PQUOTA User's Guide

PQUOTA displays information on resource usage by VMS processes.

This manual contains a description of the PQUOTA program, including usage details and an explanation of the output produced.

Software Version: 1.16

Operating System: OpenVMS/AXP and VAX Version 6.0 or higher.

First printing: October 1995
Last revised: May 1998

I would like to thank Jim Good for his feedback concerning this documentation.

1. PURPOSE OF PQUOTA

A PQUOTA program run provides information about resource utilisation for an individual VMS process. The information retrieved includes:

- The process name.
- The name of the node on which the process is running.
- The current process state.
- The CPU time that has been charged to the process so far.
- Current process resource quotas.
- The maximum quota values used by the process during the current PQUOTA run.
- Enabled and pending AST's for this process.

This information is made available in a dynamically updated terminal display. The maximum quota values used are also written to a session log file.

Runs of the PQUOTA program may be of indefinite length, and a user may run several PQUOTA sessions simultaneously on different terminal devices (e.g. in different DECterm windows).

2. STARTING THE PQUOTA PROGRAM

A PQUOTA session is started by running the PQUOTA.EXE executable image. This image does not require any logical names or symbols to be defined first. So, to run PQUOTA.EXE just enter:

```
$ RUN device:[directory]PQUOTA
```

where 'device:[directory]' is the disk device and directory in which PQUOTA.EXE is located. (As usual, if PQUOTA.EXE is in your current default directory, then the device and directory specification can be omitted. In this case, the command simply becomes $ RUN PQUOTA. This remark applies also to the other command lines shown below.)

You will then be prompted to identify the process to be monitored:

```
Please enter the target Process PID:
```

Enter the process identification (PID) of the process you want to investigate. (The PID value can be obtained from the output of the $SHOW PROCESS or $SHOW SYSTEM commands. PID values are hexadecimal numbers, e.g. '5060013B')

Alternatively you can define a DCL 'foreign command' as follows:

```
$ PQUOTA :== $device:[directory]PQUOTA
```

It is important to note that if you do define a foreign command then the device and directory specification MUST be included, even if the PQUOTA.EXE file is in your current default directory. (You may find it useful to define such a foreign command in your LOGIN.COM.)

With such a foreign command definition, a PQUOTA session can be started just by entering 'PQUOTA', and the PID of the process to be investigated can be entered directly on the command line. Thus, PQUOTA can be started as follows:

```
$ PQUOTA pid
```

(If you do not specify a process identification on the command line, then you will be prompted for a PID value as when you start PQUOTA with an explicit RUN command.)

A third alternative is to copy the PQUOTA.EXE into SYS$COMMON:[SYSEXE] (or have it copied by your system manager.) Then you can start PQUOTA as follows:

```
$ MC PQUOTA pid
```

or

```
$ MC PQUOTA
```

when you will then be prompted for the PID. Or, if you define a foreign command, you can omit the device and directory specification, i.e. when the PQUOTA.EXE program is in SYS$COMMON:[SYSEXE], the foreign command specification is simply:

```
$ PQUOTA  :== $PQUOTA
```

Placing the PQUOTA.EXE file in SYS$COMMON:[SYSEXE] makes it available to all users on the system that have CMKRNL privilege.

2.1 Single Sample Mode

In this mode the PQUOTA runs once and stores the output in the session log file PQUOTA_xxx_yyy.OUT. This mode enables you to start the PQUOTA from a program or an indirect command file for a snap shot of the target process.

The PQUOTA.EXE must be located in SYS$COMMON:[SYSEXE] or defined as a foreign command. This feature is not available in interactive mode.

The PQUOTA is started in 'single sample mode' as follows (the letter X informs PQUOTA for a single sample) :

```
$ MC PQUOTA pid X
```

or

```
$ PQUOTA pid X
```

Note: the pid must be 8 digits, even with leading zeros.

Example of a command file which can be called from a program:

```
$!
$! Input: P1 = pid of target process
$!
$ define/user sys$output NL:
$!
$ mc pquota 'P1' X
$!
```

3. SESSION LOG FILE

PQUOTA writes the process name, PID, time stamp, maximum used and percentage used quota values to the file PQUOTA_xxx_yyy.OUT on the current default directory, where ’xxx’ is the target process name and ’yyy’ is the target process PID. Any characters in the process name that are not valid file name characters are replaced by ’$’ signs. (Valid file name characters are a - z, A - Z, 0 - 9, _, -, and $).

Note: RMS will automatically convert lowercase characters into uppercase.

Entries are appended to the log file at the beginning of the PQUOTA session and then at 10 minute intervals. If the target process exits during the PQUOTA run, then the last maximum values available are written also.

A new PQUOTA_xxx_yyy.OUT file is created for each PQUOTA session. If you are running a number of PQUOTA sessions simultaneously then each will have a separate PQUOTA file open.

The PQUOTA_xxx_yyy.OUT is opened as a shared file, which allows you to type it while PQUOTA is running.

If the target process is outswapped, then some of the process information is not available. The fields ASTCNT and WSQUO in the PQUOTA_xxx_yyy.OUT file are written as " * Outswapped".

Example of a PQUOTA_xxx_yyy.OUT file:

```
_FTA15:   3600029F   31-JAN-1996 16:27:30
            Max   %
BYTCNT      576   0 Bytes
BIOCNT        1   0 IO’s
DIOCNT        0   0 IO’s
ASTCNT        3   0 Ast’s
FILCNT        0   0 Files
ENQCNT        0   0 Locks
TQCNT         0   0 Timers
PGFLQUO    1680   2 Pagelets
WSQUO      1104   3 Pagelets

_FTA15:   3600029F   31-JAN-1996 16:37:33
            Max   %
BYTCNT      576   0 Bytes
BIOCNT        1   0 IO’s
DIOCNT        1   0 IO’s
ASTCNT        3   0 Ast’s
FILCNT        0   0 Files
ENQCNT        0   0 Locks
TQCNT         0   0 Timers
PGFLQUO    1872   2 Pagelets
WSQUO      1232   3 Pagelets
```

Example of a PQUOTA_xxx_yyy.OUT file for an outswapped process:

```
CONFIGURE   36000085   16-APR-1996 19:49:14
            Max   %
BYTCNT      448   0 Bytes
BIOCNT        0   0 IO's
DIOCNT        0   0 IO's
ASTCNT * Outswapped Ast's
FILCNT        2   1 Files
ENQCNT        0   0 Locks
TQCNT         0   0 Timers
PGFLQUO    1200   3 Pagelets
WSQUO  * Outswapped Pagelets
```

4. USAGE NOTES

4.1 PRIVILEGES REQUIRED

The Change Mode to Kernel (CMKRNL) privilege is needed to run PQUOTA.

(Use $SET PROC/PRIV=CMKRNL to get this privilege. If it happens that you're not authorized to set the CMKRNL privilege, you'll have to plead with your friendly system manager.)

4.2 INTERPRETING THE QUOTA VALUES

To get reliable results for the maximum values you should usually run the PQUOTA for several hours or even longer, depending on the type of the application process you are monitoring.

Maximum resource usage values less than 5 % indicate that that particular quota may be set too high.

Usage values of 90 % or more are displayed blinking to draw attention to a quota value that may be too low.

4.3 ERROR MESSAGES

Possible error messages from PQUOTA include:

#SS$_INVARG	-Unknown MWAIT state, should never happen.
#SS$_ITEMNOTFOUND	- Can not read Process Control Block (PCB), Process Header (PHD) or Job Information Block (JIB).
#SS$_NONEXPR	- No process found on local node for the specified PID.
#SS$_REMOTE_PROC	- Target process not found on local node. PQUOTA can monitor only local processes.

4.4 RESTRICTIONS

Quotas may only be displayed for processes running on the same node.

The SWAPPER is not allowed as target process.

N.B.: PQUOTA runs partly in Kernel Mode, i.e. a program error or access violation usually leads to a system crash. Thus, PQUOTA SHOULD NOT BE EXECUTED IN A PRODUCTION ENVIRONMENT.

4.5 PQUOTA KIT FILES

The PQUOTA distribution kit consists of:

PQUOTA.EXE	Executable file for OpenVMS AXP.
PQUOTA.VAX_EXE	Executable file for OpenVMS VAX.
PQUOTA.COM	Command file to build PQUOTA on OpenVMS AXP and VAX.
PQUOTA.MAR	Common source file for OpenVMS AXP and VAX.
PQUOTA.PS	PQUOTA User's guide Postscript file (i.e. this document).
PQUOTA.TXT	PQUOTA User's guide text file.
PQUOTA.DOC	PQUOTA User's guide DECwrite source file .
ARCH_DEFS.MAR	Architectural definitions file for ALPHA.

4.6 BUILD OF PQUOTA EXECUTABLE

To build the PQUOTA executable image, set your current default directory to the one containing the PQUOTA kit files and execute the command file PQUOTA.COM as follows:

```
$ @PQUOTA
```

5. SCREEN DISPLAY

The display is updated once per second. If a maximum used value reaches 90 %, then the maximum values for that particular quota will blink.

Ctrl-W will refresh the screen, and Ctrl-Z exits the PQUOTA program.

The display is currently almost identical on both VAX and Alpha AXP platforms. The only difference is in the reporting of the PGFLQUO and WSQUO quotas, which are shown in units of 'pagelets' on an AXP, but in units of 'pages' on a VAX. However, both VAX pages and AXP pagelets are equal to 512 bytes in size. (The AXP page size is currently 8192 bytes.)

If the target process is outswapped, then some of the process information is not available. The fields CPU time, ASTCNT and WSQUO are displayed as " * Outswapped * ", the AST's enable field is displayed as "----".

VIRT SIZE is the size of current virtual address space in 512 byte page(let)s.

5.1 SAMPLE DISPLAYS

On an AXP:

```
  Node      : AXP               PQUOTA V1.15               12-MAR-1998 18:18:43
  Process   : _FTA15:           CPU :   0 00:03:27.38             Ena  Pend
  Username : SALMINEN           State : COM                AST's: KESU ----
  Virt size: 38448              I/O's : 456879             Pflts: 6158
         ---- Sysgen ----    -------------- Process ---------------
                                                 curr        max
         max/def      min       max    free    used   %    used   %
 ==================================================================
 BYTCNT  1000000  1000000   1000000  484160  515840  56  524032  57  Bytes
 BIOCNT     1000      200      3150    3097      53   1      54   1  IO's
 DIOCNT     1000      200      1250    1197      53   4      54   4  IO's
 ASTCNT      200      200      4096    3932     164   4     166   4  Ast's
 FILCNT      200      200      1000     985      15   1      15   1  Files
 ENQCNT     1000     1000     20000   19990      10   0      10   0  Locks
 TQCNT       200       64      1200    1190      10   0      10   0  Timers
 PGFLQUO   65536    65536     50000   48940    1060   2    1060   2  Pagelets
 WSQUO    327680       20     20669   18135    2534  12    2534  12  Pagelets
```

On a VAX:

```
 Node     : VAX              PQUOTA V1.9                31-JAN-1996 10:57:21
 Process  : _FTA12:          CPU :   0 00:00:33.84             Ena  Pend
 Username : SALMINEN         State : LEF                AST's: KESU ----

        ---- Sysgen ----    -------------- Process ---------------
                                                curr        max
        max/def      min       max     free     used   %    used   %
==================================================================
BYTCNT   100000    65536    100000    98336     1664   1     1664   1  Bytes
BIOCNT      100       18       100      100        0   0        0   0  IO's
DIOCNT      100       18       100      100        0   0        0   0  IO's
ASTCNT      600       24       600      597        3   0        3   0  Ast's
FILCNT      300       16       300      298        2   0        2   0  Files
ENQCNT      200       30       200      200        0   0        0   0  Locks
TQCNT       200        0       200      200        0   0        0   0  Timers
PGFLQUO   32768      512     30000    29799     1201   0     1201   0  Pages
WSQUO    236214       20    234484   233886      598   0      598   0  Pages
```

On an AXP for an outswapped process:

```
 Node     : AXP              PQUOTA V1.9                16-APR-1996 19:11:57
 Process  : CONFIGURE        CPU : * Outswapped *              Ena  Pend
 Username : SYSTEM           State : HIBO               AST's: ---- ----

        ---- Sysgen ----    -------------- Process ---------------
                                                curr        max
        max/def      min       max     free     used   %    used   %
==================================================================
BYTCNT    65536   100000    100000    99552      448   0      448   0  Bytes
BIOCNT       32      100       200      200        0   0        0   0  IO's
DIOCNT       32      100       200      200        0   0        0   0  IO's
ASTCNT   * Process is Outswapped *                                     Ast's
FILCNT      128      100       200      198        2   1        2   1  Files
ENQCNT       64      300       300      300        0   0        0   0  Locks
TQCNT        16        0       200      200        0   0        0   0  Timers
PGFLQUO   65536    32768     32768    31568     1200   3     1200   3  Pagelets
WSQUO    * Process is Outswapped *                                     Pagelets
```

5.2 QUOTA VALUES

There are two main quota value categories, "Sysgen" and "Process":

5.2.1 Sysgen Values

max /def min

The maximum/default and minimum limits per quota value that are set system-wide.
(These values correspond to the appropriate SYSGEN parameters).

5.2.2 Process Values

max

The maximum limit per quota value that is specified for the process owner in the SYSUAF.DAT file. (As defined with the AUTHORIZE utility.)

free

The currently available amount per resource.

curr
used %

The current amount used and percentage used per resource.

max
used %

The maximum amount used and maximum percentage used per resource since the beginning of the current PQUOTA run.

The rightmost column of the display indicates the units of measurement of each resource quota value. For instance, BYTCNT is shown in bytes and BIOCNT in (number of) IO's.

6. EXPLANATION OF QUOTA VALUES

The System Max(imum) and Min(imum) values shown below are the names of corresponding SYSGEN parameters.

The Process Max(imum) in each case is the name of a field in an AUTHORIZE entry. The runtime value for a process maximum is the value of this AUTHORIZE entry for the process owner.

6.1 BYTCNT - Buffered I/O Byte Limit

The maximum number of bytes of nonpaged system dynamic memory that a user's job can consume at one time.

Nonpaged dynamic memory is consumed in such operations as I/O buffering, mailboxes, and file-access windows.

System Max: PQL_DBYTLM

Min: PQL_MBYTLM

Process Max: BYTLM

6.2 BIOCNT - Buffered I/O Count Limit

The maximum number of buffered I/O operations, such as terminal I/O, that can be outstanding at one time.

System Max: PQL_DBIOLM.

Min: PQL_MBIOLM.

Process Max: BIOLM

6.3 DIOCNT - Direct I/O Count Limit

The maximum number of direct I/O operations (usually disk) that can be outstanding at one time.

System Max: PQL_DDIOLM

Min: PQL_MDIOLM

Process Max: DIOLM

6.4 ASTCNT - AST Queue Limit

The total number of asynchronous system trap (AST) operations and scheduled wake-up requests that the user can have queued at one time.

System Max: PQL_DASTLM
Min: PQL_MASTLM
Process Max: ASTLM

6.5 FILCNT - Open File Limit

The maximum number of files that can be open at one time, including active network logical links.

System Max: PQL_DFILLM
Min: PQL_MFILLM
Process Max: FILLM

6.6 ENQCNT - Lock Queue Limit

The maximum number of locks that can be queued by the user at one time.

System Max: PQL_DENQLM
Min: PQL_MENQLM
Process Max: ENQLM

6.7 TQCNT - Timer Queue / Temporary Common Event Flag Clusters Limit

The maximum number of entries in the timer queue plus the number of temporary common event flag clusters that the user can have at one time.

System Max: PQL_DTQELM
Min: PQL_MTQELM
Process Max: TQELM

6.8 PGFLQUO - Page File Quota

The maximum number of pages that the process can use in the system paging file.

System Max: PQL_DPGFLQUOTA
Min: PQL_MPGFLQUOTA
Process Max: PGFLQUOTA

6.9 WSQUO - Working Set Quota

The maximum amount of physical memory a user process can lock into its working set.

Also represents the maximum amount of swap space that the system reserves for this process and the maximum amount of physical memory that the system allows the process to consume if the systemwide memory demand is significant.

System Max: PQL_DWSQUOTA
Min: PQL_MWSQUOTA
Process Max: WSQUOTA

6.10 AST's - Enabled and pending AST's for the process

The enabled and pending AST's for this process are shown in the upper right corner. The "Ena" field shows the currently enabled AST's and the "Pend" field shows currently pending AST's. Usually the display should be as follows:

```
       Ena  Pend
AST's: KESU ----
```

If any pending AST's are shown, the process is probably either hanging or stopped at a debugger breakpoint in an AST routine.

The letters KES and U correspond to the processor modes:

K = Kernel
E = Executive
S = Supervisor
U = User

An entry of - indicates that the corresponding AST is not enabled or not pending (depending on whether it is shown under the 'Ena' heading or the 'Pend' heading.)

6.11 Virt size - Size of process virtual address space

The size of the current virtual address space in page(let)s.

6.12 I/O’s - Process total I/O’s

The total amount of buffered and direct I/O’s since the process was started.

6.13 Pflts - Process total page faults

The total amount of page faults since the process was started.