HCA Tech Note 100

Programs that operate as devices (updated 17-May-18)

Programs have many applications in HCA as they respond to events. They also can be constructed to operate more like devices. Using the techniques in this note, programs like devices, they can be turned on, turned off, set to a percentage, and have a current state.

One use for this is constructing a program to be controlled by a voice assistant. In this way, the program fits into the voice assistant model by looking like a non-dim or dimmable switch.

Program icon management

Normally when a program is running, its icon shows the ON representation of the icon selected for it, and when not running it shows the OFF representation of that icon. To make the program act like a device, the icon should instead reflect the "percentage" state of the program rather than if it is currently executing or not.

What is the "percentage" for a program? It is just a number between 0 and 100 – like a device – that is managed by HCA, and optionally by your program. HCA uses this "percentage" to show the correct representation for the program icon. Like a device, if the percentage is 0 then the OFF representation is used. If it is between 1 and 99 percent, the DIM representation is used (or if no DIM representation is available, the ON representation is used). If the percentage is 100 then the ON representation of the icon is used.

To make this happen and correctly reflect the state of the program rather than if it is running or not, select "not changed automatically" for the management of the icon. That tells HCA not to perform the default action on the icon. This is on the Advanced Tab of the program's properties.

Management of this program's icon
- ⦿ Not changed automatically
- ◯ Changed automatically - shows ON representation when program running and OFF representation when not running

Making a program operate like a non-dimmable switch

The first step is to add two triggers to the program: One to start it from an ON event and a second to start it from an OFF event. The "Generic Trigger" is perfect for this. On the triggers tab of a program press the "Add Trigger" button and select as the trigger type "Generic".

Trigger Type: Generic

Generic triggers are used to respond to a user's actions starting a program from the HCA interface, a client interface like a remote application, or by voice. Having the trigger lets you test on the starting condition.
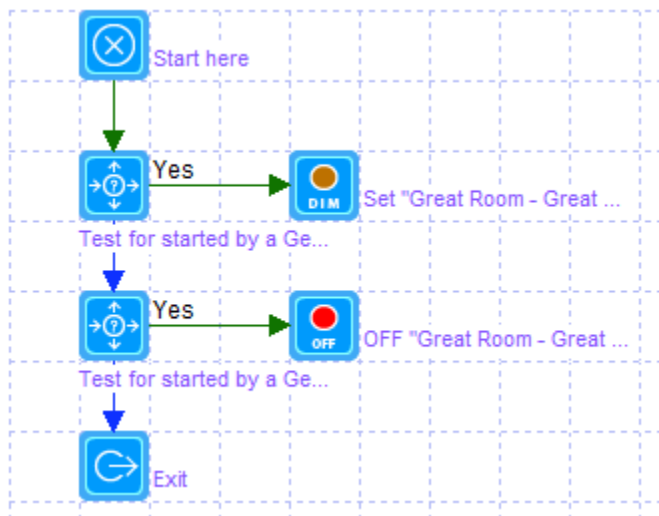
- ⦿ Started by an OFF
- ○ Started by an ON

You should add both an OFF and ON triggers. The triggers tab then shows:

Trigger	Edit	Delete
⮞ Generic Trigger: Started by OFF	Edit	Delete
⮞ Generic Trigger: Started by ON	Edit	Delete

Within the program you can then test the starting trigger and perform different actions based upon the outcome of the test. Like this:

In this example, in the "ON" case a device is controlled using the DIM element and in the OFF case the device is controlled OFF.

But this is just an example! The ON and OFF paths could do anything you want – control multiple devices, test for the time of day and do different actions at different times. It is all up to your needs and creativity.

Making a program operate like a dimmable switch

To handle DIM in addition to ON and OFF events, the program can also process events to set a percentage, increase the percentage, or decrease the percentage. To support this, several settings on the "Advanced tab" must be configured:

```
┌─ Execution Options ──────────────────────────────────────────────────────────────────────────┐
│  ☑ 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  │
│                                                                                                 │
│  ☑ The program supports Dim                                                                    │
└─────────────────────────────────────────────────────────────────────────────────────────────┘
```

The key settings are:

- The program supports parameters and/or local variables
  HCA provides the program commands about dimming and the program must process those. This is detailed in the next section

- The program supports dim
  This allows the HCA UI to properly show the DIM option for a program

In the program, open the properties of the Start-Here element and define two parameters. The names are up to you, but they must be value – not object - parameters. Like this:

Begin Here Properties

When the Start-Program element is used to start this program it provides the actual data for these parameters. The data can either be an object - a selected 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 and when the element is executed the actual values supplied by the starting program are used.

Provide each parameter a unique name and specify if it is used for an object or value.

Parameter count: 2

Parameter 1 name: action   [Value ▼]

Parameter 2 name: level   [Value ▼]

[OK]   [Cancel]

When the program executes to handle a percentage change operation, the first parameter receives a string whose contents are "dim", "dimup", or "dimdown". The second parameter receives a value that is the percent to go to - for "dim" – or the percent to change by – for "dimup" or "dimdown".

You can test the parameter value using the Compute-Test element within the program, like this:
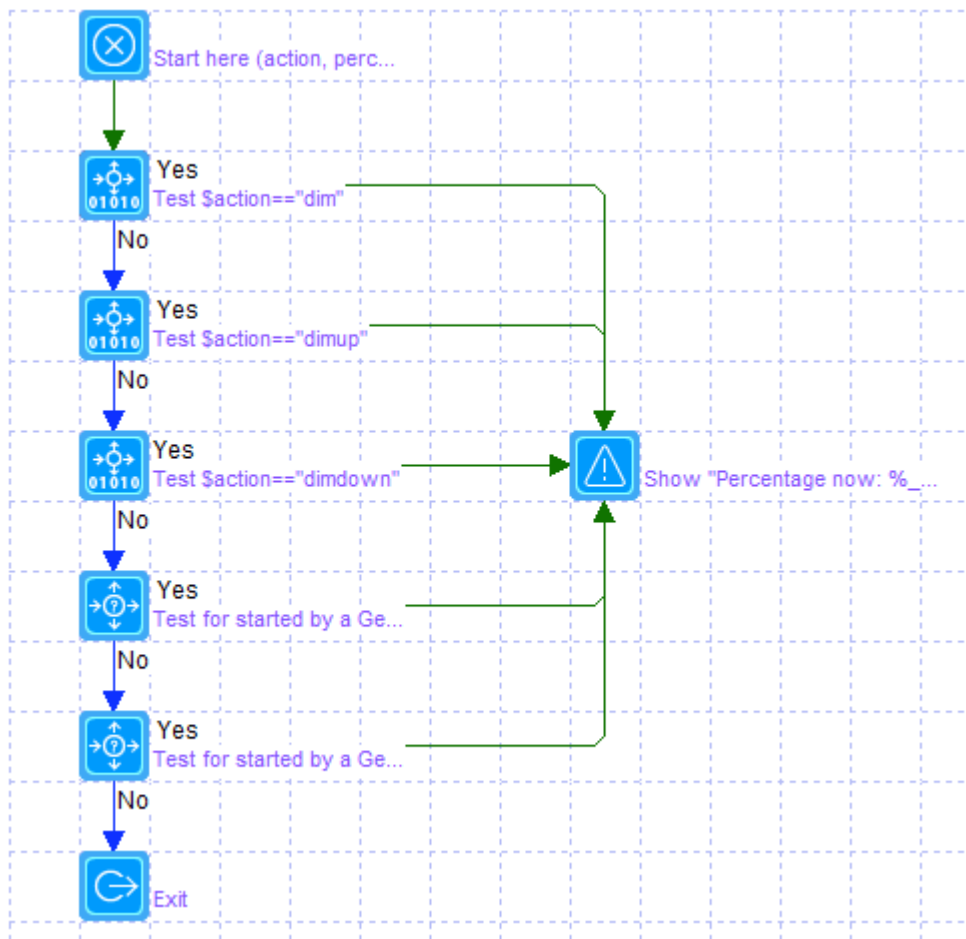

As you can see, the action argument is tested to see what it contains and there are separate tests for "dim", "dimup" and "dimdown". If none of those tests succeed, then the same trigger tests as described in the non-dim case – described in the previous section - are made. This covers all possible operations.

The program's actions in each of these cases is up to you. To go a bit deeper into this example, here is the first test element:
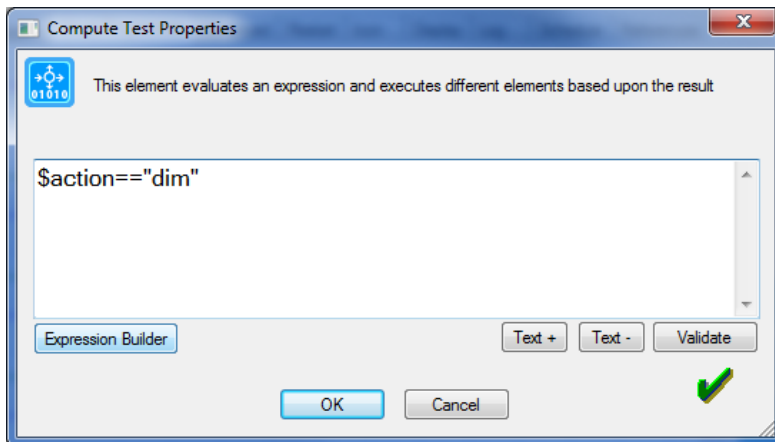
If you are not familiar the method used in expressions to refer to program parameters (hint: you prefix the name with a "$"), or create expressions, it is all covered in the "program with parameters" and "expression" technical notes. The other tests are similar to this but test for different commands.

HCA helps you maintain the programs "percentage"

As HCA does with devices, internally if the program is turned "ON", HCA updates the program's percentage to be 100. If it is turned "OFF", the percentage is set to 0. Similar adjustments are made by HCA to the program's state if dim, dim-down, or dim-up operations happen.

However, the program's percentage can be also be changed by the program itself using the _SetCurrentState expression like this:
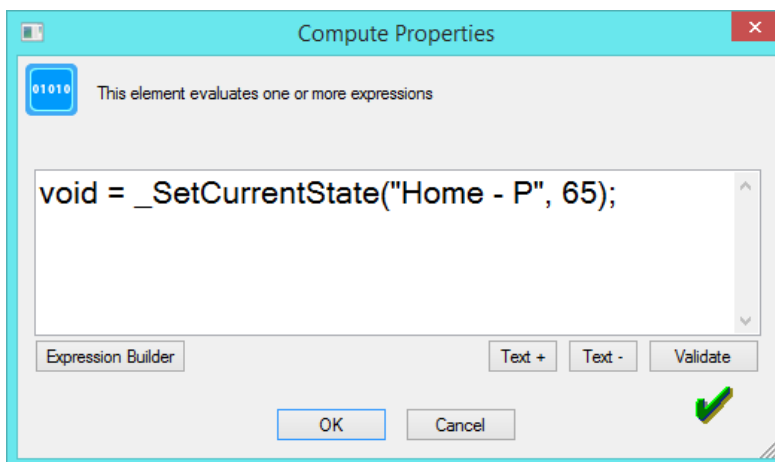
In this example, when the element executes, the program's percentage becomes 65%

A use for this, for example, is a program that implements only discrete percentage possibilities – like 0%, 33%, 66%, and 100%. If the program's percentage was at 33% and a request was made to increase by 10%, HCA would internally set the percentage to 43% before your program is started. When the program executes, it could either ignore the increase and set the percentage back to 33% or do whatever it does and then set the percentage to 66%. Again, that's all up to your needs.

Using a dimmable program in your HCA design

Once your program is marked as dimmable, then HCA makes that program available for the dim operation in several ways:

- "Dim" is now available from the right-click context menu for a program
- The program can now be scheduled for dim
- The program can now be referenced in the Multi element
- The program can now be referenced in the Dim element
- The program can now be referenced in the Test element when testing for dim percentage
- The program can now be referenced in the _State, _DimPercent, _SetCurrentState expression functions

In each of these cases, HCA starts the program, passing in the appropriate values for the two parameters, and it is the responsibility of the program to test for those values and carry out the operation.

Finally, the program can also be used with these methods in the Program object
- DimToPercent
- DimUpPercent
- DimDownPercent
- State
- DimPercent

Just remember in a program that doesn't have the "Not changed automatically" option selected for its icon management, the program's percentage is set to 100% when executing and 0% when not executing.

Using Programs with a Voice Assistant

Note: What follows uses Amazon Alexa, but the same ideas apply to all supported voice assistants.

To start a program, you can say to Alexa:

Alexa, turn on <program name>
Alexa, turn off <program name>

Using the "turn on" phase starts the program with an ON trigger. Using the "turn off" phrase starts the program with an OFF trigger. Your program should use the TEST element to test for these cases as shown above.

If you construct the program as dimmable, then in addition to "Turn on" and "Turn off", you can also use the Dim and Bright phrases with programs like this:

Alexa, increase <program name> by <percent>
Alexa, decrease <program name> by <percent>
Alexa, dim <program name>
Alexa, brighten <program name>
Alexa, set <program name> to <percent>

In each case the program is started, and arguments given to the program with the action and percent from the phrase. These values should be tested in the program to perform the necessary actions as shown in this table:

What you say	Parameter 1	parameter 2
Alexa, increase <program name> by <percent>	dimup	percent value
Alexa, decrease <program name> by <percent>	dimdown	percent value
Alexa, dim <program name>	dimdown	always 25
Alexa, brighten <program name>	dimup	always 25
Alexa, set <program name> to <percent>	dim	percent value

##end##