HP NonStop RPM

Real-time Process Monitor

User's Guide

Abstract

HP NonStop™ Real-time Process Monitor (RPM) is a high-performance, low-overhead, Cpu and Process monitoring utility specifically engineered for NonStop servers.

RPM continuously discovers busy Cpus/Processes, and sorts, prioritizes, color-encodes, and displays real-time resource statistics by Cpu, or by Node, or across a whole cluster of nodes.

This document describes how to install, configure, and use NonStop RPM. Additional information about HP NonStop RPM is available at the HP NonStop RPM technical portal http://NonStopRPM.com.

Product Version

V01

Supported Release Version Updates (RVUs)

This manual supports G06.20 and all subsequent G-series RVUs, and H06.08 and all subsequent H-series RVUs until otherwise indicated in a new edition.

Part Number	Published
545801-002	July 2008

Document History

Part Number	Product Version	Published
545801-001	V01	April 2008
545801-002	V01	July 2008

545801-002 - July 2008 Changes

- Corrected footer text in sections 4.2 and 4.3

Table of Contents

What’s New in This Manual

New and Changed Information

RPM 1.0

This is the first version of this manual.

About This Manual

Audience

The intended audience for this document is system managers, administrators, and developers responsible for maintaining and monitoring HP NonStop Servers.

Related Documents

None

Manual Organization

Section	**Description**
Overview	Overview and architecture of the HP NonStop RPM product.
Installing NonStop RPM	Procedures for installing the NonStop HP NonStop RPM product.
Configuring RPM	Procedures for configuring and managing the NonStop RPM product.
Running RPM	Procedures for running the NonStop RPM product.
RPM Commands	Procedures for using the RPM, RPM65, and RPMVT command interpreters.

Notation Conventions

Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

> This requirement is described under **Backup DAM Volumes and Physical Disk Drives** on page 25.

General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS

Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

`MAXATTACH`

lowercase italic letters

Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

file-name

computer type

`Computer type` letters within text indicate C and Open System Services (OSS) keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

`myfile.c`

italic computer type

`Italic computer type` letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are required. For example:

`pathname`

[ ] Brackets

Brackets enclose optional syntax items. For example:

`TERM [\`*system-name*`.]$`*terminal-name*

`INT[ERRUPTS]`

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [ num   ]
   [ -num ]
   [ text ]
```

```
K [ X | D ] address
```

{ } Braces

A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name  }
```

```
ALLOWSU { ON | OFF }
```

| Vertical Line

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

… Ellipsis

An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]…
```

```
[ - ] {0|1|2|3|4|5|6|7|8|9}…
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char…"
```

Punctuation

Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;
```

```
LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"[" repetition-constant-list "]"
```

Item Spacing

Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
$process-name.#su-name
```

Line Spacing

If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE

   [ , attribute-spec ]...
```

!i and !o

In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT (  segment-id                   !i
                         , error       ) ;              !o
```

!i,o

In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEDIT ( filenum ) ;                          !i,o
```

!i:i

In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ ( filename1:length               !i:i
                           , filename2:length ) ;           !i:i
```

!o:i

In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ ( filenum                             !i
                       , [ filename:maxlen ] ) ;             !o:i
```

Notation for Messages

This list summarizes the notation conventions for the presentation of displayed messages in this manual.

Bold Text

Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE
?123
CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

Nonitalic text

Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

lowercase italic letters

Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

p-register

process-name

[ ] Brackets

Brackets enclose items that are sometimes, but not always, displayed. For example:

`Event number =` *number* `[ Subject =` *first-subject-value* `]`

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

proc-name `trapped [ in SQL | in SQL file system ]`

{ } Braces

A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

obj-type obj-name `state changed to` *state*`, caused by`\
`{ Object | Operator | Service }`

process-name `State changed from` *old-objstate* `to` *objstate*\
`{ Operator Request. }`\
`{ Unknown.          }`

| Vertical Line

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

`Transfer status: { OK | Failed }`

% Percent Sign

A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400
```

```
%B101111
```

```
%H2F
```

`P=%`*p–register* `E=%`*e–register*

Notation for Management Programming Interfaces

This list summarizes the notation conventions used in the boxed descriptions of programmatic commands, event messages, and error lists in this manual.

UPPERCASE LETTERS

Uppercase letters indicate names from definition files. Type these names exactly as shown. For example:

```
ZCOM-TKN-SUBJ-SERV
```

lowercase letters

Words in lowercase letters are words that are part of the notation, including Data Definition Language (DDL) keywords. For example:

```
token-type
```

!r

The !r notation following a token or field name indicates that the token or field is required. For example:

```
ZCOM-TKN-OBJNAME      token-type ZSPI-TYP-STRING.          !r
```

!o

The !o notation following a token or field name indicates that the token or field is optional. For example:

```
ZSPI-TKN-MANAGER      token-type ZSPI-TYP-FNAME32.         !o
```

1. Overview

1.1. Introduction to NonStop RPM

NonStop **R**eal-time **P**rocess **M**onitor (**RPM**) is a software utility for NonStop servers that displays the busiest Cpus and processes by Cpu, or by node, or across a cluster of Expand nodes.

1.2. NonStop RPM Features

NonStop RPM provides a wide-range of features across a wide-range of device types. In all cases, features are equivalent on all supported devices.

Key Features and Benefits

- Discovers busy activity by Cpu, Node, or Cluster of nodes
- Continuously finds busy Cpus and Processes
- Color-encodes alerts, eg low-blue, medium-yellow, high-red alerts
- Command line configurable, can run from TACL/OSH prompt
- Fast startup, samples, displays < 1 second, very low-overhead
- ByCpu displays busiest processes in a particular Cpu
- ByNode displays busiest processes in a particular node
- "ADD *node*" command allows viewing multiple nodes at once
- Results sorted, filtered, and color-encoded in real-time
- Synchronizes statistics across Cpus, nodes, and multiple users
- Addresses wide variety of interfaces and configurations

General Features

- Easy to install, setup, and configure using the RPMWIZ wizard
- Understands both NSK and OSS processes
- Understands multiple device type interfaces
- Understands dumb terminal TTY devices, allowing output to files/smart-clients

- Understands ANSI/VT100 devices, allowing super-size 200x300 terminal I/O
- Understands T6530 devices, supporting legacy users and 24x80, 54x132

Applicability

RPM is a universally applicable operations tool for NonStop servers. It is designed to do one thing very well, that is real-time monitoring, discovery, and display of the busiest Cpus and processes executing in one or more NonStop servers. The RPM product can be used by a wide-range of systems, devices, and operations personnel. RPM capabilities include:

- Multi-node, multi-operating-system, multi-device aware
- Applicable regardless of what products or applications you are running
- Supports J-, H-, G-, D-, series operating systems
- Supports mixed-version hardware/software networks
- RPM is highly customizable, it can support both small and super-scalar devices with screen sizes ranging from 12" 80x24 x 1-Cpu to 84" 200x300 x 1000-Cpus
- Supports multi-device types TTY, T6530, VT100, and output to disk files
- Provides real-time cluster monitoring capabilities that do not otherwise exist
- Built based on long-term development experience and the requirement to understand real-time dynamics of software in network clusters
- RPM is a proven software development and operations utility that has been evolved and refined over a multi-year period in HP Labs

1.3. RPM Architecture

NonStop RPM consists of two functional components packaged together into one object file which is the RPM object file. The RPM object file acts as both a command interpreter user interface and also as a real-time Cpu and process monitor. To run RPM you simply run RPM from a TACL prompt. RPM utilizes peer-to-peer messaging capabilities that are unique to NonStop servers.

There are three versions of the NonStop RPM command interpreter (CI)

- RPM - Object file for TTY devices. This CI runs in TTY mode, displaying real-time Cpu and process information without embedding color-encoding or screen positioning information.

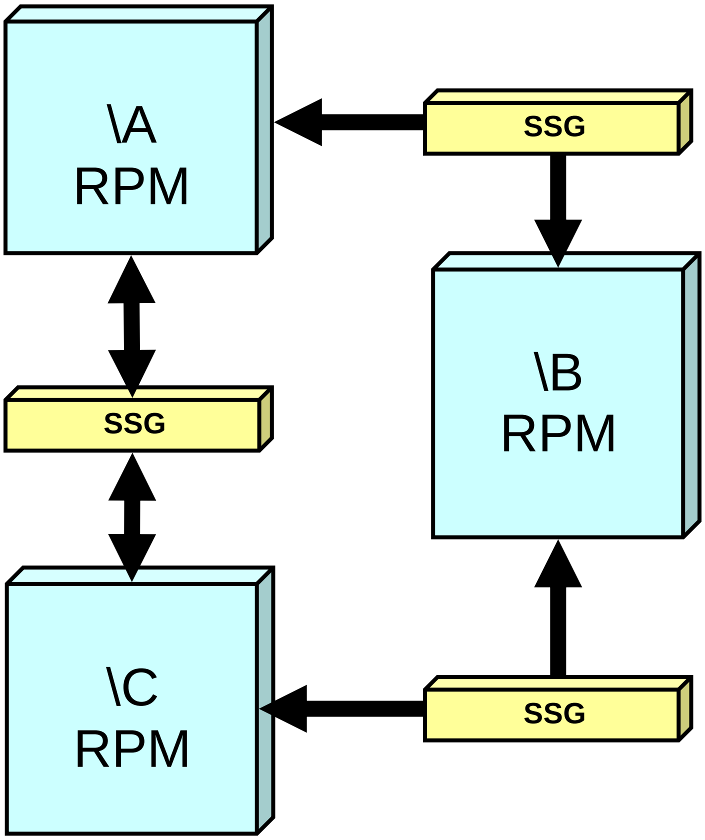
- RPMVT - Object file for ANSI or VT100 devices/emulators. The RPMVT command interpreter runs in VT100 mode, displaying data with color-encoded and screen positioning information embedded in the I/O stream that is compatible with ANSI or VT100 devices.
- RPM65 - Object file for T6530 devices/emulators. The RPM65 command interpreter runs in T6530 mode, displaying data with color-encoded and screen positioning information embedded in the I/O stream that is compatible with T6530 devices.

RPM capabilities are nearly equivalent between device types. In all cases RPM has the ability to quickly discover and display real-time Cpu and Process statistics.

Because all RPM features are available on all supported device types, you have a great deal of flexibility in how you choose to configure RPM within your network. Thus it is up to you to decide which configuration makes the most sense for your particular environment.

For example, Figure 1-1 shows how RPM has been configured to monitor busiest Cpus and processes in a 3 node network.

Figure 1-1 - RPM peer-to-peer example in a 3 node network



An RPM command interpreter can either communicate locally on a single node, or an RPM command interpreter can communicate with other RPM CIs peer-to-peer on other nodes.

As shown in the figure above, when RPM runs in a network it uses HP NonStop SSG messaging middleware to communicate with other RPM processes. The HP NonStop SSG product is standard on all NonStop servers. For more information about the SSG, see the SeeView Server Gateway Manual at http://docs.hp.com.

RPM can monitor all processes in a single CPU, or all processes in a single node, or a whole collection of nodes.

Example #1 - RPM monitoring a single Cpu

In the example below, a process busy (**PB**) command is used to monitor one Cpu, Cpu 3 in this example, on the local node. It displays the 10 busiest processes (**ENTRIES 10**), updating every 5 seconds (**RATE 5**):

```
TACL 1 > RPM PB 3, ENTRIES 10, RATE 5
```

```
Process  Cpu,Pin Busy% Name     RPM      Programs  ET= 5.0 Top Pri User
-------- ------- ----- -------- -------------------------- --- --- -------
\A        3,73   87.82 $STEP    $SYSTEM.SYSTEM.SEEVIEW       1   1  66,1
11:32:10  3,320    .30 $QAZ08   $ASAP.ASAPXQA.QADRV1T        2 168 255,34
          3,263    .20 $HIT     $SYSTEM.SYS03.TSYSDP2        3 220 255,255
          3,319    .15 $QAZ07   $ASAP.ASAPXQA.QADRV1T        4 168 255,34
          3,314    .12 $QAZ06   $ASAP.ASAPXQA.QADRV1T        5 168 255,34
          3,175    .09 $ZOOH3   $DATA2.R0403V02.ASAPXMON     6 189 255,255
          3,0      .06 $MON     $SYSTEM.SYS03.OSIMAGE        7 201 255,255
          3,192    .04 $X11W    $SYSTEM.SYSTEM.RPMVT         8 168  66,1
          3,43     .02 $ZOOM3   $SYSTEM.SYSTEM.ASAPMON       9 160 255,255
          3,312    .02 $QAZ04   $ASAP.ASAPXQA.QADRV1T       10 168 255,34
```

RPM commands can be abbreviated, options and numbers can be pushed together, and commas are optional. For example the following commands are equivalent:

```
TACL 1 > RPM PB 3, ENTRIES 10, RATE 5
TACL 1 > RPM P3 E10 R5
```

Example #2 - RPM monitoring all processes on the local node

In the example below, a process busy (**PB**) command is used to monitor all processes running in all Cpus of the local node. It displays the 10 busiest processes across all Cpus (**ENTRIES 10**), and updates every 5 seconds (**RATE 5**):

```
TACL 1 > RPM PB, ENTRIES 10, RATE 5
```

```
Process  Cpu,Pin Busy% Name     RPM      Programs  ET= 5.0 Top Pri User
-------- ------- ----- -------- -------------------------- --- --- -------
\A        3,73   98.24 $STEP    $SYSTEM.SYSTEM.SEEVIEW       1   1  66,1
11:59:05  0,41    1.25 $ZEXP    $SYSTEM.SYS03.OZEXP          2 170 255,255
          1,159    .82 $Y7Q6    $SYSTEM.SYS03.NSKCOM         3 159 255,255
          1,69     .70 $ZOOB    $SYSTEM.SYSTEM.SEEVIEW       4 160 255,255
          2,271    .58 $HIT     $SYSTEM.SYS03.TSYSDP2        5 220 255,255
          2,279    .52 $HIT     $SYSTEM.SYS03.TSYSDP2        6 220 255,255
          2,28     .38 $X90N    $SYSTEM.SYSTEM.RPM           7 168  66,1
          1,153    .31 $X11Z    $SYSTEM.SYSTEM.RPM           8 167  66,1
          3,263    .31 $HIT     $SYSTEM.SYS03.TSYSDP2        9 220 255,255
          0,0      .30 $MON     $SYSTEM.SYS03.OSIMAGE       10 201 255,255
```

Example #3 - RPM monitoring all processes on one remote node

Below a process busy (**PB**) command is used to monitor all processes in on all Cpus in a remote node, display the 10 busiest processes (**ENTRIES 10**), and update every 5 seconds (**RATE 5**):

```
TACL 1 > RPM PB \B, ENTRIES 10, RATE 5

Process  Cpu,Pin Busy% Name     RPM      Programs  ET= 5.0 Top Pri User
-------- ------- ----- -------- -------------------------- --- --- -------
\B        3,73   98.24 $STEP    $SYSTEM.SYSTEM.SEEVIEW       1   1  66,1
11:59:05  0,41    1.25 $ZEXP    $SYSTEM.SYS03.OZEXP          2 170 255,255
          1,159    .82 $Y7Q6    $SYSTEM.SYS03.NSKCOM         3 159 255,255
          1,69     .70 $ZOOB    $SYSTEM.SYSTEM.SEEVIEW       4 160 255,255
          2,271    .58 $HIT     $SYSTEM.SYS03.TSYSDP2        5 220 255,255
          2,279    .52 $HIT     $SYSTEM.SYS03.TSYSDP2        6 220 255,255
          2,28     .38 $X90N    $SYSTEM.SYSTEM.RPM           7 168  66,1
          1,153    .31 $X11Z    $SYSTEM.SYSTEM.RPM           8 167  66,1
          3,263    .31 $HIT     $SYSTEM.SYS03.TSYSDP2        9 220 255,255
          0,0      .30 $MON     $SYSTEM.SYS03.OSIMAGE       10 201 255,255
```

Example #4 - RPM monitoring all processes on all remote nodes

Below a process busy (**PB**) command is used to monitor all processes running on all Cpus of all remote nodes that have been ADDed to the RPM environment. The output displays the 5 busiest processes (**ENTRIES 5**) and updates every 5 seconds (**RATE 5**):

```
TACL 1 > RPM
Realtime Process Monitor(RPM) - T0877V01.00 - (15APR08)
+ADD \A
+ADD \B
+ADD \C
+PB  \*, ENTRIES 5, RATE 5

Process  Cpu,Pin Busy% Name     RPM      Programs  ET= 5.0 Top Pri User
-------- ------- ----- -------- -------------------------- --- --- -------
\A        3,35   94.63 $SPI1    $DATA.MMSPIN.SPIN            1   1  66,1
12:07:05  1,134    .82 $ZOOB    $SYSTEM.SYSTEM.QUERY         2 160 255,255
          1,175    .19 $RPMX    $SYSTEM.SYSTEM.SEEVIEW       3 167  66,1
          1,42     .18 $Z23V    $SYSTEM.SYSTEM.DRIVE         4 166  66,1
          1,249    .17 $Z244    $SYSTEM.SYSTEM.TEST          5 166  66,1
\B        2,271    .83 $HIT     $SYSTEM.SYS03.TSYSDP2        1 220 255,255
12:07:05  1,254    .44 $Z08N    $SYSTEM.SYSTEM.TEST          2 167  66,1
          1,153    .43 $X11Z    $SYSTEM.SYSTEM.DRIVE         3 167  66,1
          1,74     .18 $RPMX    $SYSTEM.SYSTEM.SEEVIEW       4 168  66,1
          3,320    .28 $QAZ08   $ASAP.ASAPXQA.QADRV1T        5 168 255,34
\C        0,290   3.76 $SQL     $SYSTEM.SYS00.TSYSDP2        1 220 255,255
12:07:05  1,195   2.48 $ZOOL    $SYSTEM.SYSTEM.ASAPFIL       2 160 255,255
          1,0     1.82 $MON     $SYSTEM.SYS00.OSIMAGE        3 201 255,255
          0,0     1.55 $MON     $SYSTEM.SYS00.OSIMAGE        4 201 255,255
          0,85     .07 $RPMX    $SYSTEM.SYSTEM.SEEVIEW       5 168  66,1
```

2 Installing NonStop RPM

This section provides an explanation of the steps required to install and run RPM on your system(s). Please read the other sections before fully deploying RPM in your environment.

RPM configuration and installation are greatly simplified through the use of the RPM wizard named **RPMWIZ**. Once you become a proficient RPM user, you will not need the wizard to run RPM, but you will always benefit from using the RPM Wizard to create configuration files and install RPM.

To install, configure and run RPM follow the steps shown below:

1. Install RPM files onto your NonStop server(s). This requires either loading the RPM CD, and/or copying RPM files to the system in question. Then running the RPM Wizard **RPMWIZ**. Use IP Setup to place the RPM files from the CD directly into the RPM installation subvolumes ($SYSTEM.SYSTEM and the RPM ISV) or to place the files for DSM/SCM, then use DSM/SCM to install the RPM files into the installation subvolumes if you wish to do so.

2. Volume to the RPM ISV then run the RPM wizard from TACL as shown below:

   TACL 1 > VOLUME <RPM-ISV>
   TACL 2 > **RUN RPMWIZ**

2.1 RPM Wizard - Example

Below is an example of RPMWIZ interaction for CONFIG, INSTALL, and RUN commands.

```
$DATA MYSUB 1> VOLUME $DATA.R0877V01
$DATA RPMV1 2> RUN RPMWIZ
================================================================================
== HP RPMWIZ - RPM Wizard - T0877V01.00 - (19Feb2008) http://www.NonstopRPM.com
================================================================================
==
== RPMWIZ - RPM wizard is used to configure and install RPM files.
==
== First choose CONFIG, then SAVE, and then optionally the INSTALL command:
==
== * CONFIG - Creates RPM device configuration files. When you choose CONFIG
== the RPM wizard interviews you and based on answers to questions creates
== TTY, VT100, T6530 device config files for RPM, RPMVT, and RPM65 objects.
==
== * INSTALL - First enter CONFIG and SAVE commands for devices you want to
== use with RPM. Then install RPM on $SYSTEM.SYSTEM using INSTALL command.
== Once RPM is installed you still use CONFIG and SAVE commands to adjust
== your config files, without needing to re-INSTALL RPM.
==
== Notes: 1) Default is bracketed. For example 10, 20, [100]? (100 is default)
== 2) CTL-Y escapes a prompt, and continues to next-level in RPMWIZ.
==============================================================================


CONFIG | SAVE | INSTALL | RUN | EXIT >
```

2.2 RPM Wizard - CONFIG - Example

```
CONFIG | SAVE | INSTALL | RUN | EXIT > CONFIG

-----------------------------------------------------------------------------
-- Specify RPM device TYPE ?
-----------------------------------------------------------------------------
VT100 - Builds RpmVTCNF config file used by the RPMVT (VT100) object file.
VT100 support is always present for Windows or Linux OS. For example
from MS-WINDOWS, you can run RPMVT from a standard TELNET prompt
because MS-TELNET provides native built-in support of VT100 including
color high-light escape codes. VT100 has advantages over T6530/TTY
because the VT100 display can be 100's of lines and columns long and
because VT100 colors are automatically encoded.

T6530 - Builds Rpm65CNF config file used by the RPM65 (T6530) object file.
If you want to run RPM from a T6530 emulator choose this option and
run RPM65 from TACL/OSH. T6530 emulator windows are smaller than
VT100. For example the biggest T6530 window is 54 lines x 132 chars,
where VT100 windows can be 100's of lines long x 100's of cols.

TTY - Builds RpmCNF config file used by the RPM (TTY) object file.
Use this option if you do not want any color encoding of info,
or if you want to output RPM data to a file, eg RPM / OUT file /

Enter TYPE of device VT100 | T6530 | TTY : TTY


-----------------------------------------------------------------------------
-- Specify RPM Sample RATE ?
-----------------------------------------------------------------------------

The sample RATE is in seconds. RATE determines how frequently RPM monitors
Cpus and Processes and how often it displays updates. While RPM is capable
of extremely fast sampling down to 1 second, it is often more useful to pick
an update RATE that is a little slower so that information on the screen is
less time variable. For example a value of 10 seconds is a better value.

Enter RATE interval in seconds 2, 3, 6, [10], .. : 10


-----------------------------------------------------------------------------
-- Specify RPM process ENTRIES ?
-----------------------------------------------------------------------------

The ENTRIES option indicates the maximum number of busy processes displayed
per Cpu or Node. For example ENTRIES 3 produces a report with the busiest 3
processes on each node as shown in the example output below:

Process Cpu,Pin  Busy% Name RPM Programs ET=10.0           Top Pri User
-------- ------- ----- -------- -------------------------- --- --- -------
\CENTDIV  0,331  97.94 $LOOP    /Projects/Looper/Looper      1   1  66,32
16:09:10  3,32   62.56 $SPIN    $MARS.MMSPIN.SPIN            2   1  66,1
          2,271    .76 $HIT     $SYSTEM.SYS03.TSYSDP2        3 220 255,255
\SOLAR    3,35   89.21 $SPI1    $MARS.MMSPIN.SPIN            1   1  66,1
```

```
16:09:10  1,31      .17 $ZNS1    $SYSTEM.SYS00.SCP            2 168 255,255
          1,107     .17 $ZNES    $SYSTEM.SYS00.SCP            3 168 255,255
\WINE     0,294    1.82 $SQL     $SYSTEM.SYS00.TSYSDP2        1 220 255,255
18:09:10  1,252    1.21 $ZOOL    $SYSTEM.SYSTEM.ASAPFIL       2 160 255,255
          1,0       .90 $MON     $SYSTEM.SYS00.OSIMAGE        3 201 255,255

Enter ENTRIES to display per Cpu or Node [3], 5, 10, ... : 10


------------------------------------------------------------------------------
-- Specify the REPORT that you want: PROCESS, CPU, or ZOOM report ?
------------------------------------------------------------------------------

RPM can display different real-time REPORTs.

Please choose one of the following REPORT codes: P | C | Z
P - PROCESS Report on busiest processes ByCpu or ByNode
C - CPU Only Report on Cpu stats (not often used)
Z - ZOOM Report both CPU and PROCESS stats combined

Hints about which REPORT class you might want:
VT100 users most often choose - P or Z
T6530 users most often choose - P
TTY users most often choose - P

Enter report [P] | Z : P


-------------------------------------------------------------------------------
-- Specify whether you want Process statistics displayed by CPU or NODE?
-------------------------------------------------------------------------------

RPM can sort busy process statistics by CPU or by NODE. Generally by Node is
the most useful. But in some cases sort by CPU may be useful especially with
configurations where there are a small number of nodes.

Enter whether you want to sort by CPU or [NODE] : NODE


-------------------------------------------------------------------------------
-- Specify whether you would like time-of-day in microseconds (USEC) displayed?
-------------------------------------------------------------------------------

RPM can display the time-of-day in microseconds for stats from each node.
This can be useful when analyzing overall RPM timing and/or the time-of-day
synchronization between nodes.

Enter whether you want USEC time displayed Y/[N] : N


-------------------------------------------------------------------------------
-- Specify whether you want Cpu or Process objects suppressed based on %BUSY ?
-------------------------------------------------------------------------------

RPM can filter or suppress the display of Cpu or Process objects that do not
meet a certain %BUSY threshold beyond the max ENTRIES option specified above.
This is an advanced feature and not usually recommended. The value of zero
%Busy turns off this feature and is the default recommended value below.

Specify %BUSY = 0 To NOT filter Cpu or Process objects.
Specify %BUSY > 0 To filter the Cpu or Process objects.

Enter %BUSY threshold value [0] : 0
```

```
------------------------------------------------------------------------------
-- Specify %BUSY values for Informative, Warning, and Critical COLOR alerts ?
-- Blue Yellow Red
------------------------------------------------------------------------------

RPM can color highlight CPU and PROCESS objects with %busy values over
three different threshold values. These color alert values are called:

INFO - Busy% threshold value for informative alert (BLUE on VT100 devices)
1% is the default value for this alert.

WARN - Busy% threshold value for warning alert (YELLOW on VT100 devices)
10% is the default value for this alert.

CRIT - Busy% threshold value for critical alert (RED on VT100 devices)
50% is the default value for this alert.

Enter INFO Busy% threshold value for INFO [1]  : 1
Enter WARN Busy% threshold value for WARN [10] : 10
Enter CRIT Busy% threshold value for CRIT [50] : 50

------------------------------------------------------------------------------
-- Specify NODES to monitor ?
------------------------------------------------------------------------------

RPM can monitor a single CPU, all CPUs in a node, or an entire list of nodes.
Please specify "E" to ENTER nodes, or "U" to USE nodes in existing RPMCNF file.
E - ENTER a list of node names you want to monitor by typing them in, or
U - USE the list of nodes in RPMCNF file on $SYSTEM.SYSTEM or default subvol.

Please specify "E" for ENTER, or "U" for USE [E]/U ? E


Enter nodes you want to monitor one at a time,
once you have ADD-ed all nodes to monitor,
you then terminate ADD mode by entering "//" or <null>

ADD \sysname ? \CENTDIV
ADD \sysname ? \SOLAR
ADD \sysname ? \WINE
ADD \sysname ?

!======================================================
!== RPM Config created by RPMWIZ - 08/02/22 07:20:08
!======================================================
SET TERM VT100
SET RATE 10
SET ENTRIES 10
SET SORT BYNODE
SET USEC OFF
SET CRIT 50
SET WARN 10
SET INFO 1
ADD \CENTDIV
ADD \SOLAR
ADD \WINE
!======================================================
SAVE to $DATA.RPMV1.RPMCNF Y/N ? Y
$DATA.RPMV1.RPMCNF save started.
$DATA.RPMV1.RPMCNF saved.
```

2.3 RPM Wizard - INSTALL -

```
CONFIG | SAVE | INSTALL | RUN | EXIT > INSTALL

Confirm RPM Source Distribution Volume (DSV) = $DATA.RPMV1 Y/N ? Y
======================================================================
RPM INSTALL started: 08/02/22 07:41:28
======================================================================


07:41:28 -------------> INSTALL \CENTDIV <-------------
FUP DUP $DATA.RPMV1.RPM , \CENTDIV.$SYSTEM.SYSTEM.RPM , purge,sourcedate
FUP DUP $DATA.RPMV1.RPM , \CENTDIV.$SYSTEM.SYSTEM.RPMVT , purge,sourcedate
FUP DUP $DATA.RPMV1.RPM , \CENTDIV.$SYSTEM.SYSTEM.RPM65 , purge,sourcedate
FUP DUP $DATA.RPMV1.RPMWIZ , \CENTDIV.$SYSTEM.SYSTEM.* , purge,sourcedate
FUP DUP $DATA.RPMV1.RPMWIZEE , \CENTDIV.$SYSTEM.SYSTEM.* , purge,sourcedate
\CENTDIV.$SYSTEM.SYSTEM.RPMCNF exists. Do you want to overlay *CNF files Y/N ? N
FUP SECURE \CENTDIV.$SYSTEM.SYSTEM.RPM ,nnnn
FUP SECURE \CENTDIV.$SYSTEM.SYSTEM.RPMVT ,nnnn
FUP SECURE \CENTDIV.$SYSTEM.SYSTEM.RPM65 ,nnnn
FUP SECURE \CENTDIV.$SYSTEM.SYSTEM.RPMWIZ ,nnnn
FUP SECURE \CENTDIV.$SYSTEM.SYSTEM.RPMWIZEE ,nnnn
FUP SECURE \CENTDIV.$SYSTEM.SYSTEM.RPMCNF ,nnnn
FUP SECURE \CENTDIV.$SYSTEM.SYSTEM.RPMVTCNF ,nnnn
FUP SECURE \CENTDIV.$SYSTEM.SYSTEM.RPM65CNF ,nnnn


07:41:39 -------------> INSTALL \SOLAR <-------------
FUP DUP $DATA.RPMV1.RPM , \SOLAR.$SYSTEM.SYSTEM.RPM , purge,sourcedate
FUP DUP $DATA.RPMV1.RPM , \SOLAR.$SYSTEM.SYSTEM.RPMVT , purge,sourcedate
FUP DUP $DATA.RPMV1.RPM , \SOLAR.$SYSTEM.SYSTEM.RPM65 , purge,sourcedate
FUP DUP $DATA.RPMV1.RPMWIZ , \SOLAR.$SYSTEM.SYSTEM.* , purge,sourcedate
FUP DUP $DATA.RPMV1.RPMWIZEE , \SOLAR.$SYSTEM.SYSTEM.* , purge,sourcedate
FUP SECURE \SOLAR.$SYSTEM.SYSTEM.RPM ,nnnn
FUP SECURE \SOLAR.$SYSTEM.SYSTEM.RPMVT ,nnnn
FUP SECURE \SOLAR.$SYSTEM.SYSTEM.RPM65 ,nnnn
FUP SECURE \SOLAR.$SYSTEM.SYSTEM.RPMWIZ ,nnnn
FUP SECURE \SOLAR.$SYSTEM.SYSTEM.RPMWIZEE ,nnnn
FUP SECURE \SOLAR.$SYSTEM.SYSTEM.RPMCNF ,nnnn
FUP SECURE \SOLAR.$SYSTEM.SYSTEM.RPMVTCNF ,nnnn
FUP SECURE \SOLAR.$SYSTEM.SYSTEM.RPM65CNF ,nnnn


07:41:51 -------------> INSTALL \WINE <-------------
FUP DUP $DATA.RPMV1.RPM , \WINE.$SYSTEM.SYSTEM.RPM , purge,sourcedate
FUP DUP $DATA.RPMV1.RPM , \WINE.$SYSTEM.SYSTEM.RPMVT , purge,sourcedate
FUP DUP $DATA.RPMV1.RPM , \WINE.$SYSTEM.SYSTEM.RPM65 , purge,sourcedate
FUP DUP $DATA.RPMV1.RPMWIZ , \WINE.$SYSTEM.SYSTEM.* , purge,sourcedate
FUP DUP $DATA.RPMV1.RPMWIZEE , \WINE.$SYSTEM.SYSTEM.* , purge,sourcedate
FUP SECURE \WINE.$SYSTEM.SYSTEM.RPM ,nnnn
FUP SECURE \WINE.$SYSTEM.SYSTEM.RPMVT ,nnnn
FUP SECURE \WINE.$SYSTEM.SYSTEM.RPM65 ,nnnn
FUP SECURE \WINE.$SYSTEM.SYSTEM.RPMWIZ ,nnnn
FUP SECURE \WINE.$SYSTEM.SYSTEM.RPMWIZEE ,nnnn
FUP SECURE \WINE.$SYSTEM.SYSTEM.RPMCNF ,nnnn
FUP SECURE \WINE.$SYSTEM.SYSTEM.RPMVTCNF ,nnnn
FUP SECURE \WINE.$SYSTEM.SYSTEM.RPM65CNF ,nnnn
```

2.4 RPM Wizard - RUN - Example

```
CONFIG | SAVE | INSTALL | RUN | EXIT > RUN

RUN $SYSTEM.SYSTEM.RPM
Realtime Process Monitor(RPM) - T0877V01.00 - (20FEB08)
Evaluation expires 2008/04/01 - comments - support@NonstopRPM.com
OBEY DATA.USER.RPMCNF
!======================================================
!== RPM Config created by RPMWIZ - 08/02/22 07:24:57
!======================================================
SET TERM TTY
SET RATE 10
SET ENTRIES 10
SET SORT BYNODE
SET USEC OFF
SET CRIT 50
SET WARN 10
SET INFO 1
ADD \CENTDIV
\CENTDIV.$system.system.RPM Version: 2008/02/20 11:36
ADD \SOLAR
\SOLAR.$system.system.RPM Version: 2008/02/20 11:36
ADD \WINE
\WINE.$system.system.RPM Version: 2008/02/20 11:36
!======================================================
+P\*
Process  Cpu,Pin Busy% Name RPM Programs ET=14.5           Top Pri User
-------- ------- ----- -------- -------------------------- --- --- -------
\CENTDIV 3,32     6.86 $SPIN    $MARS.MMSPIN.SPIN            1   1  66,1
07:26:40 1,250     .12 $X4AG    $SYSTEM.SYSTEM.RPM           2 166  66,1
         1,14      .05 $NCP     $SYSTEM.SYS03.NCPOBJ         3 199 255,255
         0,15      .04 $NCP     $SYSTEM.SYS03.NCPOBJ         4 199 255,255
         1,252     .04 $RPMX    $SYSTEM.SYSTEM.SEEVIEW       5 167  66,1
         1,293     .04 $ZTC04   $SYSTEM.SYS03.TCPIP          6 200 255,255
         0,12      .02 $TMP     $SYSTEM.SYS03.TMFTMP         7 204 255,255
         0,219     .02 $V03M0   $VENUS.SASAP.ASAPMONR        8 160  66,50
         0,257     .02 $SYSTEM  $SYSTEM.SYS03.OSIMAGE        9 220 255,255
         1,0       .02 $MON     $SYSTEM.SYS03.OSIMAGE       10 201 255,255
\SOLAR   3,35    38.49 $SPI1    $MARS.MMSPIN.SPIN            1   1  66,1
07:26:40 1,282     .09 $ZSD01   $SYSTEM.SYS00.NSADPR         2 199 255,255
         0,282     .07 $ZSD00   $SYSTEM.SYS00.NSADPR         3 199 255,255
         3,281     .07 $ZSD03   $SYSTEM.SYS00.NSADPR         4 199 255,255
         1,135     .06 $Z0R2    $SYSTEM.SYSTEM.RPM           5 166  66,1
         1,175     .03 $RPMX    $SYSTEM.SYSTEM.SEEVIEW       6 167  66,1
         0,5       .02 $YMIOP   $SYSTEM.SYS00.TMIOP          7 205 255,255
         0,271     .02          $SYSTEM.SYS00.NTIMEIP        8 255 255,255
         0,0       .01 $MON     $SYSTEM.SYS00.NMONTOR        9 201 255,255
         0,265     .01          $SYSTEM.SYS00.TSMSGIP       10 255 255,255
\WINE    0,12      .31 $TMP     $SYSTEM.SYS00.TMFTMP         1 204 255,255
09:26:40 0,240     .29 $Z0RD    $SYSTEM.SYSTEM.RPM           2 166  66,1
         0,242     .17 $RPMX    $SYSTEM.SYSTEM.SEEVIEW       3 167  66,1
         0,0       .08 $MON     $SYSTEM.SYS00.OSIMAGE        4 201 255,255
         1,0       .08 $MON     $SYSTEM.SYS00.OSIMAGE        5 201 255,255
         0,313     .05 $SQL     $SYSTEM.SYS00.TSYSDP2        6 220 255,255
         0,327     .02 $ZTC0    $SYSTEM.SYS00.TCPIP          7 200 255,255
         0,343     .02 $ZTC04   $SYSTEM.SYS00.TCPIP          8 200 255,255
         0,348      .02 $ZTSM    $SYSTEM.SYS00.SRM            9 150 255,255
         1,273      .02 $SQL     $SYSTEM.SYS00.TSYSDP2       10 220 255,255
```

```
+EXIT
```

CONFIG | SAVE | INSTALL | RUN | EXIT >

NOTE - If you RUN RPM via the RPM Wizard, then after you EXIT RPM, and you are still in RPMWIZ, you can then repeatedly enter **CONFIG**, **SAVE**, or **RUN** again to alter your *CNF file(s) and experiment with RPM configurations and settings.

CONFIG | SAVE | INSTALL | RUN | EXIT >

3 Configuring RPM

When any RPM program object is executed (RPM, RPMVT, RPM65, or RPMXX) the RPM program searches for a configuration file whose name is the concatenation of the RPM object file name and the suffix string "CNF". For example, if you run RPM, the program object searches for a config file named RPMCNF.

The search first occurs in the default subvolume, and if the *CNF file is not found in the default subvolume, then the $SYSTEM.SYSTEM subvolume is searched.

RPM object files MUST ALWAYS BE installed on the $SYSTEM.SYSTEM subvolume in order to perform peer-to-peer messaging using the SSG. Given the above name search rules the following describes the location of RPM configuration files for each RPM object file.

RPM - **TTY** config info is stored in a file named $System.System.**RPMCNF**.

RPM65 - **T6530** config info is stored in the a file named $System.System.**RPM65CNF**.

RPMVT - **ANSI/VT100** config info stored in a file named $System.System.**RPMVTCNF**.

3.1 RPM Config file locations

RPM Object	***Device***	***Default Configuration file location***
RPM	TTY	$System.System.**RPMCNF**
RPM65	T6530	$System.System.**RPM65CNF**
RPMVT	VT100	$System.System.**RPMVTCNF**

3.2 RPM Config file contents

RPM configuration files are edit files containing any valid RPM command(s) documented in section 5 "RPM Command Interface". Generally in order to simplify creating config files and in order to avoid incompatible configuration settings, you should use the RPM Wizard to create your config files. Once you become more knowledable about RPM configuration files you can edit your RPM config files manually.

3.3 Config file examples

The configuration files below provide examples of different types of configurations that can be generated using the RPM Wizard.

The following documents configuration file contents, this is primarily for instructive purposes only, since generally you should **use RPM Wizard RPMWIZ to create your configuration files**.

To run the RPM configuration wizard RPMWIZ enter the following command.

```
TACL 1 > RUN RPMWIZ
```

RPMCNF - TTY Configuration file Example #1

The example config file below adds 3 nodes \node1, 2, 3; sets the terminal type to TTY, indicates the 5 busiest processes should be displayed for each node, sets sort to be by node, sets critical, warning, and information thresholds, and sets the rate to 10 seconds.

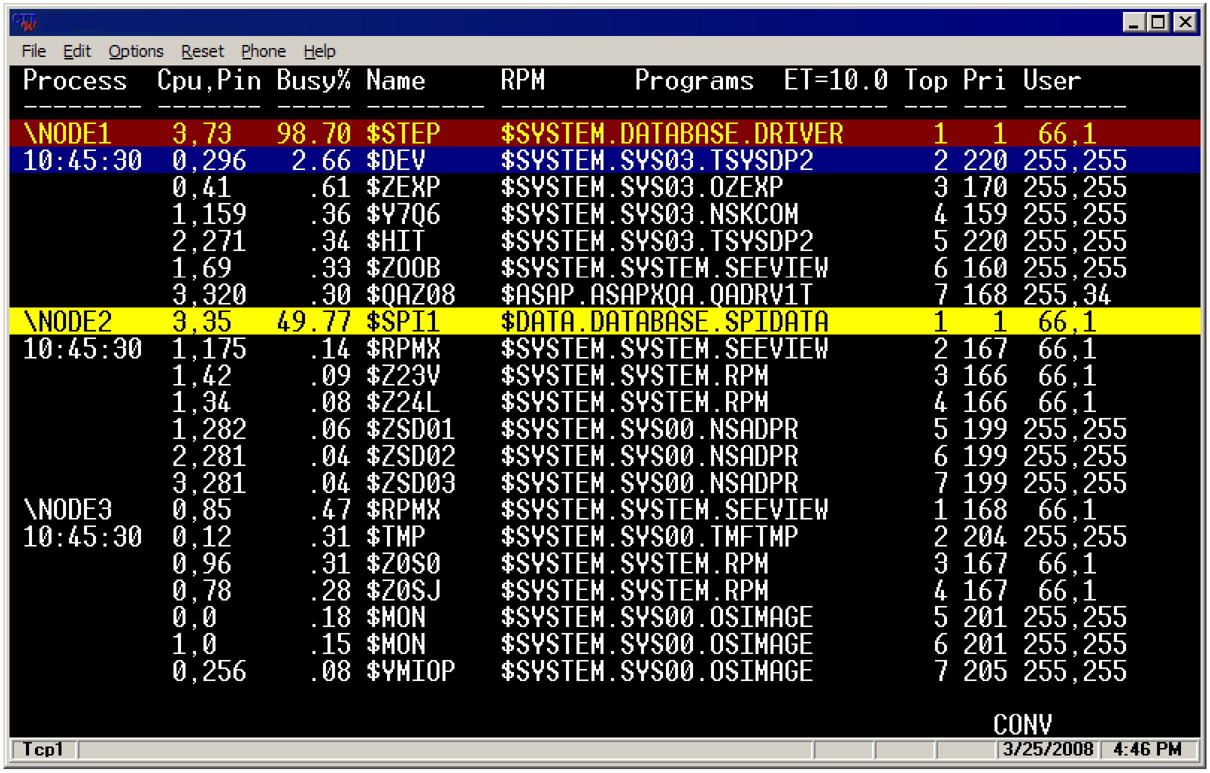
```
!===========================================================
!== RPM TTY Configuration settings - 08/04/15 10:00:00
!===========================================================
ADD \node1      ! add \node1 to list of nodes
ADD \node2      ! add \node2 to list of nodes
ADD \node3      ! add \node3 to list of nodes
SET TERM TTY    ! define default term type, VT100, T6530, TTY
SET ENTRIES 5   ! show top 5 busy processes on each cpu/node
SET SORT BYNODE ! sort across all cpus in each node
SET CRIT 50     ! set Critical alert busy threshold 50%
SET WARN 10     ! set Warning  alert busy threshold 10%
SET INFO  1     ! set Info     alert busy threshold  1%
SET RATE 10     ! set refresh rate in seconds
```

SET RATE <seconds> defines the interval between RPM samples. Although short sample intervals such as 1 second update the screen frequently and are supported by RPM, a one second update is not necessarily the best setting. Too frequent updates can be disorientating, and in particular a 10 second sample interval has some special advantages. See SET RATE for more info.

RPMCNF - T6530 Configuration file Example #2

The example below **ADD**s 3 nodes \node1, 2, 3; sets the **TERM** type to **T6530**, indicates the 7 busiest processes should be displayed for each node (**ENTRIES 7**), sets **SORT** to be **BYNODE**, sets critical, warning, and information thresholds, sets the **RATE** to 10 seconds, and executes a
PB * command which displays a Process Busy report that updates every 10 seconds.

```
!===========================================================
!== RPM T6530 Configuration settings  08/04/15 10:00:00
!===========================================================
ADD \node1       ! add \node1 to list of nodes
ADD \node2       ! add \node2 to list of nodes
ADD \node3       ! add \node3 to list of nodes
SET TERM T6530   ! define default term type, VT100, T6530, TTY
SET ENTRIES 7    ! show top 7 busy processes on each cpu/node
SET SORT BYNODE  ! sort across all cpus in each node
SET CRIT 50      ! set Critical alert busy threshold 50%
SET WARN 10      ! set Warning  alert busy threshold 10%
SET INFO  1      ! set Info     alert busy threshold  1%
SET RATE 10      ! set refresh rate in seconds
PB \*            ! show Busy Processes every 10 sec
```

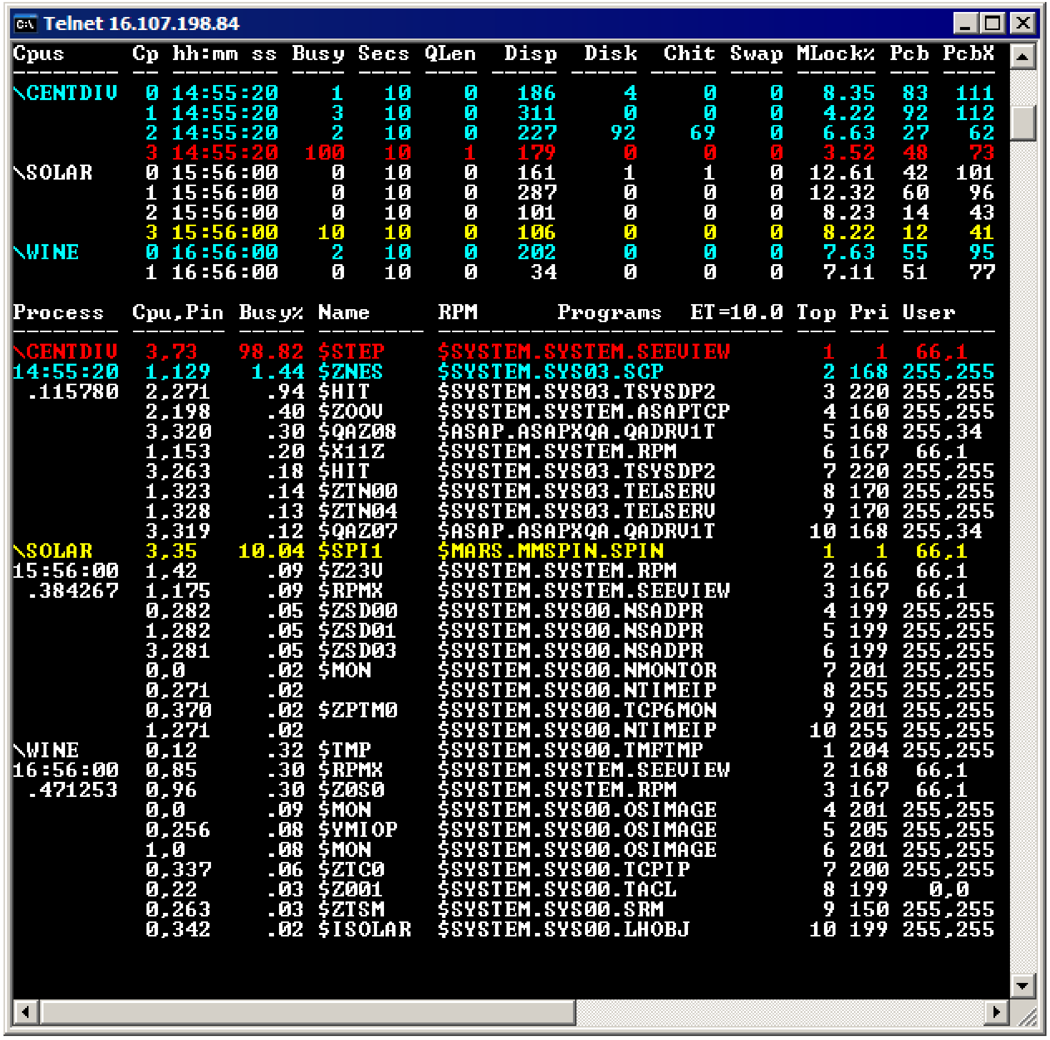


```
Process  Cpu,Pin Busy% Name     RPM       Programs  ET=10.0 Top Pri User
-------- ------- ----- -------- ------------------------ --- --- -------

\NODE1    3,73   98.70 $STEP    $SYSTEM.DATABASE.DRIVER    1   1  66,1
10:45:30  0,296   2.66 $DEV     $SYSTEM.SYS03.TSYSDP2      2 220 255,255
          0,41     .61 $ZEXP    $SYSTEM.SYS03.OZEXP        3 170 255,255
          1,159    .36 $Y7Q6    $SYSTEM.SYS03.NSKCOM       4 159 255,255
          2,271    .34 $HIT     $SYSTEM.SYS03.TSYSDP2      5 220 255,255
          1,69     .33 $Z00B    $SYSTEM.SYSTEM.SEEVIEW     6 160 255,255
          3,320    .30 $QAZ08   $ASAP.ASAPXQA.QADRV1T      7 168 255,34
\NODE2    3,35   49.77 $SPI1    $DATA.DATABASE.SPIDATA     1   1  66,1
10:45:30  1,175    .14 $RPMX    $SYSTEM.SYSTEM.SEEVIEW     2 167  66,1
          1,42     .09 $Z23V    $SYSTEM.SYSTEM.RPM         3 166  66,1
          1,34     .08 $Z24L    $SYSTEM.SYSTEM.RPM         4 166  66,1
          1,282    .06 $ZSD01   $SYSTEM.SYS00.NSADPR       5 199 255,255
          2,281    .04 $ZSD02   $SYSTEM.SYS00.NSADPR       6 199 255,255
          3,281    .04 $ZSD03   $SYSTEM.SYS00.NSADPR       7 199 255,255
\NODE3    0,85     .47 $RPMX    $SYSTEM.SYSTEM.SEEVIEW     1 168  66,1
10:45:30  0,12     .31 $TMP     $SYSTEM.SYS00.TMFTMP       2 204 255,255
          0,96     .31 $Z0S0    $SYSTEM.SYSTEM.RPM         3 167  66,1
          0,78     .28 $Z0SJ    $SYSTEM.SYSTEM.RPM         4 167  66,1
          0,0      .18 $MON     $SYSTEM.SYS00.OSIMAGE      5 201 255,255
          1,0      .15 $MON     $SYSTEM.SYS00.OSIMAGE      6 201 255,255
          0,256    .08 $YMIOP   $SYSTEM.SYS00.OSIMAGE      7 205 255,255

                                                             CONV
```

RPMCNF - VT100 Configuration file Example #3

The example below **ADD**s 3 nodes \node1, 2, 3; sets the **TERM** type to **VT100** , indicates the 10 busiest processes should be displayed for each node (**ENTRIES 10**), sets **SORT** to be **BYNODE**, sets critical, warning, and information thresholds, sets the **RATE** to 10 seconds, and executes a **ZOOM \*** command which displays a combined Cpu and PB report that updates every 10 seconds.

```
!============================================================
!== RPM VT100 Configuration settings - 08/04/15 10:00:00
!============================================================
ADD \CENTDIV     ! add \node1 to list of nodes
ADD \SOLAR       ! add \node2 to list of nodes
ADD \CENTDIV     ! add \node3 to list of nodes
SET TERM VT100   ! define default term type, VT100, T6530, TTY
SET ENTRIES 10   ! show top 10 busy processes on each cpu/node
SET SORT BYNODE  ! sort across all cpus in each node
SET CRIT 50      ! set Critical alert busy threshold 50%
SET WARN 10      ! set Warning  alert busy threshold 10%
SET INFO  1      ! set Info     alert busy threshold  1%
SET RATE 10      ! set refresh rate in seconds
ZOOM \*          ! show both Cpus and Processes every 10 sec
```



```
Cpus     Cp hh:mm ss Busy Secs QLen  Disp  Disk  Chit Swap MLock% Pcb PcbX
-------- -- -------- ---- ---- ---- ----- ----- ----- ---- ------ --- ----
\CENTDIV  0 14:55:20    1   10    0   186     4     0    0   8.35  83  111
          1 14:55:20    3   10    0   311     0     0    0   4.22  92  112
          2 14:55:20    2   10    0   227    92    69    0   6.63  27   62
          3 14:55:20  100   10    1   179     0     0    0   3.52  48   73
\SOLAR    0 15:56:00    0   10    0   161     1     1    0  12.61  42  101
          1 15:56:00    0   10    0   287     0     0    0  12.32  60   96
          2 15:56:00    0   10    0   101     0     0    0   8.23  14   43
          3 15:56:00   10   10    0   106     0     0    0   8.22  12   41
\WINE     0 16:56:00    2   10    0   202     0     0    0   7.63  55   95
          1 16:56:00    0   10    0    34     0     0    0   7.11  51   77

Process  Cpu,Pin Busy% Name     RPM      Programs  ET=10.0 Top Pri User
-------- ------- ----- -------- -------------------------- --- --- -------
\CENTDIV  3,73   98.82 $STEP    $SYSTEM.SYSTEM.SEEVIEW       1   1  66,1
14:55:20  1,129   1.44 $ZNES    $SYSTEM.SYS03.SCP            2 168 255,255
 .115780  2,271    .94 $HIT     $SYSTEM.SYS03.TSYSDP2        3 220 255,255
          2,198    .40 $ZOOV    $SYSTEM.SYSTEM.ASAPTCP       4 160 255,255
          3,320    .30 $QAZ08   $ASAP.ASAPXQA.QADRV1T        5 168 255,34
          1,153    .20 $X11Z    $SYSTEM.SYSTEM.RPM           6 167  66,1
          3,263    .18 $HIT     $SYSTEM.SYS03.TSYSDP2        7 220 255,255
          1,323    .14 $ZTN00   $SYSTEM.SYS03.TELSERV        8 170 255,255
          1,328    .13 $ZTN04   $SYSTEM.SYS03.TELSERV        9 170 255,255
          3,319    .12 $QAZ07   $ASAP.ASAPXQA.QADRV1T       10 168 255,34
\SOLAR    3,35   10.04 $SPI1    $MARS.MMSPIN.SPIN            1   1  66,1
15:56:00  1,42     .09 $Z23V    $SYSTEM.SYSTEM.RPM           2 166  66,1
 .384267  1,175    .09 $RPMX    $SYSTEM.SYSTEM.SEEVIEW       3 167  66,1
          0,282    .05 $ZSD00   $SYSTEM.SYS00.NSADPR         4 199 255,255
          1,282    .05 $ZSD01   $SYSTEM.SYS00.NSADPR         5 199 255,255
          3,281    .05 $ZSD03   $SYSTEM.SYS00.NSADPR         6 199 255,255
          0,0      .02 $MON     $SYSTEM.SYS00.NMONTOR        7 201 255,255
          0,271    .02          $SYSTEM.SYS00.NTIMEIP        8 255 255,255
          0,370    .02 $ZPTM0   $SYSTEM.SYS00.TCP6MON        9 201 255,255
          1,271    .02          $SYSTEM.SYS00.NTIMEIP       10 255 255,255
\WINE     0,12     .32 $TMP     $SYSTEM.SYS00.TMFTMP         1 204 255,255
16:56:00  0,85     .30 $RPMX    $SYSTEM.SYSTEM.SEEVIEW       2 168  66,1
 .471253  0,96     .30 $Z0S0    $SYSTEM.SYSTEM.RPM           3 167  66,1
          0,0      .09 $MON     $SYSTEM.SYS00.OSIMAGE        4 201 255,255
          0,256    .08 $YMIOP   $SYSTEM.SYS00.OSIMAGE        5 205 255,255
          1,0      .08 $MON     $SYSTEM.SYS00.OSIMAGE        6 201 255,255
          0,337    .06 $ZTC0    $SYSTEM.SYS00.TCPIP          7 200 255,255
          0,22     .03 $Z001    $SYSTEM.SYS00.TACL           8 199   0,0
          0,263    .03 $ZTSM    $SYSTEM.SYS00.SRM            9 150 255,255
          0,342    .02 $ISOLAR  $SYSTEM.SYS00.LHOBJ         10 199 255,255
```

4 Running RPM

This section describes how to run RPM on supported device types – TTY, T6530, VT100.

Since there are generally multiple mechanisms and multiple config files for starting and configuring RPM on any given system, it is important to have a basic understanding of how each mechanism works. This section provides examples of how different RPM object files and configuration files interact.

4.1 Starting RPM in TTY mode

To run RPM in TTY mode enter **RPM** from a TACL prompt.

```
TACL 1 > RPM
Realtime Process Monitor(RPM) - T0877V01.00 - (15APR08)
OBEY $SYSTEM.SYSTEM.RPMCNF
!============================================================
!== RPM Configuration settings - 08/04/15 10:00:00
!============================================================
ADD \node1       ! add \node1 to list of nodes
ADD \node2       ! add \node2 to list of nodes
ADD \node3       ! add \node3 to list of nodes
SET TERM TTY     ! define default term type, VT100, T6530, TTY
SET ENTRIES 5    ! show top 5 busy processes on each cpu/node
SET SORT BYNODE  ! sort across all cpus in each node
SET CRIT 50      ! set Critical alert busy threshold 50%
SET WARN 10      ! set Warning  alert busy threshold 10%
SET INFO  1      ! set Info     alert busy threshold  1%
SET RATE 10      ! set refresh rate in seconds
+
```

4.2 Starting RPM in T6530 mode

To run RPM in T6530 mode enter **RPM65** from a TACL prompt.

```
TACL 1 > RPM65
Realtime Process Monitor(RPM) - T0877V01.00 - (15APR08)
OBEY $SYSTEM.SYSTEM.RPM65CNF
!============================================================
!== RPM Configuration settings - 08/04/15 10:00:00
!============================================================
ADD \node1       ! add \node1 to list of nodes
ADD \node2       ! add \node2 to list of nodes
ADD \node3       ! add \node3 to list of nodes
SET TERM T6530   ! define default term type, VT100, T6530, TTY
SET ENTRIES 7    ! show top 7 busy processes on each cpu/node
SET SORT BYNODE ! sort across all cpus in each node
SET CRIT 50      ! set Critical alert busy threshold 50%
SET WARN 10      ! set Warning  alert busy threshold 10%
SET INFO  1      ! set Info     alert busy threshold  1%
SET RATE 10      ! set refresh rate in seconds
ZOOM \*          ! show both Cpus and Processes every 10 sec
```

4.3 Starting RPM in VT100 mode

To run RPM in ANSI or VT100 mode enter **RPMVT** from a TACL prompt.

```
TACL 1 > RPMVT
Realtime Process Monitor(RPM) - T0877V01.00 - (15APR08)
OBEY $SYSTEM.SYSTEM.RPMVTCNF
!=====================================================
!== RPM Configuration settings - 08/04/15 10:00:00
!=====================================================
ADD \node1       ! add \node1 to list of nodes
ADD \node2       ! add \node2 to list of nodes
ADD \node3       ! add \node3 to list of nodes
ADD \node4       ! add \node4 to list of nodes
ADD \node5       ! add \node5 to list of nodes
SET TERM VT100   ! define default term type, VT100, T6530, TTY
SET ENTRIES 10   ! show top 10 busy processes on each cpu/node
SET SORT BYNODE ! sort across all cpus in each node
SET CRIT 50      ! set Critical alert busy threshold 50%
SET WARN 10      ! set Warning  alert busy threshold 10%
SET INFO  1      ! set Info     alert busy threshold  1%
SET RATE 10      ! set refresh rate in seconds
ZOOM \*          ! show both Cpus and Processes every 10 sec
```

5 RPM Commands

RPM includes a command interpreter (CI) that can communicate with either a local copy of RPM or with any number of remote node copies of RPM.

5.1 Overview

Enter **HELP** from any **RPM** prompt to display the following summary of commands...

```
REALTIME PROCESS MONITOR (RPM) - T0877V01.00 - (15APR08)

         ----- Monitoring commands -----
CPU      Display realtime CPU statistics.    Enter HELP CPU for more info
PB       Display realtime ProcessBusy stats. Enter HELP PB  for more info
ZOOM     Displays blended CPU and PB stats, enter HELP ZOOM
         ----- Supporting commands -----
ADD      Add \<node> to list of nodes for cluster analysis
EXIT     causes program to terminate
FC       standard tandem fix command
HELP     provides description of commands
NODES    Shows nodes that have been added via the ADD command
OBEY     causes commands to be executed in an OBEY file
PAUSE    suspend until stop/abend/brk msg
RUN      runs specified program
SET      Sets/Shows option settings. Enter HELP SET for more
STATUS   displays status of SSG or process by $pid
SYSTEM   allows setting/display of system
T6530    Sets T6530 terminal support
VOLUME   allows setting/display of default volume
VT100    Sets VT100 terminal support
```

RPM commands are divided into two groups: those commands that are monitoring commands such as the **CPU**, **PB**, and **ZOOM** commands, and those commands that configure or status the environment, such as the ADD, NODES, and SET commands.

RPM Commands and abbreviations

Abbreviation	***Command***
A or ADD	ADD \<node> to list of nodes displayed whenever * is encountered
C or CPU	Displays real-time Cpu statistics, enter HELP CPU for more info
H or HELP	List commands, or if HELP <command> show command detail
P or PB	Displays real-time Process Busy stats, enter HELP PB for more info
S or SET	Sets/shows configuration settings
ST or STATUS	Displays status of all SSG's associated with this SET ID $pid
T6 or T6530	Same as SET TERMTYPE T6530
V or VT100	Same as SET TERMTYPE VT100
Z or ZOOM	Shows a combined continuously updating display of both Cpu and PB

5.2 ADD Command

```
ADD  \<node-name>
```

The **A** or **ADD** \<node-name> command adds a node to the list of RPM nodes to be monitored. Once nodes are added, whenever you use commands that support the * syntax, statistics for the added nodes are displayed. Commands that support the * construct include the CPU *, PB *, or ZOOM * commands.

For more information about commands that provide the * construct, enter:

HELP CPU
HELP PB
HELP ZOOM

EXAMPLE

```
ADD \CHICAGO
ADD \NEWYORK
ADD \SANFRAN
ADD \DALLAS
ADD \DENVER
```

```
SET ENTRIES 3
SET RATE 6
P \*
```

5.3 CPU Command

```
CPU | C  [ \* | \SYSNAME ]  [ BUSY | %     <value>   ]
                            [ DETAIL | NORMAL         ]
                            [ LAST                    ]
                            [ MEMORY | MB | PCT       ]
                            [ NOCLEAR                 ]
                            [ RATE <seconds>          ]
                            [ TAB                     ]
                            [ VT100 | T6530 | TTY     ]
```

The **C or CPU** command displays real-time CPU statistics for one or nodes.

BUSY | % <value> specifies the Cpu Busy threshold required for a Cpu to be displayed. Cpus busy must be greater than or equal to the <value> specified in order to be displayed. Cpus with a busy value less than <value> are not displayed. Default is 0, so that all Cpus are displayed by default. Using a value other than zero is considered an advanced feature and is not recommended for new or beginning users.

DETAIL | NORMAL controls how much detail is displayed. Normal is the default.

LAST causes stats to be displayed based on the requestors rate. The LAST option is an adaptive rate, if a requestor makes a request every 3 seconds, but then starts making requests every 5 seconds, the CPU command with the LAST option automatically adapts to the requestors request rate. LAST means use the stats counters from the LAST request with new stats. Display occurs once, with calculations based on time between commands.

MEMORY shows page-size, total memory, swappable, locked, and free memory in either pages or, if **MB** was specified in the command, then in units of megabytes. If **PCT** is specified, then in percent total memory.

RATE <seconds> causes stats display to repeat every <seconds>. If RATE is zero, the display is updated once, with rates and busy calculations based on 1 second sample interval unless LAST is specified. Note the default value of RATE is controlled by SET

RATE <seconds>.

TAB outputs "09" tab characters between output columns.

VT100 | T6530 | TTY - sets terminal type. Note the SET CRIT, WARN, INFO controls thresholds and the display of color-coded alerts. See HELP SET for more info.

EXAMPLES

```
CPU \*     ! show all Cpus in super-cluster
C\*        ! same as CPU \*
C RATE 6   ! show Cpu stats, repeat every 6 seconds
C MEM      ! show Cpu Memory stats in pages
C MB       ! show Cpu Memory stats in megabytes
C PCT      ! show Cpu Memory stats utilization
C\* %1     ! show Cpus greater than or equal to 1% busy
```

5.4 PB Command

```
PB  | P   [ \* | * | <cpu> ] [ BUSY | %   <value>  ]
                             [ BYCPU | BYNODE      ]
                             [ DETAIL              ]
                             [ ENTRIES <N>         ]
                             [ LAST                ]
                             [ NONULL              ]
                             [ NORMAL              ]
                             [ RATE <seconds>      ]
                             [ SAME                ]
                             [ SYNC                ]
                             [ TAB                 ]
                             [ USECS               ]
                             [ VT100 | T6530 | TTY ]
```

The **P** or **PB** command displays real-time process busy statistics. The <N> busiest processes in each Cpu, node, or cluster of nodes are displayed. For example:

```
PB      displays N busiest processes on node
P   *   displays N busiest processes on node (same as PB)
P <cpu> displays N busiest processes for <cpu>
P  \*   displays N busiest processes in \nodes (see ADD command)
```

PB command options include:

BUSY | % <value> specifies the Process Busy threshold required for a process to be displayed. Process busy must be greater than or equal to the <value> specified in order to be displayed. Processes with a busy value less than <value> are not displayed. Default value is 0, so all processes up to ENTRIES <N> count are displayed.

BYCPU | BYNODE - controls display order of the top <N> busiest processes.

> **BYCPU** displays the busiest processes in each Cpu.
>
> **BYNODE** displays the busiest processes across all Cpus in each node in one list of processes sorted from busiest to least busy process. If you do not specify this option, the busiest processes are listed ByNode, unless you specify SET SORT ByCPU.

NOTE users can globally set sort order by using the SET SORT option.

DETAIL - displays additional stats such as node name, priority, accessorid, receive queue length, and memory pages in use.

ENTRIES <N> - displays the <N> busiest processes either in all Cpus in a node, or the <N> busiest in each Cpu. Note ENTRIES can be abbreviated as E, and no space is required, thus **`P\*,E3,R6`** is valid. Note the default value for ENTRIES is controlled with the SET ENTRIES <N> option.

LAST - causes stats to be displayed based on the requestors rate. The LAST option is an adaptive rate. For example, if a requestor makes a request every 3 seconds, but then starts making requests every 5 seconds, the PB command with a LAST option automatically adapts to the requestors request rate. LAST means RPM should use the stats counters from the LAST request for the new display. Display occurs once, with calculations based on elapsed time between PB commands. See also SAME and RATE.

NONULL - suppresses display of processes consuming less than %0.01 Cpu busy.

NORMAL - displays the default output: Time, Cpu, Pin, Busy, Name, Program. Note the DETAIL option provides additional information.

RATE <seconds> - causes stats display to repeat every <seconds>. If RATE is zero, the display is updated once, with rates and busy calculations based on 1 second sample interval, unless LAST is specified. Note the default value of RATE is controlled by the SET RATE <seconds> option. Note RATE can be abbreviated and without spaces, thus **`P\*R5`** is allowed.

SAME - displays the same stats as the prior PB command, but for different <cpus> or with different DETAIL.

SAME examples:

`PB 1,LAST` could be followed by
`PB,SAME,ALL` or by
`PB 1,SAME,DETAIL` to display additional info about the same set of statistics.

SYNC - synchronizes reporting to begin at modulo seconds past the minute. For example, PB RATE 6 would report at 6, 12, 18, 24, 30, 36, 42, 48, and 54 seconds past the minute. SYNC is the default. This means that multiple RPM users will see the same percent busy since start/stop sample times will be synchronized across different copies of RPM.

TAB - outputs '09' tab control characters between output columns.

USECS - show time of day in microseconds in the NORMAL display only.

VT100 | T6530 | TTY - sets terminal type.   See HELP SET TERM for more info.

EXAMPLES

```
PB *, ENTRIES 10, RATE 6, NONULL, DETAIL, BYCPU
P *, E10, R6, NON, DET, BYC ! save as above, but abbreviated
P BYNODE, ENTRIES 22, LAST, DETAIL, TAB, RATE 10
P T6530
P VT100
P \*, E5, BYCPU ! show 5 busiest in each Cpu in all ADD nodes
P \* %1         ! show processes greater/equal to 1 percent busy
```

NOTE There are also TTY, VT100, and T6530 commands. Enter HELP for those commands to obtain more info about support for these devices. Also note that any command can be added to a file named <object>CNF file in your default subvolume, or on $system.system.*

5.5 NODES Command

```
NODES
```

The N or NODES command displays the list of ADD \<nodes> that have been added to RPM.

These nodes are the nodes that will have real-time CPU and Process Busy statistics analysis performed whenever * appears in the CPU, PB, or ZOOM command.

EXAMPLE

NODES

5.6 SET Command

```
SET | S  [ ALERTS     ON|OFF               ]
         [ CRIT       <percent-busy>       ]
         [ WARN       <percent-busy>       ]
         [ INFO       <percent-busy>       ]

         [ BUSY | %   <percent-busy>       ]
         [ BUSYCPU    <percent-busy>       ]
         [ BUSYPB     <percent-busy>       ]

         [ ENTRIES    <number>             ]
         [ LOGGING    ON|OFF               ]
         [ LOGFILE    <filename>           ]
         [ OBEYESCAPE ON|OFF               ]
         [ SORT       ByCpu | ByNode       ]
         [ TRACETOKEN ON|OFF               ]
         [ USECS      ON|OFF               ]
         [ RATE       <default-seconds>    ]
         [ TERM       TTY | VT100 | T6530 ]
```

The **S** or **SET** command controls properties of the run-time environment.

ALERTS ON|OFF - reserved for future use.

CRIT <percent> value of Cpu Busy threshold for Critical alerts.
WARN <percent> value of Cpu Busy threshold for Warning alerts.
INFO <percent> value of Cpu Busy threshold for Info alerts.

BUSY | % | BUSYCPU | BUSYPB <percent> indicates the default busy threshold value. If **BUSY|% <percent>** is specified then <percent> applies to both the CPU and PB command. Specifying separate **BUSYCPU <percent>** or **BUSYPB <percent>** defines the BUSY value for the CPU and/or PB commands respectively.

ENTRIES <number> controls the default value of the ENTRIES option for the PB command.

ObeyEscape ON|OFF controls behavior of interactive commands encountered in an OBEY file. **ON** implies interactive commands such as CPU and PB will 'escape' from the

OBEY file, eg when ObeyEscape is ON, interactive commands end the obey file steam. If ObeyEScape is ON and BREAK is pressed, the program will prompt for additional commands. If ObeyEscape is **OFF** and BREAK is pressed a TACL prompt appears. Pausing TACL will cause RPM to continue.

SORT ByCpu | ByNode - controls the display order of the top <N> busiest processes. **ByCpu** displays busiest processes in each Cpu. **ByNode** displays the busiest processes across all Cpus in each node in one list of processes sorted from busiest to least busy processes. If you do not specify this option, the busiest processes are listed ByNode. You can globally control this option with the SET SORT ByCpu | ByNode option.

RATE <default-seconds> controls the default value of the RATE <seconds> option for the CPU and PB commands. Note this value can be gloabally changed with the SET RATE <seconds> option. You can override the default value without changing the default by specifying RATE <seconds> on the CPU or PB command. Note that although short sample intervals such as 1 or 2 seconds cause the screen to update frequently and these fast sample times are supported by RPM, a one second update is not necessarily the best setting. Too frequent updates can be disorientating, and in particular a 10 second sample interval has some special advantages. When RATE is 10 seconds, the digits to the right of the %Busy decimal point represent milliseconds. No other short term RATE provides this numerical convenience. For example:

Milliseconds Used	SET RATE	Percent CPU Busy
100ms	10 seconds	1.00% Cpu Busy
90ms	10 seconds	0.90% Cpu Busy
80ms	10 seconds	0.80% Cpu Busy
70ms	10 seconds	0.70% Cpu Busy
60ms	10 seconds	0.60% Cpu Busy
and so on ...	...	...

TERM TTY | VT100 | T6530 indicates terminal type for video. TTY is text only with no video alerts/enhancement provided.

USECS - ON|OFF controls the default value of USECS in the PB command. This option shows time of day in microseconds for the NORMAL display.

EXAMPLES

```
SET TERM TTY   ! use no video
SET TERM VT100 ! use VT100 video
SET TERM T6530 ! use T6530 video
SET CRIT 50 WARN 10 INFO 1
SET %1         ! Only show Cpus/Processes if >= 1% busy
```

5.7 STATUS Command

```
STATUS [ SSG | [\<node>].$<PID> ]
```

The STATUS command displays the status of SeeView Server Gateway (SSG) processes on the ADD nodes, or for a given $PID.

EXAMPLES

```
STATUS        ! show status of all SSG's on all ADD nodes
STATUS SSG    ! same as STATUS
STATUS $ZSCX  ! show status of pid $ZSCX
```

5.8 T6530 Command

```
T6530
```

The T6530 command is equivalent to entering SET TERM T6530.

You can put commands in <object>CNF file, where <object> is the name of this program.
For example, if the program object file name is "RPM65" then you can create a file named RPM65CNF. Whenever you subsequently run RPM65 it will automatically obey all the commands that are contained in the file RPM65CNF.

File RPM65CNF:

```
ADD \node1
ADD \node2
ADD \node3
SET TERM T6530
SET RATE 10
SET ENTRIES 7
P \*
```

5.9 VT100 Command

```
VT100
```

VT100 terminal support is present generically in nearly all Windows, Linux, and Unix devices. For example, on any Windows XP device you access VT100 emulation simply by entering:

```
C:> TELNET <ip-address>
```

Entering the above from a MS-Windows DOS or "Command Prompt" will connect to the specified host, and if you click on the "control box" you can SUPER-SIZE the VT100's screen width x height to be 100s of lines long and hundreds of characters wide.

The **VT100** command is equivalent to **SET TERM VT100**.

You can put **SET** commands in a file named <object>CNF file, where <object> is the name of the RPM object file. For example if the program object file name is "RPMVT" and you create a file named RPMVTCNF, then whenever RPM runs it will automatically obey all commands in the file RPMVTCNF.

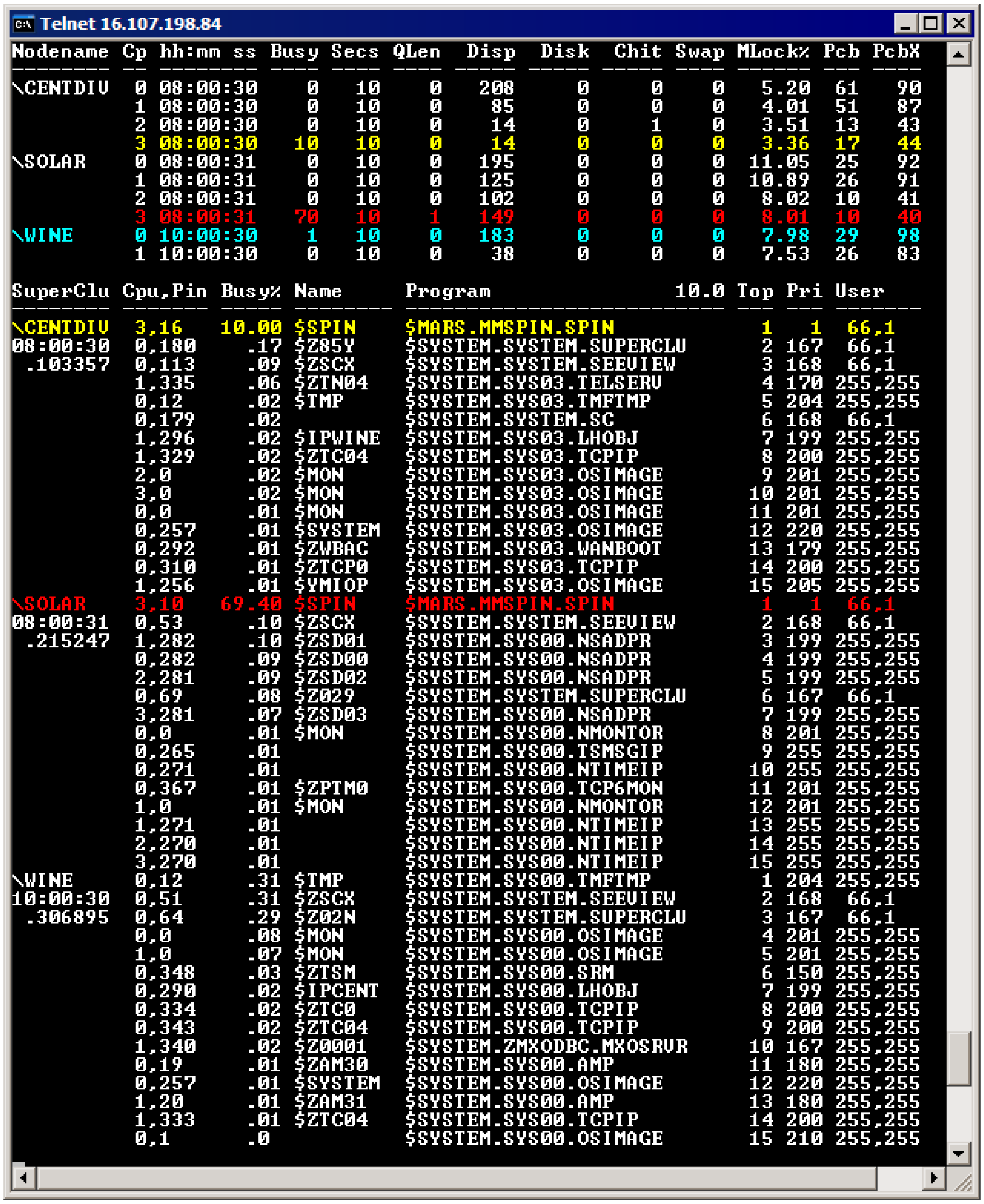
File RPMVTCNF:

```
  ADD \node1
  ADD \node2
  ADD \node3
  SET TERM VT100
  SET INFO 1 WARN 10 CRIT 50
  SET RATE 6
  SET ENTRIES 10
  Z \*
```

5.10 ZOOM Command

```
ZOOM | Z   [ \* | \sysname ]   [ BUSY | % | BUSYCPU | BUSYPB <percent> ]
                               [ DETAIL             ]
                               [ NORMAL             ]
                               [ ENTRIES <num>      ]
                               [ MEM | MB | PCT     ]
                               [ TTY|T6530|VT100    ]
                               [ RATE <secs>        ]
                               [ USECS              ]
```

ZOOM provides a blended display of both Cpu and Process statistics. Example: `Z\*`



```
Nodename Cp hh:mm ss Busy Secs QLen  Disp   Disk   Chit Swap MLock% Pcb PcbX
-------- -- -------- ---- ---- ---- ------ ------ ------ ---- ------ --- ----
\CENTDIV  0 08:00:30    0   10    0    208      0      0    0   5.20  61   90
          1 08:00:30    0   10    0     85      0      0    0   4.01  51   87
          2 08:00:30    0   10    0     14      0      1    0   3.51  13   43
          3 08:00:30   10   10    0     14      0      0    0   3.36  17   44
\SOLAR    0 08:00:31    0   10    0    195      0      0    0  11.05  25   92
          1 08:00:31    0   10    0    125      0      0    0  10.89  26   91
          2 08:00:31    0   10    0    102      0      0    0   8.02  10   41
          3 08:00:31   70   10    1    149      0      0    0   8.01  10   40
\WINE     0 10:00:30    1   10    0    183      0      0    0   7.98  29   98
          1 10:00:30    0   10    0     38      0      0    0   7.53  26   83

SuperClu Cpu,Pin Busy% Name     Program                   10.0 Top Pri User
-------- ------- ----- -------- ------------------------------ --- --- -------
\CENTDIV  3,16   10.00 $SPIN    $MARS.MMSPIN.SPIN                1   1  66,1
08:00:30  0,180    .17 $Z85Y    $SYSTEM.SYSTEM.SUPERCLU          2 167  66,1
 .103357  0,113    .09 $ZSCX    $SYSTEM.SYSTEM.SEEVIEW           3 168  66,1
          1,335    .06 $ZTN04   $SYSTEM.SYS03.TELSERV            4 170 255,255
          0,12     .02 $TMP     $SYSTEM.SYS03.TMFTMP             5 204 255,255
          0,179    .02          $SYSTEM.SYSTEM.SC                6 168  66,1
          1,296    .02 $IPWINE  $SYSTEM.SYS03.LHOBJ              7 199 255,255
          1,329    .02 $ZTC04   $SYSTEM.SYS03.TCPIP              8 200 255,255
          2,0      .02 $MON     $SYSTEM.SYS03.OSIMAGE            9 201 255,255
          3,0      .02 $MON     $SYSTEM.SYS03.OSIMAGE           10 201 255,255
          0,0      .01 $MON     $SYSTEM.SYS03.OSIMAGE           11 201 255,255
          0,257    .01 $SYSTEM  $SYSTEM.SYS03.OSIMAGE           12 220 255,255
          0,292    .01 $ZWBAC   $SYSTEM.SYS03.WANBOOT           13 179 255,255
          0,310    .01 $ZTCP0   $SYSTEM.SYS03.TCPIP             14 200 255,255
          1,256    .01 $YMIOP   $SYSTEM.SYS03.OSIMAGE           15 205 255,255
\SOLAR    3,10   69.40 $SPIN    $MARS.MMSPIN.SPIN                1   1  66,1
08:00:31  0,53     .10 $ZSCX    $SYSTEM.SYSTEM.SEEVIEW           2 168  66,1
 .215247  1,282    .10 $ZSD01   $SYSTEM.SYS00.NSADPR             3 199 255,255
          0,282    .09 $ZSD00   $SYSTEM.SYS00.NSADPR             4 199 255,255
          2,281    .09 $ZSD02   $SYSTEM.SYS00.NSADPR             5 199 255,255
          0,69     .08 $Z029    $SYSTEM.SYSTEM.SUPERCLU          6 167  66,1
          3,281    .07 $ZSD03   $SYSTEM.SYS00.NSADPR             7 199 255,255
          0,0      .01 $MON     $SYSTEM.SYS00.NMONTOR            8 201 255,255
          0,265    .01          $SYSTEM.SYS00.TSMSGIP            9 255 255,255
          0,271    .01          $SYSTEM.SYS00.NTIMEIP           10 255 255,255
          0,367    .01 $ZPTM0   $SYSTEM.SYS00.TCP6MON           11 201 255,255
          1,0      .01 $MON     $SYSTEM.SYS00.NMONTOR           12 201 255,255
          1,271    .01          $SYSTEM.SYS00.NTIMEIP           13 255 255,255
          2,270    .01          $SYSTEM.SYS00.NTIMEIP           14 255 255,255
          3,270    .01          $SYSTEM.SYS00.NTIMEIP           15 255 255,255
\WINE     0,12     .31 $TMP     $SYSTEM.SYS00.TMFTMP             1 204 255,255
10:00:30  0,51     .31 $ZSCX    $SYSTEM.SYSTEM.SEEVIEW           2 168  66,1
 .306895  0,64     .29 $Z02N    $SYSTEM.SYSTEM.SUPERCLU          3 167  66,1
          0,0      .08 $MON     $SYSTEM.SYS00.OSIMAGE            4 201 255,255
          1,0      .07 $MON     $SYSTEM.SYS00.OSIMAGE            5 201 255,255
          0,348    .03 $ZTSM    $SYSTEM.SYS00.SRM                6 150 255,255
          0,290    .02 $IPCENT  $SYSTEM.SYS00.LHOBJ              7 199 255,255
          0,334    .02 $ZTC0    $SYSTEM.SYS00.TCPIP              8 200 255,255
          0,343    .02 $ZTC04   $SYSTEM.SYS00.TCPIP              9 200 255,255
          1,340    .02 $Z0001   $SYSTEM.ZMXODBC.MXOSRVR         10 167 255,255
          0,19     .01 $ZAM30   $SYSTEM.SYS00.AMP               11 180 255,255
          0,257    .01 $SYSTEM  $SYSTEM.SYS00.OSIMAGE           12 220 255,255
          1,20     .01 $ZAM31   $SYSTEM.SYS00.AMP               13 180 255,255
          1,333    .01 $ZTC04   $SYSTEM.SYS00.TCPIP             14 200 255,255
          0,1      .0           $SYSTEM.SYS00.OSIMAGE           15 210 255,255
```