Rocket Uniface Library 10.4

$status

Return or set the current condition code.

`$status`

`$status` = *Expression*

Example: `if ($status < 0)` …

Return Values

An integer value. If a decimal value is assigned to `$status`, Uniface rounds it to the nearest integer.

Use

Allowed in all component types.

Description

`$status` returns a condition code that indicates the result of a runtime action, such as an I/O request. In general:

- A negative value in `$status` indicates an error. In this case, the function `$procerror` gives further information about the cause of the error and `$procerrorcontext` gives details about the exact location where the error occurred.
- `0` indicates a successful operation.
- A positive value indicates a warning or information.

Although you can assign a value of `$status` to pass codes to another ProcScript module or component, doing so resets the current value of `$procerror`, so the ProcScript error status and context are lost.

In the Debugger, `$status` can be accessed directly or as variable `$100`.

ProcScript Modules and `$status`

Each time a ProcScript module is activated, `$status` is set to `0`. When the ProcScript module ends, Uniface checks `$status`, because the value can influence what happens next.

The end value of `$status` has different effects with different triggers. For example, if the ProcScript code in a remove trigger ends with `$status` less than `0`, the occurrence is not removed.

If the module was invoked by a statement, such as `call` or `activate`, the return value of the module is assigned to `$status`. If the module was invoked using an inline construction (such as an instance handle or a function argument), the return value is returned inline, and the value of `$status` keeps the value as set inside the Proc module.

For more information, see Return Values, Status Values, and ProcScript Errors.

Example: Conditional Processing Based on `$status`

The following example inspects the value of `$status`, set by the `store` statement in the Store trigger. The check on `$status` allows the ProcScript to handle error situations.

```
; Store trigger
store
if ($status < 0)
 message "Store error number %%$status."
 rollback
else
 message "Store complete."
 commit
endif
```

Related concepts

- = (compute)
- return
- store
- $dberror
- $error
- $procerror
- $procerrorcontext
- $result