ViewSys User’s Guide

Abstract

ViewSys provides system managers and system operators the ability to view system resources on HP NonStop™ servers. This manual describes how the ViewSys program operates and how to interpret the program output.

Product Version

ViewSys G06

Supported Release Version Updates (RVUs)

This manual supports D39.02, G06.08, and all subsequent RVUs until otherwise indicated by its replacement publication.

Part Number	Published
526358-002	June 2004

Document History

Part Number	Product Version	Published
083999	ViewSys C00	November 1987
061038	ViewSys C30	July 1991
103491	ViewSys C30	December 1993
526358-001	ViewSys G06	May 2003
526358-002	ViewSys G06	June 2004

ViewSys User's Guide

Index Figures Tables

Figures

Figures (continued)

Tables

What’s New in This Manual

Manual Information

Abstract

ViewSys provides system managers and system operators the ability to view system resources on HP NonStop™ servers. This manual describes how the ViewSys program operates and how to interpret the program output.

Product Version

ViewSys G06

Supported Release Version Updates (RVUs)

This manual supports D39.02, G06.08, and all subsequent RVUs until otherwise indicated by its replacement publication.

Part Number	Published
526358-002	June 2004

Document History

Part Number	Product Version	Published
083999	ViewSys C00	November 1987
061038	ViewSys C30	July 1991
103491	ViewSys C30	December 1993
526358-001	ViewSys G06	May 2003
526358-002	ViewSys G06	June 2004

New and Changed Information

The fifth edition contains this change:

- ERRORPROMPT Command on page 3-4 was clarified.

The fourth edition contained these changes:

- At run time ViewSys now compiles the ViewSysC file, which contains the recommended maximum value settings for all known CPU types.
- ViewSys uses maximum values from a new ViewSysC file to determine the graph maximum for all displays of rate or length values; for example, the disk I/O rate or the CPU queue length. The ViewSysC file can also be modified, allowing users to customize these settings by using the CPU option for the MAXIMUM statement. For a detailed explanation of the ViewSysC file, see Appendix D, ViewSys

Definition Language (VDL). Whenever ViewSys is executed, it now automatically compiles the ViewSysC file and displays any errors found.

About This Manual

ViewSys gives system managers and system operators the ability to view system resources on HP NonStop systems. It also provides data to those responsible for load balancing and overall system performance.

ViewSys returns system resource information interactively, reporting data similar to that provided by PEEK, another HP NonStop resource reporting utility, and Measure, an HP NonStop performance analysis tool. PEEK displays resource use from a specific timestamp to the present. ViewSys provides the average resource use for each polling period.

Measure provides detailed resource use for each interval, but the results are generally interpreted offline after the measurement is complete. ViewSys allows the viewing of system resources in an online environment, so you can view the effects of process relocations or the altering of controller paths as the changes are made. ViewSys must be run from an HP block mode terminal or from any HP NonStop terminal emulator.

If you are familiar with either Measure or PEEK, you can translate that knowledge to this utility and need only become familiar with the commands and function keys to control the utility. If you are unfamiliar with the resources being measured, first read the *Measure User's Guide*.

Manual Organization

The manual is organized as follows:

- Section 1, Getting Started, describes the program execution and the basic operational function key commands.
- Section 2, ViewSys Overview, discusses the display modes in more detail. It contains general information as an introduction to the next section.
- Section 3, ViewSys Run Command Syntax, covers the various run commands that define and control the utility.
- Section 4, Function Key Commands, defines each function key that can be used to control the utility.
- Section 5, Interpreting ViewSys, discusses the resources measured and the interpretation of the values displayed.
- Section 6, Measurement Techniques, describes the internal measurement techniques used in the utility.
- Appendix A, ViewSys Syntax Summary, lists the command syntax and keywords.
- Appendix B, ViewSys Measurement Entities, shows the formulas used to derive the bar graph percentages as displayed.
- Appendix C, ViewSys Help Screen, explains the help screen.

- Section D, ViewSys Definition Language (VDL), discusses the ViewSysC file and the ViewSys Definition Language.

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 3-2.

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
```

Change Bar Notation

Change bars are used to indicate substantive differences between this manual and its preceding version. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

> The message types specified in the REPORT clause are different in the COBOL85 environment and the Common Run-Time Environment (CRE).
>
> The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.

1 Getting Started

ViewSys is a system resource monitor. Viewing the resource allocations across processors on a running system allows you to balance the application load more evenly. ViewSys can help you decide when to move user processes to less busy processors and disk files or relocate partitions to less busy disk volumes.

ViewSys indicates the current allocation of a given resource and the percentage used of that resource, allowing you to detect possible resource contention problems before they become serious.

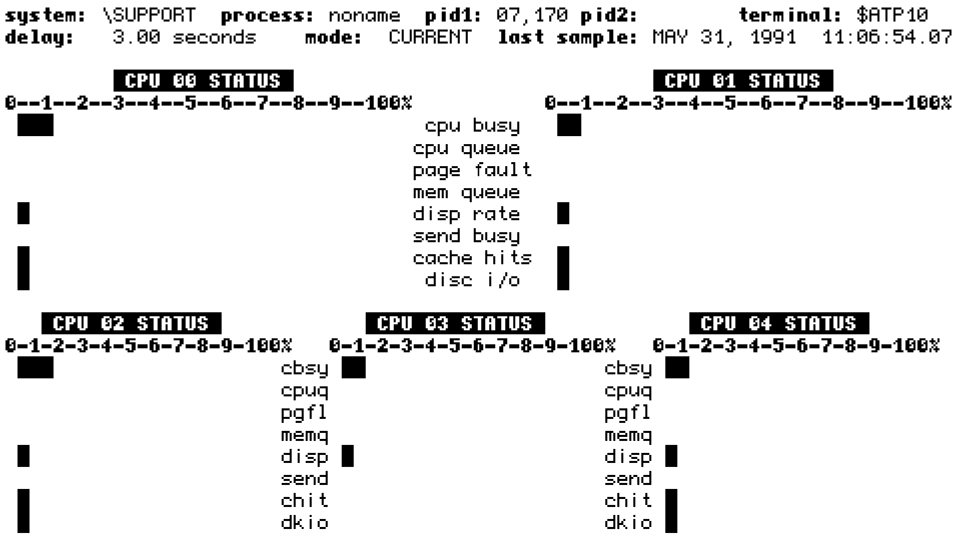
The program is nonprivileged. It requires little system overhead and is most useful when run during application processing.

Starting ViewSys

To start the program, type `VIEWSYS` at the TACL prompt. ViewSys compiles the maximum values contained in the ViewSysC file and then puts the terminal in block mode. Initially, ViewSys displays statistics for all processors that have been configured on the node. If there are more than eight processors, it alternates between two screens to display all the processors. You can use the USERCPUS and UPCPUS commands to limit the number of processors monitored.

Screen Display Description

Figure 1-1. The First ViewSys Screen



```
system: \SUPPORT  process: noname  pid1: 07,170 pid2:         terminal: $ATP10
delay:   3.00 seconds    mode:  CURRENT  last sample: MAY 31, 1991  11:06:54.07

          CPU 00 STATUS                                    CPU 01 STATUS
0--1--2--3--4--5--6--7--8--9--100%            0--1--2--3--4--5--6--7--8--9--100%
                                  cpu busy
                                 cpu queue
                                 page fault
                                 mem queue
                                 disp rate
                                 send busy
                                 cache hits
                                  disc i/o

    CPU 02 STATUS            CPU 03 STATUS            CPU 04 STATUS
0-1-2-3-4-5-6-7-8-9-100%   0-1-2-3-4-5-6-7-8-9-100%   0-1-2-3-4-5-6-7-8-9-100%
                       cbsy                       cbsy
                       cpuq                       cpuq
                       pgfl                       pgfl
                       memq                       memq
                       disp                       disp
                       send                       send
                       chit                       chit
                       dkio                       dkio
```

VST011.vsd

Several resources appear for each processor listed. An abbreviation of the resources appears opposite the bar graphs. Each time the display changes, the bar graphs display the percentages used during the polling period. The default poll delay for ViewSys is three seconds.

The system name appears in the two-line header. If the ViewSys process is named, that name appears in the process field. The primary process ID is displayed in the pid1 field; if the utility has a backup process, its ID is displayed in the next field. The name of the terminal on which ViewSys is being run is displayed in the terminal field.

The second line contains the delay between polls, followed by a mode field. ViewSys begins execution in Current mode, meaning that each display will show the resource use for that polling period. The Maximum mode will display the highest resource use levels since the resource values were last initialized. The final field in the header displays the timestamp at the time of the last poll or sample.

The terminal is read immediately after this timestamp changes. Therefore, it is best to issue a function key command immediately after this value changes.

To use Help, press `SHIFT-ROLL UP` or `BREAK`. A poll delay of 60 seconds occurs, allowing you time to read the HELP screen. Press any of the listed function keys to exit the HELP screen and to execute the corresponding command. For the function keys for the 6530 terminal and the EM6530 equivalents, see Appendix C, ViewSys Help Screen.

1. Press `F1` to display the CPUBUSY screen.
2. As each new screen appears, press the next function key to view each of the separate resource display screens.
3. To see individual processor screens, press the proper shifted function key. For example, `SHIFT-F1` yields processor 0, `SHIFT-F2` yields processor 1, and so on, through `SHIFT-F16`, which displays processor 15.

Maximum Mode

1. Press the `ROLL DOWN` function key to switch the utility environment to Maximum mode. In this mode, statistics move upward only.
2. Press the `INS LINE` function key to reset the maximums for all resources to the current poll period values. Resource values are incremented whenever a new maximum use is encountered.

These maximum values are maintained regardless of the display mode selected, so you can view current values for an hour and then select the Maximum mode to see the highest resource use since the values were last initialized.

Average Mode

1. Pressing the ROLL DOWN function key two times will switch the utility environment to Average mode. In this mode, statistics are incremented when the average usage increases since ViewSys has been running.
2. Press the INS LINE function key to reset the averages for all resources to the current poll period values. Resource values are incremented whenever a new average use is encountered.

These average values are maintained regardless of the display mode selected, so you can view current values for an hour and then select the average mode to see the average resource use since the values were last initialized.

When you press the `NEXT PAGE` function key, ViewSys rotates from display to display until the next time you press that function key. At that point, the currently selected display will continuously be updated and displayed. Rotation is toggled from state to state by subsequent function key commands.

Exiting ViewSys

To exit ViewSys, press `SHIFT-INS LINE` or `F16`.

2 ViewSys Overview

ViewSys runs on all HP NonStop servers as a nonprivileged process. It displays system resource use in block mode and bar graph form and allows you to select from a number of display screens.

ViewSys allows you to:

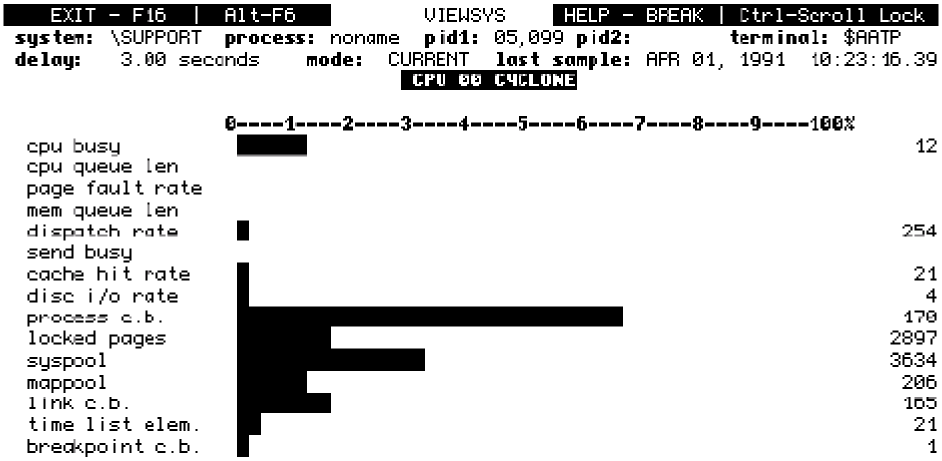
- Monitor all processors within a system
- Monitor a selected set of processors within a system

Once ViewSys is running, you can select a screen to:

- View resource use within a single processor
- View resource use within all processors being monitored (See Figure 2-1.)
- View the use of a single resource throughout all the processors being monitored

Figure 2-1 shows resource use for CPU00.

Figure 2-1. Resource Use for CPU00



```
   EXIT - F16   |  Alt-F6            VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,099 pid2:          terminal: $AATP
 delay:    3.00 seconds     mode:  CURRENT   last sample: APR 01, 1991  10:23:16.39
                                         CPU 00 CYCLONE

                    0----1----2----3----4----5----6----7----8----9----100%
  cpu busy                                                                  12
  cpu queue len
  page fault rate
  mem queue len
  dispatch rate                                                            254
  send busy
  cache hit rate                                                            21
  disc i/o rate                                                              4
  process c.b.                                                             170
  locked pages                                                            2897
  syspool                                                                 3634
  mappool                                                                  206
  link c.b.                                                                165
  time list elem.                                                           21
  breakpoint c.b.                                                            1
```

VST021.vsd

The following metrics are reported for each processor being monitored. You can tailor each metric's maximum value for controlling the graph size to the users environment. For an explanation of altering the maximum values, see Appendix D, ViewSys Definition Language (VDL). Section 5, Interpreting ViewSys, shows a sample screen for each of the entities reported:

- Processor busy
- Processor queue length
- Page fault rate
- Memory queue length

- Processor dispatch rate
- Interprocessor bus send busy rate
- Disk cache hit rate
- Disk I/O rate
- Process control blocks (PCBs) currently allocated
- Number of currently locked memory pages
- Current number of pages allocated to SYSPOOL
- Current number of pages allocated to MAPPOOL
- Link control blocks (LCBs) currently allocated
- Time list elements (TLEs) currently allocated
- Breakpoints (BPTs) currently allocated

You can collect data in two modes:

Mode	Description
Display Current	Each display reflects the average use during the last collection period.
Display Maximum	Each display reflects the maximum value reported for that particular resource since either the program began execution or the maximums were reset.

Other ViewSys display modes you can select:

- ROTATION can be set off or on, providing the repeated viewing of one particular display or the display of all screens in rotation.
- ROTATECPUS can be set off or on, providing the repeated viewing of a selected processor set screen or the display of each processor set screen in rotation. Processor sets can be defined at startup: processors 0 through 3 might form a set, and processors 4 through 7 might form another set. This command is effective only when the DISPLAY ALL CPUS command is in use.
- TIERS can be set to 1 or 2, providing two modes to display processor sets of 2, 3, or 4. When you select a single TIER is selected, the processor statistics appear in vertical alignment, side by side. When you issue the TIERS 2 command, the processors appear in two tiers, with half the processors in the upper half of the screen and the remaining processors in the bottom half. In the two-tier mode, fewer resources appear for each processor.

Compatibility Between PEEK and MEASURE

ViewSys accesses the same system tables as Measure. Because ViewSys does not write to any of these tables, you can run ViewSys and Measure simultaneously. The impact on Measure is limited to the resources used by the program itself. Most of the impact is the interprocessor communications necessary to gather individual processor values. A moderate amount of arithmetic and the display code is performed in the processor in which ViewSys executes. This version of ViewSys does not depend on Measure recorders to produce the measurement data.

PEEK also uses the system tables accessed by ViewSys. ViewSys does not affect PEEK statistics in any way. ViewSys might be slightly affected by the execution of a PEEK INIT command, because this command initializes some of the counters used by ViewSys. The impact to ViewSys is limited to only one measurement period.

Running ViewSys in NonStop Mode

Because ViewSys is an interactive terminal display utility, there is usually no need to run it in NonStop mode. If an error causes ViewSys to terminate, simply restart it.

ViewSys can be run in NonStop mode by using the run time command, `BACKUPCPU` *n*`.` However, it must be run as a named process. In NonStop mode, ViewSys checkpoints only when the display mode is altered, using little system overhead. On a switch to the backup process, the most recent display data is lost. NonStop coding was implemented to provide display and monitoring continuity.

3 ViewSys Run Command Syntax

ViewSys is run from TACL on an HP block mode terminal or from a workstation running a terminal emulation program. The run command syntax for the program follows.

You can also execute some run commands with a function key. See the commands with the notation ** FKEYS **.

Run Command

The ViewSys syntax follows:

```
VIEWSYS [ / run-option [ , run-option ] . . . / ]
[ command [ ; command ] . . . ]
```

run-option is one of:

```
NAME [ $process-name ]

CPU cpu-number

PRI priority
```

`NAME [ $process-name ]`

if present, is the symbolic process name to be given the ViewSys process. The NAME parameter is required if ViewSys will run with a backup process.

`CPU cpu-number`

if present, specifies the processor where the ViewSys process is to execute.

`PRI priority`

if present, specifies the execution priority of the ViewSys process. To ensure proper operation, run ViewSys a priority higher than the priorities of application processes.

command

is one of the options listed in Table 3-1 on page 3-1.

Startup Commands

A complete description of each startup command option follows Table 3-1.

Table 3-1. Startup Command Options (page 1 of 2)

Backupcpu	Exitkey	Rotation
Cpusets	Functionkeys	Screen
Csvfile	Help	Switchback

Table 3-1. Startup Command Options (page 2 of 2)

Delay	Multitype	Tiers
Display	Numcpus	Unstoppable
Errorprompt	Reservelcbs	Upcpus
Exitafter	Rotatecpus	Usercpus

BACKUPCPU Command

Use the BACKUPCPU command to specify the processor where the backup ViewSys process should run. ViewSys must be a named process to run as a NonStop process.

```
BACKUPCPU cpu-number
```

cpu-number

specifies the number of a running processor, other than the processor in which the primary process is running. Values can range from 0 through 15.

Example

```
VIEWSYS /NAME $vsys,CPU 2/ BACKUPCPU 3
```

CPUSETS Command

Use the CPUSETS command to specify into how many sets the monitored processors will be distributed for display. For example, selecting CPUSETS 2 with a total of 10 processors causes the CPU displays to contain 5 processors each.

```
CPUSETS number
```

number

must be in the range 0 through 15 and cannot exceed the number of processors being monitored by ViewSys.

Example

```
VIEWSYS CPUSETS 4
```

CSVFILE Command

Use the CSVFILE command to capture the ViewSys metric values in a text file. Caution should be used with this command since the EOF will increase during each ViewSys sample interval. This option is primarily intended for QA purposes.

```
CSVFILE { ON | OFF }
```

`ON`

indicates that the VIEWSYS data will be logged into a CSV file named CSVFILE on the current volume/subvolume. You can open CSVFILE in Microsoft Excel on a workstation. If CSVFILE already exists, it will be overwritten.

`OFF`

indicates that the VIEWSYS data will not be logged to a CSV file. OFF is the default setting.

Example

```
VIEWSYS CSVFILE ON
```

Note. The EOF for the CSVFILE increases during every ViewSys sample interval. Disk space consideration is a must when using this parameter. This option is primarily intended for QA purposes.

DELAY Command

Use the DELAY command to specify the approximate time between displays. The actual time between displays is the delay time plus the time spent performing the sample. If you omit this command, the delay defaults to 3 seconds.

```
DELAY number-seconds
```

number-seconds

is in the range 0 though 32767.

Example

```
VIEWSYS DELAY 10
```

DISPLAY Command

Use the DISPLAY command to specify the initial display type. If you omit this command, the default is to display all processors.

```
DISPLAY display-type ** FKEYS **
```

display-type is one of:

Entity Display	**Processor Display**
CPUBUSY	CPU0
CPUQUEUE	CPU1
PAGEFAULT	CPU2

Entity Display	**Processor Display**
MEMQUEUE	CPU3
DISPATCH	CPU4
SENDBUSY	CPU5
CACHEHITS	CPU6
DISCIO	CPU7
PCB	CPU8
LOCKEDMEM	CPU9
SYSPOOL	CPU10
MAPPOOL	CPU11
LCB	CPU12
TLE	CPU13
BPT	CPU14
(Not applicable)	CPU15

Note. You can alter the display type interactively. Function keys 1 through 16 select individual entity displays, and shifted function keys 1 through 16 select individual processor displays.

Example

```
VIEWSYS DISPLAY CPUBUSY
```

ERRORPROMPT Command

Use the ERRORPROMPT command to specify if ViewSys outputs a continuation prompt when errors are generated during the ViewSys definition language compiler phase. If you omit this command, the ERRORPROMPT default is set to ON.

```
ERRORPROMPT { ON | OFF }
```

ON

indicates that a continuation prompt appears on the ViewSys terminal.

OFF

indicates that no continuation prompt appears and ViewSys will run using internal maximum default values for the ViewSys metrics.

Example

```
VIEWSYS ERRORPROMPT OFF
```

EXITAFTER Command

Use the EXITAFTER command to specify the number of samples to be displayed before ViewSys terminates and returns you to the TACL prompt. If you omit this command, ViewSys runs continuously until you use a function key to exit.

```
EXITAFTER number
```

number

must be in the range 1 through 32767.

Example

```
VIEWSYS EXITAFTER 100
```

EXITKEY Command

Use the EXITKEY command to specify a function key to exit the program in addition to the default EXIT key. If you omit this command, the program operates continuously until you strike the predefined EXIT function key (F16 or Alt-F6).

```
EXITKEY integer
```

integer

must be an integer in the range 1 through 16.

FUNCTIONKEYS Command

Use the FUNCTIONKEYS command to specify whether the ViewSys function keys should be operational.

```
FUNCTIONKEYS { ON | OFF }
```

`ON`

indicates that the function keys with which you control ViewSys are operational. This value is the default.

`OFF`

indicates that the function keys are disabled. When running ViewSys in this mode, you must determine the display type at startup time and issue the proper DISPLAY command. When you run ViewSys in this mode, you must stop it from another terminal.

Example

```
VIEWSYS FUNCTIONKEYS OFF
```

HELP Command

Use the HELP command to display a list of the run-time commands. After the commands appear, ViewSys exits and automatically returns you to the TACL prompt.

```
HELP
```

Example

```
VIEWSYS HELP
```

MULTITYPE Command

Use the MULTITYPE command to indicate the number of entities displayed on each screen.

```
MULTITYPE { ON | OFF }
```

`ON`

indicates that when monitoring eight or fewer processors, two individual entities appear concurrently. If you are monitoring more than eight processors, this command is ignored. This value is the default.

`OFF`

indicates that only one entity appears on each screen.

Example

```
VIEWSYS MULTITYPE OFF
```

NUMCPUS Command

Use the NUMCPUS command to indicate which processors are being monitored. Processors numbered 0 through *(number* - 1) will be monitored. Subsequent USERCPUS commands cause additional processors to be monitored if the processors specified therein were not within the range specified by the NUMCPUS command.

```
NUMCPUS number
```

number

must be in the range 1 through 16.

Note. A NUMCPUS command overrides any previous USERCPUS command.

Examples

```
VIEWSYS NUMCPUS 10 ; USERCPUS 12
```

causes processors 0 through 9 and processor 11 to be monitored.

RESERVELCBS Command

Use the RESERVELCBS command to specify whether a sender's link control block is reserved for each processor being monitored. This prevents an LCB allocation failure if the processor ViewSys is running in becomes short of LCBs.

```
RESERVELCBS { ON | OFF }
```

`ON`

indicates that a sender's link control block is reserved for each processor being monitored. This prevents an LCB allocation failure if the processor ViewSys is running in becomes short of LCBs.

`OFF`

indicates that no LCBs will be reserved. This is the default.

Note. In a properly configured system, this command should not be needed. Use this command only if ViewSys has had an LCB allocation failure.

Example

```
VIEWSYS RESERVELCBS ON
```

ROTATECPUS Command

Use the ROTATECPUS command to specify whether CPU sets will be rotated.

```
ROTATECPUS { ON | OFF } ** FKEYS **
```

`ON`

indicates that when the ALLCPUS display type is selected, the display rotates sequentially from CPU set to CPU set. This is the default if more than one CPUSET is configured.

`OFF`

indicates that when the ALLCPUS display type is selected, the current CPU set is displayed repeatedly. To select the next CPU set in sequence, pressing the PREV PAGE function key.

Note. To toggle ROTATECPUS interactively, press the PREV PAGE function key.

Example

```
VIEWSYS CPUSETS 2; ROTATECPUS ON
```

ROTATION Command

Use the ROTATION command to specify that ViewSys should rotate sequentially through display types, both individual entities and CPU sets.

```
ROTATION { ON | OFF } ** FKEYS **
```

`ON`

indicates that ViewSys will rotate sequentially through all the display types, both individual entities and CPU sets.

`OFF`

indicates that ViewSys will repeatedly display the currently selected display. If the ALLCPUS set display is selected, the setting of the ROTATECPUS variable is tested to determine if ViewSys will rotate from CPU set to CPU set. This is the default.

Note. To toggle ROTATION interactively, press the NEXT PAGE function key.

Example

```
VIEWSYS ROTATION ON
```

SCREEN Command

Use the SCREEN command to replay a previous logged ViewSys session. If different screens are selected for display during the logging of the ViewSys session, on playback only the default ViewSys screen will be displayed. This option is primarily intended for QA purposes.

```
SCREEN { LOG | REPLAY | OFF }
```

`LOG`

indicates that the VIEWSYS data will be logged to a disk file named DISPFILE in the current volume/subvolume. The DISPFILE can be used to replay the data. If the file already exists, it is overwritten.

Note. The EOF for the CSVFILE increases during every ViewSys sample interval. Disk space consideration is a must when using this parameter. This option is primarily intended for QA purposes.

`REPLAY`

indicates that VIEWSYS will be running on replaying data from a disk file named DISPFILE in the current volume/subvolume. This parameter causes ViewSys to replay the all processors screen (ViewSys default screen) displaying the bar graph percentages of each metric for each configured CPU on the node.

`OFF`

indicates that the VIEWSYS data will not be logged to a file called DISPFILE and VIEWSYS will not be running on replaying data. OFF is the default setting.

Example

```
VIEWSYS SCREEN LOG
```

SWITCHBACK Command

Use the SWITCHBACK command to indicate whether the backup (or secondary) processor should retain responsibility once the primary processor has recovered from a failure.

```
SWITCHBACK { ON | OFF }
```

`ON`

indicates that should the backup process take over due to the failure of the primary's processor, once the original primary processor is again available, the original backup will return the responsibility to the original primary process.

`OFF`

indicates that following a processor failure which causes the backup process to take over, the new primary (former backup) will maintain responsibility even after the original primary processor is again available. OFF is the default.

Example

```
VIEWSYS SWITCHBACK ON
```

TIERS Command

Use the TIERS command to indicate the number of columns when two, three, or four processors are displayed in one CPU set.

```
TIERS { 1 | 2 } ** FKEYS **
```

`1`

indicates that whenever two, three, or four processors are being displayed in one CPU set, they will be displayed in two, three, or four columns respectively. This is the default.

`2`

indicates that whenever two, three, or four processors are displayed in one CPU set, they appear in two tiers. The upper half of the display contains half the processors, and the bottom half contains the remaining processors. When CPU sets are displayed in this way, fewer resource entities appear for each processor.

Note. To toggle the number of TIERS interactively, press a shifted NEXT PAGE function key.

Example

```
VIEWSYS TIERS 2
```

UNSTOPPABLE Command

Use the UNSTOPPABLE command to specify how ViewSys can be stopped.

```
UNSTOPPABLE { ON | OFF }
```

`ON`

indicates that once ViewSys is started, it can only be stopped by issuing a STOP command from a command interpreter at another terminal. All the program control function keys will be operational with the exception of the DEL LINE key, which is normally used to exit ViewSys.

`OFF`

indicates that the DEL LINE function key can be used to exit the ViewSys program. This is the default.

Example

```
VIEWSYS UNSTOPPABLE ON
```

UPCPUS Command

Use the UPCPUS command to specify whether processors that are not running should be monitored.

```
UPCPUS { ON | OFF }
```

`ON`

indicates that you want to monitor any processors that are currently running. Down processors that are subsequently reloaded will not be monitored.

`OFF`

indicates that you that to monitor all processors that are configured in the current SYSGEN configuration. Down processors are displayed as being down. Once they are reloaded, they are displayed normally. This is the default.

Example

```
VIEWSYS UPCPUS ON
```

USERCPUS Command

Use the USERCPUS command to indicate which processors you want to monitor. Processors numbered 0 through *(number* - 1) will be monitored.

```
USERCPUS cpu-number [ , cpu-number ]...
```

`cpu-number`

must be in the range 1 through 16.

Note. Subsequent NUMCPUS commands override the USERCPUS command, as if it were not entered. Entering a USERCPUS command after a NUMCPUS command causes monitoring of any processors specified within the USERCPUS command, that are not in the range specified by the NUMCPUS command.

Examples

```
VIEWSYS NUMCPUS 10 ; USERCPUS 12
```

causes processors 0 through 9 and processor 11 to be monitored.

```
   VIEWSYS USERCPUS 7, 8 ; NUMCPUS 4
```

causes processors 0 through 3 to be monitored.

4 Function Key Commands

Numbered Function Keys

Individual numbered function keys select the following display screens.

Table 4-1. Numbered Function Keys

Function Key	Function	Function Key	Function
F1	CPU BUSY	SF1	CPU0
F2	CPU QUEUE LENGTH	SF2	CPU1
F3	PAGE FAULT RATE	SF3	CPU2
F4	MEMORY QUEUE LENGTH	SF4	CPU3
F5	DISPATCH RATE	SF5	CPU4
F6	SEND BUSY	SF6	CPU5
F7	CACHE HIT RATE	SF7	CPU6
F8	DISC I/O RATE	SF8	CPU7
F9	LOCKED MEMORY	SF9	CPU8
F10	LOW PIN PCB	SF10	CPU9
F11 or ALT-F1	HIGH PIN PCB	SF11	CPU10
F12 or ALT-F2	MEMORY PRESSURE	SF12	CPU11
F13 or ALT-F3	PROCESS TIME LIST ELEMENT	SF13	CPU12
F14 or ALT-F4	TLEs	SF14	CPU13
F15 or ALT-F5	BPTs	SF15	CPU14
F16 or ALT-F6	EXIT	SF16	CPU15

Unnumbered Function Keys

ViewSys has several unnumbered function keys:

`ROLL UP` Displays all processors being measured. If the current setting of ROTATECPUS is ON and more than one CPUSET is configured, the displays alternate from CPUSET to CPUSET to display all processors.

`Shift ROLL UP` Displays HELP screen.

`ROLL DOWN` Displays the Maximum use values since the maximums were last reset.

`Shift ROLL DOWN` Displays the Current use values.

`NEXT PAGE` Toggles ROTATION. Pressing this function key toggles the state between ON and OFF. ROTATION ON causes the utility to sequentially display all screens. ROTATION OFF repeatedly displays the currently selected screen.

`Shift NEXT PAGE` Toggles TIERS between 1 and 2 tiers. Whenever 2, 3, or 4 processors are being displayed and TIERS is set to 1, the processors appear in vertical columns, side by side. When 2, 3, or 4 processors are displayed and TIERS is set to 2, the upper half of the screen displays half the processors, and the bottom half displays the remaining processors. TIERS 2 yields fewer entities per processor in the display, but the bar graph is longer, giving finer resolution to the percentage used.

`PREV PAGE` Selects next CPUSET. When there is more than one CPUSET, and the setting for ROTATECPUS is OFF, this function key causes the next CPUSET to be selected and displayed continuously.

`Shift PREV PAGE` Toggles ROTATECPUS between ON and OFF. This command is operational only after the ROLL UP function key is pressed to select the all processors display.

`INS LINE` Resets the maximum value counters. With each new sample period, these counters are set to the maximum of their current contents, or the current poll period value.

`DEL LINE` Exits the program.

Table 4-2. Unnumbered Function Keys

Function Key	Function
ROLL UP or Alt-Up Arrow	Displays all measured processors
Shift ROLL UP, BREAK, or Control-Scroll Lock	Displays HELP information
ROLL DOWN or Alt-Down Arrow	Displays Maximum use statistics
Shift ROLL DOWN or Alt-Shift-Down Arrow	Displays Current use statistics
NEXT PAGE or Alt-Pg Down	Toggles rotation
Shift NEXT PAGE or Alt-Pg Down	Toggles tiers
PREV PAGE or Alt-Pg Up	Selects next CPU set
Shift PREV PAGE or Alt-Pg Up	Toggles ROTATECPUS
INS LINE or Control-Ins	Reset maximums
DEL LINE or Control-Del	Exit

5 Interpreting ViewSys

There are no absolute rules to follow for performance tuning HP NonStop systems. Recommended rates for various system resources vary from system to system, based on the application being run. These rates all interrelate to determine the number of transactions performed per second and the average response time for those transactions. Experience with the system and the applications being run allows you to detect imbalances and inefficiencies within that system. The ability to view a single resource use over all the processors in a system you to balance the disposition of processes across the system.

Appendix B, ViewSys Measurement Entities, lists the formulae that determine the percentages used, which are then displayed on bar graphs. Section 6, Measurement Techniques, explains what constitutes 100% of a resource. Generally, if a resource display approaches 80% used, you should monitor that resource. It is common to see poll periods where this threshold is reached or exceeded, but a continuous reading at that level indicates potential system degradation.

ViewSys also monitors over time the maximum percentage ever used on the measured resources. The maximum values are initialized at program start-up and again whenever the `INS LINE` function key is pressed. Selecting the Maximum mode by pressing the `ROLL DOWN` function key, then displays the highest percentage used since the last initialization point. ViewSys can run for a set period; for example, two hours; and display processors that might be underused.

Table 5-1 describes the entities that are measured and reported. Figures 5-1 to 5-15 show each of the ViewSys reports.

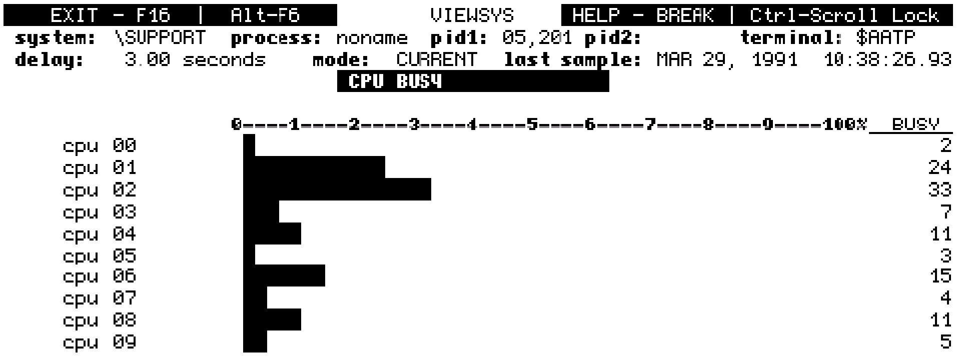
Table 5-1. ViewSys Reports

Measured Entities	**Description** (page 1 of 2)
CPUBUSY	The percentage of time a processor is not idle during the polling period. (Processor busy + interrupt busy time)
CPUQUEUE	The average number of processes on the ready list during the polling period. (Processes that are ready to run)
PAGEFAULT	The average number of page faults that occurred, per second, during the polling period.
MEMQUEUE	The average number of processes awaiting page fault servicing during the polling period.
DISPATCH	The average number of process dispatches per second during the polling period.
SENDBUSY	The percentage of time during the polling period that a send was being performed within a processor.
CACHEHITS	The average number of disk cache hits per second during the polling period.

Table 5-1. ViewSys Reports

Measured Entities	Description (page 2 of 2)
DISCIO	The average number of disk I/Os per second during the polling period. This number is an accumulation of all disk I/Os for all the disk processes within the processor being reported.
PCB	The number of process control blocks allocated at the time the processor is polled.
LOCKEDMEM	The number of pages of physical memory that are locked at the time the processor is polled.
SYSPOOL	The number of pages of SYSPOOL allocated at the time the processor is polled.
MAPPOOL	The number of pages of MAPPOOL allocated at the time the processor is polled.
LCB	The number of link control blocks allocated at the time the processor is polled.
TLE	The number of time list elements allocated at the time the processor is polled.
BPT	The number of break points allocated at the time the processor is polled.

Figure 5-1. The CPUBUSY Screen

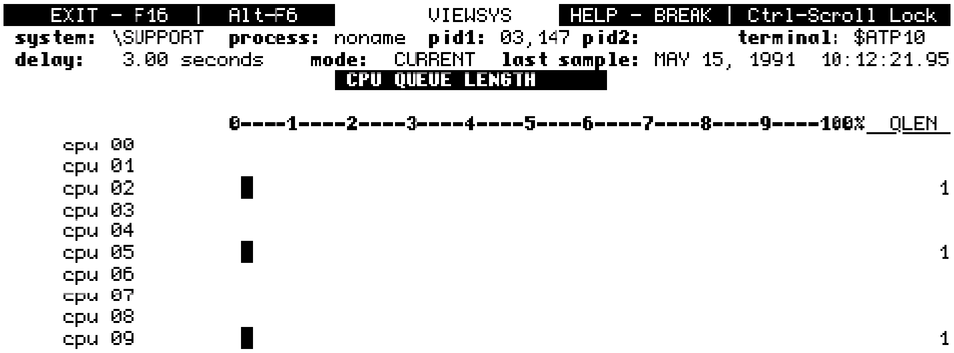


```
   EXIT - F16  |  Alt-F6          VIEWSYS      HELP - BREAK | Ctrl-Scroll Lock
 system:  \SUPPORT  process: noname  pid1: 05,201 pid2:        terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:38:26.93
                                   CPU BUSY

                      0----1----2----3----4----5----6----7----8----9----100%  BUSY
     cpu 00                                                                       2
     cpu 01                                                                      24
     cpu 02                                                                      33
     cpu 03                                                                       7
     cpu 04                                                                      11
     cpu 05                                                                       3
     cpu 06                                                                      15
     cpu 07                                                                       4
     cpu 08                                                                      11
     cpu 09                                                                       5
```

VST501.vsd

Figure 5-2. The CPUQUEUE Screen

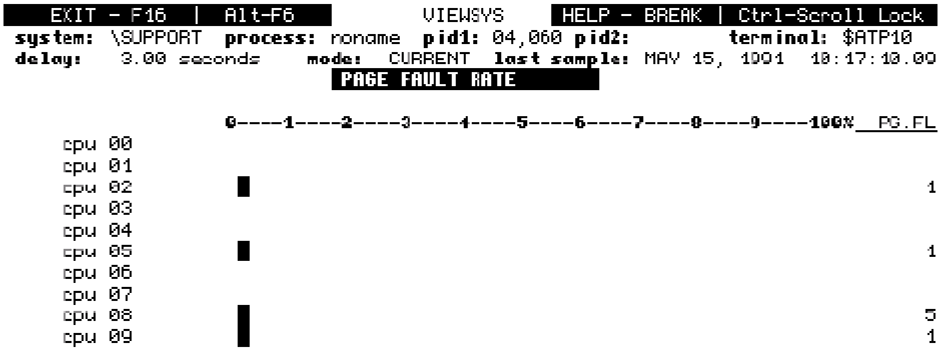


```
     EXIT - F16   |   Alt-F6              VIEWSYS      HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 03,147 pid2:         terminal: $ATP10
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAY 15, 1991  10:12:21.95
                                  CPU QUEUE LENGTH

                           0----1----2----3----4----5----6----7----8----9----100%__QLEN_
     cpu 00
     cpu 01
     cpu 02                 █                                                          1
     cpu 03
     cpu 04
     cpu 05                 █                                                          1
     cpu 06
     cpu 07
     cpu 08
     cpu 09                 █                                                          1
```

VST502.vsd

Figure 5-3. The PAGEFAULT Screen



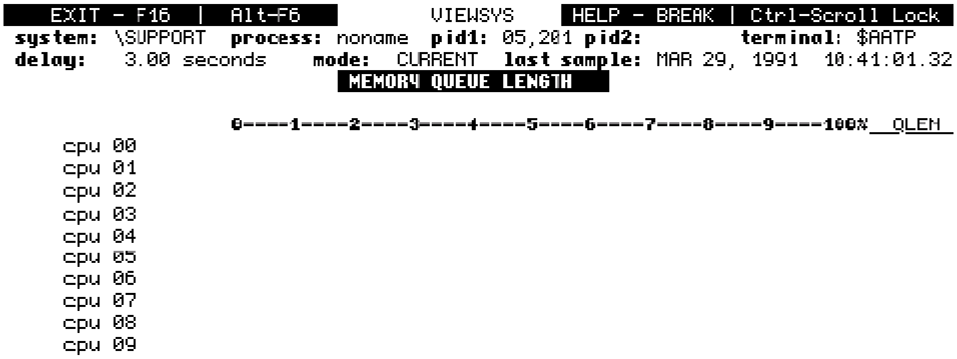
```
     EXIT - F16   |   Alt-F6              VIEWSYS      HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 04,060 pid2:         terminal: $ATP10
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAY 15, 1991  10:17:10.09
                                  PAGE FAULT RATE

                           0----1----2----3----4----5----6----7----8----9----100%__PG.FL
     cpu 00
     cpu 01
     cpu 02                 █                                                         1
     cpu 03
     cpu 04
     cpu 05                 █                                                         1
     cpu 06
     cpu 07
     cpu 08                 █                                                         5
     cpu 09                 █                                                         1
```

VST503.vsd

Note. The next screen shows no current activity. This reflects the state of the system when the screen was captured.

Figure 5-4. The MEMQUEUE Screen

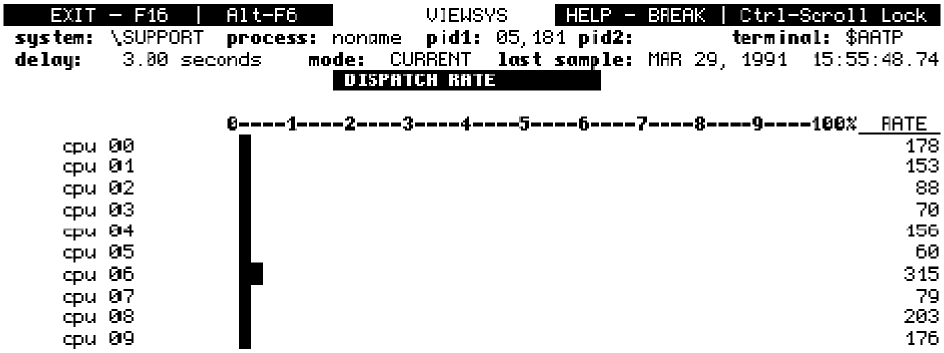


```
   EXIT - F16  |  Alt-F6          VIEWSYS    HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,201 pid2:        terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:41:01.32
                                  MEMORY QUEUE LENGTH

                     0----1----2----3----4----5----6----7----8----9----100%__QLEN_
     cpu 00
     cpu 01
     cpu 02
     cpu 03
     cpu 04
     cpu 05
     cpu 06
     cpu 07
     cpu 08
     cpu 09
```

VST504.vsd

Figure 5-5. The DISPATCH Screen

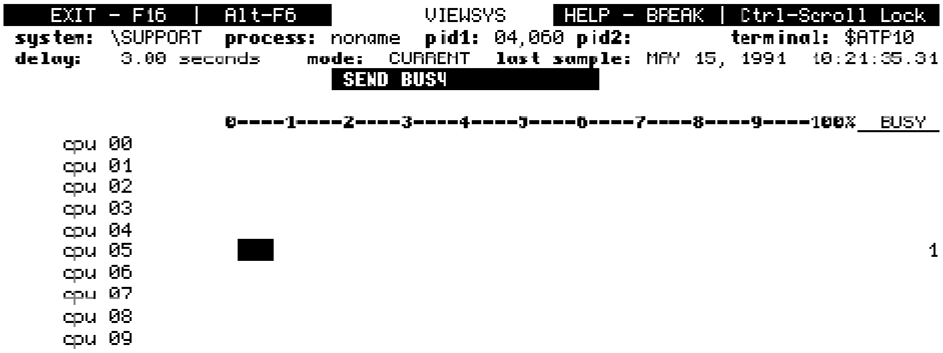


```
   EXIT - F16  |  Alt-F6          VIEWSYS    HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,181 pid2:        terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  15:55:48.74
                                  DISPATCH RATE

                     0----1----2----3----4----5----6----7----8----9----100%__RATE_
     cpu 00                                                                  178
     cpu 01                                                                  153
     cpu 02                                                                   88
     cpu 03                                                                   70
     cpu 04                                                                  156
     cpu 05                                                                   60
     cpu 06                                                                  315
     cpu 07                                                                   79
     cpu 08                                                                  203
     cpu 09                                                                  176
```

VST505.vsd

Figure 5-6. The SENDBUSY Screen

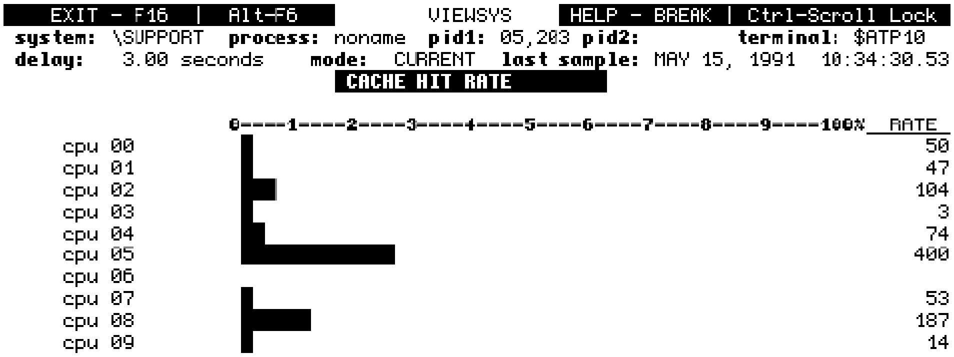


```
   EXIT - F16  |  Alt-F6          VIEWSYS    HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 04,060 pid2:        terminal: $ATP10
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAY 15, 1991  10:21:35.31
                                  SEND BUSY

                     0----1----2----3----4----5----6----7----8----9----100%__BUSY_
     cpu 00
     cpu 01
     cpu 02
     cpu 03
     cpu 04
     cpu 05                                                                    1
     cpu 06
     cpu 07
     cpu 08
     cpu 09
```

VST506.vsd

Figure 5-7. The CACHEHITS Screen

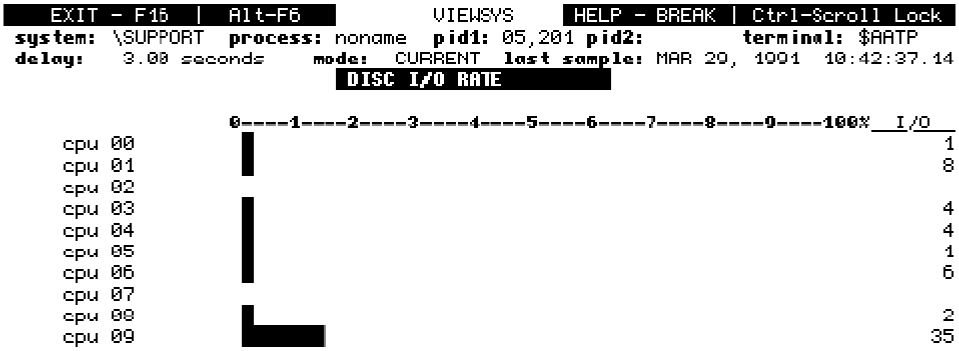


```
  EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,203 pid2:         terminal: $ATP10
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAY 15, 1991  10:34:30.53
                                  CACHE HIT RATE

                     0----1----2----3----4----5----6----7----8----9----100%  RATE
      cpu 00                                                                   50
      cpu 01                                                                   47
      cpu 02                                                                  104
      cpu 03                                                                    3
      cpu 04                                                                   74
      cpu 05                                                                  400
      cpu 06
      cpu 07                                                                   53
      cpu 08                                                                  187
      cpu 09                                                                   14
```

VST507.vsd

Figure 5-8. The DISCIO Screen

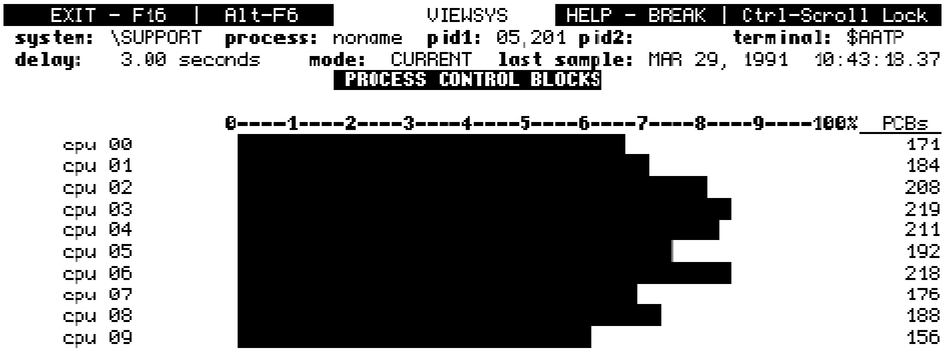


```
  EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,201 pid2:         terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:42:37.14
                                  DISC I/O RATE

                     0----1----2----3----4----5----6----7----8----9----100%  I/O
      cpu 00                                                                    1
      cpu 01                                                                    8
      cpu 02
      cpu 03                                                                    4
      cpu 04                                                                    4
      cpu 05                                                                    1
      cpu 06                                                                    6
      cpu 07
      cpu 08                                                                    2
      cpu 09                                                                   35
```

VST508.vsd

Figure 5-9. The PCB Screen

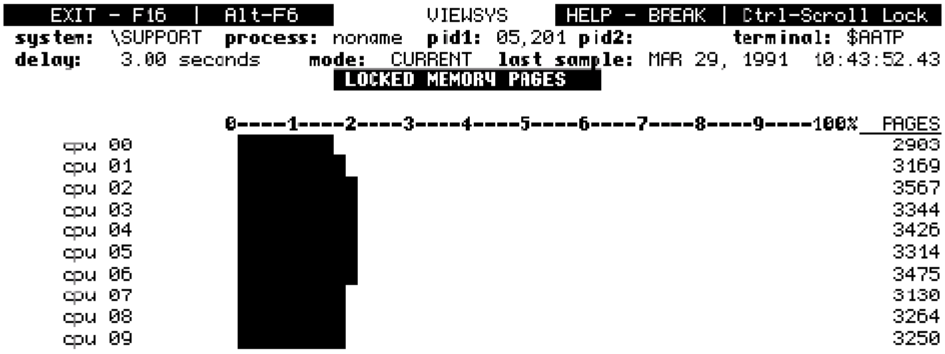


```
  EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,201 pid2:         terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:43:18.37
                                  PROCESS CONTROL BLOCKS

                     0----1----2----3----4----5----6----7----8----9----100%  PCBs
      cpu 00                                                                  171
      cpu 01                                                                  184
      cpu 02                                                                  208
      cpu 03                                                                  219
      cpu 04                                                                  211
      cpu 05                                                                  192
      cpu 06                                                                  218
      cpu 07                                                                  176
      cpu 08                                                                  188
      cpu 09                                                                  156
```

VST509.vsd

Figure 5-10. The LOCKEDMEM Screen

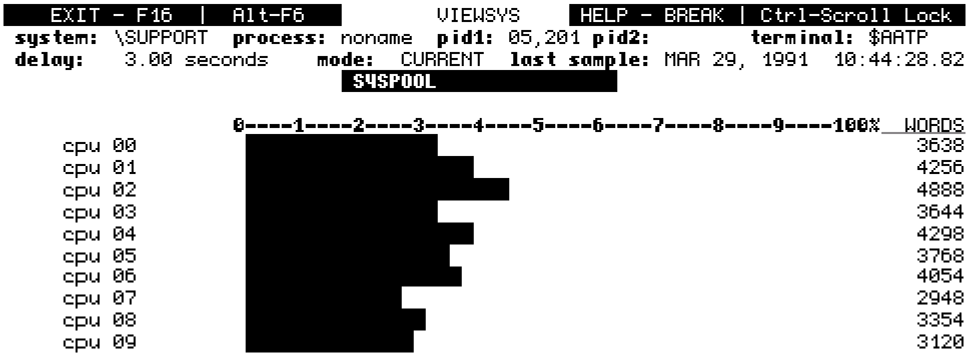


```
 EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,201 pid2:        terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:43:52.43
                                 LOCKED MEMORY PAGES

                      0----1----2----3----4----5----6----7----8----9----100%  PAGES
     cpu 00                                                                    2903
     cpu 01                                                                    3169
     cpu 02                                                                    3567
     cpu 03                                                                    3344
     cpu 04                                                                    3426
     cpu 05                                                                    3314
     cpu 06                                                                    3475
     cpu 07                                                                    3130
     cpu 08                                                                    3264
     cpu 09                                                                    3250
```

VST510.vsd

Figure 5-11. The SYSPOOL Screen

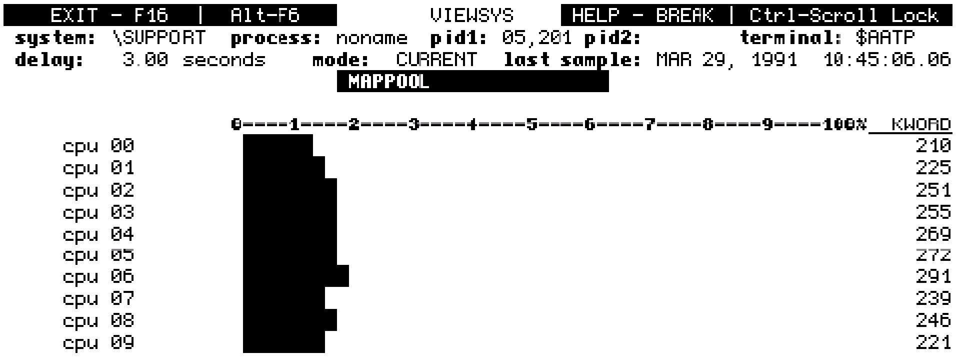


```
 EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,201 pid2:        terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:44:28.82
                                 SYSPOOL

                      0----1----2----3----4----5----6----7----8----9----100%  WORDS
     cpu 00                                                                    3638
     cpu 01                                                                    4256
     cpu 02                                                                    4888
     cpu 03                                                                    3644
     cpu 04                                                                    4298
     cpu 05                                                                    3768
     cpu 06                                                                    4054
     cpu 07                                                                    2948
     cpu 08                                                                    3354
     cpu 09                                                                    3120
```

VST511.vsd

Figure 5-12. The MAPPOOL Screen

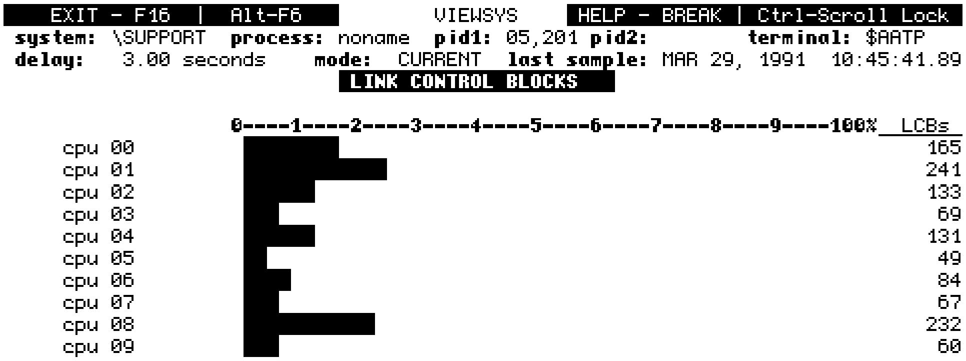


```
 EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,201 pid2:        terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:45:06.06
                                 MAPPOOL

                      0----1----2----3----4----5----6----7----8----9----100%  KWORD
     cpu 00                                                                    210
     cpu 01                                                                    225
     cpu 02                                                                    251
     cpu 03                                                                    255
     cpu 04                                                                    269
     cpu 05                                                                    272
     cpu 06                                                                    291
     cpu 07                                                                    239
     cpu 08                                                                    246
     cpu 09                                                                    221
```

VST512.vsd

Figure 5-13. The LCB Screen

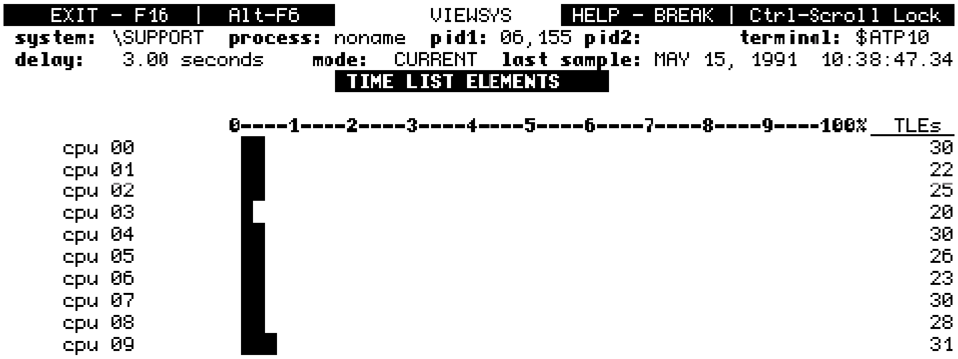


```
    EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,201 pid2:        terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:45:41.89
                               LINK CONTROL BLOCKS

                  0----1----2----3----4----5----6----7----8----9----100%   LCBs
     cpu 00                                                                 165
     cpu 01                                                                 241
     cpu 02                                                                 133
     cpu 03                                                                  69
     cpu 04                                                                 131
     cpu 05                                                                  49
     cpu 06                                                                  84
     cpu 07                                                                  67
     cpu 08                                                                 232
     cpu 09                                                                  60
```

VST513.vsd

Figure 5-14. The TLE Screen

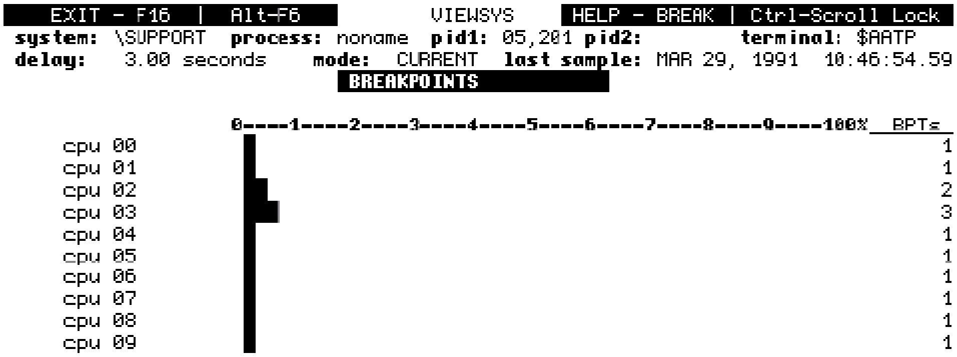


```
    EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 06,155 pid2:        terminal: $ATP10
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAY 15, 1991  10:38:47.34
                               TIME LIST ELEMENTS

                  0----1----2----3----4----5----6----7----8----9----100%   TLEs
     cpu 00                                                                  30
     cpu 01                                                                  22
     cpu 02                                                                  25
     cpu 03                                                                  20
     cpu 04                                                                  30
     cpu 05                                                                  26
     cpu 06                                                                  23
     cpu 07                                                                  30
     cpu 08                                                                  28
     cpu 09                                                                  31
```

VST514.vsd

Figure 5-15. The BPT Screen



```
     EXIT - F16  |  Alt-F6          VIEWSYS     HELP - BREAK | Ctrl-Scroll Lock
 system: \SUPPORT  process: noname  pid1: 05,201 pid2:         terminal: $AATP
 delay:    3.00 seconds    mode:  CURRENT  last sample: MAR 29, 1991  10:46:54.59
                                BREAKPOINTS

                    0----1----2----3----4----5----6----7----8----9----100%  BPTs
     cpu 00                                                                    1
     cpu 01                                                                    1
     cpu 02                                                                    2
     cpu 03                                                                    3
     cpu 04                                                                    1
     cpu 05                                                                    1
     cpu 06                                                                    1
     cpu 07                                                                    1
     cpu 08                                                                    1
     cpu 09                                                                    1
```

VST515.vsd

6 Measurement Techniques

ViewSys obtains the system resource configuration for each processor being monitored. For measured entities that have a finite number of blocks or memory pages available, the actual number configured is determined. For some measurements, such as DISPATCHES or DISC I/O RATE, an arbitrary maximum has been determined, so the percentage-used display indicates the remaining percentage of usable resource. In these cases, the maximum arbitrary configuration can be exceeded, but this indicates that a threshold has been reached, after which processor efficiency is likely to be affected.

Table 6-1 lists the maximum values selected for the seven processor types. To estimate what percentage of a particular resource your system is using, compare the value that ViewSys displays to the maximum value listed in Table 6-1. For example, if your system is a Cyclone and your CPU queue is 10 (half of the maximum value of 20), your system is at 50% of maximum capacity.

Table 6-1. Maximum Values for Processors

	VLX	**CLX 600**	**CLX 700**	**CLX 800**	**NSR-L CLX-R**	**Cyclone**	**NSR-N K10000**
CPU QUEUE	14	6	9	12	20	20	20
PAGE FAULT	42	18	28	42	42	126	126
MEMORY QUEUE	13	5	8	11	19	20	20
DISPATCHES	2870	660	1260	1840	2920	6900	11000
CACHE HITS	500	115	250	315	800	1500	2000
DISC I/O	70	26	40	60	60	250	250

When the configured amounts have been determined, ViewSys determines the actual amount of each entity in use once each measurement period. For control blocks, or pages of memory, the actual number in use is returned. For entities measuring a rate, such as CACHE HIT RATE, the average number per second during the measurement period is returned. The CPU BUSY value returned is the actual percentage of the measurement period that the processor was busy. CPU BUSY is *total time* less *idle time* (total time - idle time).

The technique used to obtain the rates and percentage busy values is similar to that used by Measure. The current value of a counter being incremented within the operating system is taken at the beginning of a measurement period. At the end of a measurement period, the value is taken from the counter, and the beginning value is subtracted from it. This yields the amount of activity for that entity for that measurement period. To arrive at the average for the measurement period, the elapsed time is then divided into this value.

A ViewSys Syntax Summary

```
VIEWSYS [ / run-option [ , run-option ]... / ]
   [ command [ ; command ]... ]
```

run-option is one of:

```
   NAME [ $process-name ]

   CPU cpu-number

   PRI priority
```

command is one of:

```
   BACKUPCPU cpu-number

   CPUSETS number

   DELAY number-seconds
*  DISPLAY display-type

   ERRORPROMPT { ON | OFF }

   EXITAFTER number

   EXITKEY integer

   FUNCTIONKEYS { ON | OFF }

   HELP

   MULTITYPE { ON | OFF }

   NUMCPUS number

   RESERVELCBS { ON | OFF }
*  ROTATECPUS { ON | OFF }
*  ROTATION { ON | OFF }

   SWITCHBACK { ON | OFF }
*  TIERS { 1 | 2 }
```

```
UNSTOPPABLE { ON | OFF }

UPCPUS { ON | OFF }

USERCPUS cpu-number [ cpu-number ]...
```

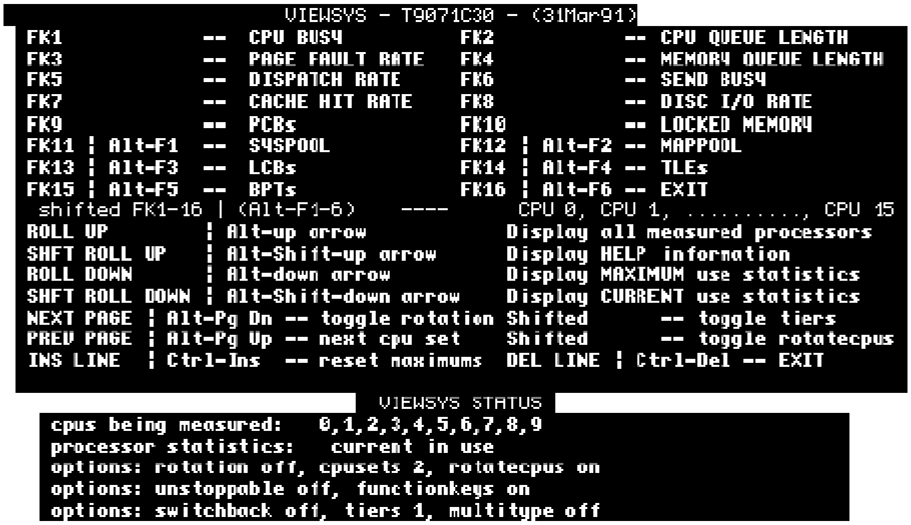
You can set commands marked with an asterisk (*) at run time or during program execution using function keys.

B ViewSys Measurement Entities

Function Key	Entity Represents	Bar Graph Represents	Number Represents
1	CPU BUSY	% CPU BUSY	% CPU BUSY
2	CPU QUEUE LENGTH	# PCBs ready/ Table 6-1	# PCBs ready
3	PAGE FAULT RATE	# per second/ Table 6-1	# per second
4	MEMORY QUEUE LENGTH	Memory manager page faults queued/ Table 6-1	# reg queued
5	DISPATCH RATE	See Table 6-1	
6	SEND BUSY	IPB send busy percentage	% IPB busy
7	CACHE HIT RATE	# per second/ Table 6-1	# per second
8	DISC I/O RATE	See Table 6-1	# per second
9	PROCESS CONTROL BLOCKS (PCBs)	# in use/ # sysgened	# in use
10	LOCKED MEMORY PAGES	Locked pages/ Total memory pages	# locked pages
11	SYSPOOL	Words in use/ Sysgened	Words in use
12	MAPPOOL	Words in use/ Sysgened	Words in use
13	LINK CONTROL BLOCKS (LCBs)	# in use/ # sysgened	# in use
14	TIME LIST ELEMENTS	# in use/ # sysgened	# in use
15	BREAKPOINTS	# in use/ # sysgened	# in use

C ViewSys Help Screen

Figure C-1. The ViewSys Help Screen



```
                           VIEWSYS - T9071C30 - (31Mar91)
 FK1              --  CPU BUSY          FK2            -- CPU QUEUE LENGTH
 FK3              --  PAGE FAULT RATE   FK4            -- MEMORY QUEUE LENGTH
 FK5              --  DISPATCH RATE     FK6            -- SEND BUSY
 FK7              --  CACHE HIT RATE    FK8            -- DISC I/O RATE
 FK9              --  PCBs              FK10           -- LOCKED MEMORY
 FK11 ¦ Alt-F1    --  SYSPOOL           FK12 ¦ Alt-F2 -- MAPPOOL
 FK13 ¦ Alt-F3    --  LCBs              FK14 ¦ Alt-F4 -- TLEs
 FK15 ¦ Alt-F5    --  BPTs              FK16 ¦ Alt-F6 -- EXIT
  shifted FK1-16 | (Alt-F1-6)    ----      CPU 0, CPU 1, .........., CPU 15
 ROLL UP          ¦ Alt-up arrow          Display all measured processors
 SHFT ROLL UP     ¦ Alt-Shift-up arrow    Display HELP information
 ROLL DOWN        ¦ Alt-down arrow        Display MAXIMUM use statistics
 SHFT ROLL DOWN ¦ Alt-Shift-down arrow    Display CURRENT use statistics
 NEXT PAGE ¦ Alt-Pg Dn -- toggle rotation Shifted      -- toggle tiers
 PREV PAGE ¦ Alt-Pg Up -- next cpu set    Shifted      -- toggle rotatecpus
 INS LINE  ¦ Ctrl-Ins  -- reset maximums  DEL LINE ¦ Ctrl-Del -- EXIT

                              VIEWSYS STATUS
   cpus being measured:   0,1,2,3,4,5,6,7,8,9
   processor statistics:   current in use
   options: rotation off, cpusets 2, rotatecpus on
   options: unstoppable off, functionkeys on
   options: switchback off, tiers 1, multitype off
```

VST999.vsd

Table C-1. Numbered Function Keys

Function Key	Function	Function Key	Function
F1	CPU BUSY	SF1	CPU0
F2	CPU QUEUE LENGTH	SF2	CPU1
F3	PAGE FAULT RATE	SF3	CPU2
F4	MEMORY QUEUE LENGTH	SF4	CPU3
F5	DISPATCH RATE	SF5	CPU4
F6	SEND BUSY	SF6	CPU5
F7	CACHE HIT RATE	SF7	CPU6
F8	DISC I/O RATE	SF8	CPU7
F9	PCBs	SF9	CPU8
F10	LOCKED MEMORY	SF10	CPU9
F11 or ALT-F1	SYSPOOL	SF11	CPU10
F12 or ALT-F2	MAPPOOL	SF12	CPU11
F13 or ALT-F3	LCBs	SF13	CPU12

Table C-1. Numbered Function Keys

Function Key	Function	Function Key	Function
F14 or ALT-F4	TLEs	SF14	CPU13
F15 or ALT-F5	BPTs	SF15	CPU14
F16 or ALT-F6	EXIT	SF16	CPU15

Table C-2. Unnumbered Function Keys

Function Key	Function
ROLL UP	Display all measured processors
Shift ROLL UP	Display HELP information
ROLL DOWN	Display MAXIMUM use statistics
Shift ROLL DOWN	Display CURRENT use statistics
NEXT PAGE	Toggle rotation
Shift NEXT PAGE	Toggle tiers
PREV. PAGE	Next CPU set
Shift PREV. PAGE	Toggle rotatecpus
INS LINE	Reset maximums
DEL LINE	Exit

ViewSys Status

ViewSys shows which CPUs are being measured, indicates whether ViewSys is running in Current or Maximum mode, and gives the status of the ViewSys options listed in Section 3, ViewSys Run Command Syntax.

Example

CPUs being measured:	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Processor statistics	Current in use
Options:	rotation off, cpusets 2, rotatecpus on
Options:	unstoppable off, functionkeys on
Options:	switchback off, tiers 1, multitype off

D

ViewSys Definition Language (VDL)

ViewSys provides the ViewSys Definition Language (VDL), which is used to define the maximum values for system metrics. ViewSys uses maximum values to determine the graph maximum for all displays of rate and length values; for example, the disk I/O rate or the CPU queue length. This approach reduces the likelihood that new software product (SPR) versions of ViewSys will be needed whenever a new NonStop server becomes available. ViewSys users can simply download the latest ViewSysC file to the same subvolume as the ViewSys object file, stop ViewSys and then restart it. For the latest ViewSysC file:

http://www.NonstopViewSys.com

VDL recognizes these system metric keywords:

Metric Name	**Description**
BREAKPOINT	Number of allocated break points
CACHEHIT	Disk cache hit rate for this CPU in hits per second
CPUQUEUE	Average CPU queue length
CPUBUSY	Percent CPU busy
DISPATCH	Average number of process dispatches per second
DISKIO	Disk I/O rate for this CPU per second
MEMLOCKED	Number of physical memory pages currently locked
MEMQUEUE	Average memory manager queue length
MEMPRESSURE	Memory pressure rate
PAGEFAULT	Page fault rate
PCBLOW	The number of PCBs allocated in low pin process
PCBHIGH	The number of PCBs allocated in high pin process
SENDBUSY	The percentage of time a send was being performed
PTLE	The number of time-list elements in use
TLE	The number of time-list elements allocated

ViewSysC file

The ViewSysC file contains various statements that provide ViewSys with information about CPU types and metric maximum values.

The two most common statements are DEFINE and MAXIMUM.

The syntax for the configuration file is keyword driven. Metric keywords in a MAXIMUM statement are optionally comma separated. All statements must be terminated with a semicolon. Statements can span multiple lines.

The ViewSysC file contains a DEFINE statement for each possible CPU type. DEFINES must appear at the beginning of the file preceding any statements that reference DEFINES.

DEFINE Statement

The DEFINE statement allows the definition of easily remembered processor symbolic names to be substituted for the TYPE and SUBTYPE values of the processor in a MAXIMUM statement. The DEFINE string information is substituted whenever the defined keyword is found in a statement.

DEFINE must be declared in the ViewSysC file before they are used in a statement.

DEFINE Syntax

The syntax of the DEFINE statement is:

```
DEFINE keyword = string#;
DEFINE K10000 = CPUTYPE 7:2#;
```

keyword

Defines an alphanumeric symbolic name that is used to refer to the replacement text defined by *string*.

Notes

- DEFINES must be terminated by a # symbol.
- All statements contained in a ViewSysC file must be terminated by a semicolon.

MAXIMUM Statement

The maximum values configuration statement is called the MAXIMUM statement. The abbreviated form of the statement is MAX.

The MAXIMUM statement provides a mechanism for mapping CPU TYPES and SUBTYPES to predefined maximum values in order to normalize the graph maximum for displays of rate or scalar values.

The MAXIMUM statement can also be used to map user-selected maximums for individual CPU numbers.

A single MAXIMUM statement can span multiple lines.

The syntax of the Maximum statement is:

```
MAX [ CPUTYPE type:subtype ]  metric value [,] ...
    [ CPU     cpu-num        ]

Where:

cpunum    ::=   *   |   0, 1, 2...15
       *  ::=  Specifies all CPUs.
metric    ::=  A monitored performance metric such as: busy,
                diskio, pagefault,
value     ::=  CONFIG | number   DEFAULT
number    ::=  Any positive number.
```

CPUTYPE

Specifies the *type* and *subtype* of a processor.

type

Specifies the processor type in the range 0 through 9.

subtype

Specifies the processor subtype in the range 0 through 11.

CPU

Indicates maximum values are being specified by *cpu-num* instead of by CPUTYPE. The *cpu-num* provides an alternate way of specifying the MAX values for a processor. This form defines the maximum values for either a single CPU or, when the alternate form CPU * is used, the maximum values for all processors. *cpu-num* specifies the CPU number for the maximum values.

By using the MAX CPU *number* form of the MAXIMUM statement, you can customize the maximum normalization values for a node for each CPU based on empirical observation.

CPUTYPE or CPU are followed by *metric value* pairs.

metric

Specifies the monitored performance metric property.

Metric Name	**Default Value**	**Description** (page 1 of 2)
BREAKPOINT	CONFIG	Number of allocated break points
CACHEHIT	9000	Disk cache hit rate for this CPU in hits per second
CPUQUEUE	60	Average CPU queue length

Metric Name	Default Value	**Description** (page 2 of 2)
CPUBUSY	100	Percent CPU busy
DISPATCH	20000	Average number of process dispatches per second
DISKIO	1200	Disk I/O rate for this CPU per second
MEMLOCKED	CONFIG	Number of physical memory pages currently locked
MEMQUEUE	60	Average memory manager queue length
MEMPRESSURE	7	Memory pressure rate
PAGEFAULT	250	Page fault rate
PCBLOW	255	The number of PCBs allocated in low PIN process
PCBHIGH	CONFIG	The number of PCBs allocated in high PIN process
SENDBUSY	160	The percentage of time a send was being performed
PTLE	CONFIG	The number of time-list elements in use
TLE	CONFIG	The number of time-list elements allocated

value

Specifies the value of the metric property; for example, CPUQUEUE 10.

When *value* equals CONFIG, the MAX value for the metric property is the configured system value for that metric property.

When *value* equals DEFAULT, the MAX value for the metric property is the default value specified in the following table.

The following table lists the metrics that you can use with the CONFIG value:

Metrics for CONFIG Option	Description
BREAKPOINT	Number of allocated break points
CPUBUSY	Percent CPU busy
MEMLOCKED	Number of physical memory pages currently locked
PCBHigh	Number of PCBs allocated in high PIN process
PCBLow	Number of PCBs allocated in low PIN process
PTLE	Number of time-list elements in use
TLE	Number of time-list elements allocated

Considerations

Default values from the preceding table are used when the CPU *cpu-num* has been specified for any metric property not defined, unless the CPU TYPE and SUBTYPE of *cpu-num* were previously defined using the MAXIMUM CPU TYPE statement. In that case, the maximum values from that statement are used. The default values from the preceding table are used only when the CPU TYPE and SUBTYPE are not known.

VDL Examples

```
DEFINE K10000 = CPUTYPE 7:2 #;
DEFINE DEFAULTS = CPUTYPE *.*#;
MAX K10000 CACHEHIT 2000,
   CPUQUEUE 25,
   DISKIO 300,
   DISPATCHES 1000,
   SENDBUSY 175,
   MEMQUEUE 20,
   SENDBUSY 165,
   PAGEFAULTS 100;
```

This example instructs ViewSys to use the maximum metric values from the MAXIMUM statement K1000 for a CPU TYPE 7, SUBTYPE 2 configuration.

Example of ViewSys Configuration File

This ViewSysC Maximum Values Configuration file was last modified December 1, 2002.

```
DEFINE DEFAULTS = CPUTYPE *.*#;
MAX  DEFAULTS,
   PAGEFAULTS 225,
   CACHEHITS 9000,
   CPUQUEUE 60,
   MEMQUEUE 60,
   DISPATCH 20000,
   SENDBUSY 160
   DISKIO 1200,
   PCBLOW 255,
   BREAKPOINT 25;
DEFINE NonStopII  = CPUTYPE 1:0
  .
  .
DEFINE S86000    =  CPUTYPE 9:4#;
DEFINE busy      =  CPUBUSY 50#;
MAX  NONSTOPII,
   PAGEFAULTS 126,
   CACHEHITS 2000,
   CPUQUEUE 20,
   MEMQUEUE 20,
   DISPATCH 11000,
   SENDBUSY 160
   DISKIO 250,
   PCBLOW CONFIG,
   BREAKPOINT CONFIG;
MAX  S86000,
PAGEFAULTS 192,
   CACHEHITS 7000,
   CPUQUEUE 50,
   MEMQUEUE 50,
   DISPATCH 16000,
   SENDBUSY 160
   DISKIO 1000,
   PCBLOW CONFIG,
   BREAKPOINT CONFIG;
MAX CPU 2, PAGEFAULTS 160, busy, DISKIO DEFAULT, PCBLOW CONFIG,
CPUQUEUE 15;
```

The preceding example uses the factory settings based on the sysgen configured Cpu Type/Subtype for the node, except CPU 2 an S86000 processor will use the values PageFaults 160, busy 50, Diskio 1200, PCBHIGH will use the sysgen configured values for PCBLow values. When a metric does not have a sysgened configured value the default for that metric will be used, in this case 255 is used, CPUQUEUE will use a value of 15 as the maximum metric instead of 40. All the other metrics that are not defined in the Maximum CPU 2 statement will use the undefined metric values from the Maximum statement S86000 PAGEFAULTS 192, CACHEHITS 7000, MEMQUEUE 50, DISPATCH 16000, SENDBUSY 160, BREAKPOINT 25.

Index