



# HCA Tech Note 100

## Programs as devices

In HCA it is possible to create a program that in all aspects appears to operate as a device. Some possible uses are:

- Programs that interface with specific hardware but not appropriate for using the “class” mechanism
- A virtual device useful for controlling state when invoked from a voice assistant
- A program that adds extra features to the existing support of a device not available from the hardware itself

Programs that operate as devices support the operations of ON, OFF, and Get-Status. If they also configured to support dim, then Set to percent, change up by percent, and change down by percent, can be done.

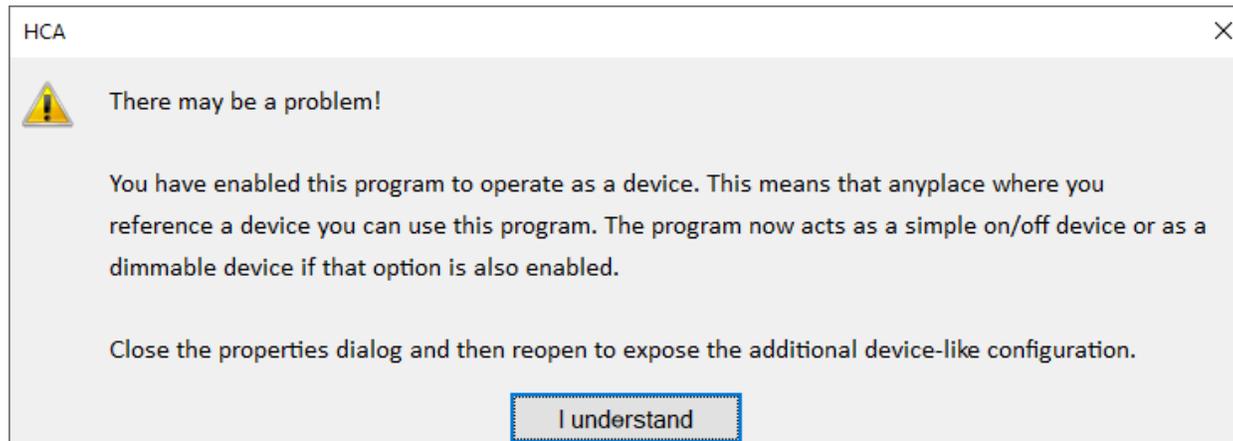
To create a program that operates like a device, an option on the Program properties is enabled:

Program Construction

- This program supports parameters and/or local variables  
Parameters are defined in the Begin-Here element. When started from another program using the Start-Program element, the actual objects or data are selected to use when elements in this program operate upon one of its parameters.
- The program makes a result available to the program that started it. Result returned to the calling program is the value in the first local variable
- This program operates as if it were a device supporting "on", "off", and "getstatus" operations. Requires processing supplied arguments
  - The program also supports "dim", "dimup", and "dimdown"

In addition to that option, the others in this group must also be enabled as the implementation must use program parameters and return a result. If the Dim option is not enabled, then the popup menus for device control show the Dim option disabled.

When this option is first enabled, and the device properties dialog is closed, this message appears:



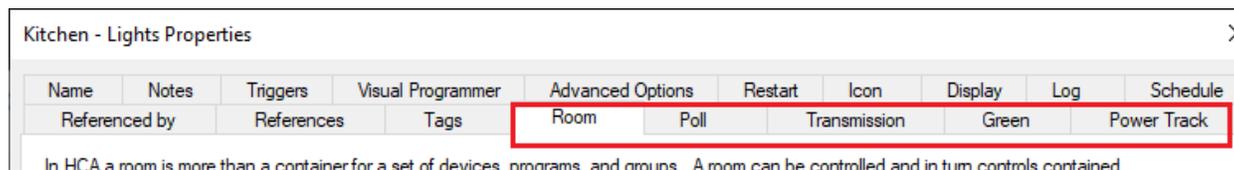
TechSup@HCATech.com

[www.hcatech.com](http://www.hcatech.com)



# HCA Tech Note 100

As it says, now that the program is configured to operate as a device. When opening the program's properties, additional tabs are now available giving access to configuration normally not available with programs.



These new tabs provide:

- (Room tab) The program-as-device can now participate in a room. That is, to be controlled by the room and optionally participate in determining the room state.
- (Poll tab) The program-as-device can be periodically polled for its state.
- (Transmission tab) When the program-as-a-device is controlled on, off, or to a percent, HCA can follow that command with a status check and reinvoke the program if the status is not as expected. This, obviously, is mostly useful with a program that controls hardware.
- (Green tab) An auto off specification can be configured for the program-as-a-device
- (Power Track) The device can be configured to keep track of its power usage and participate in power track charts.

In addition to these options, a program-as-a-device also:

- Is treated as a device on mobile clients. Rather than "Start" and "Stop" the client shows on, off, and set percentage.
- The restart tab of a program-as-device now has options more appropriate for a device and can be configured for how the device acts upon HCA startup: On, Off, Schedule catch-up, or no action.
- The program-as-a-device appears both on "All devices" and "All programs" displays showing its dual nature.
- The right-click menu on a program-as-device from the display and design panes now show device control operations rather than program operations.

## The program implementation

While the implementation of the program – how it does whatever it does - is up to you, there is a basic structure that must be followed. First, the program must accept parameters like this:

TechSup@HCATech.com

[www.hcatech.com](http://www.hcatech.com)



# HCA Tech Note 100

Begin Here Properties

When the Start-Program element is used to start this program it provides the actual data for these parameters. That can either be an object - a device, program, group, or room - or a simple value - a number or string.

In the elements of this program you refer to the parameters by their name. When the element executes the actual object or value provided by the starting program is used.

Parameters

Count: 2

Name	Type	Default for value parameter if not supplied by calling program
Parameter 1: action	Value	HCA invokes with "on", "of", "dim", "dimup", "dimdown", "ξ
Parameter 2: percent	Value	Percent to go to for "dim", percent inc/dec for dimup and dir

There must be two value parameters: The action and the percent.

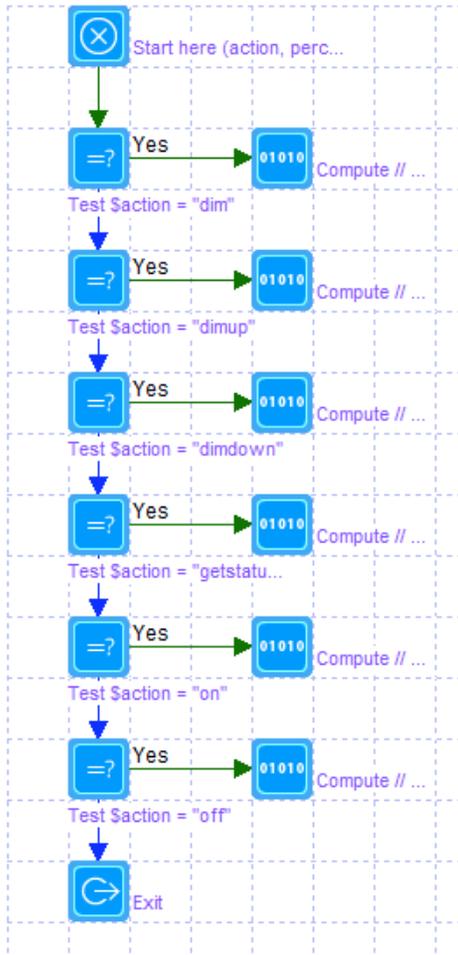
HCA invokes the program to carry out the action and passes these values to the program:

"Action" parameter value	"Percent" parameter value
on	Ignore for "on"
off	Ignore for "off"
dim	% to go to: 0 - 100
dimup	% to increase: 0 - 100
dimdown	% to decrease: 0 - 100
getstatus	Ignore for "get-status"



# HCA Tech Note 100

The general structure of the program is:



Each Variable-Test element checks the value of the parameter, for example:

TechSup@HCATech.com

[www.hcatech.com](http://www.hcatech.com)



# HCA Tech Note 100

The dialog box is titled "Variable Test Properties" and contains the following fields:

- Test:** A dropdown menu with the value "\$action".
- For:** A dropdown menu with the value "Equal to".
- With value:** A text input field containing the value "dim".

At the bottom, there are "OK" and "Cancel" buttons.

Each Compute element carries out the operation however the program wants to. The program must, as part of that action, set the state of the program with the new state. For example, the dim operation:

The dialog box is titled "Compute Properties" and contains the following elements:

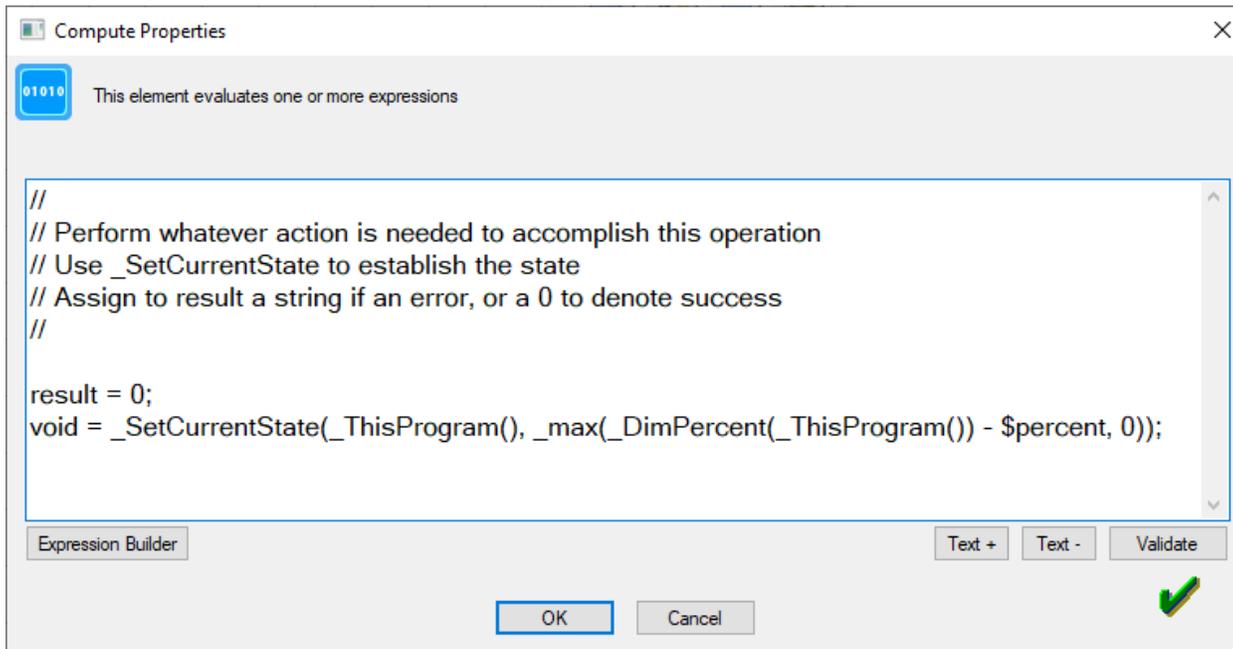
- Icon:** A blue square with the binary code "01010".
- Description:** "This element evaluates one or more expressions".
- Code Editor:** A text area containing the following code:

```
//  
// Perform whatever action is needed to accomplish this operation  
// Use _SetCurrentState to establish the state  
// Assign to result a string if an error, or a 0 to denote success  
//  
result = 0;  
void = _SetCurrentState(_ThisProgram(), $percent);
```
- Buttons:** "Expression Builder", "Text +", "Text -", "Validate", "OK", and "Cancel".
- Checkmark:** A green checkmark icon in the bottom right corner.

**The use of the `_SetCurrentState` is required as HCA doesn't automatically update the state like it does for a device.** Note that programs must ensure that the percent doesn't exceed the range of 0 to 100. For example, the action of the dimdown operation is done in this Compute element:



# HCA Tech Note 100



Use of the `_max` function ensures that the percentage change doesn't cause the state to exceed 100%.

## Handling errors

**The first local variable of the program is special as this is used to communicate to HCA if the operation was successful or not.** Return a numerical value of zero if the operation is successful or a text result if it failed. The string is logged, and an alert records that the operation failed. The UI also designates that there was device problem – with a red box around the icon.

## A few important considerations

It might be a good idea, once the program operates as expected, to suppress the logging of the program elements. The log then shows the expected operations for a device when the program-as-device is controlled but not the elements that comprise the program.

TechSup@HCATech.com

[www.hcatech.com](http://www.hcatech.com)



# HCA Tech Note 100

---

Programs that implement devices always have their actions “serialized”. That is, there is never more than one instance of the program running at one time. For example, if a program is controlled ON by a schedule and at the same time a background poll for its status happens, one of the operations begins and the other waits until the program has completed the first operation, then the program starts again to handle the second operation. This is all handled internally to HCA and nothing needs to be done to enable this. This means that the choice of the option on the “triggers” tab for multiple triggers is not important.

## How does a program-as-device differ from other programs and class programs?

A program is a program, but their intended use determines how HCA interacts with them.

### Plain vanilla programs

- Used to respond to events
- Used to sequence complex actions

### Class programs

- Typically used to implement hardware not supported directly by HCA
- Implements actions that look like a device: On, Off, Set-to-%, Get-Status
- Takes 4 parameters: Class name, device id, operation, data
- The class program exists separate from the devices that it controls
- A design typically contains one or more devices that are handled by the single class program
- When adding a new device, the class that supports it is one of the choices at that step in the Add Wizard.
- Devices are tagged with some id that the class program uses to direct its action to that specific device

### Programs-As-Devices

- Implements actions that look like a device: On, Off, Set-to-%, Get-Status
- Takes 2 parameters: Operation, data
- The program and the device are the same
- Unlike a class program, each program operates like a single device.
- For more than one “device” the program must be duplicated
- If the program implements hardware, it understands how to control a single hardware device. This differs from a class program which is invoked with information about which hardware device performs its action on
- Truly acts as a device with all the facilities of a device: room, poll, status, startup action, etc.

##end##

TechSup@HCATech.com

[www.hcatech.com](http://www.hcatech.com)