

# gpproc reference documentation v1.21 (since v0.95)

Sebastien CAUX

March 3, 2017

## 1 Introduction

gpproc is a command line tool that allows to create new process block.

## 2 Use

gpproc always takes the process file in the current directory, so you only can have one process definition per directory. A process file have the ‘.proc’ extension.

At the beginning, you need to create a process with the *new* command. After that, you can use all the commands set on this process.

Refer to the tutorial ‘GPStudio command line quick start’ to learn how to use this tool.

Under linux, you have a completion script to help you writing commands.

## 3 Commands

### 3.1 brief

The commands of this tools are used to define inputs and outputs of the block, flows, Parameter Interconnect interface with internal registers, clocks and resets. Finally, high level properties for the controller of the block can be specified.

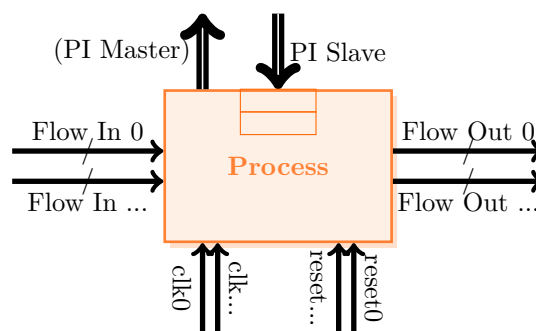


Figure 1: Process block

Commands of gpproc always have the same naming convention. This table is a brief commands list depending on the model instance to be modified.

	add	del	show	rename	set
file	addfile	delfile	showfile		
flow	addflow	delflow	showflow	renameflow	setflow
param	addparam	delparam	showparam	renameparam	setparam
bitfield	addbitfield	delbitfield	showbitfield	renamebitfield	setbitfield
property	addproperty	delproperty	showproperty	renameproperty	setproperty
enum	addenum	delenum	showenum		
reset	addreset	delreset	showreset	renamereset	setreset
clock	addclock	delclock	showclock	renameclock	setclock

Few over commands can be used to create the block and generate the skeleton defined in project section. The command [sethelp](#) is useful to document the block. In addition, [setinfo](#) saves the informations about the block.

[setpropertymap](#) defines link between properties.

For Parameters Interconnect interface configuration, you should use [setpysizeaddr](#) and [setpimastercount](#).

## 3.2 project

### 3.2.1 new

```
gpproc new -n <process-name>
```

Creates a process file inside the current directory named '`<process-name>.proc`'.

Option	Description	Example
-n	process name without space	my_process

- **Example:**

```
> gpproc newprocess -n my_process
```

Creates a new process named *my\_process*. After that, you have a file project *my\_process.proc* in the current directory.

### 3.2.2 showblock

```
gpproc showblock
```

Shows all informations about the block.

### 3.2.3 generatetop

```
gpproc generatetop [-o <dir>]
```

Generates the skeleton of the block top level in the subdirectory `build/`. This top level instantiate the process and the slave if you need. The generated file is named '`<process-name>.vhd`'.

Option	Description	Example
-o	output directory	build

- **Example:**

```
> gpproc generatetop -o build/
```

Generates *myproc.vhd* in the subdirectory *build/*.

### 3.2.4 generatesslave

```
gpproc generatesslave [-o <dir>]
```

Generates the slave for dynamic parameters with read and write. The generated file is named '`<process-name>_slave.vhd`'.

Option	Description	Example
-o	output directory	build

- **Example:**

```
> gpproc generatesslave -o build/
```

Generates *myproc\_slave.vhd* in the subdirectory *build/*.

### 3.2.5 generateprocess

```
gpproc generateprocess [-o <dir>]
```

Generates an empty skeleton to implement your custom process. The generated file is named '`<process-name>_process.vhd`'.

Option	Description	Example
-o	output directory	build

- **Example:**

```
> gpproc generateprocess -o build/
```

Generates *myproc\_process.vhd* in the subdirectory *build/*.

### 3.2.6 generate

```
gpproc generate [-o <dir>]
```

This command is an alias for the three previous command. Generates all the three files.

Option	Description	Example
-o	output directory	build

- **Example:**

```
> gpproc generate -o build/
```

Generates *myproc.vhd*, *myproc\_process.vhd* and *myproc\_slave.vhd* in the subdirectory *build/*.

### 3.2.7 generatetb

```
gpproc generatetb [-o <dir>]
```

Generates a testbench that implement your process. This generated test bench simulates clocks and resets and sends stimuli file named '`<flow-name>.stim`' for each input flows and saves output data to '`<flow-name>.data`'. To obtain stim file and retrieve data from data file you should use **convert**.

Option	Description	Example
-o	output directory	build

- **Example:**

```
> gpproc generatetb -o build/
```

*Generates myproc\_tb.vhd in the subdirectory build/.*

See also :

**gpproc convert**

### 3.2.8 convert

```
gpproc convert -i <input-file> [-o <output-file>]
```

Converts file for test bench purpose. Can convert picture files to stim file or data files to picture files.

Option	Description	Example
-i	input file	img1.png out.data
-o	output file	in.stim out.jpg

- **Example 1:**

```
> gpproc convert -i imgage.gif -o in.stim
```

*Transforms this gif picture into stimuli file to send to in flow*

- **Example 2:**

```
> gpproc convert -i out.data -o out.png
```

*Transforms the output of the flow 'out' to a png image*

## 3.3 files

### 3.3.1 addfile

```
gpproc addfile -p <path> -t <type> -g <group>
```

Adds the file with the path **<path>** to the list of files of the current process block. The type and group of the file depends of the tool-chain that you use. If the file does not exist, prints a warning.

Option	Description	Example
-p	process path relative to the file process	hdl/my__process.vhd
-t	type of the file	[vhdl - verilog ...]
-g	group of the file	[hdl - doc ...]

- **Example:**

```
> gpproc addfile -p hdl/my__process.vhd -t vhdl -g hdl
```

*Adds the file 'my\_\_process.vhd' as a file needed by the hdl implementation of the block.*

See also :

**gpproc delfile** and **gpproc showfile**

### 3.3.2 delfile

```
gpproc delfile -p <path>
```

Removes the file with the path <path>.

Option	Description	Example
-p	process path relative to the file process	hdl/my__process.vhd

- **Example:**

```
> gpproc delfile -p hdl/my__process.vhd
```

*Removes the file 'my\_\_process.vhd'.*

See also :

**gpproc addfile** and **gpproc showfile**

### 3.3.3 showfile

```
gpproc showfile
```

Prints the list of files in the current block project.

## 3.4 flows

### 3.4.1 addflow

```
gpproc addflow -n <flow-name> -d <direction> -s <size>
```

Adds a flow interface named <flow-name> to the list of flows. You need to specify the direction of the flow (input or output) and the default size.

Option	Description	Example
-n	name of the flow	in1
-d	direction of the flow	[in - out]
-s	default size of the flow interface in bit	9

- **Example:**

```
> gpproc addflow -n out0 -d out -s 8
```

*Adds a new flow output named 'out0' with a default size of 8 bits.*

See also :

**gpproc delflow** and **gpproc showflow**

### 3.4.2 delflow

```
gpproc delflow -n <flow-name>
```

Removes the file with the name <flow-name>.

Option	Description	Example
-n	name of the flow	in1

- **Example:**

```
> gpproc delflow -n out0
```

*Removes the flow named 'out0'.*

See also :

**gpproc addflow** and **gpproc showflow**

### 3.4.3 showflow

```
gpproc showflow
```

Prints the list of flows in the current block project.

### 3.4.4 renameflow

```
gpproc renameflow -n <flow-name> -v <new-name>
```

Renames the flow named <flow-name> with the name <new-name>.

Option	Description	Example
-n	name of the flow	in1
-v	new name of the flow	in0

- **Example:**

```
> gpproc renameflow -n out0 -v out1
```

*Renames the flow 'out0' to 'out1'.*

See also :

**gpproc delflow**

### 3.4.5 setflow

```
gpproc setflow -n <flow-name> [-d <direction>] [-s <size>]
```

Allows to modify one or more flow parameters like the direction or the default size.

Option	Description	Example
-n	name of the flow	in1
-d	direction of the flow	[in - out]
-s	default size of the flow interface in bit	9

- **Example:**

```
> gpproc setflow -n out0 -d in
> gpproc setflow -n out0 -s 9
```

*Sets the flow 'out0' as input and then sets the default size to 9 bits.*

## 3.5 params

### 3.5.1 addparam

```
gpproc addparam -n <param-name> [-t <type>] [-r <relative-address>] [-v <default-value>] [-m <property-map>]
```

Adds a param named <param-name> to the list of params.

Option	Description	Example
-n	name of the param	param0
-t	type of the param for fixed param	toolchain-type
-r	relative address of the param for dynamic param	2
-v	default value of the param	TRUE 1024
-m	javascript property map expression	enable.value mode.bits Math.ceil( in.width.value/4)+4

- **Example 1:**

```
> gpproc addparam -n MAX_LINE_WIDTH -t INTEGER -v 1280
```

Adds a new fixed param named 'MAX\_LINE\_WIDTH' of type INTEGER with a default value of 1280. This example is typically used to define the size of internal buffer when you have to save a complete line of picture in your process.

- **Example 2:**

```
> gpproc addparam -n offset_reg -r 1 -m offset.value
```

Adds a new register named 'offset\_reg' with the relative address 1. The value of the register is linked to the value of the property 'offset'.

See also :

**gpproc delparam, gpproc showparam and gpproc setpropertymap**

### 3.5.2 delparam

```
gpproc delparam -n <param-name>
```

Removes the param with the name <param-name>.

Option	Description	Example
-n	name of the param	param0

- **Example:**

```
> gpproc delparam -n param0
```

Removes the param named 'param0'.

See also :

**gpproc addparam and gpproc showparam**

### 3.5.3 showparam

```
gpproc showparam
```

Prints the list of params in the current block project.

### 3.5.4 renameparam

```
gpproc renameparam -n <param-name> -v <new-name>
```

Renames the param named <param-name> with the name <new-name>.

Option	Description	Example
-n	name of the param	param0
-v	new name of the param	param1

- **Example:**

```
> gpproc renameparam -n param0 -v param1
```

*Renames the param 'param0' to 'param1'.*

See also :

**gpproc delparam**

### 3.5.5 setparam

```
gpproc setparam -n <param-name> [-t <type>] [-r <relative-address>] [-v <default-value>]
```

Allows to modify one or more param parameters like the type, the default value or the relative address.

Option	Description	Example
-n	name of the param	param0
-t	type of the param for fixed param	toolchain-type
-r	relative address of the param for dynamic param	2
-v	default value of the param	TRUE

- **Example 1:**

```
> gpproc setparam -n status -v 1 -r 4
```

*Sets the default value of the dynamic param (register) 'status' to 1 and sets the relative address to 4.*

- **Example 2:**

```
> gpproc setparam -n mode -v LINEAR -t Mode
```

*Sets the type of the generic 'mode' to 'Mode' with a default value set to 'LINEAR'*

### 3.5.6 fixparam

```
gpproc fixparam -n <param-name> -v <true/false>
```

Transforms a param to a fixed parameter if -v is true or a dynamic parameter else.

- **Example:**

```
> gpproc fixparam -n mode -v true
```

*Sets the parameter 'mode' to a fixed parameter, eg generic value for a VHDL backend*

### 3.5.7 setpysizeaddr

```
gpproc setpysizeaddr -v <pysizeaddr>
```

Sets the width of PI address bus in bits. For example, with the highest address 9 you should use 4 as the pysizeaddr to have a range of 16 addresses.



Option	Description	Example
-v	width of PI slave address port in bits	3

- **Example:**

```
> gpproc setpysizeaddr -v 5
```

*Sets the parameter pysizeaddr to 5 bits to have 32 possible registers*

### 3.5.8 setpimastercount

```
gpproc setpimastercount -v <pimastercount>
```

Sets the number of PI master port.

Option	Description	Example
-v	number of master port	1

- **Example:**

```
> gpproc setpimastercount -v 1
```

*Adds a PI master port to the current process*

## 3.6 bitfields

### 3.6.1 addbitfield

```
gpproc addbitfield -n <param-name.bitfield-name> -b <bitfield> [-m <property-map>]
```

Adds a bit field named <bitfield-name> to the param <param-name>.

Option	Description	Example
-n	name of the bit field composed by the param name, a dot and the bits field name	param.b0
-b	bits field	1 8-4 2,5,10-9
-m	javascript property map expression	enable.value mode.bits Math.ceil( in.width.value/4)+4

- **Example 1:**

```
> gpproc addbitfield -n status.enable -b 0
```

*Adds a bit 'enable' positioned on the bit 0 in the register status*

- **Example 2:**

```
> gpproc addbitfield -n status.mode -b 5-3
```

*Adds a bit field 'mode' positioned on the bits 5 downto 3 in the register status*

See also :

**gpproc delbitfield, gpproc showbitfield and gpproc setpropertymap**

### 3.6.2 delbitfield

```
gpproc delbitfield -n <param-name.bitfield-name>
```

Removes the bit field with the name <param-name.bitfield-name>.

Option	Description	Example
-n	name of the bit field composed by the param name, a dot and the bits field name	param.b0

- Example:

```
> gpproc delbitfield -n param0
```

*Removes the param named 'param0'.*

See also :

**gpproc addbitfield** and **gpproc showbitfield**

### 3.6.3 showbitfield

```
gpproc showbitfield -n <param-name>
```

Prints the list of bit fields associated to the parameter<param-name> .

### 3.6.4 renamebitfield

```
gpproc renamebitfield -n <param-name.bitfield-name> -v <new-name>
```

Renames the bit field named <param-name.bitfield-name> with the name <new-name>.

Option	Description	Example
-n	name of the bit field	status.active
-v	new name of the bit field	enable

- Example:

```
> gpproc renamebitfield -n status.active -v enable
```

*Renames the bit field 'status.active' to 'status.enable'.*

See also :

**gpproc delbitfield**

### 3.6.5 setbitfield

```
gpproc setbitfield -n <bitfield-name> -b <bitfield>
```

Allows to modify bit field bits selection.

Option	Description	Example
-n	name of the bit field	param0.bf0
-b	bits field	1 8-4 2,5,10-9

- Example:

```
> gpproc setparam -n param0.bf0 -b 2-0
```

*Sets the bit selection of 'param0.bf0' to bits 2 down to 0.*

## 3.7 properties

### 3.7.1 addproperty

```
gpproc addproperty -n <property-name> -t <type> [-v <default-value>] [-m <property-map>]
```

Adds a property named <property-name> to the list of properties. You need to define a type to define the widget adapted to this data and optionally a default value. You can also specify a label for high level interface.

Option	Description	Example
-n	name of the property. Can be	enable
-t	type of property	[int-sint-bool-group-matrix-enum]
-v	default value at the beginning of the program	true
-m	javascript property map expression	enable.value mode.bits Math.ceil( in.width.value/4)+4

- **Example:**

```
> gpproc addproperty -n enable -l "enable process" -t bool -v true
```

Adds a new property to enable the process. At this point, the property is not linked to a register, you need to use **setpropertymap** to do that.

See also :

**gpproc delproperty**, **gpproc showproperty** and **gpproc setpropertymap**

### 3.7.2 delproperty

```
gpproc delproperty -n <property-name>
```

Removes the property with the name <property-name>.

Option	Description	Example
-n	name of the property	enable

- **Example:**

```
> gpproc delproperty -n status.enable
```

Removes the property named 'enable' in the 'status' property.

See also :

**gpproc addproperty** and **gpproc showproperty**

### 3.7.3 showproperty

```
gpproc showproperty [-n <property/flow-name>]
```

Prints the list of properties of the current block project if no name is specified. If a name is given, list the properties of the given property/flow.

### 3.7.4 renameproperty

```
gpproc renameproperty -n <property-name> -v <new-name>
```

Renames the property named `<property-name>` with the name `<new-name>`.

Option	Description	Example
-n	name of the property	mode
-v	new name of the property	mode_img

- **Example:**

```
> gpproc renameproperty -n mode -v mode_img
```

*Renames the property 'mode' to 'mode\_img'.*

See also :

**gpproc delproperty**

### 3.7.5 setproperty

```
gpproc setproperty -n <property-name> [-l <label>] [-t <type>] [-v <default-value>] [-r <min:max>] [-s <step>]
```

Allows to modify one or more property parameters like the label, the type or the default value.

Option	Description	Example
-n	name of the flow	enable
-l	high level label property (can contain space and special characters)	"enable process"
-t	type of property	[int-sint-bool-group-matrix-enum]
-v	default value at the beginning of the program	true
-r	range of the value (for int or sint type)	0:100
-s	step of the value (for int or sint type)	2

- **Example 1:**

```
> gpproc setproperty -n threshold -t sint -v 0
```

*Sets the property 'threshold' to slider int type with 0 as default value.*

- **Example 2:**

```
> gpproc setproperty -n threshold -r 0:255 -s 2 -v 128
```

*Sets the property 'threshold' with a range value of 0 to 255, a step of (pair values only) and default value setted to the half range.*

### 3.7.6 setpropertymap

```
gpproc setpropertymap -n <param/property/bitfield-name> -v <property-map>
```

Allows to set the property map of a dynamic parameter, a property or a bit field. It's the mathematical expression to dynamically link a property value to another value. High level interface or API only have access to property, so if you need to modify a register from the external, creates a property and link his value to the register. When the property value change, the property map expression is re evaluated to set the new computed value to the register.

This expression can be more complex than a simple mathematical expression. Can contains all the standard javascript functions.

Option	Description	Example
-n	name of the property, register or bit field	enable_bf status_reg mode
-v	javascript property map expression	enable.value mode.bits Math.ceil( in.width.value/4)+4

- **Example 1:**

```
> gpproc setpropertymap -n status.enablebit -v enable.value
```

Sets the bit field 'enablebit' of the register 'status' linked to the value of the 'enable' property. When this property change, the new value of the property will be set in the register.

- **Example 2:**

```
> gpproc setpropertymap -n out.width -v "in.width.value-2"
> gpproc setpropertymap -n out.height -v "in.height.value-2"
```

Sets property map on the flow 'out'. The output image size is the size of the input minus 2.

- **Example 3:**

```
> gpproc setpropertymap -n shift_reg -v "Math.floor(Math.log2(div.value))+1"
```

Sets property map on the flow 'out'. The output image size is the size of the input minus 2.

## 3.8 enums

### 3.8.1 addenum

```
gpproc addenum -n <enum-name> -v <value> [-l <label>]
```

Adds a new enumeration item named <enum-name> to the list of enums in the property. The associated value will be set to the parent property when the enum will chosen. You can also specify a label for high level interface.

Option	Description	Example
-n	property path with a dot following by the name of the enum item	mode.eroode
-v	value of the enum item	2
-l	label of the enum item (optional, take the value of the name)	"Eroode mode"

- **Example:**

```
> gpproc addproperty -n width -l "image width" -t enum -m width_reg
> gpproc addenum -n width.w480 -l "480 px" -v 0
> gpproc addenum -n width.w640 -l "640 px" -v 1
> gpproc addenum -n width.w860 -l "860 px" -v 2
```

Adds a new property 'width' associated to the register 'width\_reg' and three enumeration items to it.

See also :

**gpproc delenum** and **gpproc showenum**

### 3.8.2 delenum

```
gpproc delenum -n <enum-name>
```

Removes the enum with the name <enum-name>.

Option	Description	Example
-n	name of the enum	mode.ero

- Example:

```
> gpproc delenum -n mode.ero
```

*Removes the enum named 'ero' in the 'mode' enum.*

See also :

**gpproc addenum** and **gpproc showenum**

### 3.8.3 showenum

```
gpproc showenum [-n <property-name>]
```

Prints the list of enumeration item in the property <property-name> of the current block project if no name is specified. If a name is given, list the enums of the given property.

## 3.9 resets

### 3.9.1 addresset

```
gpproc addresset -n <reset-name> -d <direction> -g <group>
```

Adds a reset named <reset-name> to the list of resets. You need to specify the direction of the reset (input or output) and the group. All resets with the same group are connected to the same net.

Option	Description	Example
-n	name of the reset	reset_n
-d	direction of the reset	[in - out]
-g	group of the reset	reset_n

- Example:

```
> gpproc addresset -n reset_n -d in -g reset_n
```

*Adds a new reset input named 'reset\_n' connected to the 'reset\_n' group*

See also :

**gpproc delreset** and **gpproc showreset**

### 3.9.2 delreset

```
gpproc delreset -n <reset-name>
```

Removes the reset with the name <reset-name>.

Option	Description	Example
-n	name of the reset	in1

- Example:

```
> gpproc delreset -n out0
```

Removes the reset named 'out0'.

See also :

**gpproc addresset** and **gpproc showreset**

### 3.9.3 showreset

```
gpproc showreset
```

Prints the list of resets in the current block project.

### 3.9.4 renamereset

```
gpproc renamereset -n <reset-name> -v <new-name>
```

Renames the reset named <reset-name> with the name <new-name>.

Option	Description	Example
-n	name of the reset	reset_n
-v	new name of the reset	reset

- **Example:**

```
> gpproc renamereset -n reset_n -v reset
```

Renames the reset 'reset\_n' to 'reset'.

See also :

**gpproc delreset**

### 3.9.5 setreset

```
gpproc setreset -n <reset-name> [-d <direction>] [-g <group>]
```

Allows to modify one or more reset parameters like the direction or the group.

Option	Description	Example
-n	name of the reset	reset_n
-d	direction of the reset	[in - out]
-g	group of the reset	reset_n

- **Example:**

```
> gpproc setreset -n reset_n -d out  
> gpproc setreset -n reset_n -g reset_all
```

Sets the reset 'reset\_n' as output and then sets the reset group to 'reset\_all'.

## 3.10 clocks

### 3.10.1 addclock

```
gpproc addclock -n <clock-name> -d <direction> [-g <clock-domain>]
```

Adds a clock named <clock-name> to the list of clocks. You need to specify the direction of the clock (input or output) and eventually the clock domain.

Option	Description	Example
-n	name of the clock	clock
-d	direction of the clock	[in - out]
-g	clock domain of the clock	clk_proc

- **Example:**

```
> gpproc addclock -n clock -d in -g clk_proc
```

*Adds a new clock input named 'clock' connected to the 'clk\_proc' clock domain*

See also :

**gpproc setclock, gpproc delclock and gpproc showclock**

### 3.10.2 delclock

```
gpproc delclock -n <clock-name>
```

Removes the file with the name <clock-name>.

Option	Description	Example
-n	name of the clock	clk_proc

- **Example:**

```
> gpproc delclock -n clk_proc
```

*Removes the clock named 'clk\_proc'.*

See also :

**gpproc addclock and gpproc showclock**

### 3.10.3 showclock

```
gpproc showclock
```

Prints the list of clocks in the current block project.

### 3.10.4 renameclock

```
gpproc renameclock -n <clock-name> -v <new-name>
```

Renames the clock named <clock-name> with the name <new-name>.

Option	Description	Example
-n	name of the clock	clock
-v	new name of the clock	clock2

- **Example:**

```
> gpproc renameclock -n clock -v clock2
```

*Renames the clock 'clock' to 'clock2'.*

See also :

**gpproc delclock**

### 3.10.5 setclock

```
gpproc setclock -n <clock-name> [-d <direction>] [-g <clock-domain> [-m <multiplier>]] [-f <clock-frequency>] [-r <min:max>]
```



Allows to modify one or more clock parameters. It exists many ways to define a clock :

- ▶ defines only the clock domain to have the clock domain frequency specified in the final project with -g
- ▶ defines the clock domain and the multiplier to have the clock domain frequency specified in the final project multiplied by the multiplier with -g and -m
- ▶ directly defines a fixed frequency with -f
- ▶ defines a range for the clock with -r. The final frequency could be set in the final project with a verification of the range or automatically chosen by the CI system

Option	Description	Example
-n	name of the clock	clock
-d	direction of the clock	[in - out]
-g	clock domain of the clock	clk_proc
-m	multiplier of the clock in the clock domain	0.5 2
-f	frequency to give to the clock. It is possible to use multiplier suffix like 'G', 'M' or 'k'.	0.25G 62M 5.5k
-r	range of the clock	5.5k:12k

- **Example 1:**

```
> gpproc setclock -n clock -d in
```

*Sets the clock 'clock' as input.*

- **Example 2:**

```
> gpproc setclock -n clock -g clock_img
```

*Moves it to the clock domain 'clock\_img'.*

- **Example 3:**

```
> gpproc setclock -n clock -g clock_img -m 2
```

*Defines the frequency to the double of the clock domain 'clock\_img' frequency.*

- **Example 4:**

```
> gpproc setclock -n clock -f 12.3M
```

*Defines the clock frequency to 12.3MHz.*

- **Example 5:**

```
> gpproc setclock -n clock -r 5.5k:12k
```

*Defines the clock frequency in the range of 5.5k<f<12k.*

### 3.11 global

#### 3.11.1 sethelp

```
gpproc sethelp [-n <instance-name>] -v <help-text>
```

Sets the help text for an instance of the block if a name is specified, or the block itself is name is omitted. This is useful for the automatic documentation generation.

Option	Description	Example
-n	name of the property, parameter, bit field, flow, file, clock or reset	enable_bf status_reg mode ...
-v	help text	"enable the process"

- **Example:**

```
> gpproc sethelp -v "Process to remove the noise in the input image."
> gpproc sethelp -n in -v "input image flow"
> gpproc sethelp -n out -v "output image flow"
```

*Few documentation for the block.*

### 3.11.2 setdraw

```
gpproc setdraw -v <svg-draw-content>
gpproc setdraw -f <svg-file>
```

Sets the picture of the block from svg commands or a svg file.

Option	Description	Example
-v	svg commands	<ellipse fill='red' cx='283.5' cy='487.5' rx='259' ry='80'/>
-f	svg file	"hog.svg"

- **Example 1:**

```
> gpproc setdraw -v "<ellipse fill='red' cx='283.5' cy='487.5' rx='259' ry='80'/>"
```

*Put an ellipse as draw of the block*

- **Example 2:**

```
> gpproc setdraw -f doc/hog.svg
```

*Put the content of 'doc/hog.svg' file as draw of the block*

### 3.11.3 setcateg

```
gpproc setcateg -v <categ-value>
```

Sets the category of the block. IPs block can be sorted by usage and utility and be more easy to find.

Option	Description	Example
-v	value to set	"filter/blur" "descriptor/hog"

- **Example:**

```
> gpproc setcateg -v "filter"
> gpproc setcateg -v "filter/blur"
```

### 3.11.4 categ

```
gpproc categ
```

Returns the category of the block.

- **Example:**

```
> gpproc categ
descriptor/lbp
```

### 3.11.5 setinfo

```
gpproc setinfo -n <info-name> -v <info-value/content>
```

Sets infos of the block. When you develop IPs, it is very important to set some information like author, licence, contact, email and version.

Option	Description	Example
-n	name of the info	"author" "email" "company"
-v	value to set	"John Doe" "john.doe@company.tld" "The famous company (R)"

- **Example:**

```
> gpproc setinfo -n "author" -v "John Doe"
> gpproc setinfo -n "email" -v "john.doe@company.tld"
> gpproc setinfo -n "company" -v "The famous company (R)"
```

### 3.11.6 info

```
gpproc info [-n <info-name>]
```

Returns infos of the block.

Option	Description	Example
-n	name of the info	"author" "email" "company"

- **Example:**

```
> gpproc info -n "company"
The famous company (R)
> gpproc info
author: John Doe
email: john.doe@company.tld
company: The famous company (R)
```