

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 "+" delimiter 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 "+" delimiter 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=1718804313

Last update: 2024/06/19 13:38

