

## Markers, Events and Parameters

There are three devices in ALCW that look as if they have similar functions, but have subtle differences.

**Parameters** are external variable/value pairs obtained from permanent storage for a particular profile and are offered to the profile at the point of execution. If the profile references these variable identifiers then they will [initially] take those values. If the profile changes those variables' values then those values are written back to the parameter storage, to be accessed on the next run of the profile.



Parameters can be useful where a profile takes different execution paths depending upon a value, where it might not be convenient or appropriate to change the profile code just to change that value. Other uses may be as a counting mechanism.

Consider this code:


```
$my_param_1 default 10;
#my_glob_1 default "Unknown";

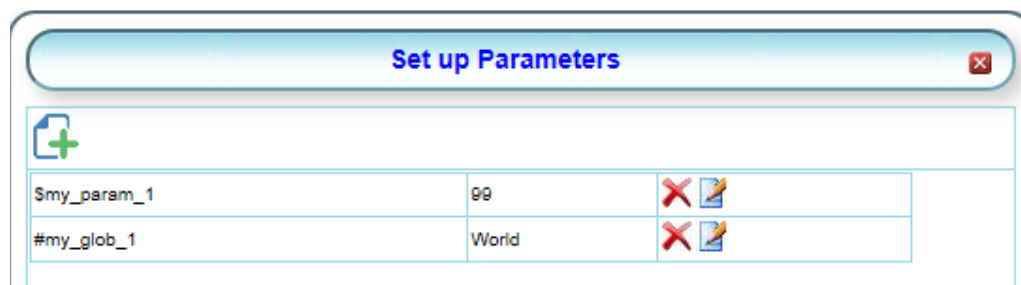
console("Hello {#my_glob_1} we are {$my_param_1}");

exit true;
```

Without parameters this runs with the result:

```
Hello Unknown we are 10
```

Now consider we have parameters for this profile set up using the parameter dialog, under the icon:  the set up looks like this:



Now the run results in this:

```
Hello World we are 99
```

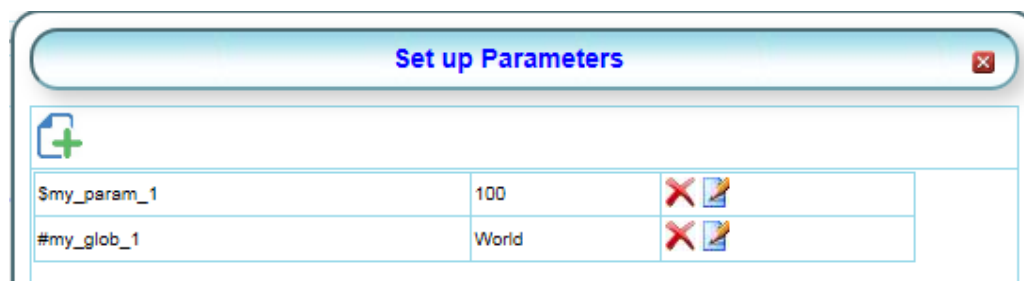
Now consider that the code updates the parameter, \$my\_param\_1, in this way:

```
$my_param_1 default 10;  
$my_param_1 = $my_param_1 + 1;
```

Then the code would result in:

```
Hello World we are 100
```

We can see that the parameter has returned to permanent storage:



And if we run again, then we continue to increment \$my\_param\_1. Like this next run:

```
Hello World we are 101
```

Note: the parameter values are only returned on successful runs – that is, when there is an exit true.

## Markers



Markers have a very similar functionality to parameters. The differences are that a marker maintains a date-time stamp against the marker; it can have an expiry date; and it is returned to permanent storage at the point of call of marker\_value, whether or not the code completes successfully.

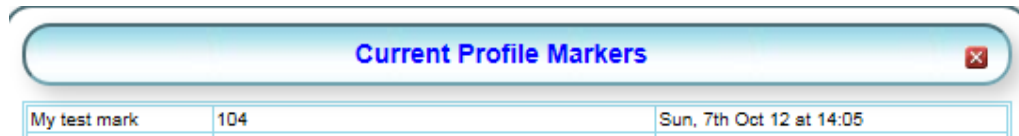
Consider the code above extended:

```
$my_param_1 default 10;  
#my_glob_1 default "Unknown";  
  
$my_param_1 = $my_param_1 + 1;  
  
console("Hello {#my_glob_1} we are {$my_param_1}");  
  
$my_mark = marker_value("My test mark");  
  
console("My last run was {$my_mark}");  
  
marker_value("My test mark", $my_param_1);
```

Generates the result:

```
Hello World we are 105  
My last run was 104
```

And we can observe the marker in the marker dialog:



## Events



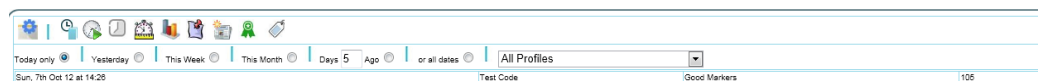
Events differ from the above parameters and markers in that they are accessible from other profiles. They are generally used to provide validation over a set of profiles, or as an alert to an error condition to other profiles – perhaps to prevent a profile running if it detects an event from another profile with a particular value.

An Event can be considered a true global, in the sense that it extends beyond the scope of the issuing profile.

We further extend our example code to include the issue of an event:

```
$my_param_1 default 10;  
#my_glob_1 default "Unknown";  
  
$my_param_1 = $my_param_1 + 1;  
  
console("Hello {#my_glob_1} we are {$my_param_1}");  
  
$my_mark = marker_value("My test mark");  
  
console("My last run was {$my_mark}");  
  
marker_value("My test mark", $my_param_1);  
  
if $my_mark is < $my_param_1 issue_event("Good Markers",  
$my_mark) else issue_event("Good Markers", 0);  
  
exit true;
```

We can observe that marker in Monitor=>



We will create another profile to collect that marker from the issuing profile:

```
$events = get_events("Good Markers", true, 0, 'Test Code');  
dump($events);
```

Here we are saying collect (all) the events named 'Good Markers' issued by the profile 'Test Code' today. See the details for get\_events for the other parameters, but they are effectively:

```
get_events("event name", today[true or false], days ago [if  
today is false], "issuing profile name");
```

Our collecting profile code produces the following in the structure \$events:

```
.[0].[ItemId] PF4fe3151d2b250  
.[0].[IssuedDate] 2012-10-07 14:26:54  
.[0].[EventCode] Good Markers  
.[0].[EventValue] 105
```