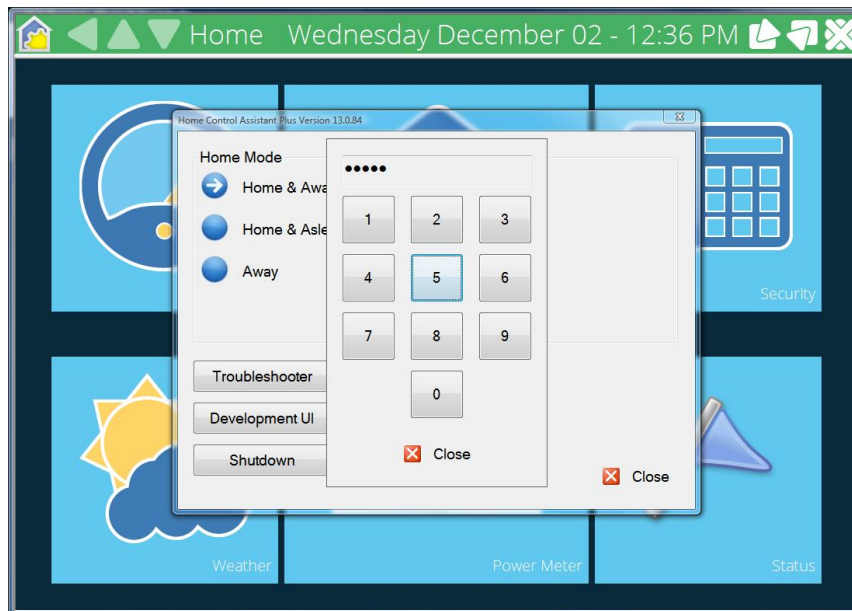
The three options in the Control Interface box are:

- **Show the Control Interface at Startup.**  
When this option is enabled and when a file is loaded at startup, then the Control Interface is immediately shown.
- **Start with the Control Interface Maximized.**  
When this option is enabled and when a file is loaded at startup, then HCA takes over the complete Windows display. This is the same maximize option when using the status bar maximize option. That is, it covers the whole screen – Windows Task Bar included.
- **Lock options**  
These options are similar to those available in the Control UI control panel as described above.

In addition to these options, you can also assign a lock password to the Control Interface. This code must be entered to unlock the Control Interface. You can assign this password by entering it on the Home Properties dialog security tab.

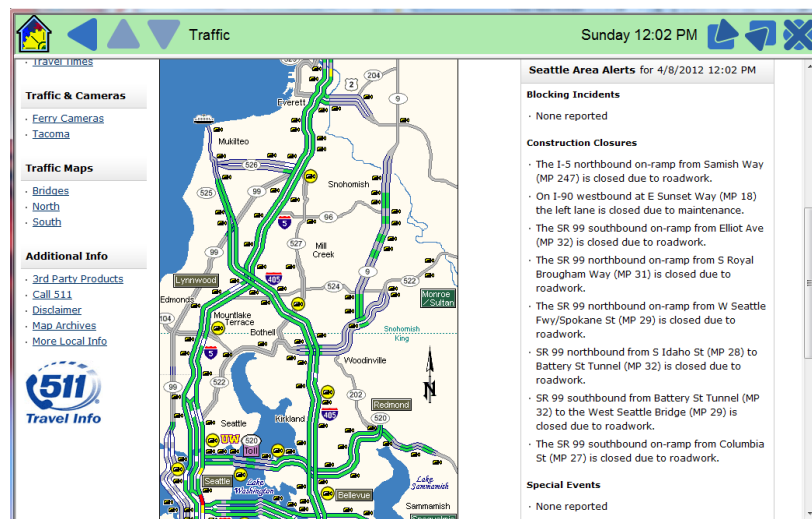
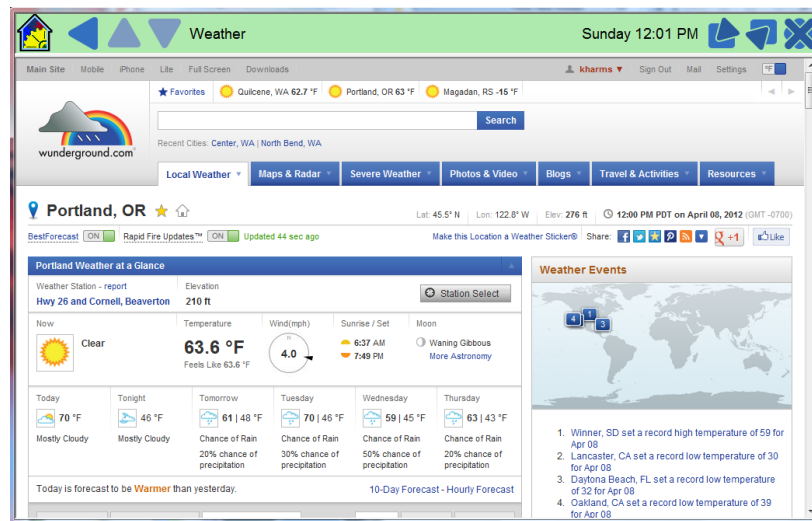


Note that the unlock password must only be numeric digits. This is because the unlock password is entered using an on-screen keypad:



## Using HTML Displays with the Control Interface

One exciting feature of the Touch Screen User Interface is using HTML displays. With HTML displays you can integrate other kinds of information into the display. Here are some examples:



HTML Displays are described in greater detail in the chapter on Displays.





## Chapter 24

# Design Tools

There are several tools in HCA that can make managing your home design a bit easier.

This chapter describes these tools and includes:

- Using the Program Flag Inventory
- HCA Status dialog
- Communications Monitor
- Interface Viewer
- File Inventory

Also in this chapter are three tools for the older X10 protocol.

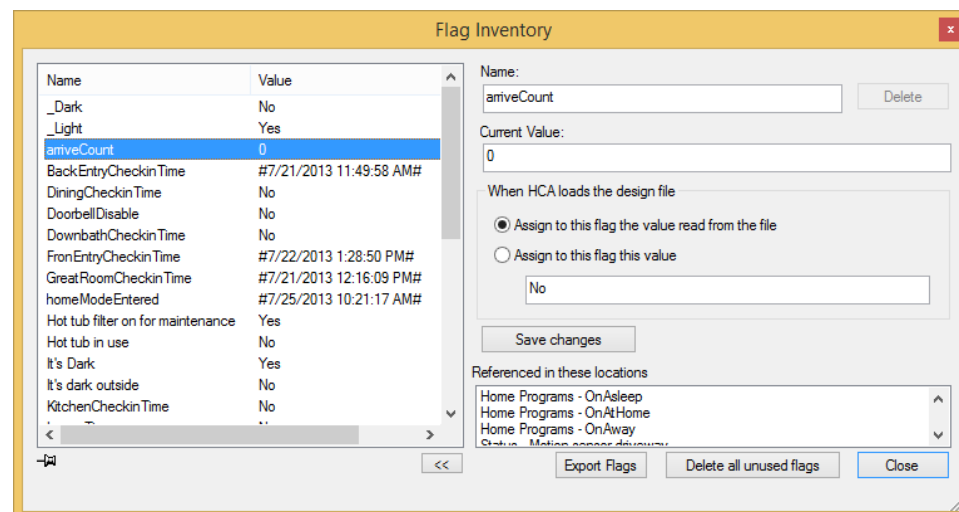
- Using the X10 House and Unit Code Inventory
  - Viewing X10 settings
  - Changing X10 settings
- X10 Reception
- X10 Test

---

### Program Flag Inventory

As explained in the chapter on programs, there are objects in HCA that you create that have both a name and a value. These objects are called flags. In a traditional programming language they would be called variables.

All the flags in an HCA design can be seen and managed in a dialog accessed by pressing the *Flags* button in the *Control* ribbon category.



This dialog can be left open while you perform other actions with HCA and the value of each flag updates as programs that reference them are executed. If you want to temporarily stop the display from updating, click on the push-pin icon in the lower left corner of the screen. When the dialog is pinned, no updates happen to the display. Click it again to resume having the dialog update.

This dialog contains these parts:

- A list of all flags in your design with their name and current value.
- Information about the currently selected flag's value
- A list of all uses of the flag.

When the program flag inventory dialog is open you can modify the value of any flag by selecting its name in the list then changing its value in the value edit box, and pressing the *Save Changes* button. What you place in the value edit box need not be a simple thing like 100, Yes or "abc" – it can be an expression. The expression is evaluated and the result is assigned to the flag. This is an easy way to immediately see the effect of what an expression evaluates to. Enter the expression in the value edit box and press the *Save Changes* button.

When you save your design file the current values of each flag is saved. When your design file is loaded you can have HCA set the flag to that saved value or assign it a new value. This is specified in the flag inventory dialog “When HCA loads the design file” box. If you are using the restart feature of HCA, the value of the flags can come from the state file – see the chapter on Restart.

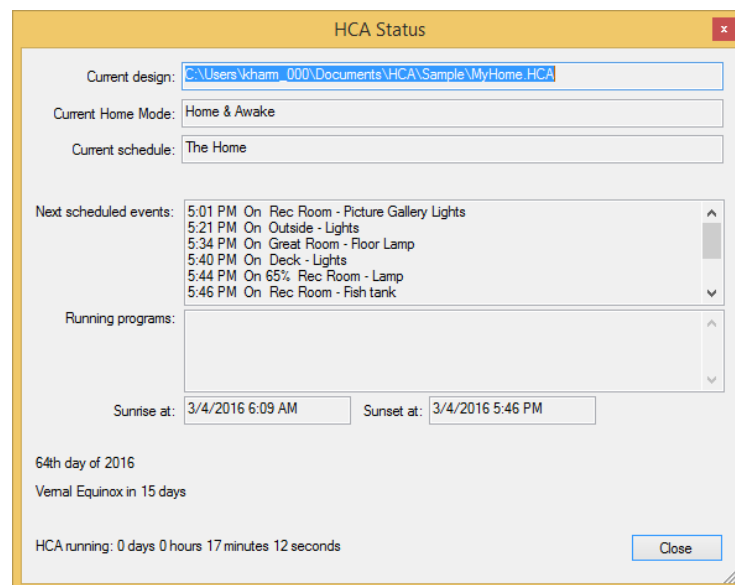
To remove a flag from your design, use the Delete button. You can't delete a flag unless it is no longer referenced any place in your design.

**Hint:** For more information on flags see the chapter on programs and the appendix on expressions.

**Hint:** When keeping the flags inventory on the screen you can use the button labeled “<<” to shrink and expand the dialog to use less screen space.

## HCA Status dialog

The HCA status dialog is a quick place to go to check on the state of your design. Open the status dialog by pressing the *HCA Status* button in the *Control* ribbon category.



Shown in the dialog is the current schedule, the next few scheduled events and any programs currently running and what they are doing.

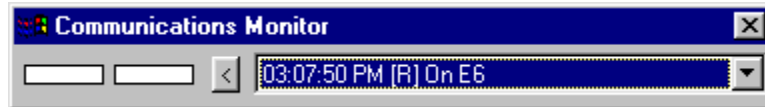
This dialog can be left open while you perform other tasks and HCA keeps it updated as things change.

---

## Communications Monitor

The Communications monitor is a small window that shows when commands are sent or received from any of the interfaces connected to your computer's communications ports used by HCA to send and receive commands.

The Communications monitor is opened by pressing the *Comm Monitor* button in the ribbon *Control* category. This Window appears as:



The parts of this window are:

- A rectangle that lights up when data is received from the interface
- A rectangle that lights up when data is sent to the interface
- A button that can reduce the window to just the send and receive lights or expand it to show all the information
- A drop down list that shows the last few commands with the date and time, if the command was a receive [R] or send [S], the name of the command, and the house and unit code of the command.

The communications monitor can be very useful if the interface in use doesn't have any status lights as part of its hardware.

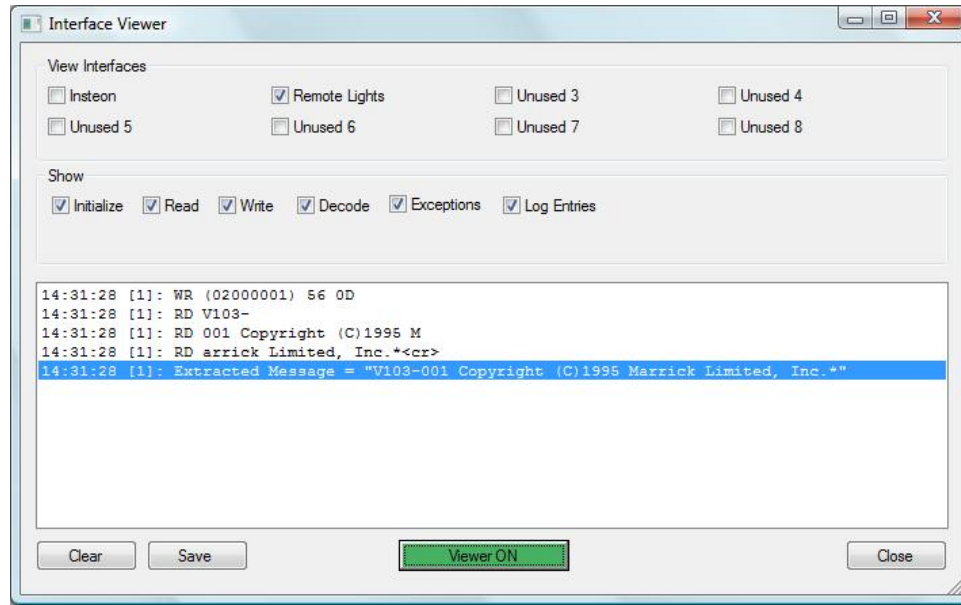
---

## Interface Viewer

In general, HCA takes care of all the details of interacting with the interfaces and sending, receiving, and decoding their protocols. There usually isn't a reason for you to want to see the lowest level communications between HCA and the interface but if you do, the Interface Viewer does that.

One place where you may really need to see all the data between HCA and an interface is when working with the Generic Serial and IP interfaces – detailed in the *Generic Interfaces* appendix. It can be complex working with these interfaces since all the encoding of messages you send and decoding of messages received is implemented in programs you create.

Open the *Interface Viewer* from the *Interfaces* ribbon category. This viewer shows the messages sent to and received from one or more interfaces. This gives you a window into the communications at the lowest level.



The names of the eight interfaces that can be configured are the first set of checkboxes. Tick the ones that you want to see data from. Interfaces not configured are listed as *Unused*.

The *Show* box determines what sort of information you want to view. Tick or clear the boxes for the type of data you want to see.

You can turn the viewer on or off with the button at the bottom of the dialog. When green the viewer is enabled and when red it isn't.

In the list reads from the interface are prefixed with RD, writes with WR, and if the message is decoded – some interfaces are but as explained above the receptions from the Generic Interfaces are not – they are prefixed with DC. Enclosed in []'s is the number of the interface the list entry is for. The first interface is interface 0 and the last is 7.

This interface is at a very low level so you may see, because of the nature of network and serial reads, a single reception in multiple RD lines.

You can clear the list with the *Clear* button and save the list into a text file with the *Save* button.

---

## File Inventory

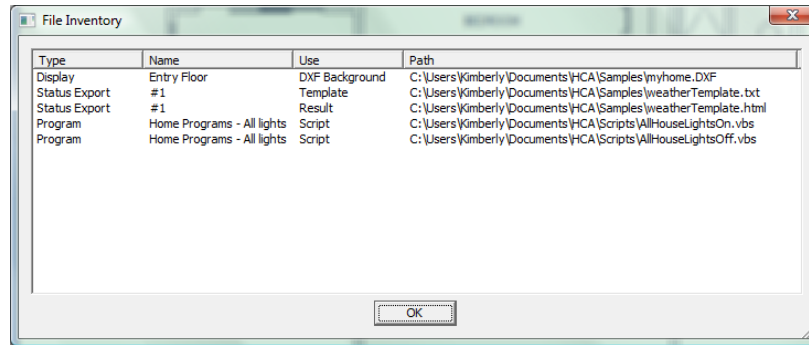
In addition to your design file - the .HCA file – there may be several other files that are referenced by your design. These files are:

- Display background DXF files.
- Display background image files
- HTML displays template and result files.
- IR key definition files for Global Cache IR interfaces
- Sound files referenced in "Play Sound" Visual Program elements.
- Data files referenced in "Read Data" Visual program elements.
- Files referenced in "Status Export" Visual Program elements
- Script files referenced in "Script" visual program elements

- Windows programs referenced in Visual Program "Run" elements
- The path of files used by the Weather system - import file paths, database folder paths
- Files referenced in periodic status exports

With so many files referenced from so many places it may be complex to try and keep track of them all. HCA helpfully provides one place where you can find all these files and where they are referenced.

To display this dialog, press the *File Inventory* button in the ribbon *Design Tools* category.



This dialog shows in one place all your external files that are part of your design.

## Using the X10 House and Unit Code Inventory

You need to think carefully before setting the house codes and unit codes for modules in your home, because the way you assign codes can make it harder or easier to work with your design. It is smart to assign the same house code to items in the same room, or with similar functions (such as outdoor lights), and easier to control your home if you choose sequential numbers for devices that are located or used together. The logical codes and sequential listings are easier to remember, and easier to control.

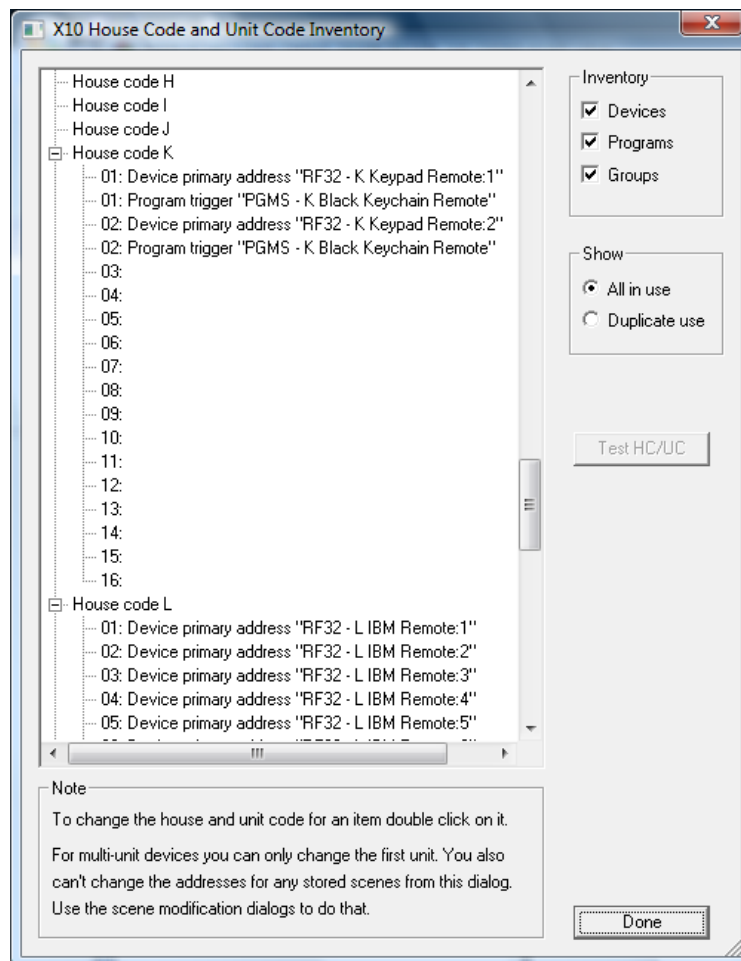
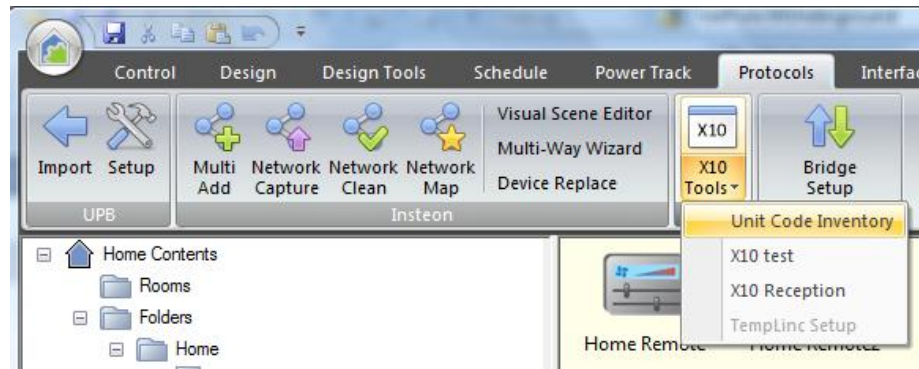
Keeping track of all your devices, groups, and programs can be a daunting task. You could walk around and check each device and its codes, writing each on a master list, or you could use the device, program, and group properties dialogs to construct a list of all house and unit codes.

However, HCA makes it easy, and does this for you. The House Code and Unit Code Inventory is a fast and efficient way to check the codes that you have used in your home design. It shows you which device, group, or program you have assigned each code to. It also gives you a way to change the codes as you review them.

There are several reasons that this tool can be a helpful for you.

- You may find that after installing new devices in your home for the first time, your choices need to be changed—perhaps more than once.
- If you later add a new device to your home, it is important that you don't choose settings that conflict with another module.

The inventory dialog provides easy access to all the codes. To open this tool, select *Unit Code Inventory* from the *X10 Tools* button menu in the *Protocols* category.



Since there are many ways that a House and Unit code can be used in your design, the Inventory displays not on the name of the item but how it is used. The possibilities are:

- Device primary address. This is what is assigned to the device on the X10 address tab.
- Device trigger. A device retransmission trigger.
- Program trigger. An X10 trigger for a program.
- Group trigger. A retransmission trigger for a group.
- Device scene. The X10 address of a scene stored in a device.

In the case of a multi-unit device, the name is suffixed with a colon and the unit number.

---

## Viewing settings

The **display area** on the left of the dialog box lists each of the 16 house codes. Each house code that you have already used is marked with a minus sign (-) indicating that it has unit codes already assigned.

Beneath each house code, you will see its 16 unit codes. Each unit code that is used by a device, program, or group in your design, shows the name of the item it is assigned to.

You can expand or contract the house codes to hide or view the assigned unit codes.

On the right side of the dialog box are some useful controls:

- The Inventory checkboxes allow you to only show those types of items that are checked off.
- The show checkboxes allow you to show all house and unit code assignments or just ones that are assigned to more than one item. This may show you a problem in your design.
- A Test button sends an ON then and OFF command to the house and unit code selected. This may help you determine if you have the correct house and unit code selected in the module or switch. You do this by using the Test button and seeing if the light or appliance goes on.
- A Done button that closes the dialog box.

---

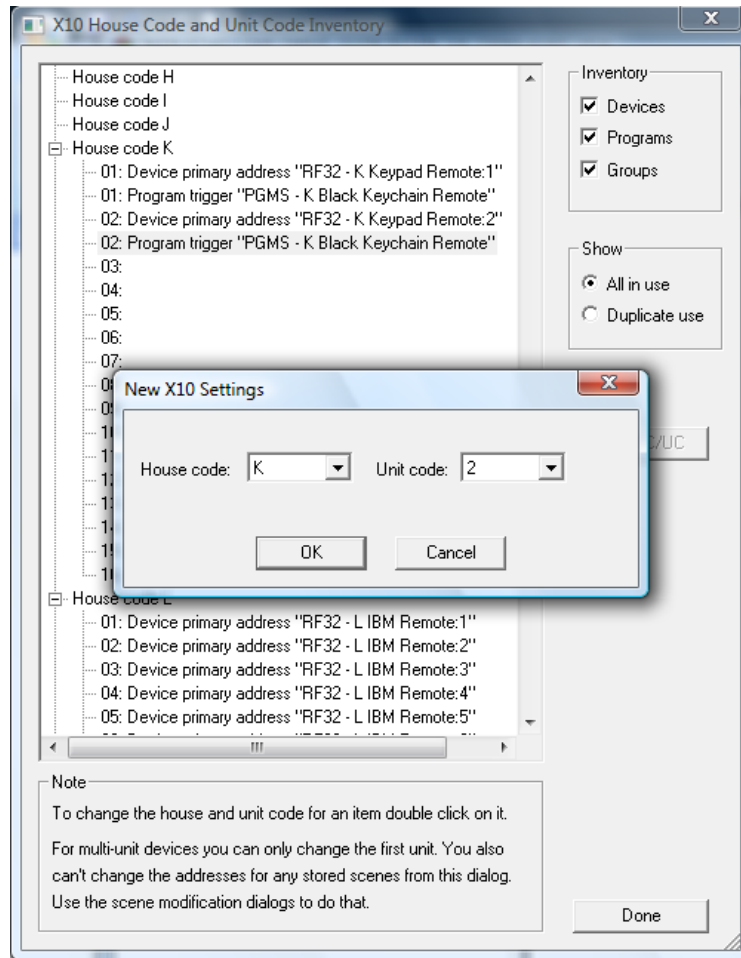
## Changing settings

The House Code and Unit Code Inventory dialog box not only displays X10 settings for devices, programs, and groups, but also lets you change them.

### To change a setting

1. Select the object in the tree and double-click the left mouse button.  
The New X10 Settings dialog box appears.





The dialog box shows the house and unit code for the device, program, or group you selected. You can change the setting with the *House code* and *Unit code* lists.

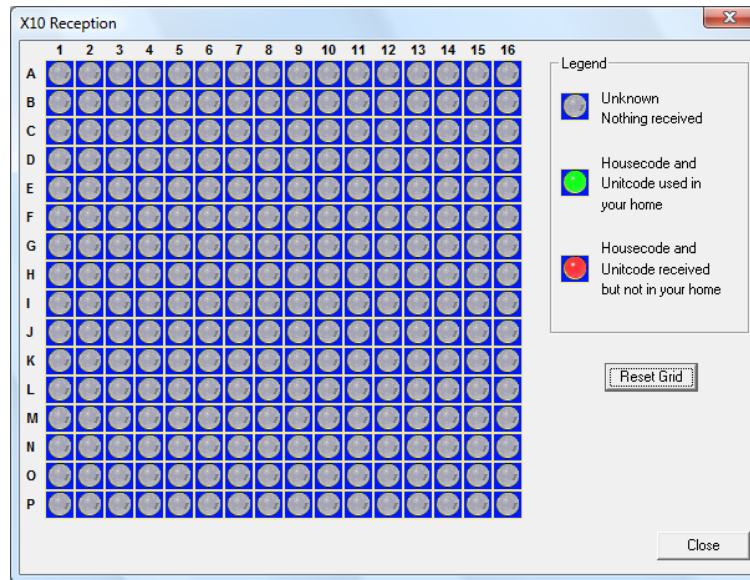
1. Simply click the arrow, and select the new house or unit code that you want assigned to the object you have selected.
2. Click the OK button to close this dialog box.  
HCA updates the properties of the chosen device, program, or group, and the inventory dialog box reflects these changes.

**Hint:** Remember that changing the X10 settings for a device only changes your design as managed by HCA. You still need to make the corresponding changes on the module itself.

---

## X10 Reception

The X10 reception tool provides an alternative view of information that appears in the log. To open this tool, select *X10 Reception* from the *X10 Tools* button menu in the *Protocols* category.



The large grid that appears in this tab shows, for each housecode and unitcode, if any X10 commands have been received by HCA and when. This could have been an ON, OFF, DIM or Status command.

The color coding tells you at a glance if the code has been received or not. If no command has been received for a housecode-unitcode, then the corresponding cell is gray. However, if a command for a housecode-unitcode has been received, the corresponding cell is green if that device is in your home design or red if it is not in your design.

To see the time when the last command was received, move the mouse over any of the 256 cells in the grid and a popup window shows the date and time of that reception. If you right click on a cell, a popup menu allows you to send an On, Off, or Status Request command to the house and unit code for that cell.

#### **Why is this important?**

This can be a useful tool to make sure that some controller in your home is not sending signals that you have not planned for. If X10 commands are being received by HCA and they don't correspond to a device, program, or group in your design, they have no effect. This may not be what you intended.

To reset the grid and have all cells show as Unknown (gray), press the *Reset Grid* button.

**NOTE:** This grid only shows receptions and not transmissions from HCA.

## X10 Test

The X10 test dialog lets you send continuous X10 commands on to the power line for test purposes. To open this tool, select *X10 Test* from the *X10 Tools* button menu in the *Protocols* category.

The purpose of this test dialog is to assist you in locating possible X10 reception problems by continuously sending X10 commands to a selected house and unit code. Test equipment can be used to monitor the X10 signal. If the device assigned to the house and unit code is not responding reliably, you could try turning other appliances and lights on and off to try and locate what might be interfering or generating noise

Step 1. Select a house and unit code  
House code:  Unit code:

Step 2. Select the command sequence to send  
 Send On - delay - send Off - delay - repeat  
 Send On - delay - send Dim - delay - send Off - delay - repeat

Step 3. Select the time delay between commands  
 seconds

Step 4. Some X10 test and monitoring equipment displays signal strength when commands are sent to the P1 address. You may want the command sequence sent to P1 in addition to the selected house and unit code  
 Also send to P1

Step 5. Some switches and modules respond to Compose commands. These can be sent instead of the standard X10 commands for On and Off, and Scene A can be used for the Dim  
 Use Compose commands

Step 6. Start the test

As it says in the dialog, the purpose of this dialog is to send X10 commands using your X10 interface in an effort to assess how X10 communications are working in your home.

This tool can be particularly useful if you are attempting to diagnose x10 signal strength throughout your home with an X10 signal meter. The x10 test pattern will continue to operate until you click the *Done* button.



## Chapter 25

# Status Export

Up until now most of what we have been concerned with was getting messages in and out of HCA. Messages from events – powerline transmissions, weather data, clock time – into HCA, worked on, and then messages sent from HCA.

In this chapter we look at getting information out of HCA that doesn't control anything but rather is a way for you to see what is happening in your home.

---

### Status Export Overview

What status export does is very simple, why you want to use it may be a bit harder to grasp.

Status exports reads what HCA calls a Template File, processes it, and produces what HCA calls a Result file. That's all that it does.

What's in the template file? Anything you want. Anything that is not a placeholder is copied directly from the template file to the destination file.

Placeholders are evaluated and the result is placed in the destination file. Status export knows what a placeholder is because they all start and end with the percent character. Between the starting and ending percent signs is an HCA expression – the same sort of thing you use in the Compute element. See the expression chapter to learn about HCA expressions.

Some example place holders are:

```
%_weather(InsideTemp)%
```

```
%_State("Table lamp")%
```

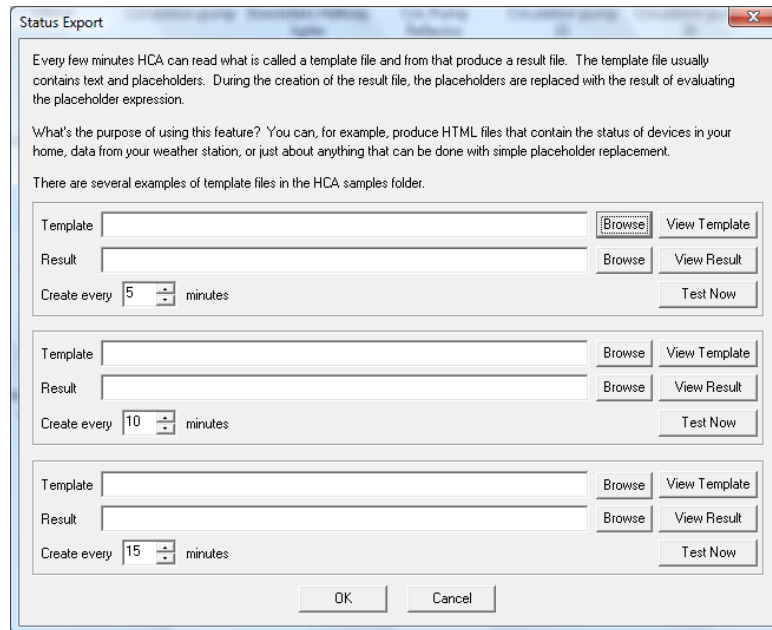
When HCA processes the template, it takes whatever is between percent signs and evaluates it in the same manner that the Compute and Compute test element does. Then it changes the result to text and writes that to the Result file.

In the above example, the first might evaluate to "80" and the second to "1".

---

### Configuring Status Export

To setup status export, press the *Status Exports* button in the ribbon *Design Tools* category. The status export setup dialog opens:



As you can see in this dialog, you can configure up to three different status files with different settings on how often they happen. In this example, only the first one is configured and that happens every 10 minutes.

The View Template button starts the Windows Notepad program to view the template file. The View Result button starts whatever program is appropriate to the kind of file you are producing. If it's a text file then Notepad is used, if it's an HTML file then your browser is used, if a CSV file is used then a spreadsheet program starts, etc.

The Test Now button performs the status export – copy the template to the result file with placeholders replaced – right now, so you can test your work.

**Hint:** In the configuration for an export, if you make the setting for how often the export is done to zero, the status export doesn't happen automatically until caused to happen by the Visual Programmer Status Export element. See the Visual Programmer chapter for the Status Export element.

---

## Using Status Export

Status export allows you to create files that contain almost anything you want as long as it's all text.

One use would be to periodically produce a file that contains the status of all your lights and appliances. This file can be stored someplace on your network for easy access or display. If you have a web server it can be placed into a folder managed by the web server and seen from any browser.

In addition to the status exports you configure here, you can also use the Status Export element in programs. See the Visual Programmer chapter for more info on the status export element.

**Hint:** If you want to move your status export files to a remote web server using, for example FTP, HCA can't do it directly. But it is simple to do! First configure HCA status export and set the time to zero. Next create a program that causes the status export to use the status export element and then use the Run program element to invoke the Windows FTP program. Finally schedule this program to run whenever you want.

**Hint:** There are good FTP shareware programs you can find available on the internet.

## Chapter 26

# Design Import and Export

This chapter covers a topic much different than other chapters. Rather than discussing features in HCA that you can use in your own automation solution, this chapter covers a feature that will help you and others exchange parts of automation designs.

Using *Export Design Elements* you select parts of your design to extract and create a new file with. Using *Import Design Elements* you can add to your design a piece of a design created by someone else. This is also helpful if you are taking something from one of the sample files for incorporation into your design.

---

### Design Import and Export Overview

Suppose your design contains a thermostat and you want to have an icon on a display show the thermostat picture and below it the current temperature. This isn't too hard. You need to have in your design a thermostat device and a program that triggers on current temperature reports. Also that program needs to manage its own icon and use the change icon to select the thermostat icon and change the text below it to show the temperature using an embedded expression.

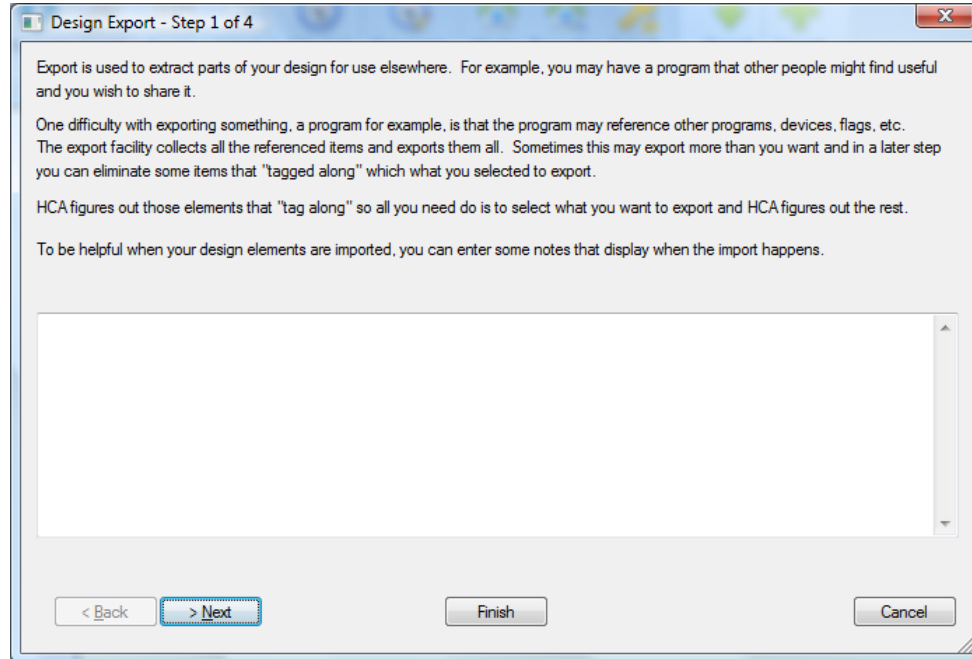
Not hard but it takes a bit of work. Suppose you want to share it with other HCA users. This is what Export Design Elements is for.

The export process is not quite as simple as it would be for exporting some data from a spreadsheet or database program. This is because so many things in HCA are connected. For example, a schedule could reference a program and that program references another program, and those programs reference some devices, etc. To export the schedule requires assembling a list of all the items referenced by the schedule, any things they reference, etc. This is what the Export Design Elements wizard does.

Importing, on the other hand, is not very complex. All things imported go into an Import folder.

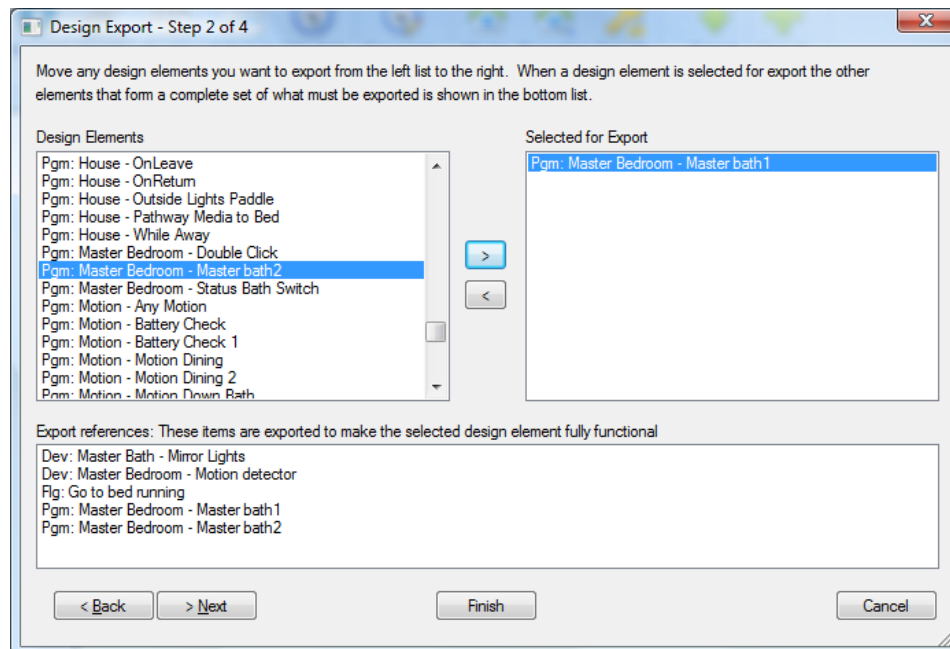
## Export Wizard

The export wizard is a simple three step wizard. To start the wizard, press the *Export* button in the ribbon *Design* category. The first step of the wizard appears as:



In this first step, you can enter any notes you have about what this export is all about. This can be very helpful for other HCA users when they import. When the import occurs, these notes display and the user is given the option to save them in a file – so your notes can be as descriptive as needed.

The second step of the wizard is where the action is: you select what you want to export:



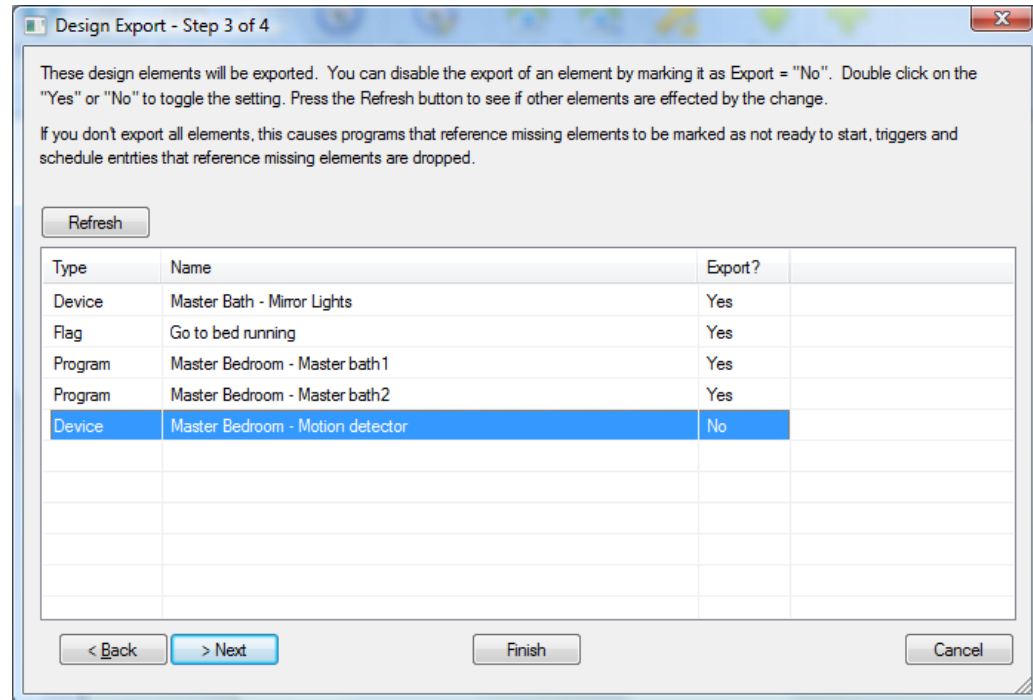


In the top left list are all objects in your design that can be exported. Move the ones that you want to export to the left hand list using the → button.

In the lower list are all the referenced objects that have to be exported. In this example, the one program to be exported *Master Bath1* references two devices and one flag, and two other programs, all of which will be exported to make it all work.

As described above, doing an export of one object – a program for example – also “brings along” lots of other pieces of your design. Sometimes that’s ok and sometimes it can result in an unwieldy mess. Too many objects being exported renders it practically useless for someone to import.

In the next wizard step you have control over what is exported.



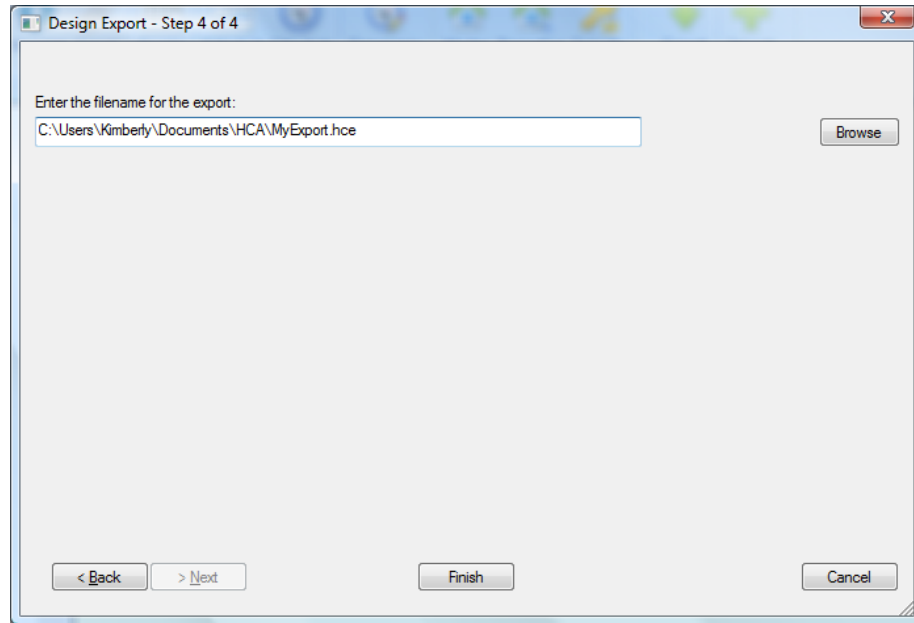
When this step first appears, everything that the wizard has marked for export is listed. As the dialog says, you can double click on a row in the list to toggle the export from Yes to No and from No to Yes.

Sometimes when you disable one item from being exported then the entire list of what to export changes. For example, suppose you wanted to export program A. Program A references programs B and C. Program C references practically every device in your design. Even though program A uses program C, if you don’t export program C then all those devices it references are also not exported.

After you make changes to the export list, the Update button reexamines what to export. In the above example after you disable export of program C then all the rows for the devices that are referenced by program C disappear.

But what happens when you take things away from export that are needed? The simple answer is that what is imported is not 100% operational. In the above example, program A had a *Start-Program* element from program C. When program C not exported the *Start-Program* element in program A is not correctly configured and so program A is marked “Not ready to start”. It is up to whoever imports the file to “fix things up”. But that is often ok since the major work or concept of the export is preserved.

In the final step of the wizard you select the filename to write to:

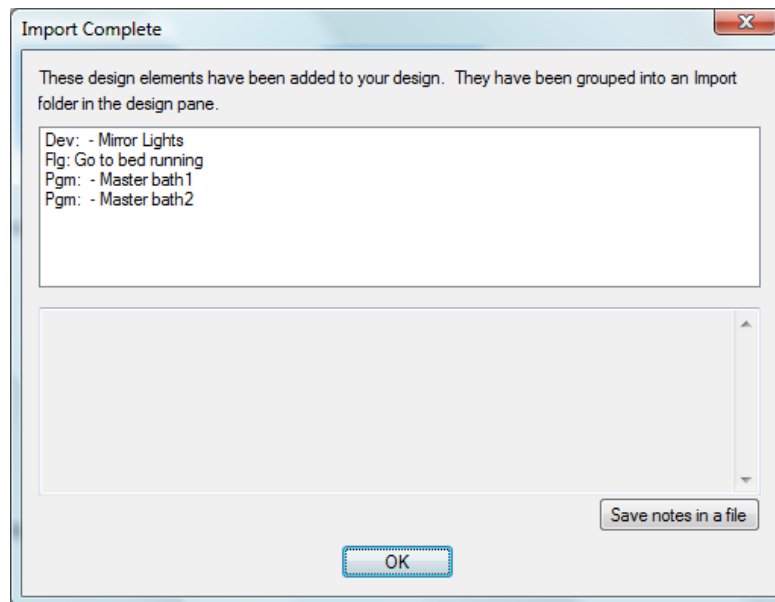


The expected file type for HCA exports is .HCE

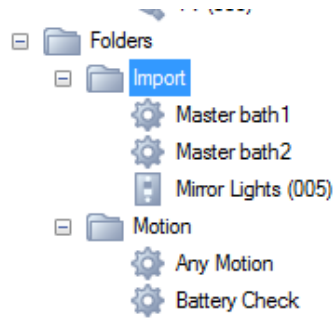
---

## Importing

Importing is much simpler than exporting. To import design elements into your design, press the *Import* button in the ribbon *Design* category. Then browse to the file you want. When the import is complete this popup displays:



As explained in the dialog text, everything imported goes into an Import folder:



Once it's in the import folder you can look at all the parts and see what you want to keep, delete, modify or whatever.



## Chapter 27

### HCA Client-Server

Everything described in the User Guide up to this point describes what is called *stand-alone HCA*. It works just like most other Windows applications – you install it on your Windows computer, start it, work with it, and leave it running so it can control your home.

But HCA can also operate in client-server mode. This is what is called *Client-Server HCA*. If you are unfamiliar with the term “client-server” here is a definition:

*Client/server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request. Although the client/server idea can be used by programs within a single computer, it is a more important idea in a network. In a network, the client/server model provides a convenient way to interconnect programs that are distributed efficiently across different locations*

Client-Server HCA consists of these components:

- The **HCA Server** loads and “executes” your automation design. This application has a very minimal user interface. It communicates with all your attached interfaces, runs schedules and programs contained in your design, and handles triggers for programs and devices. You can think of the server as giving the same function as stand-alone HCA after you loaded your design and then if you never touched the UI. Your design just runs in the background.
- Once the server is running and loaded your design then you access it using a client. There is a client for iPhone/iPad, a client for Android phones and tablets, and a client for Windows. All of these clients are control only. That is, you can control items but not make changes to your design.
- You can also run HCA itself in client mode - the same HCA executable works in both stand-alone mode and as a client to the HCA Server. What can HCA do when operating as a client? It can do anything that you can do with stand-alone HCA except for creation of a whole new design. From a client you can modify programs, add new devices, scan an Insteon network, control devices, etc. The experience working with the client should be the same as when working with stand-alone HCA.

A very important point is that multiple clients can be connected to the HCA server at one time. When you do this, any change made on one client to the state of an object – on, off, or dim – or a change to the design – adding a new device, changing a program etc - automatically updates the other clients with those same changes.

Why would you want to use HCA Client-Server? The major advantage of HCA is that it can integrate all your different powerline technologies, weather, IR, wireless, and the internet with your programming and schedules. With client-server you can get access to HCA from any client – phone, tablet, laptop – from anywhere.

You also may want to run a client on a shared computer in your home. If you configure that client to use the Touch-Screen User Interface – described in its own chapter – then everyone can control just those devices you want to expose. And they are prevented from making any design changes.

**Note:** Don’t be confused with the term “HCA Server”. The HCA Server is not a “Windows Service”. The HCA Server is just another Windows application.

## First some words about security

Before even describing how to use the HCA Server and connect a client it is important to first consider security.

***HCA controls real things in your home and you should do whatever it takes to limit that control to authorized users.***

Using client-server you can expose your home automation design to external access. You may want to do this so you can access your home from anyplace – from your work computer or a laptop when on the road - and control devices and programs. But if you can do that, anybody can do that unless you work to prevent it!

There are many ways to limit access and you should use all of them:

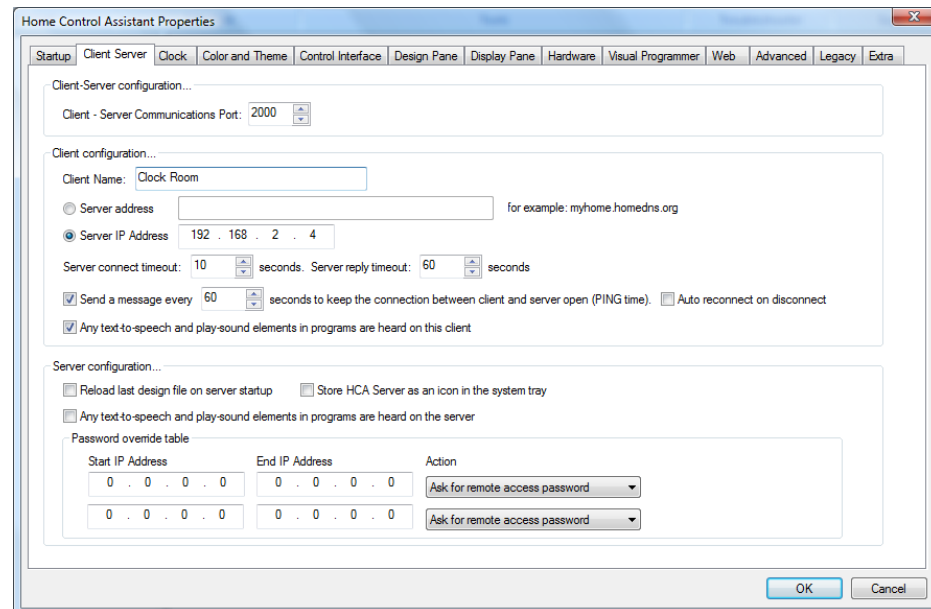
- You should choose a non-default port number.
- You should assign a remote access password to your home design.
- You should enable any password protection features that your router may provide.

***Remember: HCA controls real things in your home and you should do whatever it takes to limit that control to authorized users. Security is your responsibility!***

## Using the HCA Server

Before you can use the HCA Server you have to configure it. All server configuration is performed from the *HCA Options* dialog. This means that even if you want to run only the HCA Server on a computer you must install and run HCA at least once to get access to all the options.

To configure the server press the *HCA Options* button in the HCA application menu.



As the dialog says, some of the sections configure the server, some the client, and some configure both. The parameters that configure the server are:

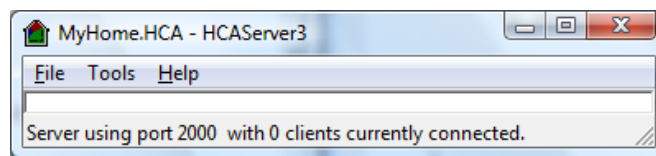
- Port  
Communication happens between the client and the server using a designated port number. You can choose any port number you would like.

- **Reload on startup**  
This option directs the HCA Server to reload your design file when it starts. This is similar to the reload option in stand-alone HCA on the startup tab.
- **System notification area on startup**  
This option directs the HCA Server to minimize itself to the system notification area upon startup. This is similar to the system notification area option in stand-alone HCA on the startup tab.
- **Text to speech**  
If any program executes a text to speech element, if the option is enabled then the server's audio system is used. A message is also sent to the connected clients and they too can use their auto systems for the text to speech operation.

The last section of the client-server tab has to do with the HCA Security model. When you open the home properties, on the security tab you can assign several different passwords. The one password important for this discussion is the Remote Access Password. This password is checked each time client attempts to connect to the server

In order to make it simpler for you on an internal network – your house-side of the firewall – you can assign a password to your design but configure the server to not request the password from some computers – identified by a range of IP addresses. You should, of course, only use this password suppression on internal IP addresses.

You can start the HCA server from the Windows Start button. Once started its main window appears:



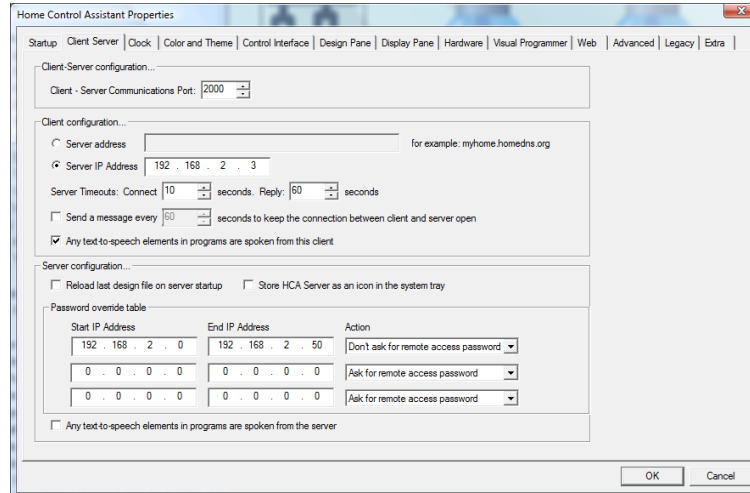
This is the whole interface for the HCA Server! We did say it has a very minimal user interface! The status bar shows the port it is using as well as the number of currently connected clients. The title bar shows the filename of the loaded design file. The menu selections are:

- **File – Open**  
Open a selected file
- **File – Minimize to notification area**  
Store HCA Server as an icon in the Windows notification area. This is different than using the minimize box in the title bar which minimizes the application to the task bar.
- **Tools – Remote Access Viewer**  
This opens a window where the various messages that flow between the client and the server display. This is largely for diagnostic purposes of technical support. Also displayed in this viewer are any log entries generated by the server in a compressed form. You can use this dialog if you want to be reassured that the server is doing something.
- **Tools – Show Connected Clients**  
Opens a dialog that shows the IP address of all connected clients
- **Tools – DDSN Update and DDNS Test**  
See the DDNS technical note for more information.

Once the design is loaded by the server, it is operational. Schedules run, triggers are responded to, weather observations taken, etc.

## Connecting a Client

Once the Server is started and your design loaded, then a client can be connected. Before the connection can be made the client must first be configured. To configure HCA to run in client mode, run HCA and open *HCA Options* and choose the Client-Server tab.



As the dialog says, some of the sections configure the server, some the client, and some configure both. The parameters that configure the client are:

- **Port**  
Communication happens between the client and the server using a designated port number. You can choose any port number you would like. The port selected for the server and client must match.
- **Client Name**  
Assigning a name to each client is necessary for some places in HCA where you can explicitly have an action carried out by the server with only a single client. For example, the Visual Program element that requests user input when executed by the server can be routed to a named client.
- **Server IP Address**  
This can be entered by using a four part IP address – a.b.c.d – or using a text string that can be translated into an IP address. Depending upon how you have Windows configured, it may be possible to use a computer name here.

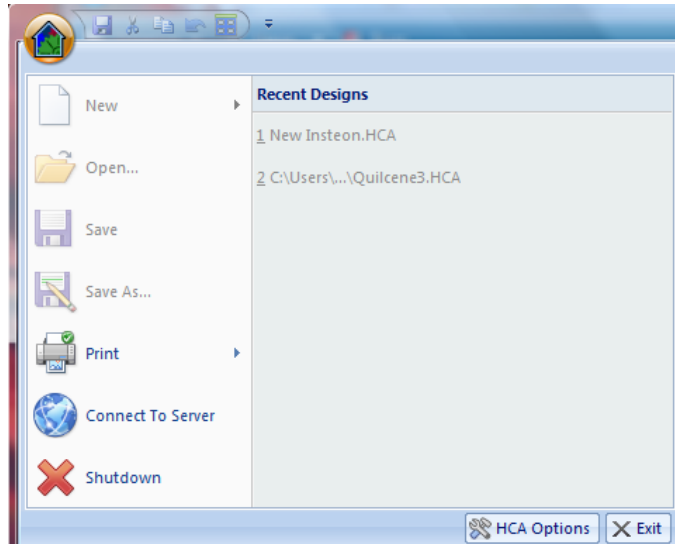
If your internet access gets assigned an IP address that can change over time you probably have setup an account with one of the services – like DynDNS.org – that give you the ability to reference your home network using something like “myhome.HomeDNS.org”. You can also use the HCA DDNS Service. See the DDNS Tech note for more info.

- **Server Connect Time**  
The number of seconds to allow for a connection to be made to the server
- **Server Reply Timeout**  
The number of seconds to wait for a reply to any request sent to the server
- **Periodic messages**  
The client can periodically send a short message to the server every ‘n’ seconds. Depending upon the hardware and operating system settings, this may be necessary to keep the server to client communication connection open over time.



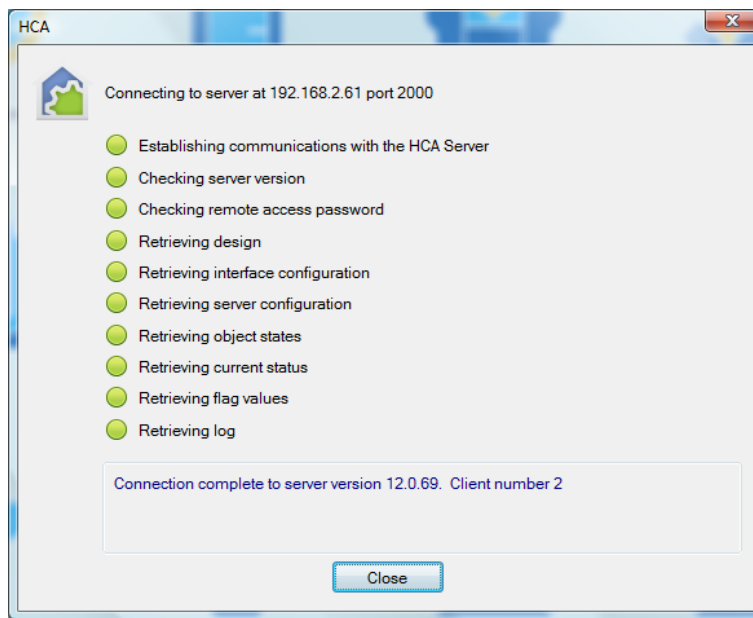
- Auto reconnect  
If the client becomes disconnected for any reason, if this option is enabled then the client tries to reconnect

Once you have configured the server and the client, started the server, and loaded your file you can now connect the client. To do this start HCA and select from the application menu *Connect To Server*.



Note: If you are running the client and the server on the same computer, when you start HCA it determines that the HCA Server is running and configures the menu to disable all the options for stand-alone operation - the actions that work with a file directly like Open, New, Save, etc.

Once *Connect to Server* is selected HCA establishes a connection to the server and then requests your design, configuration settings, the state of all your devices, and any log entries generated since you last connected.



And that's it! You can now interact with your design. To terminate HCA Client just select *Exit* from the application menu or click on the close box in the title bar.

**Note:** As your design is being loaded, any external files that are referenced – DXF files, IR definition files, icon files, etc are requested from the server and moved to the client. These files are all stored in the Temp subfolder in your HCA documents area.

---

## What can you do with HCA Client – Server

As stated above, the goal of the HCA Client is that you can do anything you would normally do with HCA except for creation of a new file.

This means that each of your actions results in a communication with the server. If you control a light, for example, the client sends a message to the server to turn on the light.

Any operation that happens in the background when you are using stand-alone HCA happens on the server when in client-server mode. This includes:

- Schedules
- Program execution
- Trigger response for programs, devices, and groups
- Periodic status export
- Home Mode changes and Power Track statistics gathering
- Web Component
- Alert Manager
- Weather observations
- Power failure and restore
- Startup status polling
- Protocol Bridges
- Any device control initiated from the user interface

From the client, you can do more than just action operations. You can also change schedules and programs, add new devices, etc.

One important point to keep in mind as you work with your design are those areas that reference external files. For example, suppose you modify a program that uses the Read-Data element. This element reads an external file. Where does that file need to reside and what is the path that goes in the element properties?

Since all programs are executed on the server, the path in the element needs to be a *path to the file on the server* and not a path on the client computer. Those dialogs that accept a path contain a warning to remind you of this when HCA is working in client mode.

You can also work with those settings that are made in HCA – Properties. For example, you can test the connection to your hardware interfaces, start and stop the web component, and change program execution options. All these result in communication from the client to the server to carry out the action.

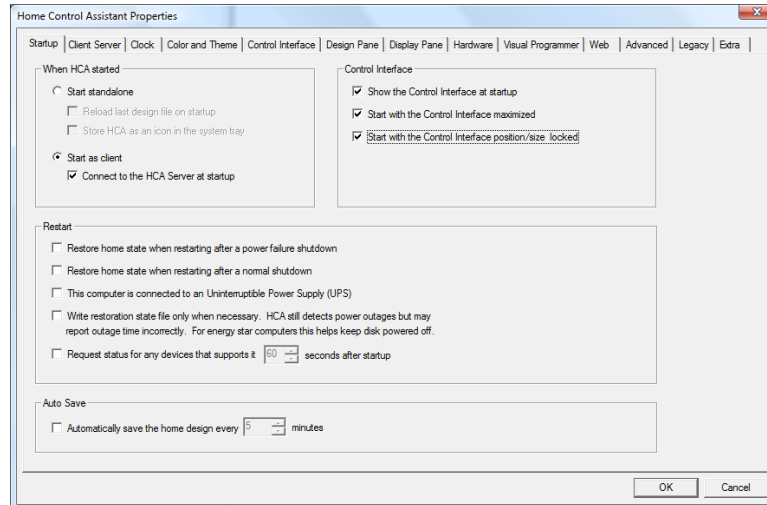
Should your connection to the server be lost – this can happen with network communications for a variety of reasons – you can reconnect to the server by selecting from the menu File – Reconnect to server. This causes your design to be reloaded.

**Note:** Some HCA settings are local and others are global. For example, you can configure a client’s display pane settings (HCA – Properties Display Pane tab) and those settings are local to that client. Other settings, like the Advanced Settings (HCA – Properties Advanced tab) are communicated to the server when changed on the client as they effect the action of the server.

---

## Automatically Starting as a Client

Depending upon your application it might be convenient to have HCA when started to automatically connect to the server. This is accomplished by settings in HCA Options on the Startup tab.



In the *When HCA started* box, enable the options “Start as a client” and “Connect to the HCA Server at startup”. Once enabled, HCA automatically connects at startup and closes the connection progress dialog when the connection is completed.

---

## Remote Access to the HCA Server

Everything described above in configuring the server and the client may be only part of what you need to do to establish communication between them.

You may need to configure your firewall and router to allow the selected port through and pass request from that port to the computer on your network running the HCA Server.

Configuring your firewall and router – software and hardware – is outside the scope of HCA and, we admit, can be a real challenge. All we can do is to suggest that you look carefully over your router documentation and consult the many different helpful guides you can find on the internet.

---

## Other Clients

In addition to using HCA as a client to the server, you can also use the other clients. These are described in the next chapter on Clients.



## Chapter 28

# HCA Clients

As described in the Client-Server chapter, the HCA Server loads and executes your design. In that chapter it was also described how you can run HCA in client mode to access the server and have full read/write access your design.

This chapter describes applications that connect to the HCA Server and let you control your design but not make modifications to it. There are three clients available:

- HCA Window Client
- Application for Android mobile platforms – tablets and phones
- Application for iOS mobile platforms – iPhone and iPad

Using any of these applications and the HCA Server you can access your home from any location around the world as long as you have access to the internet.

However, *before* even considering installing and using these clients it is first necessary to install HCA, configure the HCA Server, and use HCA in client mode to test your work.

Also, each of these clients use the settings in the design that configures the Control Interface so working with that as well is also strongly suggested.

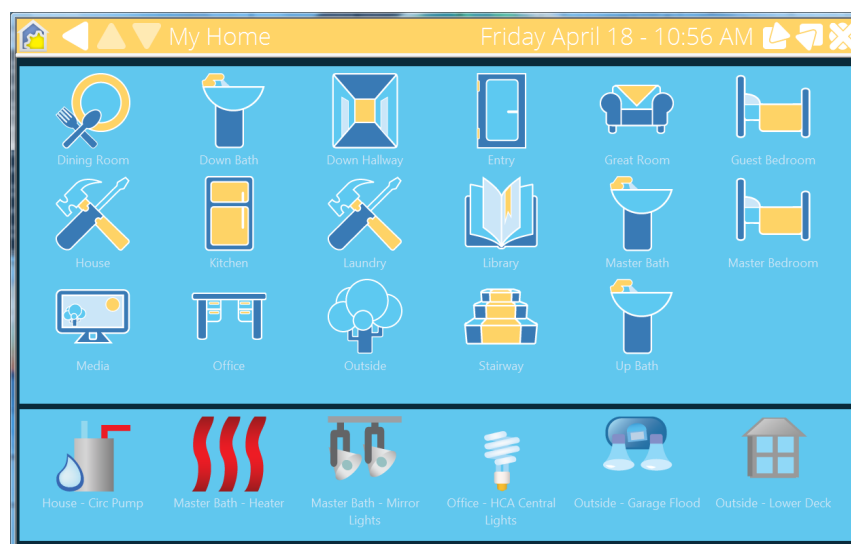
NOTE: Getting any HCA client to connect to the HCA server may not be simple. There are issues of network access, network security, firewalls, anti-virus programs, etc. All of these “security” mechanisms try and prevent exactly what you are trying to do. Since each network is different, technical support will only be of limited assistance.

---

### HCA Client for Windows – Quick Tour

The HCA client is available as a separate install downloaded from the HCA Support web site. It can be installed on any Windows computer. No registration key is required.

If you are familiar with the Control User Interface you will quickly recognize how similar *the Windows HCA Client* is to that interface.

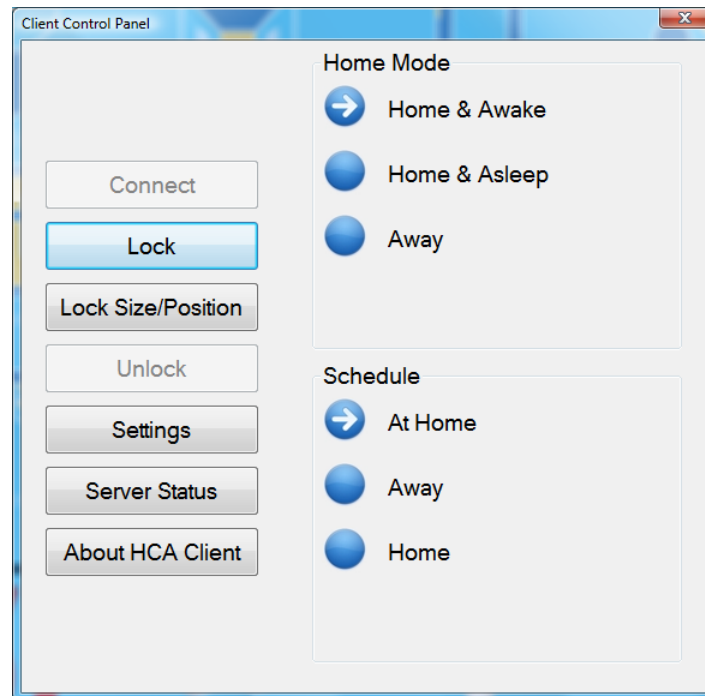


Like the Control User Interface, the same short and long tab methods are used to control a device, program, or group. Also, like the Control User Interface, the status bar color reflects the Alert level – red, yellow, or green, and the HCA icon used at the left end of the status bar changes color to show the home mode.

---

## Client Control Panel

To open the Client Control Panel, click on the HCA icon at the left edge of the status bar at the top of the window.



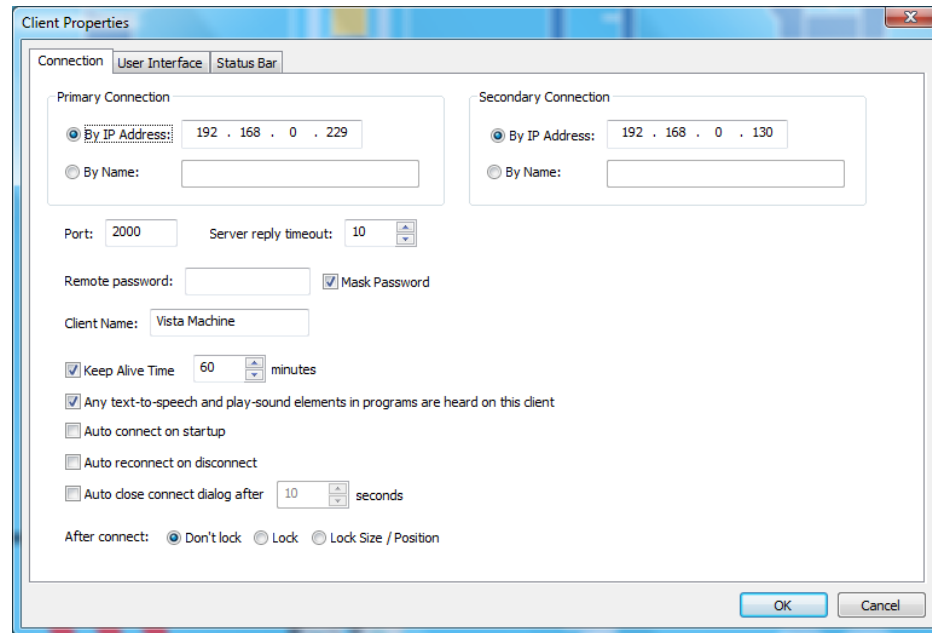
The control panel shows the home modes and schedules. They can be changed by tapping on the round button to the left of the mode or schedule name.

The *Lock*, *Lock Size/Position*, and *Unlock* buttons are used to prevent the movement and resizing of the client window. The *Lock* button prevents changing the window size and also removes the minimize, maximize, and close buttons from the status bar. *Lock Size/Position* also locks the window but retains the minimize button.

The *Server Status* button shows the current server status – time, versions, etc – and also a brief display of the current Alert Report.

## Configuring the client

To configure the client, press the *Settings* button. A multi-tabbed dialog opens.



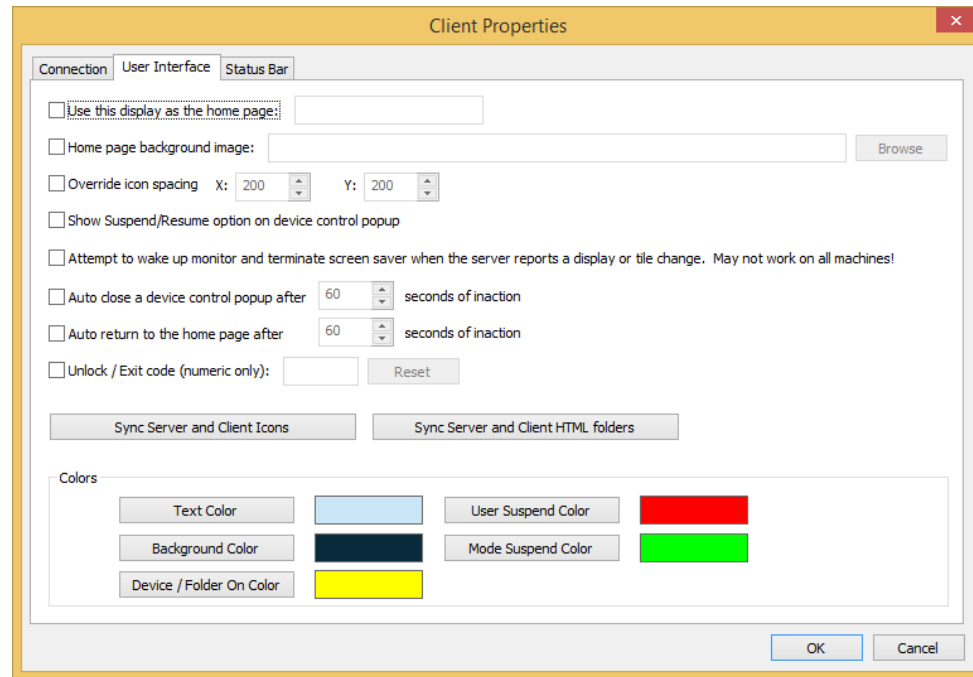
## Connection tab

The connection tab lets you modify these settings:

- Primary address, Secondary address, and Port. The address and port number of the server.
- The remote access password. If your design has a password you can enter it here or enter it when prompted on each connection.
- Client name. Each client should have a unique client name. This is used in the server “connected clients” display and also in programs that have elements that can effect only a selected client.
- Keep alive Time: How often a message is sent from the client to the server to ensure that the connection is open. If it isn’t answered by the server then the client may try and reconnect.
- Auto connect at startup. When the client is started, if this option is enabled then the client immediately tries to connect to the server,
- Auto reconnect on disconnect. This option works with the “Keep Alive” setting. If enabled, then the client automatically tries to reconnect if it is found to be disconnected.
- Auto close connect dialog. When connecting a popup shows the connection process. If this option is enabled then this dialog is automatically closed when the connection completes.
- After connect lock option. After a successful connection the client window can be locked in the selected lock option.

---

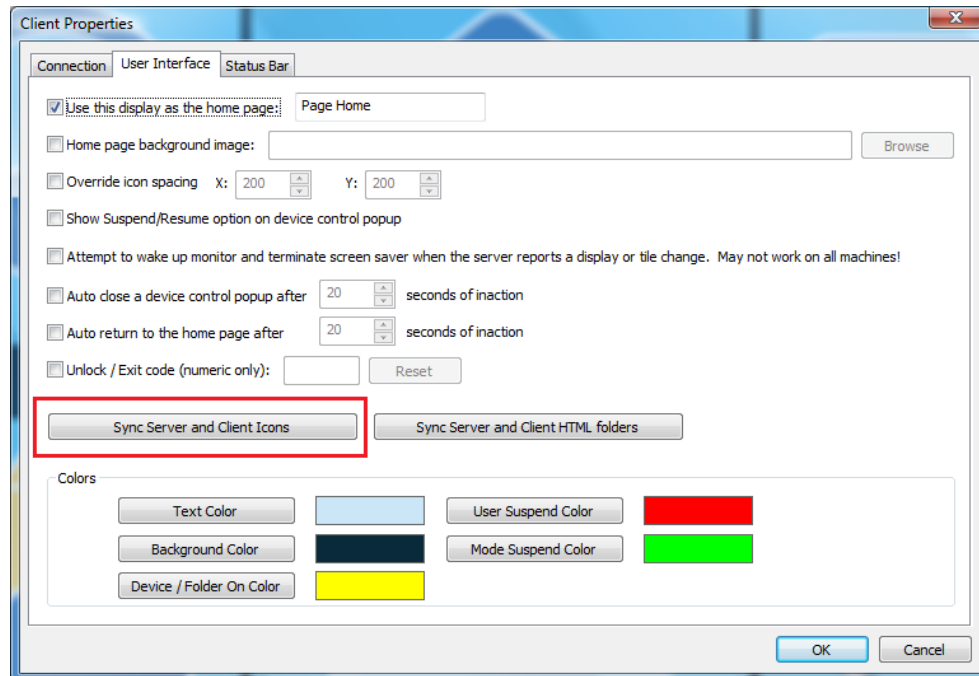
## User Interface tab



The options on this tab configure the user interface appearance and actions.

- Home page. The name of the display to use as the home page.
- Home page background. This option lets you choose a display background for the home page. This image can be unique to each client and overrides any display background for the HCA display named as the home page.
- Icon spacing. This option overrides the default icon spacing that is normally used by the theme selected for the display page.
- Show a Suspend/Resume button on the device control page so that you can suspend or resume a device, group, or program.
- Wake up monitor. As the text says when this option is enabled the monitor on the computer the client is running on will attempt to be “wake up”. Also, as it says this may not work on all monitors.
- Auto close times. These settings are used to automatically close the action popup – typically from a long tap on a device – and to return to the display home page after a selected number of seconds where you perform no action.
- Exit code. You can enter a code that must be entered to exit or unlock the client. This only is useful if you also lock the client so that the exit button isn’t seen on the status bar.
- The *Sync server and client icon* button: 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. If you create your own icons all you need do 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 Sync server and client HTML folders: Described below.



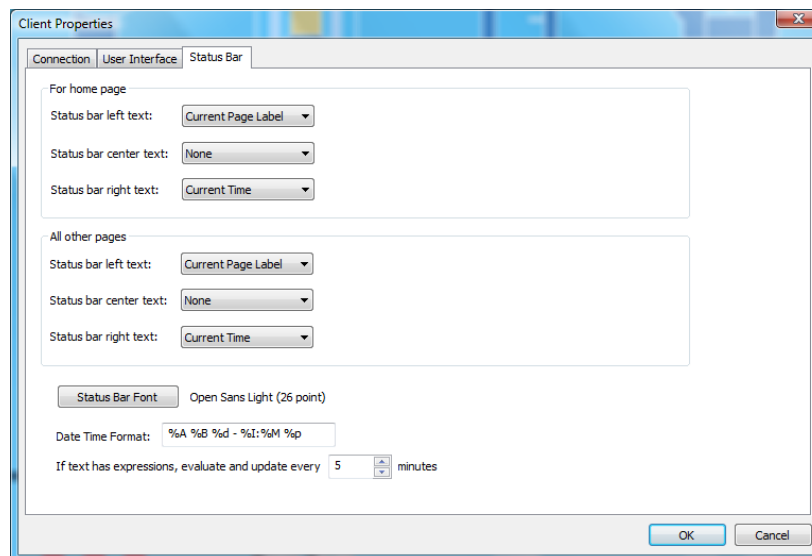


- Colors. The various colors used for user interface elements.

---

## Status Bar tab

The status bar tab is very similar to the Control User Interface setup in HCA. You can specify the text that displays in the left, middle, and center of the status bar.



In addition to choosing the data in the status bar sections you can also choose the status bar font and point size as well as the date format. The format pattern characters are the same as used in the `_FormatTime` function and is documented in the User Guide Expressions chapter.

---

## 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 file, result file, 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:

Result file produced from the template:

*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.*

If you set things up as these guidelines specify then the client “HTML Folder Sync” operation moves to the client computer all the auxiliary files – style sheets, images file, etc. – that your HTML display uses.

---

## HCA for Android – Quick Tour

The Android HCA client is available for download from the Android market (aka “Google Play Store”).

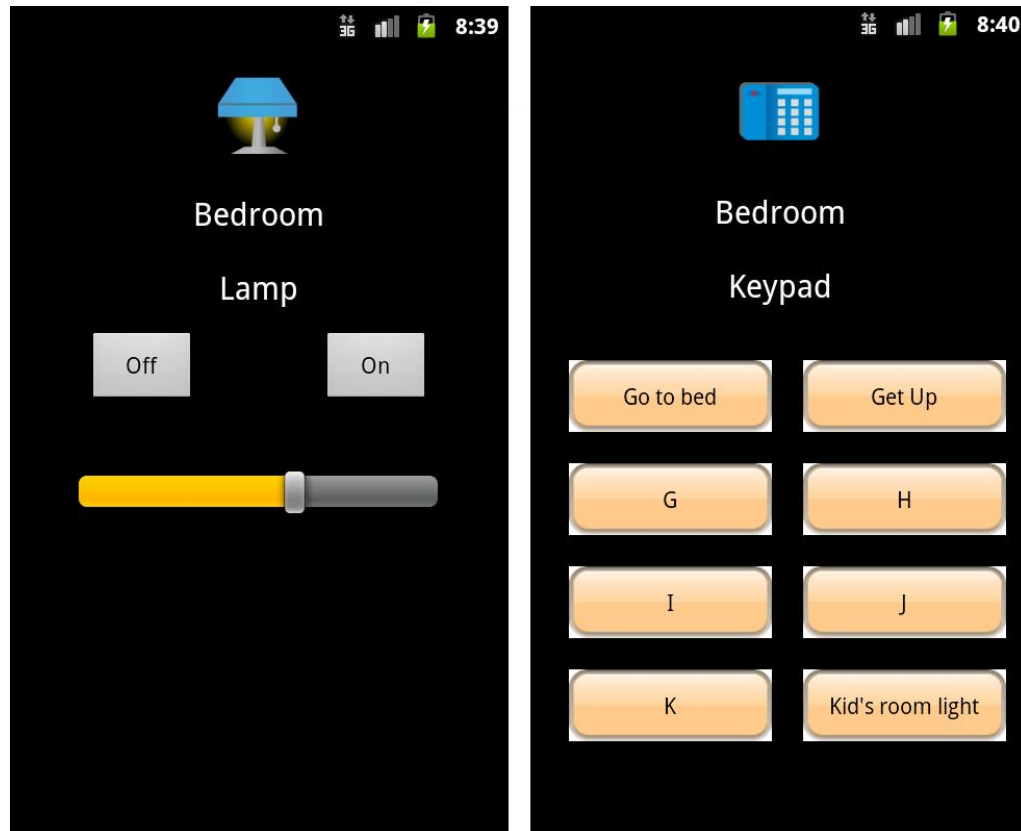


If you tap on display icon, the page for that display appears – the image on the right above.

Depending on how the device's properties are set, when you tap on a device icon it can toggle the device's state or open a page for controlling the device.

There also are options for each device, group, or program that are set in your HCA design that say if an icon for it should appear on a display when shown in the Control UI. This same option is used by the Android client when showing the contents of a display.

Tap on an icon and a page for that kind of device, group, or program opens. The image shown below on the left is for a dimmable device and the display on the right is for an eight button keypad.



The “Glass Keypad” features in the Control Interface are also in the HCA Android client. The image on the right shows such a glass keypad.

To navigate back a page, the dedicated back button is used.

---

## Android Settings

The first action to take after starting the application is to configure it for access to the HCA Server. Use the menu button at the top – looks like a wrench -to open the settings menu.

Choose HCA Server Settings and set the primary IP address to the address of the server computer. Also set the port number. The port number must, of course, match the port number that the server was configured for.

**Hint:** You can enter the address as a.b.c.d or something that resolves to an IP address like “myhome.homedns.org”.

The use for secondary IP address is explained in the next section.

---

## Android Server Settings

- Server Settings - Remote Password  
This is the password configured in your home design in HCA – Properties on the security tab. You must enter it here before you connect. It remains part of the configuration so you need not enter it each time you connect.
- Server Settings – Server Reply Timeout  
Each transaction with the server times out after this many seconds if no reply is received. If a timeout occurs it is assumed that connection to the server is lost.

- Server Settings – Keep Alive Timeout  
Like the HCA Windows client, *HCA for Android* can periodically send a message to the server and get a response back. This may help to keep the connection open. This setting enables this.
- Server Settings – Keep Alive Timeout value  
This is the number of minutes between each “keep alive” message sent.

---

## Connecting and Disconnecting

Once all the sever settings are configured, you can connect by pressing the *Connect* on the home page – looks like an arrow pointing at a dot. To disconnect, press the disconnect button – looks like an X.

The connection to the server lasts until you shutdown the phone, the connection is lost for some reason, or disconnected by using the *Disconnect* option from the menu, or you *Back* out of the application to the Android main home page. If you leave the HCA Android client by pressing and holding the Home button until the running applications popup appears and you select another application, the connection to the server is not closed.

As long as the connection to the server is open, the phone receives status updates from the HCA server and your devices change to show appropriate state.

And that’s really all there is. The various displayed pages follow the stock Android methods so if you rotate the phone to landscape mode then the displays change to use the space differently.

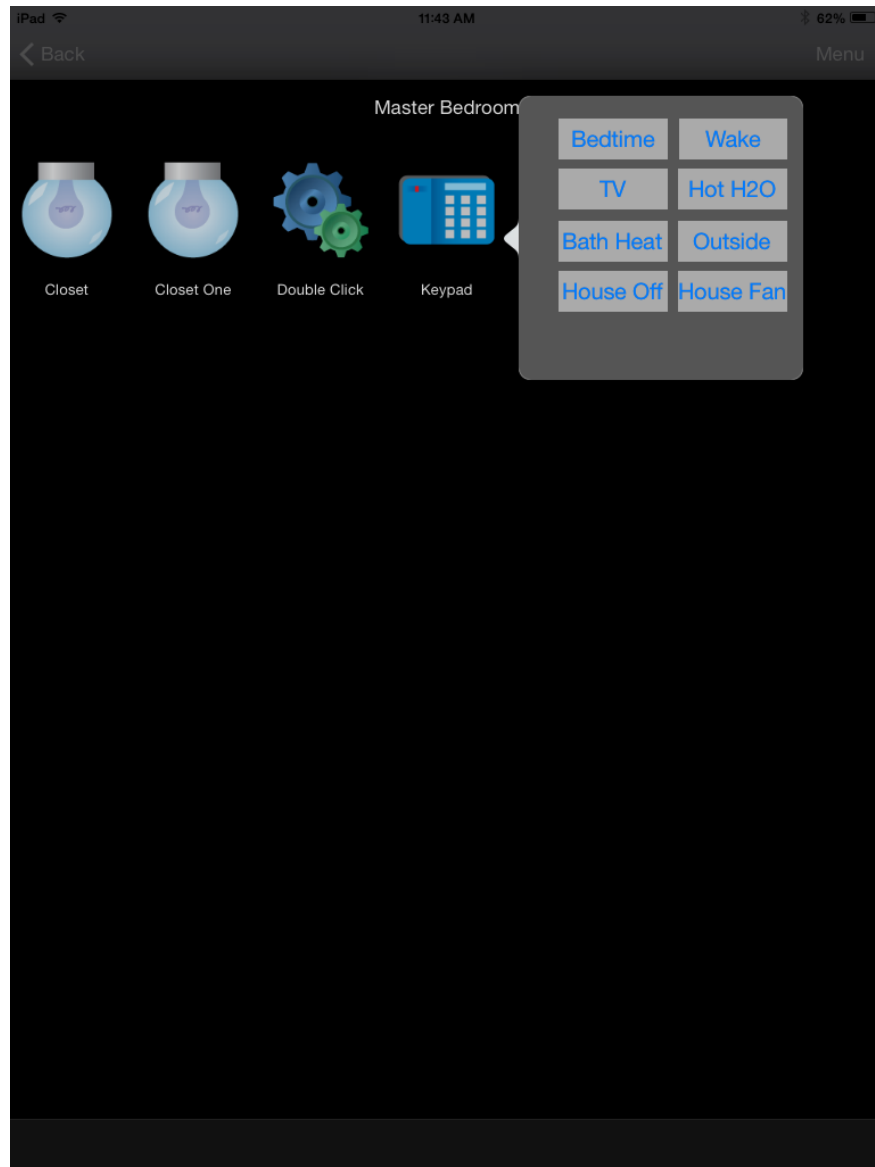
In addition to this description, there is also a brief Help file. Select the Help option from the menu popup.

## HCA for iOS

An iPad and iPhone client can be used with the HCA Server available from the Apple App Store (aka iTunes). This client operates very similarly to the Windows HCA client and the HCA Android client.

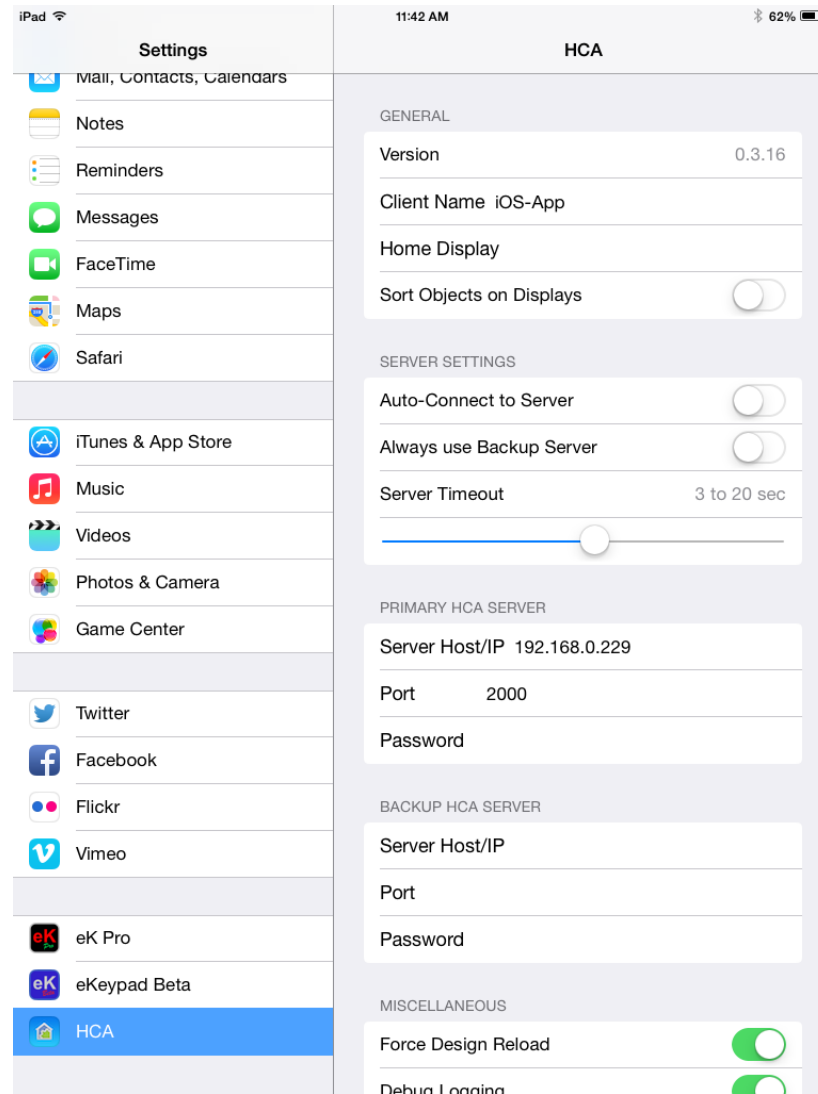


This example display, on an iPad mini, shows the familiar set of HCA icons. The same short and long taps on an icon like the other clients are implemented. To toggle the status of a device, a short tap toggles its state. Long taps are a bit different than the HCA Windows client and HCA Android client as the popup display doesn't appear until you release the tap.



## Configuring the iOS application.

The iOS settings application allows access to the HCA application settings.



The settings are:

- Client name. Each client should be given a unique name.
- Connection information. Like the other HCA clients, the primary and secondary sever settings of the address and port of the server.
- Password. If a remote access password is assigned to the design loaded into the server then that password can be entered here and will be supplied to the sever upon connection.



---

## Primary and Secondary IP Addresses

As part of the settings for each client, the IP address of the server is specified. The application settings have a place for a primary and a secondary IP address.

When you make a client connection, an attempt is made using the Primary IP address. If that fails, the secondary IP address is tried.

The reason for having two IP addresses is best explained by an example. Let's see how a user called Bob uses the primary and secondary IP addresses.

When Bob is at home he accesses his home server using wireless. The IP address of that machine is always 192.168.2.100. It never changes. So he makes that his primary IP address. At home with wireless on, the connection is made using this IP address.

But when Bob is not at home he wants to access his server using BobsHome.homeDNS.org. So he makes this his secondary IP address. Now when he is out of wireless range, the connection is made using the phone 3/4G connection. The DynDNS service converts "BobsHome.homeDNS.org" to an IP address and then connects to the server.

Why couldn't Bob just use "myhome.homeDNS.org" as the one and only IP address? Because of the way his internet access works that just doesn't work. The "BobsHome.homeDNS.org" resolves to the address of his DSL/Cable modem – as it should – and then the modem gets confused by a home-side reference to its own address. This may or may not be similar to your situation and we offer it only as an example.



## Chapter 29

# Web Component

The HCA Web Component is an application installed when HCA installs that allows HCA to generate HTML pages from browser requests. By using a network and a browser, you can access your design to control devices, start and stop programs, look at the HCA log and many other actions. You have full control except you can't modify your design – add new devices, change programs, modify schedules, etc.

Once the HCA Web Server is started, http references to a configurable port number show a display which will allow you to interact with your HCA design.

With the HCA Web Server started you can view the web interface by directing your web browser to the following address: `http://xxx.xxx.xxx.xxx:yyyy` where `xxx.xxx.xxx.xxx` is the IP address of the computer running HCA and `yyyy` is the port number you've configured.

For example,

`http://169.254.180.32:9955/`

Hint: There are many methods to reference your computer by name rather than by IP address.

This can be especially helpful if your IP address changes over time. This is beyond the scope of the user guide. The HCA Web Server technical note will have more information on this. See the final section of this chapter.

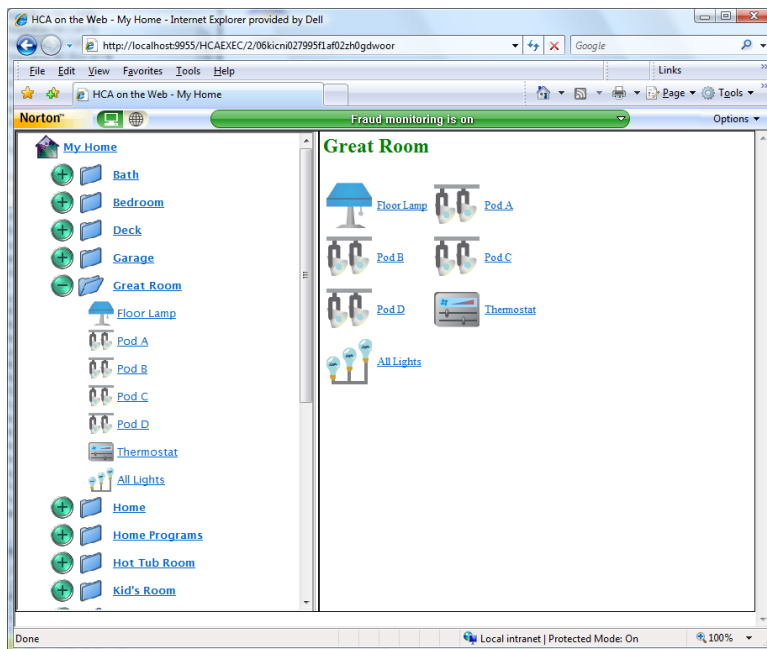
The Web Server creates pages that are either optimized for a big screen – the high resolution of a laptop or desktop computer – or the small screen on a handheld device.

Configuration of the HCA Web Server is simple and described later in this chapter.

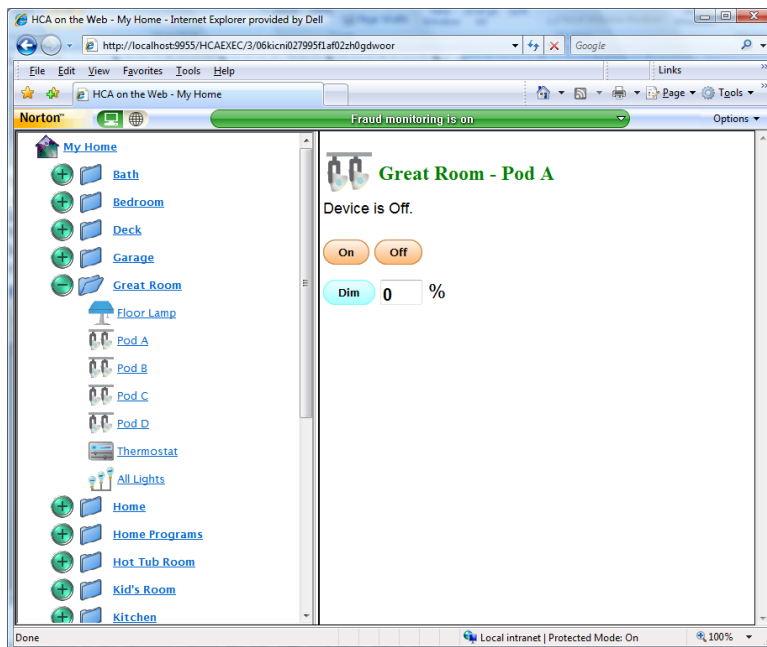
**NOTE:** This feature is in “legacy” support. This means that no support, improvements, or modifications to correct programs will happen. We suggest you use the Android, iOS or Windows clients instead of this.

## Accessing your design from a web browser

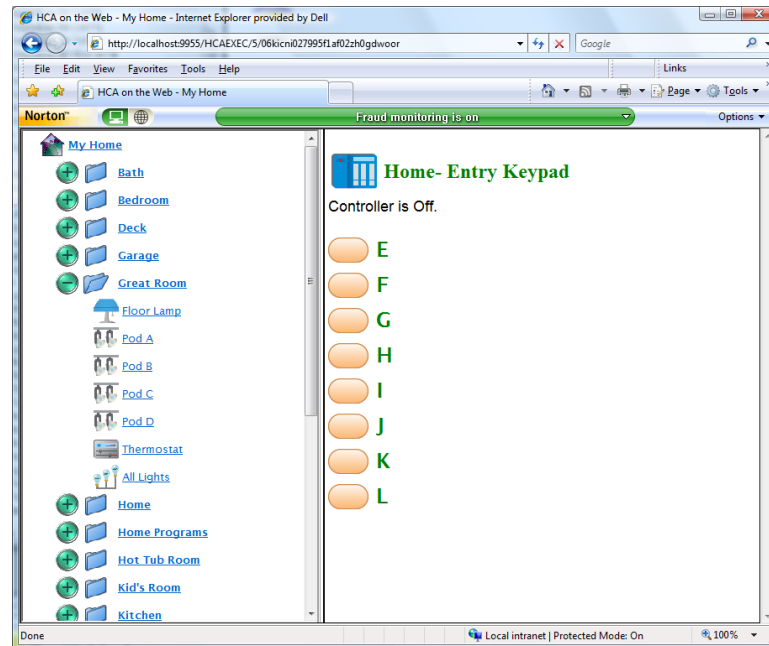
Once the Web Server is started your design appears in the browser when you connect to the correct port. For example:



Note that the left hand pane is very similar to displayed in HCA. The only difference is that there is a branch of the view for Logs and Flags. In this example, the right hand pane displays the contents of the folder. When you click on a device, program, group, or controller the right hand pane changes to display appropriate actions. For example, clicking on a dimmable device shows this:



In addition to the obvious actions like On, Off, and Dim, when you click on a controller a virtual keypad displays. Here is an example:



What is the purpose of a virtual keypad? After all a keypad just controls devices or sends signals to HCA that start programs.

The purpose of the virtual buttons is to simulate a physical button push. Imagine that in your home you have a keypad by the front door. You know that if you press the "A" button on the keypad a whole sequence of things happen. Just what are those actions are easy to forget. All you need remember is that when you want to do something – like get the house ready for a late night arrival – you push *this* button on *that* keypad.

When working remotely you can use the Virtual Keypads to initiate those actions. HCA both initiates the actions that the keypads would have controlled and also simulates reception of what the keypad sent. An example: You have a simple X10 keypad. If you pressed the 3<sup>rd</sup> key on the actual keypad it would send a B3.

Using the Virtual Keypad you press the 3<sup>rd</sup> button. HCA sends a B3 - to cause whatever the keypad controls to be controlled. And it also simulates receiving a B3. This allows programs that trigger on that reception to happen.

Virtual Keypads support simple X10 keypads, X10 KeypadLincs, Insteon ControlLinc and KeypadLincs, and UPB keypads. Virtual Keypads also implement switches that can be programmed to initiate scenes on rocker action – UPB and Insteon types.

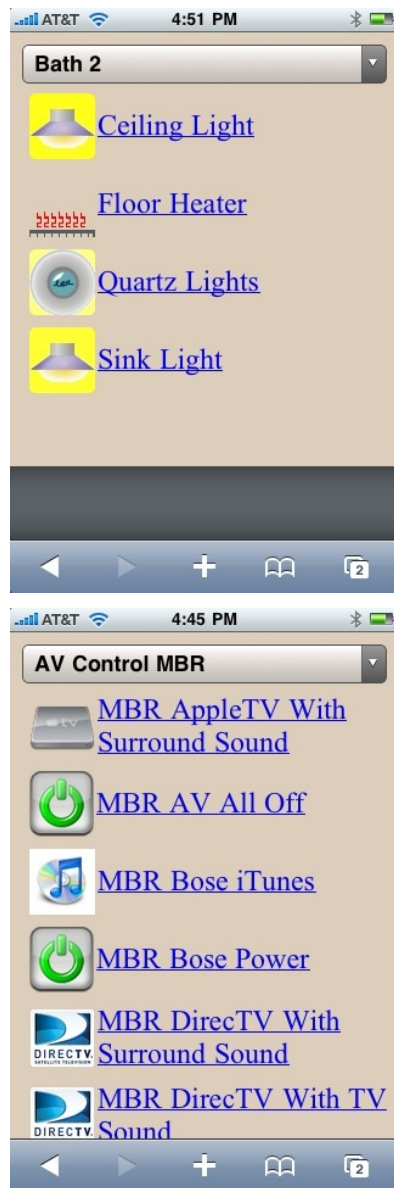
---

## Accessing your design from a mobile browser

You can also access your HCA design on a pocket PC, PDA, or other mobile devices. Once the Web Server is started, your design appears in a smaller format that the desktop browser would.

Across the top of the display is a dropdown that allows for selection of any folder or display in your design.

Here are some sample screens shown on a non-Android phone





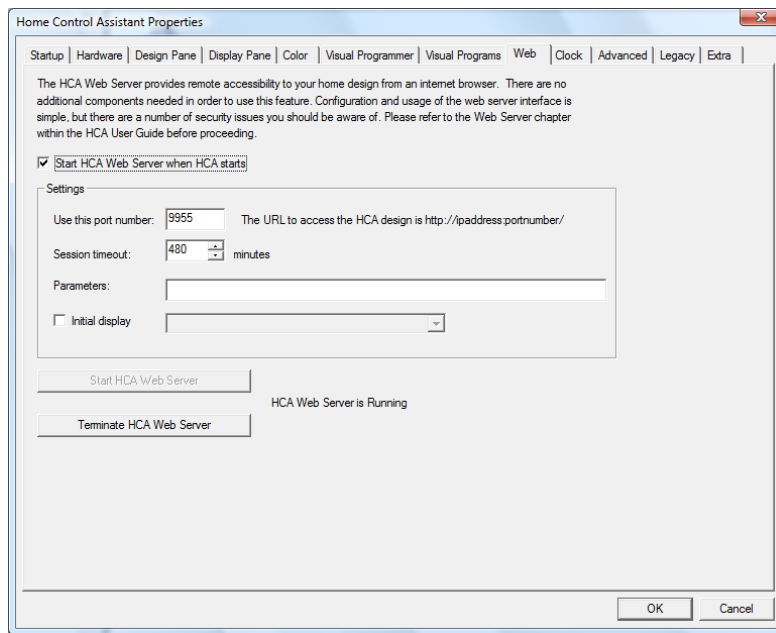
The mobile browser web interface has been optimized for the limited screen size. The main change is that the “design pane” has been removed and replaced drop down box. The basic actions of controlling devices on and off, starting and stopping programs, are all supported in the small screen display.

Hint: How does the Web Server know to display in large screen or small screen mode? When a browser first connects to the web server it supplies information necessary to make this decision. Normally you need do nothing to affect this but in some cases you may need to. The next section on configuration will describe this.

---

## Configuring the HCA Web Server

To start, stop, and configure the web server, press the *HCA Options* button from the application menu and choose the Web tab.



The options on this tab are:

- An option to start the Web Server each time HCA starts. When HCA terminates, the web server is also terminated.
- The Port number to use for the web server.
- The session timeout allows you to set the session length, in minutes. If a web browser is open with the web interface and is idle for the session length time, the session will expire. This prevents users from reusing a session that may have been left open unintentionally.
- Web Server parameters. These are described in the Web Server technical note.
- Initial display. You can chose a display or folder that forms the “home” page for the Web Server.
- Buttons to start and terminate the Web Server.

---

## Security Considerations

Using the web server opens up your home design to remote access. And that can be good! When “on the road” you can use the internet to get to your home and adjust lighting, temperature, A/V equipment, or just check status.

The potential problem is that without password control it may be possible for anyone to get access. There are two methods of adding security to your home design

---

## HCA Design Passwords

As explained in an earlier chapter of this User Guide, you can add passwords to your HCA design. In brief, the three possible passwords are:

- Remote Access password. Before the Web Server can even view your design, this password must be entered.
- Control password. No action – on, off, dim, start program, stop program, execute schedule entry, etc. may be done without this password being entered.
- Design password. No change in the design can be made unless this password is entered.

If you are using the Web Server you are strongly urged to add a Remote Access password to your design. To be even safer, adding a Control password prevents changes even if the remote access password is compromised.

---

## Hardware Security

In addition to the HCA security, the hardware that connects your home to the internet often has password facilities built-in. You should investigate them and if supported by your hardware you should enable them. This will block unauthorized access at the basic connect level.

---

## Web Server – More information

There are a number of parameters and URL options that can be used to tailor the web interface. These are explained in a technical note on the HCA Support web site.

Also in this technical note are some helpful notes about getting your browser to display your HCA design in the best possible manner.



## Chapter 30

# Getting Help

The Home Control Assistant can be a complex program. As you use HCA, you may find that you have questions about how to do some things, or you may find an area where HCA doesn't behave as you expect.

Outlined below are some procedures and resources available to help resolve problems if you find yourself in a situation where you think you need help.

Before you look too far for answers, make sure that you don't already have the information you are looking for. Check the User Guide carefully for the feature with which you are having problems. Make sure that you are following the directions completely and carefully.

---

### Our web site

If you are still having problems, the next place to look is the HCA Technical Support web site that contains extensive resources for HCA. The web site is located at:

**<http://www.HCATech.com>**

Some of the information available on the web site includes:

- Technical Notes on specific hardware.
- Troubleshooting guide for help resolving common problems.

If none of the Internet resources help you fully resolve your problem, the HCA technical support staff will work closely with you to solve any problems related to our software.

---

### Other considerations

While our support personnel do their best to help you with your software related problems, we know that on occasion a problem can be traced to hardware or to another software application. We will supply as much help as we can, but we can't provide support for products manufactured or published by another company. If you are having problems with your video display or printer, please make sure that you have the most current drivers for them. These can usually be found by contacting your computer manufacturer directly.

Finally, this product is designed to be used on PCs configured with the current versions of Microsoft Windows. This means that we tested for those configurations, and not every operating environment that you might encounter. Any non-standard hardware or software you have may be at the root of your problem. If possible, please disable all such devices to be sure that they are not causing your problem.

---

### Technical support

Unlike other companies you may have worked with, HCA provides technical support that is available using e-mail. You might find that e-mail is a more satisfying way to get technical support. You won't be put on hold, and the technician will have time to fully consider your problems and formulate an answer.

Technical support can be reached either from the bug reporting form on the support web site, or directly by sending e-mail to: **TechSup@HCATech.com**

To receive the fastest response to your technical questions, please include the responses to ALL of the following items in your e-mail.:

- What is the exact sequence of events that created the problem? Make sure that you can reproduce the problem by following the same series of steps.
- What is the version number of HCA? To get the version number from the Help menu, choose About HCA.
- What is the type of computer you are using and the version of Windows in use? The operating system version number can be found using the control panel *System* applet.
- Provide the exact wording of any error messages.

Software is very complex and a program like HCA is a very complex program. While the programmers try extra hard to get it right, sometime problems happen. If you find a problem please report it in a way that helps us find it and fix it.

Reporting a bug is an art that everyone should learn regardless of what software you are working with. It is vital that you provide all the information you can. The absolutely most important thing you can do is to tell us how to reproduce the bug. Sometimes this is easy. "Open my design file, select the program called *Bath Motion* select the display tab and HCA dies". Or "The empty speech queue option in the Visual Programmer *Speak* element doesn't work".

Now sometimes you can't reproduce the problem. So give us suggestions. For example, if you have a problem where occasionally HCA crashes try and determine what might have been happening at that time. Were you working with the user interface? Was HCA just sitting there? Do you think that it died when some program or schedule got triggered off? Guess. Theorize. Be a detective. Try and reproduce the problem yourself. Give us all the information you have: "HCA dies sometimes at night. I think it has to do with my bathroom motion sensor program. Attached is my HCA file. Send a M1 on the powerline to get my motion sensor program to start. It usually seems to die only around sunrise".

The more information you provide the quicker the problem can be identified and fixed.

Hardworking programmers everywhere thank **good** bug reporters.

## Glossary

**controller**

Something in your home that generates signals. This could be a keypad or a motion sensor for example.

**current**

In HCA terminology, current means the schedule that is in effect, the one that is working or active at the time.

**design**

In this user guide, refers to what you create using HCA to run your home. The design includes the displays, devices, programs, groups, schedules, schedule entries, and so on. Your design is stored in a file with the type HCA.

**design pane**

The left hand pane of the HCA main window. The design pane shows the names of all your displays, devices, programs, groups, controllers, schedules, and schedule entries

**device**

The term used in HCA to represent something that receives signals. A device can be any item that you want to control: a lamp, toaster, heater, etc. For more information, see the Devices Chapter.

**display pane**

The right hand pane of the HCA main window. This pane shows displays. Each display can be an arrangement of icons, messages, or HTML.

**DXF**

Abbreviation for drawing interchange format. This is a widely used file format that was originally developed for use with the AutoCAD program. DXF is also the standard file type/extension used for typical floor plan drawing programs.

**element**

*See program elements*

**entries**

Schedules have entries. There are two ways to create schedule entries: through the Schedule Entry Wizard, and with the Visual Scheduler. Refer to the Visual Scheduler chapter for more details.

**execute**

While a program is running, each element is executed, that is, whatever the element is supposed to do, is done. If it is an element that turns on a light, when the element is executed, the light comes on.

**expression**

A programming language like piece of text that assigns a value to a flag or tests the value of a flag. See the expression chapter

**flag**

A persistent state object that can be manipulated in a program. In a traditional programming language, this is called a variable. A flag can take on values that are dates, times, yes or no value, numbers, or text. A program can use flags like pieces of note paper. Each flag has a name and a value.

In the way that you might make notes to yourself while doing a complex task, programs can use flags to record things. A program can set a flag to a value and that value remains true as long as HCA is running. For example, one program can set a flag to Yes while it runs. Much later another program can test the value of that flag to see if it's Yes or No.

**group**

A collection of devices that you want to control as one system (unit). A group can have its own house code and unit code, and be controlled from the HCA or with a control panel. For more information, see the Groups chapter.

**HCA**

An abbreviation for the Home Control Assistant.

**home**

What the Home Control Assistant refers to your design as—your house, your home.

**house code**

Part of the address for your device, this is set on the module, and referenced in HCA. The house code works with a unit code.

**Insteon**

A protocol for sending and receiving signals over the powerline.

**Magic Module**

The name for a specific piece of hardware manufactured by Elk Products and supported by HCA. Also used to refer to all the various hardware items in the Magic Module family.

**members**

Groups have members. These are the individual devices within a group. *Group members* can be devices or programs, but **not** another group.

**module**

Refers to the electric switch that you plug in to the wall socket, and to which you connect a device. See the Devices chapter for more information.

**Network Device**

A device on your network – wired or wireless – that has an IP address that can be queried to see if it can be contacted or not. This can be useful to know, for example, if a mobile phone is in the house or not and from that if the phone's owner is home or not.

**PIM**

Powerline Interface Module. The name given to the UPB powerline interface. Also called a CIM

**PLC**

Powerline controller. The PIM is often referred to as a PLC. The PowerLinc also is called a PLC

**Powerlinc**

The name very over used by SmartHome to name just about anything – from a power adapter to the Insteon PowerLinc.

**primary address**

The address assigned to a device or controller.

**program**

A series of instructions for HCA that allow it to control several devices in sequence and be scheduled or activated remotely. Programs are logical, and sequential, and can be set up to have a delays or waits, and may have conditional steps. Programs have names, and can be started when HCA receives one of the triggers defined for it. For more information on creating programs, see the Programs chapter.

**program elements**

Each action that the program executes (does) is an element.

You draw programs by placing elements in the programming canvas, and link them together by drawing connecting lines. The program begins with the “Start Here” element and flows from element to element following the connecting lines in the direction of the arrows.

**properties**

Nearly all things in HCA have properties, that allow you to determine how the design works.

**repeat**

An element that allows one or more elements to be executed (done) a number of times.

**right-click**

Click the right mouse button, not the left, that you usually use. Of course, if your mouse is set up for left-handed use, click the left mouse button.

**run**

Between the time a program is started and the time it finishes, it is said to be running.

**scene**

A preset illumination level and (optionally) the rate at which the light changes illumination levels. Scenes are stored in the switch hardware and can be programmed by HCA.

**schedule**

Tells when things will happen. When devices will turn on and off, when programs will start, what the “time plan” for HCA is. HCA can have only one schedule current at any time. You can suspend a group or device or program so the current schedule doesn’t see it. For more information on schedules, see the Schedule and Visual Scheduler chapters.

**start**

When a program is started it begins running. Programs can be started in several ways. The first is from the HCA display by using the popup menu from right clicking a program icon in the display pane or on a program icon in the design pane. You can also start a program that has a house code/unit code address by sending that home code/unit code from a control panel.

**sun-relative time**

As used in the Visual Scheduler, refers to a time based upon sunrise or sunset. For example, you can set a schedule entry as 10 minutes before sunrise, or 30 minutes after sunset.

**test**

A test is an element that allows the program to analyze a condition and execute different elements based upon the outcome of that test.

**time markers**

On the Visual Scheduler, the three sets of markers below each bar. The ones on the left of the bar are used to create specific times (10 a.m., 3:15 p.m., etc.) and the ones to the right are used to create times relative to sunset and sunrise (at sunset, at 30 minutes before sunrise, etc.).

**trigger**

A event used to cause a program to start, a group to pass the signal to it’s members, or a device to send the command to it’s primary address.

**UPB**

Universal Powerline Bus. A protocol for sending messages over the powerline.

**unit code**

Part of the address for your device, this is set on the module, and referenced in HCA. The unit code works with a house code.

**Visual Programmer**

Is an accessible, visual method of constructing programs for HCA. It helps you see the sequential steps you are creating and placing in a grid that graphically represents the program.

**Visual Scheduler**

Provides a graphical manner for adding entries to a schedule. You can see the current on/off entries for all devices in a particular schedule. And you can see markers for the entries you are adding.

**X10**

A protocol for sending commands over the powerline.