

# Conditional Statement

This command was modelled after the [For\\_Each](#) command in the sense that there is a start (Conditional\_Statement) and an ending event (Conditional\_Statement\_End).

When the pipeline comes to an Conditional\_Statement command, it will evaluate the expression using the same syntax as the [Evaluate\\_Expression](#) command. If the expression is TRUE (returns any value other than zero), all of the commands between the Conditional\_Statement and Conditional\_Statement\_End commands will be executed. If the expression is FALSE (returns a zero), all the commands that are between the Conditional\_Statement and Conditional\_Statement\_End commands will be skipped.

## Pipeline Command

The **Conditional Statement** command can be found in the Pipeline Workshop within the Pipeline Control folder as so:

```
Conditional_Statement
/ITERATION_PARAMETER_NAME=
! /EXPRESSION=
! /AS_INTEGER=TRUE
;

Conditional_Statement_End
/ITERATION_PARAMETER_NAME=
;
```

## Command Parameters

The following table shows the command parameters and descriptions:

| Parameter                       | Description                                                                                     |
|---------------------------------|-------------------------------------------------------------------------------------------------|
| <b>Iteration Parameter Name</b> | The name of the parameter that the conditional statement will apply to (defined before command) |
| <b>Expression</b>               | Expression for the operation to be applied to the selected signals                              |
| <b>As Integer</b>               | True or False: Expression value is stored as integer                                            |

## Pipeline Example: Processing Walking and Running Trials

This script will prompt the user to specify if walking or running trials are being processed. Based on the user's selection, different processing will be completed. In this example, the pipeline parameter TEST will be set to 1 if the condition is Walk, and set to 2 if the condition is Run.

```
Prompt_For_Pipeline_Parameter_Value
/PIPELINE_PARAMETER_NAME=CONDITION
```

```
/PROMPT=Select Condition:
/DATA_TYPE=STRING
/DEFAULT_VALUE=Walk
/USE_POSSIBLE_VALUES=TRUE
/POSSIBLE_VALUES=Walk+Run
! /USE_UNITS=FALSE
! /DEFAULT_UNITS=
! /USE_POSSIBLE_UNITS=FALSE
! /POSSIBLE_UNITS=
! /CONVERSIONS=
;

Conditional_Statement
/ITERATION_PARAMETER_NAME=STATEMENT_1
/EXPRESSION= "&::CONDITION&" == "WALK"
;

Set_Pipeline_Parameter
/PARAMETER_NAME=TEST
/PARAMETER_VALUE=1
! /PARAMETER_VALUE_SEARCH_FOR=
! /PARAMETER_VALUE_REPLACE_WITH=
! /PARAMETER_VALUE_PREFIX=
! /PARAMETER_VALUE_APPEND=
! /MULTI_PASS=FALSE
;

Conditional_Statement_End
/ITERATION_PARAMETER_NAME=STATEMENT_1
;

Conditional_Statement
/ITERATION_PARAMETER_NAME=STATEMENT_2
/EXPRESSION= "&::CONDITION&" == "RUN"
;

Set_Pipeline_Parameter
/PARAMETER_NAME=TEST
/PARAMETER_VALUE=2
! /PARAMETER_VALUE_SEARCH_FOR=
! /PARAMETER_VALUE_REPLACE_WITH=
! /PARAMETER_VALUE_PREFIX=
! /PARAMETER_VALUE_APPEND=
! /MULTI_PASS=FALSE
;

Conditional_Statement_End
/ITERATION_PARAMETER_NAME=STATEMENT_2
;
```

## Notes

The `ITERATION_PARAMETER_NAME` is used to distinguish between different `Conditional_Statement` commands so that `Conditional_Statement` and `Conditional_Statement_End` commands can be unambiguously associated with each other.

**NOTE:** If there is more than one active file then the `Conditional_Statement` command will be skipped. This is because the expression might be `TRUE` for some files and `FALSE` for others, leaving Visual3D in an ambiguous place.

**NOTE:** This command was previously called the `Expression_Loop`. Scripts referring to this old command name will still work.

From:  
<https://has-motion.com/wiki/> - Wiki Documentation Site

Permanent link:  
[https://has-motion.com/wiki/doku.php?id=visual3d:documentation:pipeline:pipeline\\_commands:conditional\\_statement](https://has-motion.com/wiki/doku.php?id=visual3d:documentation:pipeline:pipeline_commands:conditional_statement)

Last update: 2026/02/04 16:38

