

Set Pipeline Parameter From Expression

This command can be used to set the value of a specified parameter by defining it with an expression. This allows the value of a parameter to be set to anything from an explicit value to an expression containing other signals.

Pipeline Command

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

```
Set_Pipeline_Parameter_From_Expression
/PARAMETER_NAME=
/EXPRESSION=
! /AS_INTEGER=TRUE
;
```

Command Parameters

The following table shows the command parameters that can be used to define the command and their descriptions:

Parameter	Description
Parameter Name	The name of the Pipeline Parameter. The parameter is referred to by other commands using two colons (::) in front of the parameter name
Expression	The expression used to define the parameter value
As Integer	True or False: Express the resulting parameter value as an integer

For more information on how expressions can be used and defined see [Expressions Overview](#).

Syntax Notes

```
Set_Pipeline_Parameter_From_Expression
/PARAMETER_NAME=t_Mass
/EXPRESSION=METRIC::PROCESSED::HEIGHT * 4.6
/AS_INTEGER=TRUE
;
```

The pipeline parameter contains the value of the expression.

Getting the syntax correct can be challenging because there can be a conflict between the & as a concatenation parameter in a pipeline parameter and the & as a boolean “and” operator.

For many pipeline commands Visual3D recognizes the conflict and handles the situations in a context specific way.

Visual3D cannot make that distinction and “bad” things happen.

Consider the following which uses string comparisons and a pipeline parameter in the expression:

```
Set_Pipeline_Parameter_From_Expression
/PARAMETER_NAME= MVAL
/EXPRESSION=(&::TAG_NAMES& == Utility) * 4.6
/AS_INTEGER=FALSE
;
```

The goal was to set MVAL to 4.6 when the pipeline parameter TAG_NAMES was equal to the string “Utility”;

One solution is to separate the pipeline parameter from the expression.

```
Set_Pipeline_Parameter
/PARAMETER_NAME=TEMP
/PARAMETER_VALUE=(&::TAG_NAMES& == Utility)*4.6
;
```

```
Set_Pipeline_Parameter_From_Expression
/PARAMETER_NAME=t_Mass
/EXPRESSION=: :TEMP
/AS_INTEGER=FALSE
;
```

Another solution, which is even trickier in terms of syntax is:

```
Set_Pipeline_Parameter_From_Expression
/PARAMETER_NAME=SCOTT
/EXPRESSION=("&::TAG_NAMES&"==" Utility")
! /AS_INTEGER=TRUE
;
```

This forces Visual3D to recognize the strings as strings and the & is treated correctly.

Example: Get global metrics

Global metrics can be set to a pipeline parameter by setting “GLOBAL” in front of the metric name. More information about expressions can be found [here](#).

```
Set_Pipeline_Parameter_From_Expression
/PARAMETER_NAME=KATIE
/EXPRESSION=GLOBAL::METRIC::PROCESSED::MASS
```

```
/AS_INTEGER=FALSE  
;
```

From:

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

Permanent link:

https://has-motion.com/wiki/doku.php?id=visual3d:documentation:pipeline:pipeline_commands:set_pipeline_parameter_from_expression

Last update: 2026/02/06 21:01

