

[OpenVMS] How To Troubleshoot A Process In RWAST

Last Technical Review: 11-OCT-2000

Copyright (c) 1997, 2000 Compaq Computer Corporation. All rights reserved.

PRODUCT: DIGITAL OpenVMS VAX, All Versions
DIGITAL OpenVMS Alpha, All Versions

COMPONENT: Scheduler

SOURCE: Compaq Computer Corporation

OVERVIEW:

The DCL command "SHOW SYSTEM" displays one or more processes in an RWAST state, and the process seems to be hung, or locked. How can you determine why the process is in this state?

BACKGROUND:

The RWAST is a general purpose 'Resource Wait' state. It indicates the wait is expected to be satisfied by the delivery and/or enqueueing of an AST to the process.

Reasons why a process goes into RWAST include:

- o waiting for an I/O to complete on a channel
- o running out of a process or system quota
- o waiting for a file system or lock request to complete
- o waiting for a subprocess to terminate

Other database articles discuss BACKUP and print symbiont processes in RWAST and RWAST states on systems running OpenVMS Workstation Software (VWS).

Processes in the RWAST state CAN'T be deleted until the condition they are waiting for is satisfied. If you can not identify what the process is waiting for, you will have to reboot the system to eliminate the process.

If the process in RWAST state is a user-written program, it is possible to receive an error status rather than OpenVMS putting the process into resource wait for some system service calls. Usually, the status indicates either a quota problem or insufficient pooled memory. This is accomplished by using the SYS\$SETRWM system service call as described in the "VAX/VMS Version 4.4 System Service Reference Manual", (April 1986), page SYS-376.

Note:

This article is for both OpenVMS VAX and Alpha. Commands relating to the specific platform will be preceded with either (VAX) or (Alpha).

PROCEDURE:

To find out why processes are in an RWAST state, use the System Dump Analyzer (SDA):

- 1. Invoke SDA:

```
$ ANALYZE/SYSTEM
VAX/VMS System analyzer
```

Define data structures and symbols:

```
SDA> READ/EXECUTIVE
(VAX) SDA> READ SYS$SYSTEM:SYSDEF
(ALPHA) SDA> READ SYS$LOADABLE_IMAGES:SYSDEF
```

Note:

This step must be executed to ensure that the offsets SDA may display during the analysis of this problem match the offsets presented in this article, e.g.; if the READ command isn't executed EXE\$DELPRC_C+143C will be displayed as PROCESS_MANAGEMENT_NPRO+0ABBC.

Find the RWAST process and its index (Indx):

```
SDA> SHOW SUMMARY
```

Current process summary

Extended -- PID	Indx --	Process name	Username	State	Pri	PCB	PHD	Wkset
20200080	0000	NULL		COM	0	800024A8	80002328	0
20200081	0001	SWAPPER		HIB	16	80002748	800025C8	0
20201005	0005	JONES	JONES	LEF	4	80363C50	82CEEE00	211
20200086	0006	ERRFMT	SYSTEM	HIB	7	8030CA80	80A2FA00	88
20200087	0007	CACHE_SERVER	SYSTEM	HIB	16	80317F70	80C3AE00	62
2020104F	004F	SMITH	SMITH	RWAST	6	8036CE90	82DF4800	200

//
 |
 +-----+
 |
 |
 |

- 2. Set your default to the RWAST process using its INDEX value.

```
SDA> SET PROCESS/INDEX=4F <-----+
|
```

Note:

If you have tried to delete the process, SDA may not permit you to set your process to the RWAST process. In this case, you would receive the following error:

```
%SDA-E-NOTINPHYS, xxxxxx: not in physical memory
```

If you receive this error, you may have to format the PCB and/or JIB to figure out the problem. The address for the PHD and PCB can be found from the SHOW SUMMARY display. The address for the JIB will be at offset PCB\$L_JIB in the formatted PCB. Keep this in mind if SDA will not allow you 'normal' access to the data structures that follow.

If you can get no access to the process data structures, for example the process header is outswapped, you may have to reboot the system and wait for the problem to occur

again. If it happens again, you may be able to catch the data structures in memory and analyze the Resource Wait state more thoroughly.

3. Find the process Program Counter (PC) and see if it evaluates to one of the following symbols:

```
(VAX) SDA> EXAMINE @PC
(Alpha) SDA> SHOW CALL
```

Note:

For Alpha systems, use the "Return address" from the from the "Call Frame" as the PC.

It's also important to note that even if the PC doesn't equate to any of those listed in step 3, the rest of this article may be used to further evaluate the process.

- a. (VAX) EXE\$DASSGN+84 (V5.0 through 7.0)
(VAX) EXE\$DASSGN+9D (V7.1 through 7.2)
(Alpha) EXE\$DASSGN_C+00194 (V1.5 and 6.1)
(Alpha) EXE\$DASSGN_C+0018C (V6.2)
(Alpha) EXE\$DASSGN_C+00204 (V7.0)
(Alpha) EXE\$DASSGN_C+00240 (V7.1)
(Alpha) EXE\$DASSGN_C+0023C (V7.1-1H2)
(Alpha) EXE\$DASSGN_C+00224 (V7.2, V7.2-1)

This means the process is waiting for an I/O request to complete. (V4 offset: EXE\$DASSGN+6D)

*** Go to Step 4 ***

- b. (VAX) EXE\$MULTIQUOTA+5E (V5.0 through 7.2)
(Alpha) EXE\$MULTIQUOTA_C+00180 (V1.5)
(Alpha) EXE\$SNGLEQUOTA_LONG_C+0016C (V1.5)
(Alpha) EXE\$MULTIQUOTA_C+00168 (V6.1 and V6.2)
(Alpha) EXE\$SNGLEQUOTA_LONG_C+00154 (V6.1)
(Alpha) EXE\$SNGLEQUOTA_LONG_C+0015C (V6.2)
(Alpha) EXE\$MULTIQUOTA_C+00184 (V7.0)
(Alpha) EXE\$MULTIQUOTA_C+00190 (V7.1)
(Alpha) EXE\$MULTIQUOTA_C+0017C (V7.2)

The process has exhausted an AUTHORIZE or SYSGEN quota. (V4 offset: EXE\$MULTIQUOTA+032)

*** Go to Step 5 ***

- c. (VAX) EXE\$DELPRC+BD (V5.0 through V5.1)
(VAX) EXE\$DELPRC+D4 (V5.2 & V5.2-1)
(VAX) EXE\$DELPRC+EA (V5.3 through V5.3-2)
(VAX) EXE\$DELPRC+10C (V5.4 through V5.5-2)
(VAX) EXE\$DELPRC+112 (V6.0 through V7.2)
(Alpha) EXE\$DELPRC_C+00DBC (V1.5)
(Alpha) EXE\$DELPRC_C+00CAC (V6.1)
(Alpha) EXE\$DELPRC_C+00CE4 (V6.2)
(Alpha) EXE\$DELPRC_C+01250 (V7.0)
(Alpha) EXE\$DELPRC_C+01334 (V7.1)
(Alpha) EXE\$DELPRC_C+011F4 (V7.2)
(Alpha) EXE\$DELPRC_C+01204 (V7.2-1)

The process is waiting for a file system or lock request to complete. (V4 offset: EXE\$DCLEXH+0A5)

*** Go to Step 6 ***

- d. (VAX) EXE\$DELPRC+FB (V5.0 through V5.1-1)
- (VAX) EXE\$DELPRC+112 (V5.2 & V5.2-1)
- (VAX) EXE\$DELPRC+128 (V5.3 through V5.3-2)
- (VAX) EXE\$DELPRC+14A (V5.4 through V5.5-2)
- (VAX) EXE\$DELPRC+1B9 (V6.0 through V7.2)
- (Alpha) EXE\$DELPRC_C+00EB8 (V1.5)
- (Alpha) EXE\$DELPRC_C+00DA8 (V6.1)
- (Alpha) EXE\$DELPRC_C+00DE8 (V6.2)
- (Alpha) EXE\$DELPRC_C+01358 (V7.0)
- (Alpha) PROCESS_MANAGEMENT_NPRO+9798 (V7.0)
- (Alpha) EXE\$DELPRC_C+0143C (V7.1)
- (Alpha) PROCESS_MANAGEMENT_NPRO+0ABBC (V7.1)
- (Alpha) EXE\$DELPRC_C+0147C (V7.1-1H2)
- (Alpha) EXE\$DELPRC_C+012F8 (V7.2)
- (Alpha) PROCESS_MANAGEMENT_NPRO+B7B8 (V7.2)

The process is waiting for a subprocess to terminate.
(V4 offset: EXE\$DCLEXH+141)

*** Go to Step 7 ***

Other RWAST states are possible but very rare. If the PC does not evaluate to one of the above symbols, you will have to reboot the system to eliminate the hung process. Take a crash dump of the system to determine why the process was in RWAST.

Occasionally, the RWAST process will clear itself if the process waits long enough and the AST somehow gets satisfied.

On OpenVMS Alpha a PC of EXE\$KERNEL_WAIT_PS_C may indicate a resource wait associated with a resource depletion. Processes in RWAST at this PC should be evaluated as if they're in a MUTEX state. (See the RELATED ARTICLE section)

- 4. The process is waiting for an I/O request to complete. Many times, the device will be shown as "Busy" in the SHOW PROCESS/CHANNEL display:

```
SDA> SHOW PROCESS/CHANNEL
Process index: 004F Name: SMITH Extended PID: 22E0124D
```

```
-----
                Process active channels
                -----
Channel  Window      Status  Device/file accessed
-----  -
0010    00000000
00C0    00000000      Busy   LPA0: <-----+
00D0    00000000      MBAl:  |
                                     |
```

In the above example, you have only one "BUSY" channel, so this must be the channel causing the process to hang in RWAST.

For the EXE\$DASSGN+84 offset only, if you have multiple BUSY channels, you can identify which one is causing the RWAST state with the following commands:

```
SDA> READ SYS$SYSTEM:SYSDEF.STB ! read in system symbols
SDA> SHOW DEVICE/ADDRESS=@@R6
```

If the device is a printer connected to a terminal port and the symbiont is waiting for an XON to be delivered, occasionally turning the printer OFF and back ON again will cause an XON to be sent back to the VAX. This allows the I/O to complete, permitting the print symbiont to continue.

If the device is a printer connected to a printer port (LPA0, LCA0...), the VAX thinks the printer is offline. This may indicate a hardware problem with the printer or controller, if it is really online. Again, turning the printer OFF and back ON again may help.

If the device is a tape drive, setting TAPE_MVTIMEOUT to a very low value should cause the outstanding operation to timeout, allowing the process to continue. Another database article discusses modifying this parameter when a process running BACKUP hangs in an RWAST state. This article can be found using a search string of:

```
BACKUP RWAST
```

If the device is an RTA terminal created by a SET HOST command, disconnect the remote link as follows:

Note:

Another article in the DECNET-VMS database describe methods for converting NCP commands to NCL. This article can be found using a search string of:

```
CONVERSION NCL NCP COMMANDS
```

- a) Use SHOW USERS/FULL to determine the remote port information (indicated by ^^^ below):

```
$ SHOW USERS/FULL username
```

```
VAX/VMS User Processes at 18-MAY-1994 19:17:21.91
Total number of users = 1, number of processes = 1
```

```
Username  Node  Process Name  PID  Terminal
USER2     NODEA  USER2        2020566C  RTA1: (3602::_PC_0E02)
                                         ^^^^^^^^^
```

- b) Examine current remote links:

```
$ RUN SYS$SYSTEM:NCP
NCP> SHOW KNOWN LINKS
```

```
Link Node  PID  Process  Remote link  Remote user
317  3.530 20200214  REMACP      3733         _PC_0E02
                                         ^^^^^^^^^
```

- c) Disconnect link where "Remote user" matches SHOW USERS terminal port information:

```
NCP> DISCONNECT LINK 317
```

5. The RWAST state indicates the process has run out of a quota. A SHOW SYSTEM display will often show the process continuing to accumulate CPU time.

You can often determine which quota the process has exhausted by issuing the following command:

```
SDA> SHOW PROCESS          ! for the SMITH process
Process index: 004F      Name: SMITH      Extended PID: 2020104F
-----
Process status: 02040001  RES,PHDRES

PCB address      8036CE90  JIB address      8064F3C0
PHD address      82DF4800  Swapfile disk address 01002821
Master internal PID 0020004F  Subprocess count      0
Internal PID     0020004F  Creator internal PID  00000000
Extended PID     2020104F  Creator extended PID  00000000
State            RWAST    Termination mailbox   0000
Base priority    4        AST's active          NONE
UIC              [00022,000016]  AST's remaining       16
Mutex count      0        Buffered I/O count/limit 0/18 <---+
Waiting EF cluster 0        Direct I/O count/limit  18/18 <---+
Starting wait time 1B001B1B  BUFIO byte count/limit 30478/31936<--+
      .                .                |
      .                .                |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|
```

Note these fields to see if any quotas are down to zero. In the example above, you can see that "Buffered I/O count/limit" is zero.

Note:

The SHOW PROCESS display for OpenVMS Alpha is slightly different from the OpenVMS VAX display.

The number before the slash (/) is the amount of this quota left. The number after the slash is the total amount allowed. These fields correspond to the following limits in the User Authorization File (UAF), or to the SYSGEN PQL parameters if the process is a detached process:

	UAF	SYSGEN PQLs
	-----	-----
AST's remaining	- ASTLM	PQL_DASTLM
Buffered I/O count/limit	- BIOLM	PQL_DBIOLM
Direct I/O count/limit	- DIOLM	PQL_DDIOLM
BUFIO byte count/limit	- BYTLM	PQL_DBYTLM
# open files allowed left	- FILLM	PQL_DFILLM
Timer Entries allowed left	- TQELM	PQL_DTQELM

Once you have determined which quota needs to be increased for processes or subprocesses, increase that value where appropriate, as indicated in the following table:

For this type of process:	Increase this type of parameter:
Process	UAF
Subprocess	UAF
Detached	SYSGEN PQL
Process which creates the detached process	Parameter in the SYS\$CREPRC system service call

The new UAF or SYS\$CREPRC value is used when a new process is created.

The new SYSGEN PQL parameter is used when you log out and log back in again.

At this point, reboot the system to eliminate the RWAST process waiting for quota.

If the RWAST process is waiting for a quota and the quota does not appear to be any of these, you can format and display the Job Information Block (JIB), Process Control Block (PCB), and the Process Header (PHD) to locate the quota problem.

R2 contains the address of the insufficient quota. To determine the insufficient quota, do the following:

```
SDA> EXAMINE R2
R2: 8036CECA
```

Next, locate the addresses of the PCB (Process Control Block) and the JIB (Job Information Block) from the top of the SHOW PROCESS display. The value found in R2 will be pointing somewhere in one of these two data structures. Identify which data structure would contain the value in R2 and format that data structure. In this case, R2 would be in the PCB so the PCB needs to be formatted:

Note:

The PCB definition in SDA contain hexadecimal values, while the UAF values are in decimal. Also, OpenVMS Alpha uses a longword for some of the PCB definitions, whereas OpenVMS VAX uses a word, i.e; PCB\$L_ as opposed to PCB\$W_.

```
SDA> FORMAT 8036CE90
8036CE90  PCB$L_SQFL          80002180
8036CE94  PCB$L_SQBL          80002180
      .
      .
      .
8036CEC6  PCB$W_PPGCNT         008C
8036CEC8  PCB$W_ASTCNT         0010
+--> "8036CECA" PCB$W_BIOCNT     0000  <-----+
|      8036CECC  PCB$W_BIOLM     0012  |
|      .          .             .      |
|      .          .             .      |
|      .          .             .      |
```

+-- This address matches the contents of R2. The value for PCB\$W_BIOCNT is zero, indicating that the BIOLM quota is depleted. This process requires either an increase in its UAF quota or its application program modified to decrease the number of outstanding buffered I/O requests.

If the address is not found in the PCB, format the JIB. The JIB address can be found from either the SHOW PROCESS display or the PCB\$L_JIB value above:

```
SDA> FORMAT 8064F3C0          ! Formatting JIB of process
```

6. The process may be waiting for the file system to complete a request or for a lock request. EXE\$DCLEXH+0A5 is returned if you have tried to delete the process, whose former state was probably LEF.

If the system is running OpenVMS Workstation Software (VWS) and the RWAST process is a VWS process, check the following:

- Does the system have enough pool?
- Does the system have enough global pages?
- What is the error status in R11?

Check if the RWAST process is waiting for an XQP file request to complete: format the PCB and look for a non-zero value in PCB\$B_DPC (Alpha = PCB\$L_DPC). If the process is being forced to wait under these circumstances, the SDA 'SHOW PROCESS' command displays the PROCESS status as "DELPEN" or "SUSPEN". In the following example, the process is in the delete pending state with a status of "DELPEN".

```

SDA> SHOW PROCESS
Process index: 00DF   Name: Mike Mc. | Extended PID: 000007DF
-----V-----
Process status: 02040023   RES,DELPEN,RESPEN,PHDRES

PCB address      +-- 80339230   JIB address      804FE7E0
PHD address      | 8092D000   Swapfile disk address 010065A1
.               | .           .
.               | .           .
                v
SDA> FORMAT 80339230
80339230   PCB$L_SQFL           80002180
80339234   PCB$L_SQBL           8032D5C0
.         .                   .
.         .                   .
8033925A   PCB$B_DPC           01 <--+
.         .                   .
.         .                   .
+-----+
|

```

- o If the value is non-zero, then the process is waiting for XQP (file system) activity to complete.
- o If the value is zero (00), the RWAST is not waiting for the XQP and you can check for outstanding lock requests using the command SDA> SHOW PROCESS/LOCK. You will often find a lock in either "Waiting for" or "Converting to" state.

(Two other database articles describe how to trace lock requests on both clustered and nonclustered systems.)

The process holding the lock this process is waiting for is often in a RWxxx state itself, and solving that other process's problem may clear up this process's RWAST state.

If you are at this address, EXE\$DCLEXH+0A5, and the process is not getting CPU time or waiting for XQP or lock operations to complete, reboot the system to eliminate the process. Take a crash dump for later examination if the problem occurs often.

- o If the value is non-zero (01), the process is waiting for a XQP file system request. It could be waiting for a lock request, for paged pool, or for another blocked file system request (such as insufficient ACP_MAPCACHE on a badly fragmented disk).

First check the process to see if it is waiting for a file system lock by using the SDA 'SHOW PROCESS/LOCK' command. If a lock is "Waiting for" or "Converting to", start tracing the lock request to isolate the problem.

Other articles in the OPENVMS database describing how to trace lock requests on clustered and non-clustered systems can be found using a search string of:

TRACE LOCK CLUSTER

If the process is not waiting for a lock, issue the following commands to see if it might be waiting for paged pool (PAGEDYN). If it is not waiting for either a lock request or paged pool, then you may want to contact the Customer Support Center to investigate the problem further.

WARNING: Do not use this command when troubleshooting an RWAST process on a V5 system. The process that issues the command may hang.

SDA> SHOW PROCESS/CHANNELS

Process index: 00DF Name: Mike Mc. Extended PID: 000007DF

```
-----
                Process active channels
                -----
Channel  Window          Status          Device/file accessed
-----  -
0010    00000000
0040    00000000
0050    00000000
0070    00000000          Busy          DUA2:
                                         VTA52:
                                         VTA52:
                                         DUA2:
```

An RWAST process may be waiting for XQP because the system's paged dynamic memory (PAGEDYN) is depleted. Increase the value of the nondynamic SYSGEN parameter PAGEDYN. You should do this with the AUTOGEN command procedure. PAGEDYN should normally be at least 1/3 (33%) free, and up to 40% free on a busy system.

To determine how much PAGEDYN has been used, issue the following command:

SDA> SHOW POOL/SUMMARY/PAGE

Paged dynamic storage pool

Summary of paged pool contents

```
108 UNKNOWN = 357984 (20%)
  2 LOG      =  83728 (4%)
. . . . .
  1 CI       =    96 (0%)
  1 CLU      =  2384 (0%)
```

Total space used = 1719808 out of 1988608 total bytes,
268800 bytes left Total space utilization = 86% <---+

```
-----+
|
This indicates that only 14% is left
```

To determine if the RWAST process is waiting for PAGEDYN, you can do the following:

Create the following macro to set up a symbol definition to format data structures to get necessary symbols defined for the next step:

```
SDA> SPAWN
$ MACRO/OBJ=F11BDEF.STB SYS$INPUT+SYS$LIBRARY:LIB/LIB
  $F11BCDEF GLOBAL
  $F11BDEF GLOBAL <--- (VAX only)
  .END
<CNTRL Z>
$ LO
SDA> READ F11BDEF
```

Look for the "AQB address" on the 4th screen of the following display:

```
SDA> SHOW DEVICE DUA2 <--- DUA2 is the device shown "busy"
                             from the SHOW PROCESS/CHANNELS
                             command.

I/O data structures
-----
DUA2                RA80                UCB address:  80484B90
.                   .                   .
.                   .                   .
Press RETURN for more.
SDA>
I/O data structures
-----

--- Volume Control Block (VCB) 8048AD70 ---

Volume: NODE_1_PAGE      Lock name: NODE_1_PAGE
Status:  A0 extfid,system
Status2: 05 writethru,mountver

Mount count      1  Rel. volume      0  AQB address  80BA4AA0-+
Transactions     2  Max. files    29651  RVT address  80484B90 |
Free block    34020  Rsvd. files     9  FCB queue   808A9D10 |
Window size     7  Cluster size    3  Cache blk.  80768460 |
Vol. lock ID   B1  Def. extend sz.  5
Block. lock ID A7  Record size    0

SDA> FORMAT 80BA4AA0 <-----+
80BA4AA0  AQB$L_ACPQFL          80BA4AA0
.         .                   .
.         .                   .
80BA4AB6  AQB$B_CLASS          00
80BA4AB7          00
80BA4AB8  AQB$L_BUFCACHE      80274380 -----+
.         .                   .
.         .                   .

SDA> FORMAT 80274380 <-----+
80274380  F11BC$L_BUFBASE      80297400
.         .                   .
```


The following DCL command procedure can be used to check all subprocesses on the system to find which process has the RWAST parent process.

```

$ a      = ""
$ pid    = ""
$ context = ""
$ count  = 0
$Loop:
$ pid = F$PID(context)
$ IF pid .EQS. "" THEN GOTO Done
$ IF count .EQS. 1 THEN GOTO Skip
$ count = count+1
$ header = "Main      Sub      Ownr PID  Sub PID"
$ text1 = F$FAO("!35AS ",header)
$ WRITE SYS$OUTPUT text1
$Skip:
$ qpid = a+pid+a
$ owner = F$GETJPI('qpid,"OWNER")
$ IF owner .EQS. "" THEN GOTO Loop
$ username = F$GETJPI('qpid,"USERNAME")
$ prcnam = F$GETJPI('qpid,"PRCNAM")
$ imagename = F$GETJPI('qpid,"IMAGENAME")
$ imagename = F$PARSE(imagename,,,"NAME")
$ text = F$FAO("!8AS !8AS !8AS !8AS ",username,prcnam,owner,pid)
$ WRITE SYS$OUTPUT text
$ GOTO Loop
$Done:

```

If no subprocess is found for the parent process, the parent process will wait forever and the system will have to be rebooted to eliminate this process. It could be that privileged code is altering the subprocess PCB\$L_OWNER field so process termination does not know about the parent process.

The parent also hangs in RWAST if the subprocess changed its UIC, as with the DCL command "SET UIC", which requires CMKRNL privilege. If the subprocess does not return the UIC to its original value, the parent process is never notified when the subprocess terminates, even for a normal termination. The parent hangs in HIBernate. If you try to get rid of the parent with the command "STOP/PROCESS", it hangs in RWAST.

To find the subprocess in a crash dump, you need to locate the OWNER field whose PID matches that of the parent process in RWAST. This can be done by displaying every owner field in every PCB available on the system.

- 1) SDA> READ SYS\$SYSTEM:SYSDEF (VAX)
 - SDA> READ SYS\$LOADABLE_IMAGES:SYSDEF (Alpha)
 - 2) SDA> SHOW SUMMARY !to get all the PCB addresses
 - 3) SDA> EXAMINE <pcb_address>+PCB\$L_OWNER
- |
- +-- Execute this command for each PCB address.

When you find the process whose owner/parent is the PID of the process in RWAST, you can start analyzing why the subprocess is not terminating.

Note:

When a crash is forced on an OpenVMS Alpha, V6.1 and later, the process context for any RWAST process is saved in a selective dumpfile. The system manager can control which processes should be dumped early by using the following process.

1. The SYS\$SYSTEM:SYS\$DUMP_PRIORITY.TEMPLATE should be copied to .DAT.
2. Edit the template to include the set of processes to be considered as priority. The template file explains how to do this.
3. Run SYS\$SYSTEM:SYS\$SET_DUMP_PRIORITY.EXE to include these processes. This should be executed at STARTUP and at other times to replace any existing process list with a new one.