Pipeline parameters can be created in the pipeline and used as variables in other commands in the same pipeline.

Set_Pipeline_Parameter			
/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		
/Parameter_Value=	The value of the parameter. If a “+” delimeter is used in the value, the text string is maintained so that the Pipeline parameter is interpreted as multiple values when used in a command. If a “+” delimeter is used in the value, the entries in the line are concatenated, and will be treated as a single value when they are used in another command.		
/Parameter_Value_Search_For=			
/Parameter_Value_Replace_With=	A string of characters that will replace the “search for” string. This can be useful if you want to replace some of the terms within an existing Global Parameter value		
/Multi_Pass=	FALSE	-	;

Example: Set Pipeline Values via Search and Replace

This example sets pipeline parameters and new values via search and replace. The pipeline parameter **original** and the string value **replacethexxxx**, is set to the pipeline parameter **newparameter** and replaced with the string **replacethezzzz**.

```
!Set the Pipeline parameter original to the value replacethexxxx
Set_Pipeline_Parameter
/PARAMETER_NAME=original
/PARAMETER_VALUE=replacethexxxx
;

! Set the Pipeline parameter newparameter to a new value replacethezzzz by
replacing
! part of the string zzzz for xxxx
Set_Pipeline_Parameter
/PARAMETER_NAME=newparameter
/PARAMETER_VALUE=replacethexxxx
/PARAMETER_VALUE_SEARCH_FOR=xxxx
/PARAMETER_VALUE_REPLACE_WITH=zzzz
;
```

Example: Set Pipeline Parameter Value to a String

For example, the following command creates a global parameter **SIGNALS** which contains the string **RightFoot1+RightFoot2+RightFoot3**. A low pass filter is run on the pipeline parameter **SIGNALS**.

```
Set_Pipeline_Parameter
/Parameter_Name=SIGNALS
```

```
/Parameter_Value=RightFoot1+RightFoot2+RightFoot3
;

! All three signals will be processed with low pass filter
Lowpass_Filter
/Signal_Types=ORIGINAL+ORIGINAL+ORIGINAL
/Signal_Names=::SIGNALS
;
```

Example: Search and Replace + Sign

The plus sign typically indicates different a list of values, but sometimes a user may want to search and replace a plus sign. This is allowed if the user types \+. An example is below:

```
Set_Pipeline_Parameter
/PARAMETER_NAME=TEST
/PARAMETER_VALUE=A + B
/PARAMETER_VALUE_SEARCH_FOR=\+
/PARAMETER_VALUE_REPLACE_WITH=-
;
```

Result: ::TEST = A - B

```
Set_Pipeline_Parameter
/PARAMETER_NAME=TEST
/PARAMETER_VALUE= A - B
/PARAMETER_VALUE_SEARCH_FOR=-
/PARAMETER_VALUE_REPLACE_WITH=\+
;
```

Result: ::TEST = A + B

Example: Set Pipeline Parameter Multi-Pass

MultiPass was introduced january 21, 2014

The resulting parameter could be a new pipeline parameter

for example, the resulting string could be ::VALUE If MultiPass were true, this should return the value of the parameter VALUE not the string ::VALUE example:

```
Set_Pipeline_Parameter
/PARAMETER_NAME=TEST
/PARAMETER_VALUE=SCOTT
! /PARAMETER_VALUE_SEARCH_FOR=
! /PARAMETER_VALUE_REPLACE_WITH=
```

```
! /PARAMETER_VALUE_PREFIX=
! /PARAMETER_VALUE_APPEND=
;
```

Pipeline Parameter : ::TEST = SCOTT

```
--------------------------------------------------------------------------------
```

```
Set_Pipeline_Parameter
/PARAMETER_NAME=ONE
/PARAMETER_VALUE=KEITH
! /PARAMETER_VALUE_SEARCH_FOR=
! /PARAMETER_VALUE_REPLACE_WITH=
! /PARAMETER_VALUE_PREFIX=
! /PARAMETER_VALUE_APPEND=
;
```

Pipeline Parameter : ::ONE = KEITH

```
--------------------------------------------------------------------------------
```

```
Set_Pipeline_Parameter
/PARAMETER_NAME=SCOTTKEITH
/PARAMETER_VALUE=7
! /PARAMETER_VALUE_SEARCH_FOR=
! /PARAMETER_VALUE_REPLACE_WITH=
! /PARAMETER_VALUE_PREFIX=
! /PARAMETER_VALUE_APPEND=
;
```

Pipeline Parameter : ::SCOTTKEITH = 7

```
--------------------------------------------------------------------------------
```

if we set MULTI_PASS= FALSE

```
Set_Pipeline_Parameter
/PARAMETER_NAME=FINAL
/PARAMETER_VALUE=::SCOTTKEITH
! /PARAMETER_VALUE_SEARCH_FOR=
! /PARAMETER_VALUE_REPLACE_WITH=
! /PARAMETER_VALUE_PREFIX=
! /PARAMETER_VALUE_APPEND=
! /MULTI_PASS=FALSE
;
```

Pipeline Parameter : ::FINAL = ::SCOTTKEITH

if we set MULTI_PASS= TRUE

```
Set_Pipeline_Parameter
/PARAMETER_NAME=FINAL
/PARAMETER_VALUE=::SCOTTKEITH
! /PARAMETER_VALUE_SEARCH_FOR=
! /PARAMETER_VALUE_REPLACE_WITH=
! /PARAMETER_VALUE_PREFIX=
! /PARAMETER_VALUE_APPEND=
/MULTI_PASS=true
;
```

Pipeline Parameter : ::FINAL = 7

NOTE: If the above does not work (if FINAL is not set to 7, and MULTI_PASS is set to TRUE), instead of writing “::SCOTTKEITH”, use “:&:&::SCOTTKEITH”.

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&rev=1719432418

Last update: **2024/06/26 20:06**