PPD is freeware from GreenHouse Software & Consulting.
No warranty is given.
Use it at your own risk.
Version 305
09. August 2016

This version of PPD displays all named processes from the Process Pair Directory of a local or remote system.
It supports wildcards where ever meaningful.
In addition it sorts the list of found processes by

- Process name (node name and process name)
- Mom's name (node name, mom's name and process name
- PID (node name and primary PID)
- Object name (node name, object name and process name)
- Program file (node name, object name and process name)
- Start time (start time of process)

Command syntax is:

```
[run] PPD [/OUT <out-file>/] [<process>] [BY MOM|BY PID|BY NAME|BY OBJECT]
                                         [IOPROC]
                                         [WHERE MOM|ANCESTOR <$name>|<PIN>]
                                         [WHERE OBJECT <file>]
```

where

- process can start with a node name, where node name supports wildcards; e.g.:

```
\ginkgo
\*be?
```

When node name part is missing, the local node is used.
The node name can be followed by a process name, starting with the $-sign,
or a PID number.

The process name supports wildcards, while the PID does not; e.g.:

```
$W*
12,5632
```

When the process is not defined, all named processes from \node are displayed.

Supported process definitions:

\node	node supports wildcards; all processes from matching \node are displayed
\node.$name|name	node and name support wildcards; all matching processes from all matching nodes are displayed
\node.nn,mm	no wildcard support here; whether the node nor the PID supports wildcards
$name|name	name supports wildcards; all matching processes from the local node are displayed
nn,mm	no wildcard support here; the PID does not support wildcards

- BY MOM sorts the processes by their node name and MOM

- BY NAME sorts the processes by their node name and process NAME

- BY PID sorts the processes by their node name and PIN

- BY OBJECT sorts the processes by their object file name

- BY STARTTIME sorts the processes by their start time

- IOPROC causes PPD to list I/O processes as well

- **WHERE MON** or **WHERE ANCESTOR** displays only processes with a matching MOM/ANCESTOR.
  A no longer existing ancestor is displayed in brackets.
  A named MOM supports wildcards, e.g.:
  PPD \beech.$* WHERE MOM $SPLS

- **WHERE OBJECT** displays only processes, started from the defined object file.
  The object file name supports wildcards, e.g.:
  PPD *.$TU* WHERE OBJECT $SYS*.SYS??.TACL*

Accessing remote systems takes some time.
e.g. the command:

```
PPD \*.$*
```

may take even minutes, when there are many systems with many named processes in the EXPAND network!

The intention of adding EXPAND support for this tool is, to quickly display equal named processes from a set of defined system, e.g.:

```
PPD \DEV*.$GHSP*
```

This displays all processes, where the name starts with $GHSP from all systems, where the EXPAND name starts with \DEV.

A typical output for a local system looks like this:

```
$GHS1 SECOM700 1> ppd $s*
PPD (305) - T7172L06 - (09Aug2016)   System \OAK, running NSK L06.04
Copyright (c) GreenHouse Software & Consulting 1999-2016

Used search string: $S*

Name      Primary  Backup    Ancestor   Object File              StartTime
$SECCT     0,416              $ZPM     $GHS1.SECOM700.SECOMCTX        08:33:37
$SPLS      0,36     1,56     [$Z02R]   $SYSTEM.SYSTEM.SPOOL           08:35:32
$S         0,464    1,483     $SPLS    $SYSTEM.SYSTEM.CSPOOL          08:35:32
$SP        0,465    1,482     $SPLS    $SYSTEM.SYSTEM.CSPOOL          08:35:33
Number of matches: 4
$GHS1 SECOM700 2>
```

Addressing the process name along with the node name shows this result:

```
$GHS1 SECOM700 3> ppd \oak.$s*
PPD (305) - T7172L06 - (09Aug2016)   System \OAK, running NSK L06.04
Copyright (c) GreenHouse Software & Consulting 1999-2016

Used search string: \OAK.$S*

  Node     Name    Primary    Backup     Ancestor   Program File
\OAK      $SECCT    0,416               $ZPM       $GHS1.SECOM700.SECOMCTX
\OAK      $SPLS     0,36       1,56     [$Z02R]     $SYSTEM.SYSTEM.SPOOL
\OAK      $S        0,464      1,483     $SPLS      $SYSTEM.SYSTEM.CSPOOL
\OAK      $SP       0,465      1,482     $SPLS      $SYSTEM.SYSTEM.CSPOOL
Number of matches on \OAK: 4
Matching nodes: 1
$GHS1 SECOM700 4>
```

A typical output for a network wide PPD looks like this:

```
$SYSCT1.DE139274 7> ppd \*.$GHSP* BY NAME
PPD (304) - T7172H06 - (09May2016)   System \OL8, running NSK H06.26
Copyright (c) GreenHouse Software & Consulting 1999 .. 2016

Used search string: \*.$GHSP* sorted by PROC name

  Node     Name    Primary    Backup     Ancestor   Program File
\EN5      $GHSPI    0,1179     1,720     $ZPM       $SYSTEM.PSSHELL.PINEW
Number of matches on \EN5    : 1

  Node     Name    Primary    Backup     Ancestor   Program File
\EN6      $GHSPI    0,1249     1,1732    $ZPM       $SYSTEM.PSSHELL.PINEW
Number of matches on \EN6    : 1

  Node     Name    Primary    Backup     Ancestor   Program File
\EN7      $GHSPI    0,1591     1,702     $ZPM       $SYSTEM.PSSHELL.PINEW
\EN7      $GHSPS    0,1299               $ZSMP      $SYSTEM.PSSHELL.PSSHELL
Number of matches on \EN7    : 2

  Node     Name    Primary    Backup     Ancestor   Program File
\OL5      $GHSPI    0,1012     1,1007    $ZPM       $SYSTEM.PSSHELL.PINEW
Number of matches on \OL5    : 1

  Node     Name    Primary    Backup     Ancestor   Program File
\OL6      $GHSPI    0,976      1,1344    $ZPM       $SYSTEM.PSSHELL.PINEW
\OL6      $GHSPS    0,1166               $ZSMP      $SYSTEM.PSSHELL.PSSHELL
Number of matches on \OL6    : 2

  Node     Name    Primary    Backup     Ancestor   Program File
\OL7      $GHSPI    0,536      1,1650    $ZPM       $SYSTEM.PSSHELL.PINEW
Number of matches on \OL7    : 1

  Node     Name    Primary    Backup     Ancestor   Program File
\OL8      $GHSPI    0,2145     1,1642    $ZPM       $SYSTEM.PSSHELL.PINEW
\OL8      $GHSPS    0,2126               $ZSMP      $SYSTEM.PSSHELL.PSSHELL
Number of matches on \OL8    : 2
```

Feel free to use this tool, and in case you stumble into problems, please let me know, and I'll fix it.

Carl Weber
GreenHouse Software & Consulting
178. August 2016

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?0* NAME,CPU,PIN,ANCESTOR,...)
```