

PPD is freeware from GreenHouse Software & Consulting.  
No warranty is given.  
Use it at your own risk.

PPD displays all named processes from the Process Pair Directory.

A PPD is nothing new, but this version supports wild cards and allows the sorting by the process

- name
- mom
- PID

The command syntax is:

```
[run] PPD [/OUT <out-file>/][<$name>|<PID>] [BYMOM|BYPID|BYNAME][IOPROC]
[WHERE MOM|ANCESTOR $name|PIN]
```

where

- **\$name** is a process name, or process name template, e.g.:  
PPD \$GHS displays the \$GHS CPU(s)  
PPD \$G?S\* displays all processes, matching the given template.  
The \$-sign can be omitted: It is automatically inserted.
- **PID** is the process ID in CPU,PIN format  
PPD 3,423 displays the related processes name in case it exists.
- **BY MOM** sorts the processes by their MOM
- **BY PID** sorts the processes by their PIN
- **BY NAME** sorts the processes by their NAME
- **IOPROC** causes PPD to list I/O processes as well
- **WHERE MON** or
- **WHERE ANCESTOR** causes PPD to display only the processes with the defined MOM/ANCESTOR

A no longer existing ancestor of a process is displayed in brackets.

The SELECT utility from Kari can do this too (and a lots of other things) e.g.

```
SELECT PROCESS \node NAME,CPU,PIN,ANCESTOR,PROGRAM where NAME like $A?o*
```

This command could be hidden in a small TACL macro.

(In the next version this is even more straightforward:

```
SELECT PROCESS $A?o* NAME,CPU,PIN,ANCESTOR,...)
```