TRACK for z/VM

User's Guide

BETA VERSION

Version 1.0 - February 2005
Revision 2005-03-07 9:33:00 AM

Table of Contents

General Information

If you want information on a specific TRACK command or panel while on CMS, try entering 'HELP TRACK panel' or 'HELP TRACK command'. For a menu of available TRACK panels and commands, try entering HELP TRACK MENU.

*** * * IMPORTANT note for previous TRACK users * * ***

> A few of the features of TRACK that used to be available, such as the TRT (third-level storage read) and the ability to read from a DUMP file have been removed. These functions have not been maintained so were removed. There may still be references to these features within HELP files or the code, so keep this in mind if you stumble upon them.

Use the TRACK routine to examine one or more virtual machines as they execute via a full-screen program. TRACK is a module that runs in the CMS user area.

It requires Class E CP privileges (for DIAG X'0004'). Class A CP privileges to issue CP LOCK and UNLOCK commands and CLASS C or E for the CP LOCATE command are highly recommended but not absolutely necessary. Without them, virtual storage and other non resident pages may well be inaccessible.

TRACK ***NEVER*** modifies CP in any way. In other words, TRACK *looks* but does not *touch* or *change* your system. If pages are LOCKed by TRACK, they are quickly UNLOCKed once the storage is read.

Be sure to read the *Copyright, Support and Contact Information* section for important information.

Installing TRACK

See the README FIRST and TRACK MEMO files for details on building/installing TRACK on your system.

Invoking TRACK

See the section *TRACK command – External* for details on how to begin the TRACK utility from CMS.

Examples of using TRACK

Because certain panels are invoked via PF keys, rather than commands, you can use the panel names as pseudo-commands, as follows:

- TRACK userid VMD
  this is the default VMD panel

- TRACK userid DEV
  enters TRACK in DEV panel

- TRACK userid DBK address
  enters TRACK in DBK panel

- TRACK userid VST address
  enters TRACK in VST panel

- TRACK userid RST address
  enters TRACK in RST panel

Using the existing TRACK internal commands, you could do the following:

- TRACK userid RET OFF
  enters TRACK with RETrieve buffer OFF

- TRACK userid AUTO 5
  set auto-update interval

- TRACK userid T userid
  TRACK two (or more) users immediately

A common use of the initial command is to TRACK a user with an immediate display of the spooled console, e.g.:

- TRACK userid CON

You can also specify a panel name AND a command for that panel, for example:

- TRACK userid DEV CLASS DASD TYPE 3390

The above example enters TRACK with the DEV panel displaying only 3390's

- TRACK userid CMS SVC FILE

The above example enters TRACK with the CMS panel displaying the SVC and FILE types.

You can enter TRACK in AUTO-UPDATE mode by appending the word AUP to the TRACK command, as follows:

- TRACK userid AUP

Virtual Machine Hung Example

If a virtual machine is hung, it often has pending interrupts on virtual devices. You can enter TRACK with an immediate display of devices with pending interrupts:

- TRACK userid DEV CLASS * IO PEND

Additional Examples

More examples of entering the TRACK command from CMS:

```
track operator
track *
track system
track x'02D42'    * This will be changed to address 02D42000
track operator console
track operator cms all
```

TRACK tries to do a hash table lookup for the specified user, if it can. If it can't or fails, it searches the entire VMDBK chain. If the VMDBK cannot be found, TRACK issues an error message to that effect and exits. Otherwise, TRACK shows you the VMD information panel and waits for you to enter a command (or executes the initial command, if one was provided).

Panels And Commands

TRACK shows you information about the user in one of *many* panels. Each panel shows a different kind of information. You move between the panels via the PF keys or commands, as follows:

PFkey or Command	Action	Panel
PF9/21 or VMD cmd	Displays CMS and VMDBK fields	VMD
Names command	Displays ALL LOGGED-ON USERS	NAM
PF1 or PF13	Displays REAL STORAGE	RST
PF10 or PF22	Displays VIRTUAL STORAGE	VST
PFX command	Displays REAL and ABSOLUTE PFX(s)	PFX
PF11 or PF23	Displays ALL VIRTUAL DEVICES	DEV
PF12 or PF24	Displays DEVICE BLOCKS	DBK
CONsole command	Displays lines from SPOOLED CONSOLE	CON
PRT command	Displays lines from SPOOLED PRINTER	PRT
PUN command	Displays lines from SPOOLED PUNCH	PUN
RDR command	Displays lines from SPOOLED READER	RDR
CMS command	Displays CMS panel	CMS
FORMat command	Displays FORMATTED DSECTs	FRM
IUCv command	Displays IUCV/APPC connections	IUC

You can go FROM any panel TO any other panel via these PF keys or commands. Initially you are shown the VMD panel. Within any panel, you can take a variety of actions via the ENTER key, CLEAR key, other PF keys and other commands. PF keys 13 to 24 can be used as alternates to PF keys 1 to 12.

Updating A Panel - The ENTER Key

Pressing the ENTER key from any panel causes TRACK to re-read and re-display the information on the current panel.

Simply KEEP PRESSING ENTER to update any panel display.

TRACK can also automatically update the display at fixed time intervals - see the discussion of AUTO-MODE later in this document.

If you type something in the command entry field and press ENTER, TRACK will attempt to execute what you typed in as a command. Twenty three commands are recognized, as follows:

<table>
<tr><th>Command</th><th>Arguments</th><th>Action</th></tr>
<tr><td>CMS</td><td>&lt;ALL|*|types ...&gt;</td><td>Displays the CMS panel</td></tr>
<tr><td>Console</td><td>&lt;none&gt;</td><td>Displays the CON panel (spooled console)</td></tr>
<tr><td>FORMat</td><td>block addr file</td><td>Displays the FRM panel (control blocks)</td></tr>
<tr><td>Names</td><td>&lt;none&gt;</td><td>Displays the NAM panel (all users)</td></tr>
<tr><td>PRT</td><td>&lt;address|N|F&gt;</td><td>Displays the PRT panel (any spooled printer)</td></tr>
<tr><td>PFX</td><td>&lt;1-6|I|M|A|0&gt;</td><td>Displays the PFX panel (selected low core)</td></tr>
<tr><td>PUN</td><td>&lt;address|N|F&gt;</td><td>Displays the PUN panel (any spooled punch)</td></tr>
<tr><td>RDR</td><td>&lt;address|N|F&gt;</td><td>Displays the RDR panel (any spooled reader)</td></tr>
<tr><td>IUCv</td><td>&lt;cp_service&gt;</td><td>Displays the IUC panel (IUCV connections)</td></tr>
<tr><td>VMD</td><td>&lt;none&gt;</td><td>Displays the VMD panel</td></tr>
<tr><td>?</td><td>&lt;none&gt;</td><td>Re-displays the last command</td></tr>
<tr><td>=</td><td>&lt;none&gt;</td><td>Re-executes the last command</td></tr>
<tr><td>Auto</td><td>&lt;? | nn.nnn <nn&gt;&gt;</td><td>Re-sets the auto-update interval/limit</td></tr>
<tr><td>Bits</td><td>&lt;?|ON|OFF|ALL&gt;</td><td>Changes format of flags displays</td></tr>
<tr><td>CLASs</td><td>class &lt;class ...&gt;</td><td>Restricts DEV panel by CLASS/TYPE/IO</td></tr>
<tr><td>DROPfrm</td><td>label &lt;label&gt;</td><td>Drops selected lines from FRM display</td></tr>
<tr><td>Dump</td><td>&lt;start <end&gt;&gt;</td><td>Takes a dump of the tracked virtual machine</td></tr>
<tr><td>Help</td><td>&lt;anything&gt;</td><td>Invokes native CMS help system</td></tr>
<tr><td>HIDEfrm</td><td>label &lt;label&gt; |OFF</td><td>Hides selected lines from FRM display</td></tr>
<tr><td>MACRO</td><td>macroname</td><td>Invokes macroname TRACK as a TRACK macro</td></tr>
<tr><td>Quit</td><td>&lt;userid&gt;</td><td>Drops a tracked user</td></tr>
<tr><td>RETrieve</td><td>&lt;ON|OFF|?&gt;</td><td>Toggles display of RETRIEVE buffer</td></tr>
<tr><td>SAVEfrm</td><td>&lt;none&gt;</td><td>Save a formatted control block</td></tr>
<tr><td>Track</td><td>&lt;?|uid <cmd> <(NEW&gt;&gt;</td><td>Tracks another user</td></tr>
</table>

If the command you enter does not match one of the above entries, TRACK checks to see if the command is the file-name of a file with a file-type of TRACK. If so, the EXEC interpreter is invoked to execute the file. If not, the command is passed to CMS for execution as a SUBSET command.

Getting To Panels - PF Keys and Commands

You get to a specific panel via a PF key or a command. When you TRACK a user, you are initially shown the VMD panel. The PF keys and commands that get you to a specific panel are documented below.

VMDBK/CMS Display - VMD Panel - PF9

Syntax: Hit PF9 from any panel

The VMD panel shows information from the virtual machine's VMDBK and low core CMS areas (if the VM is not running CMS, the low core information will probably be meaningless). Hitting PF7 or PF8 in the VMD panel will shift you to the previous or next user in the CP VMDBK chain (cyclic).

See the VMD section or the helpfile (HELP TRACK VMD) for more information on the VMD panel.

CMS display - CMS panel - CMS command

Syntax: Enter CMS <ALL|*|types> from any panel

The CMS panel shows information relevant to a virtual machine running CMS. The panel can display a variable amount of information, depending on what you type after the word 'CMS'.

See the CMS section or the helpfile (HELP TRACK CMS) for details on this panel and the CMS command.

Virtual Storage display - VST panel - PF10

Syntax: Type or place cursor under a virtual address from any panel, hit PF10. Just hit PF10 to cycle to previous VST displays.

Note: TRACK imbeds addresses in unprotected fields. You can often place your cursor under an address simply by hitting the TAB key a few times.

The VST panel displays a virtual machine's storage. You can type or place your cursor under a virtual address before hitting PF10. Subsequently, pressing PF10 without typing anything (and without putting your cursor under an address) will take you back to the previous VST panels. Note that you can type a new address (or place your cursor under a new address) and press PF10 from any panel - this will take you to the new address.

You can type an address prefixed with a + or - sign to scroll the storage display forward or back (e.g., typing +10 and pressing PF10 will shift the display forward X'10' bytes). PF7 and PF8 scroll the display backward and forward a panel at a time. PF6 will display the storage in ASSEMBLER format (hit PF6 again to return to standard format).

See the VST section or the helpfile (HELP TRACK VST) for more information on the VST panel.

Real Storage display - RST panel - PF1

Syntax: Type or place cursor under a real address from any panel, hit PF1. Just hit PF1 to cycle to any previous RST display.

Note: TRACK imbeds addresses in unprotected fields. You can often place your cursor under an address simply by hitting the TAB key a few times.

PF1 takes you to the RST panel. It works like PF10. Before using it for the first time, type a storage address on the command line (or place your cursor under a real address) and press PF1. Subsequently, pressing PF1 without entering anything will take you back to previous RST displays. You can enter a new address (or position your cursor under a new address) from within any panel and press PF1 to move the RST display to that address. You can also prefix an address with a + or - sign to move the display forwards and back. As with the VST display, you can use PF7 and PF8 to scroll forwards and back a screen-full at a time, or PF6 to dis-assemble the displayed storage.

In addition to entering addresses, you can search for a hex or character string in real storage.

See the RST section or the helpfile (HELP TRACK RST) for more information on the RST panel.

Devices display - DEV panel - PF11

Syntax: Hit PF11 from ANY panel.

PF11 takes you to the DEV panel. The ADDRESS, CLASS and TYPE of all devices attached to the virtual machine are displayed (including LOGICAL DEVICES). If the devices will not fit on one panel, a MORE message is displayed. Press ENTER to scroll the display. (If you want to prevent the DEV panel from scrolling, press the PF11 key). Devices with pending interrupts are shown hilighted.

From the DEV panel, you can use PF6, PF7 and PF8 to sort the device display by CLASS, ADDRESS and TYPE, respectively, and you can restrict the display on the basis of device CLASS, TYPE and/or IO activity via the CLASS command.

See the DEV helpfile (HELP TRACK DEV) for more information on the DEV panel and the CLASS command.

Device Blocks display - DBK panel - PF12

Syntax: Type or place cursor under a device address from any panel, hit PF12. Just hit PF12 to cycle back to any previous DBK display.

Note: TRACK imbeds addresses in unprotected fields. You can often place your cursor under an address simply by hitting the TAB key a few times.

PF12 takes you to the DBK panel. PF12 works like PF10. Before using it for the first time, type a device address on the command line (or place your cursor under a device address from any panel) and press PF12. Subsequently, pressing PF12 without typing anything on the command line will take you back to previous DBK panels. You can always re-type a new address on the command line (or place your cursor under a device address) and press PF12 to display information on the new device.

You can use PF7 and PF8 to skip to the PREVIOUS or NEXT virtual device in the user's device chains. If you have used the CLASS command from the DEV panel, PF7 and PF8 will respect the current CLASS/TYPE restrictions.

See the DBK helpfile (HELP TRACK DBK) for more information on the DBK panel and the CLASS command.

Console display - CON panel - The CONSOLE command

Syntax: Type 'C' and hit ENTER from ANY panel.

Entering "Console" will show you the CON panel, which contains lines from the tracked user's spooled console buffer. If the console is not spooled, an error message is displayed. If the console is spooled, the panel shows lines in the current in-core CP console buffer. You can scroll back and forth through the console buffer via PF7 and PF8. Each time you hit the ENTER key, the entire console buffer is re-read. Thus, if CP has just written the buffer out to DASD, the display may jump from showing many lines to showing just a few.

See the CON helpfile (HELP TRACK CON) for more information on the CONsole panel.

Unit-Record displays - PRT/PUN/RDR panels & commands

Syntax: Type 'RDR', 'PRT' or 'PUN', followed by nothing, or by a device address, or by 'N' or 'F', then press ENTER.

These commands display lines from an active spooled unit-record device. You can enter PRT|PUN|RDR by itself, in which case TRACK locates the first ACTIVE unit-record device (at the lowest address). If you provide an address (PRT|PUN|RDR address), that specific unit-record device is displayed. You can enter 'Next' or 'First', rather than an address, to display the next or the first unit-record device. PF7/8 scroll you through the display.

See the PRT, RDR and PUN sections or the helpfiles (HELP TRACK RDR or PRT or PUN) for more information on unit-record displays.

Userids display - NAM panel - The NAMES command

Syntax: Type 'N', press ENTER from ANY panel.

Entering "Names" (you can abbreviate it to "N") shows you the NAM panel, which lists all users and their VMDBK addresses. Initially, the display is shown sorted by USERID. PF7 sorts the display by VMDBK address, PF6 sorts it by USERID. You can place your cursor under a USERID and hit PF8 -- this is equivalent to typing "Track userid" on the command line. If all of the names do not fit on your screen, TRACK puts up a "MORE" message - hit ENTER to scroll the display. If you want to prevent the display from scrolling when the MORE message is up, keep entering "Names".

See the NAMES section or the helpfile (HELP TRACK NAMES) for more information on the NAMES panel

PFX display - PFX panel - PFX command

Syntax: Type 'PFX', 'PFX I', 'PFX M', 'PFX A', 'PFX 0' or 'PFX n' where n is either 1|2|3|4|5|6', press ENTER from ANY panel.

Entering "Pfx" (you can abbreviate it to "P") shows you the PFX panel, which contains selected low-core fields. You can optionally provide a CPUID (1-6) if you are executing on a multi-processor. If you just enter "PFX," or if CP is UP generated, you will get IPL proc page 0. Entering "PFX n", where n = 1 thru 6, will get you a specific PFX. Entering "PFX I" will get you the IPL processor prefix page. Entering "PFX M" will get you the Master processor prefix page. Entering "PFX A" will get you the (or an) Alternate processor prefix page. Entering "PFX 0" will show you the storage at the address identified as the processor's prefix value. Note that on VM/XA or ESA systems, absolute page 0 (this page) contains very little useful information, at least as far as I (RJF) can tell. PF7 and PF8 scroll you through the PFX display a screen-full at a time. PF6 advances the display line by line. If you just press PF6, the display scrolls down 1 line. You can also type a number and press PF6 to scroll down that many lines - if you prefix the number with a "-" sign, the display scrolls up that many lines.

See the PFX section or the helpfile (HELP TRACK PFX) for more information on the PFX panel.

Formatted Control-Block display - FRM panel - FORMAT command

Syntax: See FORMAT section or HELP TRACK FORMAT

FORMat is an experimental facility. It attempts to mimic the ANALYZE FORMAT command, but has less capability. FORMAT dynamically reads control blocks (DSECTs) from maclibs, attempts to assemble these control blocks, and displays real, virtual or third-level storage as per the control block.

See the FORMAT section or the helpfile (HELP TRACK FORMAT) for more information on the FRM panel and the FORMAT command.

General Commands and PF keys

In addition to the commands and PF keys that get you into panels, TRACK supports a number of commands and PF keys that are valid from ANY panel. These are discussed below.

Tracking Multiple Users - The TRACK command

```
Syntax: Type 'Track'                               -or-

             'Track ?'                             -or-

             'Track userid <command> <(NEW>'

        Hit ENTER from ANY panel

Examples: Track operator
          Track pvm (new
          Track pvm format nucon v0
          Track pvm format nucon v0 (new
          Track pvm con (new
          Track pvm ret off
          Track pvm track rscs
          Track pvm track rscs c
          Track * (new
```

You can track up to 8 users at a time. This command takes 3 forms.

"Track" by itself will rotate you through the tracked users.

"Track ?" will display the list of user's being tracked on the message line.

"Track userid" will start tracking a new user, or shift to that user if he/she is already being tracked. You can append an initial command after the "userid." For example, "Track operator console", or "Track pvm dev".

You can track the same user multiple times by appending the option 'NEW' to your track command. For example, if you are TRACKing the operator, you can enter 'TRACK OPERATOR (NEW', and you will now have two TRACK sessions, each for OPERATOR.

You can append an initial command before the NEW option -- e.g., "Track operator format nucon v0 (new", or "Track rscs con (new".

See the TRACK helpfile (HELP TRACK TRACK) for more information on the internal TRACK command.

Quitting - The QUIT and DROP commands

Syntax: Type 'Quit <userid>' -or- 'Drop <userid>', hit ENTER from ANY panel.

If you are tracking more than one user, you can issue the Quit or Drop command. This will drop the current user from the list of tracked users. If you are only tracking one user, Quit/Drop has the same effect as PF3 (exits TRACK). You can also enter "Quit userid" or "Drop userid" to drop any of the users you are tracking.

HELP - The HELP command

Syntax: Type 'Help' or 'Help arg(s)', hit ENTER from ANY panel.

Entering Help (or 'H') invokes the CMS help facility. If you type HELP by itself, TRACK invokes the help facility with a command of the form:

```
HELP TRACK panel
```

where 'panel' is the panel name of the current panel. If you enter HELP followed by a word, TRACK looks up the word to see if it matches a TRACK panel name or command -- if so, it invokes the help facility with a command of the form:

```
HELP TRACK word
```

Finally, if the word does not match something that is known to TRACK, the help facility is invoked with the command as entered.

CMS subset - CLEAR key

Syntax: Hit CLEAR key from ANY panel.

The CLEAR key causes TRACK to clear your screen and put you into the CMS SUBSET environment. Enter RETURN to get back into TRACK. CLEAR does not cause TRACK to update the currently displayed panel. Press ENTER again to have the panel updated.

Auto-Update mode - PF2 key

Syntax: Hit PF2 key from ANY panel.

PF2 causes TRACK to go into auto-update mode. The panel you are being shown is updated every 2 seconds. This allows you to watch a virtual machine as it executes. PF2, like ENTER, is valid from any panel. Press the ENTER or any PF key to get out of auto-update mode. You can reset the auto-update interval to any value from 1 to 99.999 seconds via the AUTO command (see below).

The AUTO section and helpfile (HELP TRACK AUTO) contains detailed information about auto-update mode and the AUTO command.

Auto-Update interval - The AUTO command

Syntax: Type 'Auto <? | nn.nnn <nn>>', hit ENTER from ANY panel.

The Auto command allows you to query (via "Auto", or "Auto ?") or set (via "Auto nn") the auto-update interval from 1 to 99.999 seconds. The interval is initially set to 2 seconds. You can specify an auto-update limit as well as an interval, in which case auto-updating terminates when the limit is reached (or you press ENTER).

See the AUTO section or the helpfile (HELP TRACK AUTO) for more information about auto-update mode.

EXIT - PF3 key

Syntax: Hit PF3 from ANY panel.

This key ends execution of TRACK, returning you to the environment from which you invoked the TRACK module.

PRINT - PF4 key

Syntax: Hit PF4 from ANY panel.

This key causes TRACK to write the currently displayed panel to your spooled printer. TRACK does not close your printer after the write, so successive PF4's are appended to your printer. Close your printer from within TRACK, or after exiting the module, to have the file sent to the printer's spooled destination.

FILE - PF5 key

Syntax: Hit PF5 from ANY panel.

This key causes TRACK to write the currently displayed panel to the file "userid TRACKFIL A". This write uses default record number values, so that if the file does not exist, CMS creates it, while if it does exist, the information is appended to the end of the file. PF5 does not cause TRACK to update the information on the panel. The return code from the write is shown in the message area.

Flag Bits - The BITS command

Syntax: Type 'BITS <ON|OFF|ALL|?>', hit ENTER from VMD, DBK or CMS panel.

TRACK attempts to interpret flag bits on the VMD, DBK and CMS panels. You can control the format of this display via the BITS command. BITS ON (the default) shows only the bits that are ON in a flag. BITS ALL shows all flag bits – the ones that are ON are shown in upper-case, the ones that are OFF in lower-case. BITS OFF shows the flags as a HEX value. BITS or BITS ? shows the current BITS setting.

See the BITS section or the helpfile (HELP TRACK BITS) for more information on the BITS command.

Dumping a TRACKed machine - The DUMP command

Syntax: Type 'DUMP <start-adr <end-adr>>', hit ENTER from ANY panel.

The Dump command causes TRACK to dump the tracked user's virtual storage to your spooled printer (DCSS storage is NOT dumped). This dump is in EBCDIC format. Each line of the dump shows 8 words (32 bytes) of storage in both EBCDIC and HEX form. The dump is taken in 64K chunks. You can interrupt this command by pressing ENTER - this will abort the dump after the current 64K chunk. Dump does not close your printer. This allows you to append or prefix a dump with other TRACK screens.

You can specify a range on the dump command. 'DUMP' defaults to printing ALL of the tracked machine's storage. 'DUMP nnnnnnnn' takes a dump from address nnnnnnnn to the end of virtual storage, while 'DUMP nnnnnnnn nnnnnnnn' takes a dump from the first nnnnnnnn to the last. The DUMP command adjusts each range value to the next lower 64k boundary. Each range value must be a valid hex address.

ASCII display - The ASCII command

Syntax: Type 'ASCII <ON|OFF|?>', hit ENTER from ANY panel.

The "ASCII" command toggles the display of the characters on the RST and VST panels. Entering "ASCII" by itself, or "ASCII ?", will display the current setting. "ASCII ON|OFF" toggles the display. Note that the ASCII setting is applicable individually for each tracked user -- you must separately turn it on or off for each tracked user. By default, TRACK sets ASCII OFF for the first and each subsequent tracked user.

Retrieve Buffer display - The RETRIEVE command

Syntax: Type 'RETrieve <ON|OFF|?>', hit ENTER from ANY panel.

The "RETrieve" command toggles the display of the RETRIEVE buffer on the VMD panel. Entering "RETrieve" by itself, or "RETrieve ?", will display the current setting. "RETrieve ON|OFF" toggles the display. Note that the retrieve setting is applicable individually for each tracked user -- you must separately turn it on or off for each tracked user. By default, TRACK sets RETrieve ON for the first and each subsequent tracked user.

Notes on CP facilities

TRACK uses the following CP facilities:

DIAG code 0:	See extended identification code (uid, system, GMT)
DIAG code 4:	To read CP storage
DIAG code C:	Time function
DIAG code 24:	Get device (console) info
DIAG code 58:	Screen support
DIAG code 8C:	Get device (console) info
DISPLAY command:	To display regs & storage when op. exception in TRK004
LOCATE command:	To find CP code and force pages resident via DIAG 8, for CON, RDR, PRT, PUN displays
QUERY command:	Various uses via DIAG code 8
LOCK command:	For VMD, VST, CMS, FRM displays via DIAG code 8, as well as for the CON, DUMP, PRT, PUN & RDR commands
UNLOCK command:	For VMD, VST, CMS, FRM displays via DIAG code 8, as well as for the CON, DUMP, PRT, PUN & RDR commands
Clock-Comparator:	For auto-update function

TRACK does not attempt to bypass the standard CP security measures in any way. It includes no 'tricky' code to get access to information that the user's privilege classes would not normally allow.

Usage Summary

Special commands:

?	Re-displays last cmd	**=**	Re-executes last command
Help	Invokes CMS help facility	**Auto**	Re-sets auto-update
Track	Tracks another user	**Quit|Drop**	Drops a tracked user
CMS	Displays the CMS panel	**BITS**	Controls flags display
Names	Displays the NAM panel	**Dump**	Dumps the tracked user
Console	Displays the CON panel	**Pfx**	Displays the PFX panel
RETrieve	Toggles RETRIEVE display	**IUCv**	Displays the IUC panel
FORmat	Displays the FRM panel	**HIDEfrm**	Hides lines on FRM panel
SAVEfrm	Saves the current DSECT	**DROPfrm**	Drops lines on FRM panel
PUN	Displays the PUN panel	**PRT**	Displays the PRT panel

CLASs	Restricts DEV display	**RDR**	Displays the RDR panel
MACRO	Invokes a TRACK macro	**VMD**	Displays the VMD panel
ASCII	Toggles ASCII display		

Keys

ENTER - updates the information currently being displayed, or executes a command.

CLEAR - puts you in the CMS subset environment. Enter RETURN to get back.

PF2 --- puts TRACK in auto-update mode. TRACK updates information on panel currently being displayed every 2 seconds. Press any AID key to get out of auto-update mode.

PF3 --- ends the program.

PF4 --- appends the panel to your spooled printer.

PF5 --- appends the panel to the file "userid TRACKFIL A".

PF9 --- displays the VMD panel (VMDBK and low-core fields).

Two special keys are available within the VMD panel:

PF7 - skips to the previous user in the VMDBK chain.
PF8 - skips to the next user in the VMDBK chain.

PF10 -- displays the VST panel. Type a storage address on the command line before pressing PF10 for the first time. After that, pressing PF10 with a blank command line takes you back to the last storage areas displayed -or- you can re-type a new address and press PF10 to see a new storage area. You can place your cursor under an address instead of typing it, or prefix addresses with a + or - sign.

Three special keys are available within the VST panel:

PF6 - disassembles the displayed storage.
Note: press PF10 to get back to standard storage display.
PF7 - scrolls the displayed storage back one screen.
PF8 - scrolls the displayed storage forward one screen.

PF11 -- displays the DEV panel (list of all devices). A MORE message is put up if the devices don't fit on the screen -- press ENTER to scroll the display, or press PF11 to stay at the top.

Three special keys are available within the DEV panel:

PF6 - sorts the display by CLASS.
PF7 - sorts the display by ADDRESS.
PF8 - sorts the display by TYPE.

The CLASS command can be used to restrict the display to specific device classes and/or types.

PF12 -- displays the DBK panel. Type a device address on the command line before pressing PF12 for the first time. After that, pressing PF12 with a blank command line takes you back to the last device blocks displayed -or- you can re-type a new device address and press PF12 to see a new device block. You can place your cursor under a device address instead of typing it.

Two special keys are available within the DBK panel:

PF7 - shows information about the previous virtual device.
PF8 - shows information about the next virtual device.

The PF7/8 keys respect the current class/type restrictions, if any. The CLASS command can be used to reset the current class/type restrictions.

PF1 -- displays the RST panel. Type a storage address on the command line before pressing PF1 for the first time. After that, pressing PF1 with a blank command line takes you back to the last storage areas displayed -or- you can re-type a new address and press PF1 to see a new storage area. You can place your cursor under an address instead of typing it, or prefix addresses with a + or - sign. You can also enter "/charater-string" of "Xhex-string" and press PF1 to search for a string in real storage.

Three special keys are available within the RST panel:

PF6 - disassembles the displayed storage.
Note: press PF1 to get back to standard storage display.
PF7 - scrolls the displayed storage back one screen.
PF8 - scrolls the displayed storage forward one screen.

CMS -- This command displays the CMS panel. Fields relevant to a virtual machine running CMS are displayed. You can restrict the display be specifying one or more "types" following the word "CMS".

Names - This command displays the NAMes panel. All users and their VMDBK addresses are displayed. A MORE message is displayed if the names do not fit on the screen -- press ENTER to scroll the display, or enter "Names" to stay at the top.

Three special keys are available within the NAM panel:

PF6 - sorts the display by USERID.
PF7 - sorts the display by ADDRESS.
PF8 - tracks the user at the current cursor position.

Pfx -- This command displays the PFX panel. On an MP/AP system, you may provide an argment (1-6 | A | M) to select a specific PFX - "Pfx" by itself shows the IPL processor prefix page, as does specifying "PFX I" (or "PFX M" on a UP system).

Three special keys are available within the PFX panel:

PF6 - scrolls the display forward and back by lines.
PF7 - scrolls the displayed storage back one screen.
PF8 - scrolls the displayed storage forward one screen.

Con -- This command displays the CON panel. Lines from the spooled console buffer, if any, are displayed, preceded by a line number.

Two special keys are available within the CON panel:

PF7 - scrolls the displayed buffer back one screen.
PF8 - scrolls the displayed buffer forward one screen.

PRT/PUN/RDR -- These commands display the PRT, PUN and RDR panels. Each line of the spool buffer is displayed, preceded by a line number. You can provide an address ('PRT address'), or use the words 'Next' or 'First' to display the next or first spooled device ('PRT Next'). If you just enter 'PRT|PUN|RDR', TRACK locates the first device.

Two special keys are available within the PRT|PUN|RDR panels:

PF7 - scrolls the displayed buffer back one screen.
PF8 - scrolls the displayed buffer forward one screen.

IUCv -- This command displays the IUC panel. This panel displays any IUCV/APPC connections for the userid. The SYSTEM userid will allow a filter keyword that matches the CP Service you want to see.

PF6 - scrolls the display forward and back by lines.
PF7 - scrolls the display back one screen.
PF8 - scrolls the display forward one screen.

Format- This command displays the FRM panel. This panel displays real or virtual storage formatted as per a specified DSECT.

Three special keys are available within the FRM panel:

PF6 - scrolls forward/back or searches for a string.
PF7 - scrolls the displayed storage back one screen.
PF8 - scrolls the displayed storage forward one screen.

Three special commands, HIDEfrm, SAVEfrm and DROPFRM, are availabel in the the FRM screen.

HIDEfrm hides selected lines from the FRM display.

DROPfrm drops selected lines from the FRM display.
SAVEfrm saves the formatted DSECT for latter quick recovery.

Help - This command invokes your native HELP facility.

Macro - This command invokes a TRACK macro.

Copyright, Support and Contact Information

TRACK is copyrighted ***and*** free. What this means is you can send it to anyone you'd like as long as the package is shipped in its entirety and not sold nor charged for in any way.

No warranty is expresses or implied by using TRACK. TRACK will not harm your system in any way, but there is no guarantee it will work for *your* system or that it won't product addressing/protection exceptions on the userid running it due to code issues.

Support is handled on a free-time basis and is not guaranteed. Bugs are usually handled quickly; enhancements are considered and added to "the list" for future inclusion.

NOTE: New enhancements are currently being added to z/VM 4+ in 64 bit mode (the TRACKZ64 and TRACKV51 levels) **ONLY**. Anyone wishing to take the source updates and apply them to the 31 bit level(s) of the code is welcome and encouraged to! I currently do not have access to a system that would allow me to build and test 31 bit updates.

USE THE SOURCE LUKE! The package contains all the source code to TRACK and you are welcome to add, change, or even delete code as you see fit. If you do something that you think would be useful to others, please send it on to me and I will package it in.

Current TRACK maintainer:

James Vincent (James.Vincent@nationwide.com)
Nationwide Insurance, One Nationwide Plaza, Columbus OH 43215
614-249-5547

The following are out-of-date contacts and only contained here for credits:

Serge Goldstein (SERGE@PUCC.PRINCETON.EDU)
Princeton Computing Center, 87 Prospect Ave., Princeton NJ 08544
609-258-6059

Ross Fried (RJFCU@CUNYVM.CUNY.EDU)
CUNY/CIS, 555 West 57 St., 16th floor, New York NY 10019
212-541-0987

Commands, PF keys and Panels Reference

ASCII - Command

Syntax

ASCII	[ON | OFF | ?]

The ASCII command toggles the display of the character output on the RST and VST panels. You may enter this command from any panel, but it will not take effect until you return to either the RST or VST panels.

If you enter ASCII by itself, or followed by a "?", the current status of the ASCII display will be shown. The status of the ASCII display is remembered by TRACK even if you leave and return to the RST or VST panel.

If you are tracking more than one user, the ASCII display status is separately maintained for each user.

AUTO - Command and Key

The AUTO-UPDATE function in TRACK automatically updates any TRACK panel. It simulates pressing of the ENTER key at specific time intervals.

You enter AUTO-UPDATE mode by pressing PF2 or PF14. Once in AUTO-UPDATE mode, the screen is updated at every auto-update interval. AUTO-UPDATE mode is terminated when you press an AID key, or, optionally, when the auto-update limit is reached.

You can set the AUTO-UPDATE interval, and an optional AUTO-UPDATE limit, via the AUTO command. These settings apply across all tracked users.

The AUTO-UPDATE interval is initially set at 2 seconds. You can reset it to any value from 0 to 99.999 seconds.

The AUTO-UPDATE limit is initially unset, meaning that auto-updates continue until you explicitly stop them by pressing an AID key. You can set a limit, in which case auto-updates will stop either when you press an AID key or when the limit is reached.

Syntax - AUTO command

AUTO	[nn.nnn | ? [nnnn]]

AUTO nn.nnn sets the auto-update interval to nn.nnn seconds, where 'nn.nnn' can be any number from 0 to 99.999. It also turns off the auto-update limit.

AUTO nn.nnn nnnn sets the auto-update interval and limit.

AUTO, or AUTO ?, displays the current settings.

```
For example: Auto 1      (auto-update interval=1      sec; limit = NONE)
             Auto 2 4    (auto-update interval=2      sec; limit = 4)
             Auto 14.215 (auto-update interval=14.215 sec; limit = NONE)
             Auto .3 10  (auto-update interval=.3     sec; limit = 10)
             Auto 0.5    (auto-update interval=0.5    sec; limit = NONE)
```

If you specify an auto-update interval of 0 seconds, and do not specify an auto-update limit, TRACK will set a limit of 10 seconds. This is a safety feature, as it is generaly not possible to interrupt continuous auto-updating from the terminal.

You can append an AUTO command to your initial, or internal, TRACK command.

For example, from CMS,   enter: Track pvm auto 3
For example, from TRACK, enter: Track rscs auto 1 20

Syntax - AUTO-UPDATE key

Hit PF2 or PF14 to enter AUTO-UPDATE mode.

Hit ENTER, or any PF key, to terminate AUTO-UPDATE mode. If you hit a PF key, AUTO-UPDATE mode is terminated, but the PF key action (if any) is not executed (for example, hitting PF3 in AUTO-UPDATE mode terminates auto-update, but does not exit TRACK; you have to hit PF3 again to exit TRACK).

If you have set an AUTO-UPDATE limit, AUTO-UPDATE mode will terminate when the limit is reached OR when you press an AID key. AUTO-UPDATE is also terminated if CP steals the screen, or if the tracked user logs off.

You can enter TRACK in AUTO-UPDATE mode by appending the word AUP to an initial or internal TRACK command.

For example, from CMS, enter: Track pvm aup

Use

See the FORMAT HELPTRAC file (enter HELP TRACK FORMAT) for special uses of the AUTO-UPDATE function in chaining through formatted control blocks.

Depending on system load, TRACK may or may not be able to update the screen within the specified auto-update interval. Thus the actual screen refreshes may take up to a second longer than the current auto-update interval setting.

BITS - Command

Syntax

BITS	[ON | OFF | ALL | ?]

You can append a BITS command to your initial, or internal, TRACK command.

For example, from CMS,   enter: Track pvm bits all
For example, from TRACK, enter: Track rscs bits off

Description

TRACK attempts to interpret flag bits on the VMD, DBK and CMS panels. You can control the format of this display via the BITS command.

BITS ON (the default) shows only the bits that are ON in a flag.

BITS ALL shows all flag bits -- the ones that are ON are shown in upper-case, the ones that are OFF in lower-case.

BITS OFF shows the flags as a HEX value.

BITS or BITS ? shows the current BITS setting. The default is BITS ON.

BITS ALL has limited utility - the screen is not always wide enough to show all of the interpreted flag bits. If this occurs, TRACK displays trailing dots at the end of the interpreted flags. BITS ALL may be especially useful in the CMS panel.

Use

BITS OFF may be useful if you are planning to modify CP real storage, and you need to see the FLAG value in HEX.

CLASS – DBK and DEV subcommand

The CLASS command may only be issued from the DEV and DBK panels.

See the DEV and DBK sections for complete information on these panels.

Syntax

CLASS	[* | class … [TYPE * | type … [IO * | <active> <pending> <idle>]]]

The CLASS command can be used to restrict which devices will be displayed on the DEV panel on the basis of device CLASS, device TYPE and device IO activity. For example:

```
CLASS term
CLASS dasd
CLASS dasd TYPE 3380 3390
CLASS spol dasd
CLASS term IO active
CLASS term dasd IO pending
CLASS term dasd IO active pending
CLASS * IO pending
CLASS * IO active
CLASS * IO active pending
CLASS * IO *
CLASS dasd term IO *
CLASS term term TYPE 3278 3215
CLASS tape TYPE 3420 3480 IO idle
```

Neither of the words CLASS, TYPE or IO need be entered in capitals. They are shown in capitals above for clarity only. This command is ONLY valid from the DEV or DBK panels, or as an optional suffix to the TRACK command (initial or internal).

A value of '*' for CLASS, TYPE or IO selects ALL of that CLASS/TYPE/IO. CLASS * TYPE * IO * displays all devices, and this is the initial setting. If you specify CLASS *, the TYPE and IO are reset to *, although you can override this reset by including a TYPE or IO specification in the same command, e.g. 'class * type 3390', or 'class * io pending'. Entering the command 'CLASS *' is a quick way to reset all class restrictions.

CLASS/TYPE/IO restrictions are remembered if you are tracking multiple users on a user-by-user basis. You can still use the SORT pf-keys after restricting the display via the CLASS command. You can also use the CLASS command following the word 'dev' on an initial or internal TRACK command - this causes the DEV display to initially show only the selected devices. For example:

```
TRACK pvm dev class term type 3278
```

TRACK pvm dev class * io pending

You can also type a CLASS command on the command line, then hit PF11. This will enter the DEV panel with only the specified devices displayed.

The following is a list of the classes and, for each class, the types currently recognized by TRACK:

```
CLASS   TYPEs
-----   -----
TERM    2700  3215   TTY  IBM1  2741  1050 UNDEF   BSC
GRAF    3277  3278 CLUST  3271 R3274  3275  3276  3287  3284  5080  2250
SPOL    RDR   2501 2540R  3505   PUN 2540P  3525   PRT  1403  3211  3203
        3800  3262  4245  4248  3820  AFP1   SYS
TAPE    3420  3430  3480  3422
DASD    3330  3340  3350 3350C 3350D  2305  3380  3390  3375  3370
SPEC    CTCA  3704  3851  3890 UNSUP
```

LDEVs are CLASS GRAF. TYPE and IO restrictions, if any, are ignored for LDEVs. Note that CLASS and TYPE specifications in CP are not always bit-significant. This means that certain class/type combinations may not be unique - thus, devices whose class/type do not exactly match a class/type specification may occasionally be displayed.

The IO operand allows you to select devices on the basis of io activity. You can specify one or more of the words 'active', 'pending' or 'idle' following the word IO. 'Active' selects devices which are active with an IO operation, or which have an IO operation queued (pending device activity). 'Pending' selects devices which have pending interrupts. 'Idle' selects devices which are neither active nor have pending interrupts. Finally, 'IO *' is equivalent to 'IO active pending idle', and selects devices regardless of IO activity.

Please note that IO is an operand of the CLASS command, and cannot be entered by itself. If you want to restrict the device display only on the basis of IO activity (ignoring CLASS/TYPE), precede your IO specification with 'CLASS *'. For example 'CLASS * IO pending'. You can also combine IO with other operands. For example, to see which DASD devices are currently active, you would enter 'CLASS DASD IO ACTIVE'.

TRACK checks the bits in the VDEVBLOK flag field VDEVSTAT to determine the device's IO status, as well as other VDEVBLOK fields. These are:

- ü VDEVAIOR and/or VDEVACTV bits in VDEVSTAT - Signals device ACTIVE
- ü VDEVNIOR and/or rest of bits in VDEVSTAT - Signals device ACTIVE (This actually indicates pending ("queued") device activity, and the VDEVCATT bit is the only bit not checked. This may be split out into a QUEUED io activity at some later time.
- ü VDEVPIOR - Signals device interrupt PENDING

If none of the above fields/bits are set, the device is treated as IDLE.

The current CLASS/TYPE restrictions, if any, are respected when the BACK or FORWARD pf-keys (PF7/PF8) are used from the DBK panel.

Using TRACK's ability to track the same user multiple times (via the NEW option of the internal TRACK command), you can create multiple sessions, each displaying a different set of devices. Simply enter a unique CLASS command from the DEV panel of each session, and then enter TRACK by itself to jump between the various device displays.

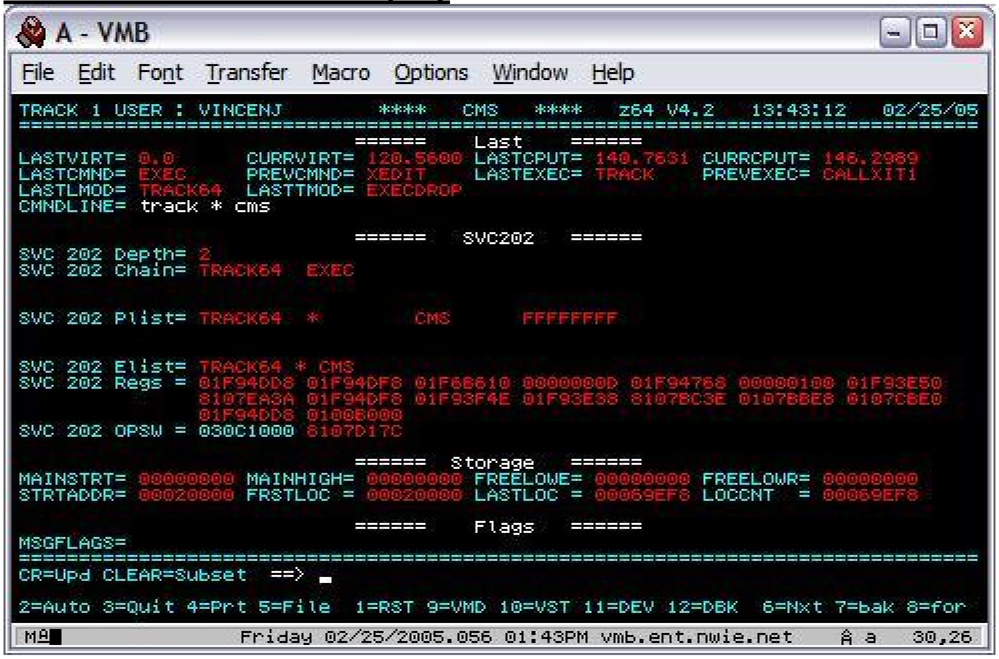
CMS – Command and Panel

Notes

You need the privilege class necessary to issue the CP LOCK and UNLOCK commands for TRACK to be able to completely build the CMS panel.

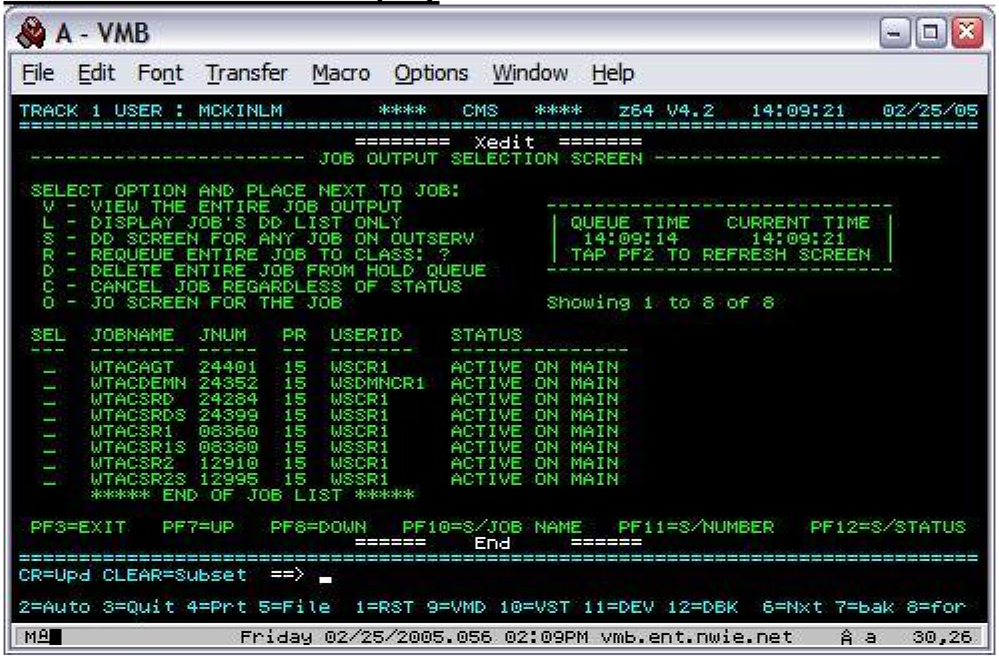
The code is highly release and user-mod sensitive. Expect problems (inaccurate data) if you are running a highly modified version of CMS or you try the CMS panel on non-CMS IPLed userids.

Screen Shot – default display



```
TRACK 1 USER : VINCENJ        ****   CMS   ****   z64 V4.2   13:43:12   02/25/05
================================================================================
                            ======    Last    ======
LASTVIRT= 0.0      CURRVIRT= 120.5600 LASTCPUT= 140.7631 CURRCPUT= 146.2989
LASTCMND= EXEC     PREVCMND= XEDIT    LASTEXEC= TRACK    PREVEXEC= CALLXIT1
LASTLMOD= TRACK64  LASTTMOD= EXECDROP
CMNDLINE= track * cms

                            ======   SVC202   ======
SVC 202 Depth= 2
SVC 202 Chain= TRACK64  EXEC


SVC 202 Plist= TRACK64  *        CMS      FFFFFFFF


SVC 202 Elist= TRACK64 * CMS
SVC 202 Regs = 01F94DD8 01F94DF8 01F6B610 0000000D 01F94768 00000100 01F93E50
               8107EA3A 01F94DF8 01F93F4E 01F93E38 8107BC3E 0107BBE8 0107CBE0
               01F94DD8 0100B000
SVC 202 OPSW = 030C1000 8107D17C

                            ======  Storage   ======
MAINSTRT= 00000000 MAINHIGH= 00000000 FREELOWE= 00000000 FREELOWR= 00000000
STRTADDR= 00020000 FRSTLOC = 00020000 LASTLOC = 00069EF8 LOCCNT  = 00069EF8

                            ======    Flags   ======
MSGFLAGS=
================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=Nxt 7=bak 8=for
 MA          Friday 02/25/2005.056 01:43PM vmb.ent.nwie.net    A a   30,26
```

Screen Shot – XEDIT display



```
TRACK 1 USER : MCKINLM        ****   CMS   ****   z64 V4.2   14:09:21   02/25/05
================================================================================
                              ========  Xedit  =======
 ----------------------- JOB OUTPUT SELECTION SCREEN -----------------------

 SELECT OPTION AND PLACE NEXT TO JOB:
  V - VIEW THE ENTIRE JOB OUTPUT             -----------------------------
  L - DISPLAY JOB'S DD LIST ONLY             | QUEUE TIME   CURRENT TIME |
  S - DD SCREEN FOR ANY JOB ON OUTSERV       |  14:09:14     14:09:21    |
  R - REQUEUE ENTIRE JOB TO CLASS: ?         | TAP PF2 TO REFRESH SCREEN |
  D - DELETE ENTIRE JOB FROM HOLD QUEUE      -----------------------------
  C - CANCEL JOB REGARDLESS OF STATUS
  O - JO SCREEN FOR THE JOB                  Showing 1 to 8 of 8

 SEL  JOBNAME  JNUM   PR  USERID    STATUS
 ---  -------- -----  --  --------  ---------------
  _   WTACAGT  24401  15  WSCR1     ACTIVE ON MAIN
  _   WTACDEMN 24352  15  WSDMNCR1  ACTIVE ON MAIN
  _   WTACSRD  24284  15  WSCR1     ACTIVE ON MAIN
  _   WTACSRDS 24399  15  WSSR1     ACTIVE ON MAIN
  _   WTACSR1  08360  15  WSCR1     ACTIVE ON MAIN
  _   WTACSR1S 08380  15  WSSR1     ACTIVE ON MAIN
  _   WTACSR2  12910  15  WSCR1     ACTIVE ON MAIN
  _   WTACSR2S 12995  15  WSSR1     ACTIVE ON MAIN
      ***** END OF JOB LIST *****

 PF3=EXIT   PF7=UP   PF8=DOWN   PF10=S/JOB NAME   PF11=S/NUMBER   PF12=S/STATUS
                              ======     End     ======
================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=Nxt 7=bak 8=for
 MA█           Friday 02/25/2005.056 02:09PM vmb.ent.nwie.net    Å a   30,26
```

Syntax

CMS	[ALL | * | types]

The CMS panel is invoked via the CMS command. You can enter the CMS command from ANY track panel, including the CMS panel itself. You can also append the CMS command to your initial or internal TRACK command.

For example, from CMS,   enter: Track operator cms
For example, from CMS,   enter: Track operator cms svc

For example, from TRACK, enter: Track vmmail cms
For example, from TRACK, enter: Track vmmail cms all
For example, from TRACK, enter: Track vmmail cms files stack
For example, from TRACK, enter: Track vmmail cms xedit

From within the CMS panel, enter: cms all
From within the CMS panel, enter: cms last files

From any TRACK panel, enter   : cms
From any TRACK panel, enter   : cms all
From any TRACK panel, enter   : cms svc last
From any TRACK panel, enter   : cms last svc

Once in the CMS panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Types

You can specify one or more 'types' after the word 'CMS', as follows:

```
LAST       - Selects various NUCON LAST and PREVIOUS fields
SVC202     - Select SVC 202 PLIST-related fields
STORAGE    - Selects various NUCON STORAGE fields
FLAGS      - Selects various NUCON FLAG fields
LIBRARIES  - Selects MACLIBs, TXTLIBs, DOSLIBs, LOADLIBs, CSLLIBs
DISKS      - Selects QUERY DISK display (all accessed disks)
STACK      - Selects PROGRAM and CONSOLE stack display
FILES      - Selects OPEN FILES display
FILEDEFS   - Selects FILDEFS display
XEDIT      - Selects XEDIT buffer-image display
NUCX       - Displays nucleus-extension SCBLOCK chain
SUBCOM     - Displays subcom SCBLOCK chain
SFS        - Displays Shared file directories for CMS 6 and above
```

By default, types LAST, SVC202, STORAGE, FLAGS, LIBRARIES, DISKS and SFS are displayed, in that order. You can abbreviate the types to their first two or three letters.

Please note that types STACK, FILES, FILEDEFS, XEDIT, NUCX and SUBCOM are NOT displayed unless you ask for them, or ask for ALL.

You can specify as many or as few types as you want, following the word 'CMS'. The types are displayed IN THE ORDER you specify them. You can also specify a type of 'ALL' or '*', which causes ALL fields to be displayed, in the order in which they are listed above.

Once you have specified a set of types, TRACK remembers your specification, and will re-display the same types if you leave and re-enter the CMS panel. If you are tracking multiple users, the selected types are remembered on a user-by-user basis.

You can re-specify a set of types from any panel, including the CMS panel itself. You can also specify the same types in a different order if you wish to change the panel display layout.

TRACK allocates 200 lines of buffer space for the CMS panel. If that space is insufficient, TRACK will show as much of the panel as will fit in the buffer, and will then issue a message. If this happens, you should restrict the display by selecting a shorter set of types.

Types LAST, SVC202, STORAGE, FLAGS and LIBRARIES use a fixed amount of buffer space. Type DISKS uses up to 29 lines (depending on how many disks are accessed). Types STACK, FILES, FILEDEFS, XEDIT, NUCX, SUBCOM and SFS use varying amounts of buffer space. Be especially wary of STACK; if the VM has just read a large file onto the stack, the STACK display will very often use up the buffer space. You may want to specify STACK and/or XEDIT as the last, or only, type.

The XEDIT display shows ZFOBUFIM, XEDIT's buffer image. This is NOT identical to the user's screen display when in XEDIT; it is what WOULD be displayed if file-lines were not hidden by a command line and other non-file-related data.

Description

The CMS panel shows fields that are significant to a virtual machine running CMS. TRACK does not check to see if the current tracked user is actually running CMS - if that is not the case, the display will contain meaningless information.

Whenever TRACK encounters a field which is invalid, or which cannot be read, it displays question-marks on the screen. You will see a great many question-marks if you invoke the CMS panel for a virtual machine which is not running CMS.

To insure that all types are correctly displayed, you should have the privilege class needed to issue the LOCK command. This is because TRACK must repeatedly read virtual storage to build the display. If TRACK is unable to read virtual storage, the relevant fields on the display will be shown as question marks. TRACK will not even attempt to build the CMS panel if it unable to read virtual page 0.

Use

Building the CMS panel involves a considerable amount of overhead. You should expect that panel refreshes will be slow compared to other TRACK panels. For example, it takes aproximately 5 times as long for TRACK to build the CMS DISK display as for CMS to execute a QUERY DISK command.

It is advantageous to restrict the display to the types you are interested in -- this will reduce the amount of work TRACK has to do to build the display.

There is no way for TRACK to prevent the tracked user from being dispatched by CP during the time that the CMS panel is being built. Thus it is possible, even likely, that the tracked virtual machine will be changing its storage as TRACK is reading it. When this happens, certain of the CMS fields may contain invalid or inaccurate information. This will generally be obvious. Press ENTER a few times to force TRACK to re-build the panel with fresh information.

The FLAGS display can be toggled using the BITS command (see the BITS section).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - Scrolls CMS display forward or back by lines
PF7  - Scrolls back    one screen's worth of CMS display
```

```
PF8  - Scrolls forward one screen's worth of CMS display
PF9  - Displays VMDBK (VMD panel)
PF10 - Displays VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

PF7 and PF8 scroll the displayed lines BACK and FORWARD a screen-full at a time. PF6 scrolls forward or back in lines - type a number, or a number preceded by a '-' sign, before hitting PF6 (if you don't type anything, PF6 scrolls the display forward one line).

Commands

See the General Information section for a description of commands valid from ANY panel. There are no commands specific to the CMS panel, except for the CMS command itself, whose syntax is detailed above.

CONSOLE – Command and Panel

Syntax

Type CONsole from ANY panel, then press ENTER. CONsole can be abbreviated to C.

You can start tracking a user in the CON panel by appending the word CON command to an initial or internal TRACK command.

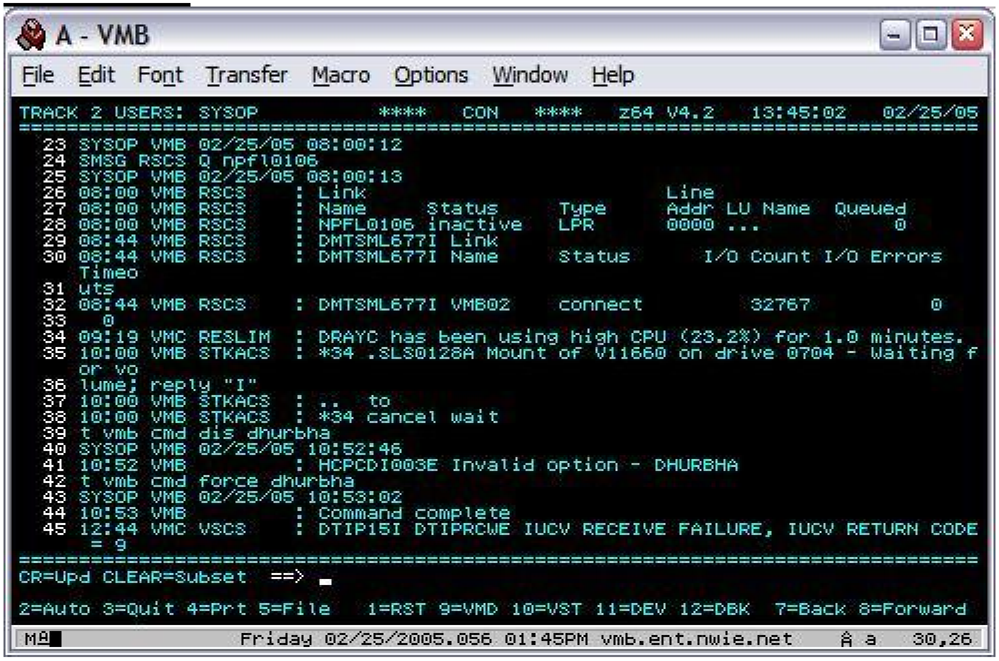
For example, from CMS, enter: Track rscs con
For example, from TRACK, enter: Track mvs c

Once in the CON panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Description

The CON panel displays lines from the active spooled console, if any.

Screen Shot



```
TRACK 2 USERS: SYSOP          ****   CON   ****   z64 V4.2   13:45:02   02/25/05
===============================================================================
  23 SYSOP VMB 02/25/05 08:00:12
  24 SMSG RSCS Q npfl0106
  25 SYSOP VMB 02/25/05 08:00:13
  26 08:00 VMB RSCS     : Link                        Line
  27 08:00 VMB RSCS     : Name     Status     Type      Addr LU Name  Queued
  28 08:00 VMB RSCS     : NPFL0106 inactive   LPR       0000 ...           0
  29 08:44 VMB RSCS     : DMTSML677I Link
  30 08:44 VMB RSCS     : DMTSML677I Name     Status       I/O Count I/O Errors
     Timeo
  31 uts
  32 08:44 VMB RSCS     : DMTSML677I VMB02    connect          32767          0
  33    0
  34 09:19 VMC RESLIM   : DRAYC has been using high CPU (23.2%) for 1.0 minutes.
  35 10:00 VMB STKACS   : *34 .SLS0128A Mount of V11660 on drive 0704 - Waiting f
     or vo
  36 lume; reply "I"
  37 10:00 VMB STKACS   : ..  to
  38 10:00 VMB STKACS   : *34 cancel wait
  39 t vmb cmd dis dhurbha
  40 SYSOP VMB 02/25/05 10:52:46
  41 10:52 VMB          : HCPCDI003E Invalid option - DHURBHA
  42 t vmb cmd force dhurbha
  43 SYSOP VMB 02/25/05 10:53:02
  44 10:53 VMB          : Command complete
  45 12:44 VMC VSCS     : DTIP15I DTIPRCWE IUCV RECEIVE FAILURE, IUCV RETURN CODE
       = 9
===============================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK  7=Back 8=Forward
 MA          Friday 02/25/2005.056 01:45PM vmb.ent.nwie.net    A a   30,26
```

Use

TRACK displays the lines in CP's current in-core spool buffer (if any). Each line is preceded by a line number, showing that line's offset from the top of the display. Every time you press

ENTER, the entire CP spool buffer is re-read. Thus, the display may jump from showing many lines, to showing just a few (or none).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the console)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - No special action, updates the panel
PF7  - Scrolls back    one screen's worth of console lines
PF8  - Scrolls forward one screen's worth of console lines
PF9  - Displays VMDBK (VMD panel)
PF10 - Display VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

PF7 and PF8 scroll the displayed lines BACK and FORWARD a screen-full at a time.

Commands

See the General Information section for a description of commands valid from ANY panel. There are no commands specific to the CON panel.

DBK - Panel

Syntax

Type a device address on the command line, or place your cursor under a device address (use the TAB key to move your cursor), then press PF12 or PF24 from any TRACK panel (including the DBK panel itself).

You can also hit PF12/24 without typing a device address (or placing your cursor under a device address) -- in this case, the last displayed devices are re-displayed in a cyclic fashion. If you are tracking multiple users, the DBK device addresses are remembered separately for each user.

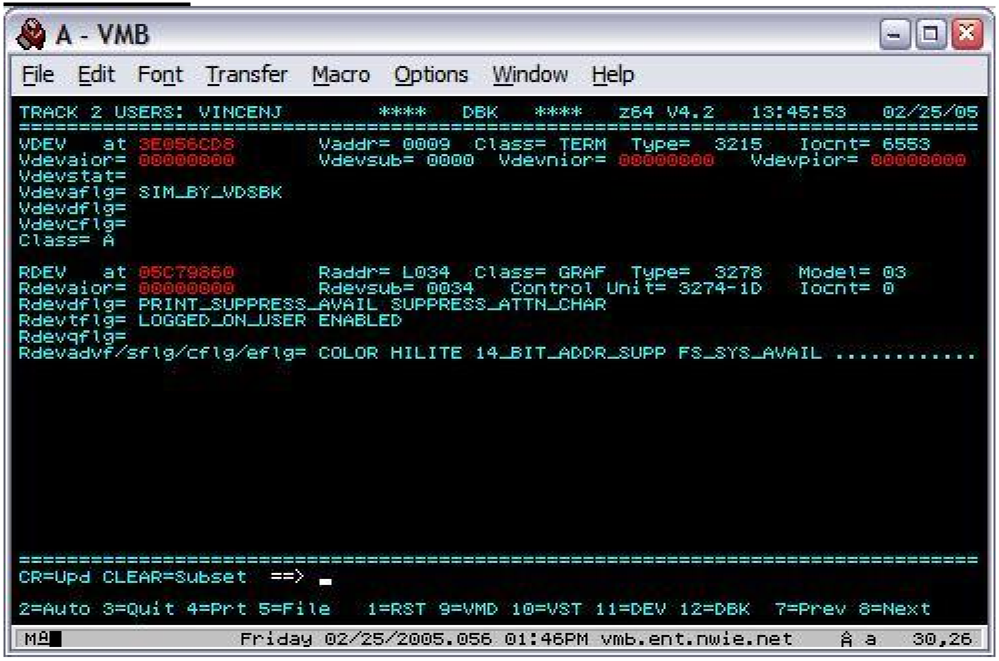
If you hit PF12 without specifying an address, and you have not previously used PF12, the lowest-address device is displayed.

You can start tracking a user in the DBK panel by appending the word 'dbk' and a device address to an initial or internal TRACK command.

For example, from CMS,   enter: Track operator dbk 009
For example, from TRACK, enter: Track pvm dbk L001

Once in the DBK panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot



```
TRACK 2 USERS: VINCENJ        ****   DBK   ****   z64 V4.2   13:45:53   02/25/05
================================================================================
VDEV   at 3E056CD8        Vaddr= 0009  Class= TERM  Type=  3215   Iocnt= 6553
Vdevaior= 00000000        Vdevsub= 0000  Vdevnior= 00000000   Vdevpior= 00000000
Vdevstat=
Vdevaflg= SIM_BY_VDSBK
Vdevdflg=
Vdevcflg=
Class= A

RDEV   at 05C79860        Raddr= L034  Class= GRAF  Type=  3278   Model= 03
Rdevaior= 00000000        Rdevsub= 0034   Control Unit= 3274-1D   Iocnt= 0
Rdevdflg= PRINT_SUPPRESS_AVAIL SUPPRESS_ATTN_CHAR
Rdevtflg= LOGGED_ON_USER ENABLED
Rdevqflg=
Rdevadvf/sflg/cflg/eflg= COLOR HILITE 14_BIT_ADDR_SUPP FS_SYS_AVAIL ...........








================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK  7=Prev 8=Next
 MA                 Friday 02/25/2005.056 01:46PM vmb.ent.nwie.net    A a   30,26
```

Description

The DBK panel shows selected information from device control blocks. Generally, information is displayed from the RDEV and VDEV, although in some cases (LDEVs and CTCAs), the information comes from other control blocks. The information is labelled, so that you can determine which control block it came from.

PF8 and PF7 move the DBK display forward and back through all virtual devices.

Use

A number of the values displayed on the DBK panel are imbedded in unprotected fields. You can place your cursor under these values simply by pressing the TAB key. This is useful if you want to display a real or virtual address (TAB to the address, hit PF1 or PF10), display a device (TAB to the device address, hit PF12), or format the address (type 'FORMAT dsect', TAB to the address, hit ENTER).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - No special action, just updates the panel
PF7  - Shifts to PREVIOUS device in the user's virtual device chain
PF8  - Shifts to NEXT     device in the user's virtual device chain
PF9  - Displays VMDBL (VMD panel)
PF10 - Displays VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Shifts to a new device if you type, or place cursor under, an
       address; else shifts to any previously displayed devices.
```

Commands

See the General Information section for a description of commands valid from any panel. What follows are commands specific to the DBK panel.

CLASS subcommand

The CLASS subcommand is valid from the DBK panel. See the section *CLASS – DBK and DEV subcommand* for more information.

DEV - Panel

Syntax

Hit PF11 or PF23 from ANY panel.

The DEV panel is invoked via PF11/23. You can also start tracking a user in the DEV panel by appending the word 'dev' to your initial, or internal, TRACK command.

For example, from CMS, enter: Track pvm dev
For example, from TRACK, enter: Track rscs dev

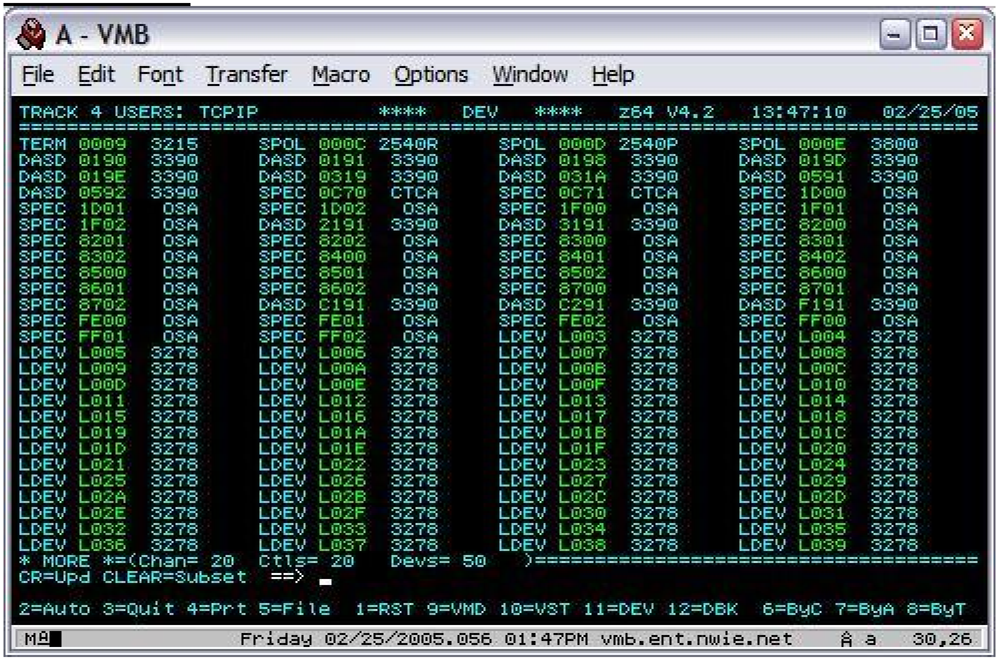
You can also append a CLASS command after the word 'dev' (see below for a description of the CLASS command).

For example, from CMS, enter: Track mvs dev class dasd
For example, from CMS, enter: Track mvs dev class * io pending
For example, from TRACK, enter: Track mvs dev class dasd io active

Hung users often have IO interrupts pending from CP. You can quickly display any such pending interrupts by appending the words 'dev class * io pend' to your TRACK command (see second example above).

Once in the DEV panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot



```
TRACK 4 USERS: TCPIP          ****   DEV   ****   z64 V4.2   13:47:10   02/25/05
================================================================================
TERM 0009  3215      SPOL 000C 2540R      SPOL 000D 2540P      SPOL 000E  3800
DASD 0190  3390      DASD 0191  3390      DASD 0198  3390      DASD 019D  3390
DASD 019E  3390      DASD 0319  3390      DASD 031A  3390      DASD 0591  3390
DASD 0592  3390      SPEC 0C70  CTCA      SPEC 0C71  CTCA      SPEC 1D00   OSA
SPEC 1D01   OSA      SPEC 1D02   OSA      SPEC 1F00   OSA      SPEC 1F01   OSA
SPEC 1F02   OSA      DASD 2191  3390      DASD 3191  3390      SPEC 8200   OSA
SPEC 8201   OSA      SPEC 8202   OSA      SPEC 8300   OSA      SPEC 8301   OSA
SPEC 8302   OSA      SPEC 8400   OSA      SPEC 8401   OSA      SPEC 8402   OSA
SPEC 8500   OSA      SPEC 8501   OSA      SPEC 8502   OSA      SPEC 8600   OSA
SPEC 8601   OSA      SPEC 8602   OSA      SPEC 8700   OSA      SPEC 8701   OSA
SPEC 8702   OSA      DASD C191  3390      DASD C291  3390      DASD F191  3390
SPEC FE00   OSA      SPEC FE01   OSA      SPEC FE02   OSA      SPEC FF00   OSA
SPEC FF01   OSA      SPEC FF02   OSA      LDEV L003  3278      LDEV L004  3278
LDEV L005  3278      LDEV L006  3278      LDEV L007  3278      LDEV L008  3278
LDEV L009  3278      LDEV L00A  3278      LDEV L00B  3278      LDEV L00C  3278
LDEV L00D  3278      LDEV L00E  3278      LDEV L00F  3278      LDEV L010  3278
LDEV L011  3278      LDEV L012  3278      LDEV L013  3278      LDEV L014  3278
LDEV L015  3278      LDEV L016  3278      LDEV L017  3278      LDEV L018  3278
LDEV L019  3278      LDEV L01A  3278      LDEV L01B  3278      LDEV L01C  3278
LDEV L01D  3278      LDEV L01E  3278      LDEV L01F  3278      LDEV L020  3278
LDEV L021  3278      LDEV L022  3278      LDEV L023  3278      LDEV L024  3278
LDEV L025  3278      LDEV L026  3278      LDEV L027  3278      LDEV L029  3278
LDEV L02A  3278      LDEV L02B  3278      LDEV L02C  3278      LDEV L02D  3278
LDEV L02E  3278      LDEV L02F  3278      LDEV L030  3278      LDEV L031  3278
LDEV L032  3278      LDEV L033  3278      LDEV L034  3278      LDEV L035  3278
LDEV L036  3278      LDEV L037  3278      LDEV L038  3278      LDEV L039  3278
* MORE *=(Chan= 20  Ctls= 20   Devs= 50   )====================================
CR=Upd CLEARSubset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=ByC 7=ByA 8=ByT
MA              Friday 02/25/2005.056 01:47PM vmb.ent.nwie.net    A a   30,26
```

Description

The DEV panel shows all, or selected virtual devices in a virtual machine's device configuration. For each device, the device CLASS, ADDRESS and TYPE are displayed. See the COMMANDS section for setting device display restrictions.

Devices with pending interrupts are shown highlighted.

If the screen is not large enough to display all of the devices, TRACK puts out a MORE message. The screen is scrolled when you press ENTER. If you want to update the screen WITHOUT scrolling, press PF11 instead of ENTER.

Use

You can sort the displayed devices by CLASS, ADDRESS or TYPE via PF6, PF7 and PF8, respectively. Once you have sorted the devices, the sort order is remembered by TRACK, even if you leave and return to the DEV panel. If you are tracking more than one user, the sort order is separately maintained for each user.

The default (initial) sort order is by ADDRESS.

The device addresses are placed in unprotected fields -- you can TAB to these addresses, then hit PF12 to invoke the DBK panel for a specific device.

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - Sorts the device display by CLASS
PF7  - Sorts the device display by ADDRESS
PF8  - Sorts the device display by TYPE
PF9  - Displays VMDBK (VMD panel)
PF10 - Displays VIRTUAL storage (VST panel)
PF11 - Updates the panel, without scrolling if MORE message displayed
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

Commands

See the General Information section for a description of commands valid from ANY panel. What follows are commands specific to the DEV panel.

CLASS Command

The CLASS subcommand is valid from the DBK panel. See the section *CLASS – DBK and DEV subcommand* for more information.

DROPFRM – FRM panel subcommand

The DROPFRM command may only be issued from the FRM panel in TRACK.

See the FORMAT section on "DROPRM command for the FRM panel" for more information.

DUMP - Command

Syntax

DUMP	[start-address [end-address]]

DUMP can be entered on any of the panels.

You can append a DUMP command to your initial, or internal, TRACK command.

For example, from CMS, enter: Track pvm dump
For example, from TRACK, enter: Track rscs dump 0 fff000

Description

The Dump command causes TRACK to dump the tracked user's virtual storage to your spooled printer (DCSS storage is NOT dumped).

The dump is in EBCDIC format. Each line of the dump shows 8 words (32 bytes) of storage in both EBCDIC and HEX form.

The dump is taken in 64K chunks. You can interrupt this command by pressing ENTER which will abort the dump after the current 64K chunk. Dump does not close your printer. This allows you to append or prefix a dump with other TRACK screens.

You can specify a range on the dump command. 'DUMP' defaults to printing ALL of the tracked machine's storage. 'DUMP nnnnnnnn' takes a dump from address nnnnnnnn to the end of virtual storage, while 'DUMP nnnnnnnn nnnnnnnn' takes a dump from the first nnnnnnnn to the last. The DUMP command adjusts each range value to the next lower 64k boundary. Each range value must be a valid hex address.

FORMAT – Command and Panel

Overview

The FORMAT facility of TRACK allows you to display storage (real, virtual or third-level) as mapped by a dsect. The FORMAT display is dynamic – the displayed storage is updated every time you hit the ENTER key, so that you can watch storage change as a virtual machine, or CP, executes.

FORMAT reads dsects directly from your CP, CMS, or any other MACLIB. It then displays the dsect, with the actual storage values interpreted as per the dsect statements. For example, you can display a user's VMDBK as per the CP VMDBK dsect, with the current values of each operand displayed and interpreted.

FORMAT displays a dsect on a panel labeled FRM. Each line of the dsect is displayed, along with its address (real) and offset from the start of the dsect. You can scroll through the dsect via PF7 and 8. PF6 takes you to a specific label in the dsect.

FORMAT does not show the repetition or length values of operands (no room) - instead, it interprets the actual storage value as per the length and type of the operand. If the interpreted value will not fit on one line, it is continued on the following line(s).

Generally, FORMAT shows a dsect statement exactly as it is coded. FORMAT generates its own labels for unlabelled DS statements and for unlabelled MACRO directives (e.g., ANOP, AGO). These internally-generated labels are of the form ':Ln', where 'n' is a sequential number. If a single DS statement contains multiple operands, FORMAT displays each operand on a separate line.

FORMAT handles flag fields (equated bits) in a special way. Whenever FORMAT encounters a sequence of the form:

```
label1   DC    X'xx'
label2   EQU   X'80'
label3   EQU   X'40'
.
.
.
```

-OR-

```
label1   DC    X'xx'
.
.
.
* BITS DEFINED IN label1
label2   EQU   X'80'
label3   EQU   X'40'
.
.
.
```

FORMAT assumes that the EQUATES are flags defined on the preceding DC or DS, and it will tell you whether or not the FLAG is currently ON or OFF. If ON, TRACK will also highlight the flag. In the second case above, FORMAT will also position the flag definitions immediately after the defining flag, regardless of where they occur in the dsect.

Syntax

You invoke FORMAT via the FORMAT command from within TRACK. The syntax of the command is:

FORMAT	[dsect [address | * [maclib-file <MACLIB>]]]

Examples:

```
FORMAT vmdbk 2d42000
FORMAT nucon v0 dmsgpi
FORMAT myblock v2c850 mylib maclib
FORMAT rdev * hcpom1    (then place cursor on address and tap ENTER)
```

The 'dsect', 'address' and 'maclib-file' operands may or may not be required, depending on the context in which you are invoking the command. Each operand is reviewed separately below:

Maclib-File

This operand is optional.

FORMAT needs to know which MACLIB files it should read to locate the dsect you wish to display. You can specify either the filename of a MACLIB file, or the filename of a CNTRL file. In the latter case, FORMAT will pick up the name(s) of the MACLIB file(s) from the MACS record of the CNTRL file.

If you have two files, one a CNTRL file and the other a MACLIB file, both with the same filename, you can select the MACLIB file by appending the word MACLIB following the filename. By default, TRACK will first search for a CNTRL file, then for a MACLIB file, whose fn matches the specified 'maclib-fn'.

If you do not specify a 'maclib-fn', TRACK will search a default set of CNTRL files depending on the build level of TRACK and the VM system you are running on. The TRACK installation instructions detail how you can override these defaults (e.g., make your CP/CMS cntrl files the defaults).

Once you have issued a FORMAT command that specifies a MACLIB or CNTRL filename, or if TRACK locates the default CNTRL/MACLIB files, it will remember the MACLIB or MACLIBs searched, and will search those same MACLIBs for any further FORMAT commands. You can always re-specify a CNTRL or MACLIB fn on any further FORMAT command to reset the defaults.

Address

This operand is optional. You can also specify addresses by typing a "*" and placing your cursor under them, rather than typing them out.

The address field of the FORMAT command specifies the real or virtual address that you want displayed as per the specified dsect. You can enter an address in one of two ways:

1. Type the address following the dsect name in the FORMAT command.
2. Type 'FORMAT dsect' on the command line and place your cursor under an address field in any TRACK panel (use the TAB key to move your cursor). **Note** that if your FORMAT command is going to include a 'maclib-fn', you must type a "*" character for the address and then the 'maclib-fn'.

If you are typing in an address, you can prefix the address with a 'V' or an 'R' to specify whether the address is virtual or real (default = real). For example:

```
FORMAT VMDBK 2D42000
FORMAT VMDBK R2D42000
FORMAT NUCON  V0
```

If you are specifying the address by placing your cursor under an address field, TRACK will assume that the address is real, unless you are in the VST panel, in which case it will assume that the address is virtual. If you are already displaying a dsect (you are in the FRM panel), FORMAT will assume that the address type matches that of the currently displayed dsect.

Placing your cursor under an address is convenient if you are chaining through control blocks, or if you are picking up the address of one control block from another. For example, if you have just entered TRACK and are in the VMD screen, you can type FORMAT VMDBK on the command line, hit TAB to move your cursor to the VMDBK address field of the VMD panel, and hit ENTER -- TRACK will pick up the address from you cursor location.

You can follow this procedure from any TRACK panel, including the FRM panel itself. The latter possibility allows you to chain through a set of linked control blocks. For example, after entering the FRM panel for the RDEV dsect, you can type FORMAT VMDBK on the command line, TAB to the RDEVUSER field, and hit ENTER. TRACK will now show you the VMDBK in dsect form.

Dsect

This operand is only required when you wish to display a new dsect.

The 'dsect' field of the format command specifies the dsect that you want displayed. You must specify this field on your first FORMAT command, and whenever you want to display a different dsect. TRACK remembers the name (and storage addresses) of all the dsects that you have displayed. You can get back to any previous dsect display simply by entering

FORMAT (which cycles you around the previous displays), or by entering 'FORMAT dsect', which takes you to a specific dsect display.

Use

TRACK first searches all accessed disks for a file named 'dsect TRACKFRM' (see discussion of the SAVEFRM command below). If not found, TRACK searches your specified, or default, maclib(s) for the dsect, parses the assembler statements, builds control blocks in storage, and shows the specified storage as per the dsect. Once TRACK has read a dsect into storage, it does not need to re-read the maclib(s) -- it will use the in-storage control blocks.

TRACK shows the dsect in a panel labeled FRM. You can go from this panel to any other TRACK panel simply be entering a command or pressing a PF key. Enter FORMAT by itself again to jump back to your previous FRM panels. If you are tracking more than one user, you can display separate dsects for each user, and jump between the FRM displays simply be entering 'Track <userid>'.

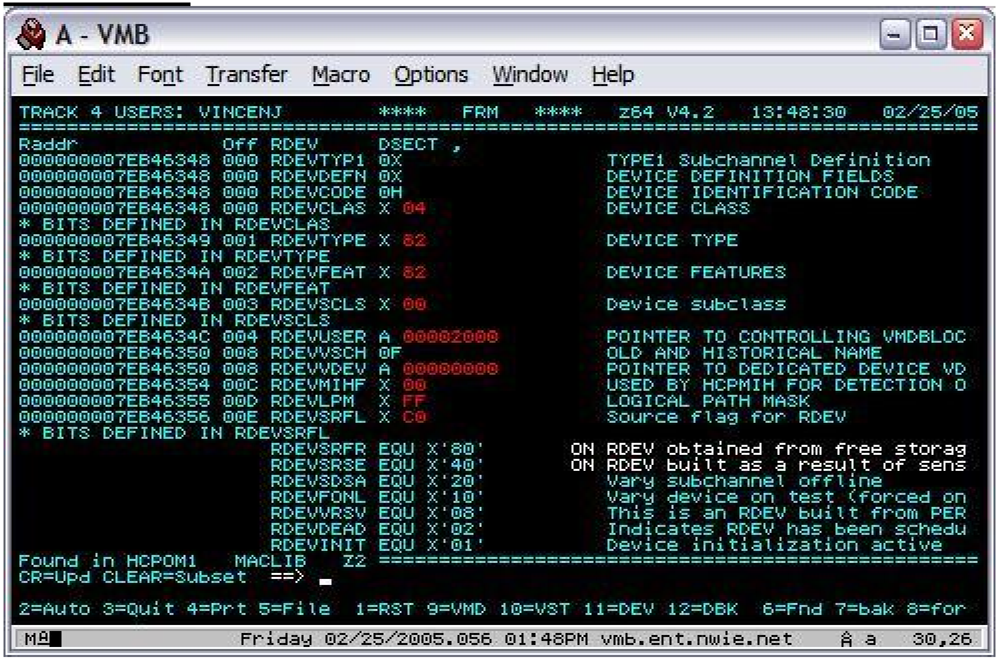
For example, you can TRACK one user and format that user's VMDBK. Then you can enter 'TRACK userid' to track another user, use the FORMAT command to display a different (or the same) dsect, and jump between the two displays by entering 'TRACK.' You can track and format up to 8 separate dsects in this way.

TRACK remembers all previously displayed dsects and storage addresses. You can cycle between these dsects simply by entering the FORMAT command with no operands.

For example, you TRACK a user, type FORMAT VMDBK on the command line, TAB to the VMDBK address field, and hit ENTER. You are now in the FRM panel displaying the VMDBK. You type VMDRTERM and hit PF6. Your display has now shifted so that VMDRTERM is at the top of the screen. Now type FORMAT RDEV on the command line, hit TAB so your cursor is under the VMDRTERM field, and hit ENTER. You are now in FRM displaying the RDEVBK. Simply entering FORMAT will now cycle you between these two dsects.

From within the FRM panel, you can do a number of special things. Two new commands are available, and there is a special facility for chaining through control blocks. There are also special PF keys for moving around the FRM display. Like all TRACK panels, you can simply keep pressing ENTER within FRM to have the displayed storage updated, or you can enter a command or hit a PF key to jump to a different panel.

Screen Shot



```
TRACK 4 USERS: VINCENJ        ****   FRM   ****   z64 V4.2   13:48:30   02/25/05
================================================================================
Raddr            Off RDEV     DSECT ,
000000007EB46348 000 RDEVTYP1 0X                 TYPE1 Subchannel Definition
000000007EB46348 000 RDEVDEFN 0X                 DEVICE DEFINITION FIELDS
000000007EB46348 000 RDEVCODE 0H                 DEVICE IDENTIFICATION CODE
000000007EB46348 000 RDEVCLAS X 04               DEVICE CLASS
* BITS DEFINED IN RDEVCLAS
000000007EB46349 001 RDEVTYPE X 82               DEVICE TYPE
* BITS DEFINED IN RDEVTYPE
000000007EB4634A 002 RDEVFEAT X 82               DEVICE FEATURES
* BITS DEFINED IN RDEVFEAT
000000007EB4634B 003 RDEVSCLS X 00               Device subclass
* BITS DEFINED IN RDEVSCLS
000000007EB4634C 004 RDEVUSER A 00002000         POINTER TO CONTROLLING VMDBLOC
000000007EB46350 008 RDEVVSCH 0F                 OLD AND HISTORICAL NAME
000000007EB46350 008 RDEVVDEV A 00000000         POINTER TO DEDICATED DEVICE VD
000000007EB46354 00C RDEVMIHF X 00               USED BY HCPMIH FOR DETECTION O
000000007EB46355 00D RDEVLPM  X FF               LOGICAL PATH MASK
000000007EB46356 00E RDEVSRFL X C0               Source flag for RDEV
* BITS DEFINED IN RDEVSRFL
                     RDEVSRFR EQU X'80'       ON RDEV obtained from free storag
                     RDEVSRSE EQU X'40'       ON RDEV built as a result of sens
                     RDEVSDSA EQU X'20'          Vary subchannel offline
                     RDEVFONL EQU X'10'          Vary device on test (forced on
                     RDEVVRSV EQU X'08'          This is an RDEV built from PER
                     RDEVDEAD EQU X'02'          Indicates RDEV has been schedu
                     RDEVINIT EQU X'01'          Device initialization active
Found in HCPOM1   MACLIB   Z2 ==================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=Fnd 7=bak 8=for
 MA            Friday 02/25/2005.056 01:48PM vmb.ent.nwie.net    A a   30,26
```

Chaining

Within the FRM panel, you can jump to a different storage address simply by placing your cursor under an address without typing anything on the command line. For example, if you are in the FRM panel displaying the VMDBK dsect, you can tab to the VMDCYCLE field value and hit ENTER. TRACK will pick up the address from your cursor, and will display that storage as per the VMDBK dsect. Thus you can chain through the same control blocks simply by tabbing to a chaining-field address and pressing ENTER. Note that if you want to display a DIFFERENT control block, you must enter 'FORMAT dsect' on the command line.

To further facilitate chaining, you can tab your cursor to a chaining field and hit PF2 (auto-update). TRACK will now take you through the control-block chain, skipping to the next control block at every auto-update interval (use AUTO nn to set the auto-update interval from 1 to 99 seconds). Press ENTER to terminate the chaining.

Commands

See the General Information section for a description of commands valid from any panel. What follows are commands specific to the FRM panel.

HIDEFRM command for the FRM panel

A problem with chaining through control blocks is that the information you want displayed may not all fit on one screen. The HIDEfrm command allows you to tailor the display. HIDE, which is only valid from the FRM panel, allows you to hide (not display) any range of lines. Thus you can reduce the display to contain only those fields you want to see. If they all fit on one screen, you can then chain through the control blocks with all needed information displayed.

The syntax of the HIDE command is:

```
HIDE OFF
HIDE label <label>
```

'HIDE OFF' unhides all lines - the display is returned to it's full form.

'HIDE label' hides the one line containing that label.

'HIDE label label' hides all lines between the two labels, inclusive.

If you want to hide a line whose label if 'OFF', specify 'HIDE OFF OFF'.

You can use a '*' for either the first or second label. For the first label, a '*' stands for the top of the dsect. For the second label, it stands for the end of the dsect.

You can also precede the second label with a minus sign (-). This causes the hide range to EXCLUDE the second label; e.g., HIDE VM1 -VM4 would hide everything from VM1 up to, but not including, VM4.

HIDE commands are additive. You can issue multiple HIDE commands to HIDE more and more lines. Each HIDE hides all currently hidden lines PLUS the ones specified. Once you have hidden lines, TRACK remembers that they are hidden even if you leave and return to the FRM panel. If you TRACK multiple users and display the same dsect for these users, each display will respect hidden lines -- that is, HIDE affects all references to the dsect in any FORMAT command.

DROPRM command for the FRM panel

TRACK does not support AIFs and AGOs in DSECTs. This can result in incorrect mapping of storage areas. For example, consider the following section of a DSECT:

```
         AIF   (NOT &AP).NOTAP1
LABEL1   DS    F
LABEL2   DS    F
LABEL3   DS    F
         AGO   .NOTAP2
.NOTAP1  ANOP
LABEL4   DS    F
LABEL5   DS    F
LABEL6   DS    F
.NOTAP2  ANOP
LABEL7   DS    F
```

This DSECT maps 3 words of storage in two different ways, depending on whether the &AP global variable is ON or OFF. Because TRACK ignores AIF's and ANOP's, it will include labels 1 to 6 (totaling 24 bytes), when it should have included only labels 1 to 3 or 4 to 6 (totaling 12 bytes). The offsets and values of everything following label 3 will be incorrect.

The DROPFRM command enables you to fix such incorrect mappings by DROPping the labels that should not be included. In the above example, you would either DROP labels 1 to 3 (if &AP is OFF in your system), or labels 4 to 6 (if &AP is ON in your system).

The syntax of the DROPFrm command is:

```
DROPFrm label <label>
```

If you specify only one label, or if the two labels are the same, only that one line is dropped. If the labels are different, all lines between the two labels (inclusive) are dropped. You can also precede the second label with a dash (-), to drop everything up to, but not including, the second label.

Unlike HIDEfrm, DROPfrm actually drops the lines, and adjusts the offsets of all following lines to account for the dropped storage (if any). An attempt is also made to adjust alignments, if they are changed. You can specify any label, or set of labels, on a DROPfrm command, including the dummy labels that TRACK generates for things like ANOPs and AIFs. To continue the above example, if you were to display storage with the FORMAT command using the above DSECT, TRACK's display would look something like this (the addresses and offsets shown below are purely illustrative):

```
               :L8      AIF   (NOT &AP).NOTAP1
00080040 050   LABEL1   DS    F   00000034
00080044 054   LABEL2   DS    F   000010F7
00080048 058   LABEL3   DS    F   00023456
               :L9      AGO   .NOTAP2
               .NOTAP1  ANOP
0008004C 05C   LABEL4   DS    F   10204556
00080050 060   LABEL5   DS    F   000203FF
00080054 064   LABEL6   DS    F   FF018976
              .NOTAP2  ANOP
00080058 068   LABEL7   DS    F   FF011112
```

The labels that start with ':L' are generated by TRACK. Suppose the &AP variable is ON in your system. This means that the storage from 80040 to 8004B should be mapped by LABEL1 to LABEL3, and LABEL7 should pick up after LABEL3 (in brief, LABEL4 to LABEL6 should be dropped). You would issue the following two commands:

```
DROPFrm :L9 .NOTAP2
DROPFrm :L8
```

The first command would drop labels 4 to 6, as well as the AGOs and ANOPs that surround these labels. The second command would simply clean-up the DSECT by dropping the first AIF. The storage mapping would now look like this:

```
00080040 050   LABEL1   DS    F   00000034
00080044 054   LABEL2   DS    F   000010F7
00080048 058   LABEL3   DS    F   00023456
0008004C 05C   LABEL7   DS    F   10204556
```

Note that the offset of LABEL7 has been adjusted, and the corresponding value now shows what is at that offset.

You should always drop the AGOs and AIFs that surround DS statements, along with the DS statements themselves. This will prevent TRACK from issuing it's warning message about 'unresolved AIFs and AGOs'.

You can issue multiple DROPFrm commands to drop discontinuous sets of labels.

Once you have DROPped labels, you cannot recover them, except by quitting TRACK, re-entering TRACK and re-issuing the FORMAT command. You can issue a SAVEfrm command following your DROPFrm command(s); this will save the DSECT as modified by the DROPFrm command(s). Indeed, the purpose of the DROPFrm command is to allow you to fix a DSECT mapping, and save the corrected mapping via the SAVEfrm command.

SAVEFRM command for the FRM panel

The SAVEFRM command saves the currently displayed dsect in a file named 'dsect TRACKFRM A'. This file is in a special format that only TRACK can read. Once you have saved a dsect in this way, TRACK will look for it on any future FORMAT commands. The idea is to avoid having to re-read your MACLIBs and re-build the in-storage control blocks. You can build up a library of TRACKFRM files. Then, whenever you go into TRACK, those dsects will be immediately available without going through the overhead of reading maclibs and parsing the assembler statements.

The syntax of the SAVEFRM command is:

```
SAVEFRM
```

SAVEFRM will only save a dsect onto your A disk, and it will NOT replace a saved dsect from your A disk. You must manually erase the 'dsect TRACKFRM A' file before issuing a new SAVEFRM command for the same dsect. TRACK will read a saved dsect file from ANY accessed disk, however, so that you can make saved dsects available on a public disk.

TRACK will look for a 'dsect TRACKFRM' file before attempting to read your MACLIBs. If you have modified a control block, you should erase the old TRACKFRM file, issue FORMAT to force a read of the MACLIB, then issue another SAVEFRM command to save the newly formatted control block (you can do all of this from within TRACK). The fact that TRACK looks for a SAVEFRM file before searching MACLIBs, means that you can use the FORMAT command without accessing any MACLIBs.

Before you save a dsect with SAVEfrm, you should go through it (as displayed by TRACK) and fix-up incorrect mappings with the DROPFrm command (see above).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
```

```
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - Scrolls forward or back by lines, or scrolls to a label
PF7  - Scrolls display FORWARD  one screen's worth
PF8  - Scrolls display BACKWARD one screen's worth
PF9  - Displays VMDBK (VMD panel)
PF10 - Displays VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

Three special PF keys are available with the FRM screen, PFs 6, 7 and 8.

PF7 and PF8 scroll you back and forth through the dsect a screen's worth at a time.

PF6 is a multi-purpose PFkey. When you hit PF6, TRACK looks at the command line. If it finds a number, it scrolls FORWARD that many lines. If the number is preceded by a '-' sign, it scrolls BACK that many lines.

If it finds the word 'top' or 'bot', it scrolls to the top or bottom of the dsect, respectively.

If it finds any other word, it assumes that you have entered a LABEL, and it does the equivalent of an XEDIT FIND - it searches for a label in the dsect that starts with the characters you typed.

If TRACK finds nothing on the command line, it repeats the last PF6 argument (default = '1'). So:

```
        Type       Hit     Action
        ----       ---     ------
         1         PF6     Scroll forward 1 line
        -12        PF6     Scroll back 12 lines
         vmdu      PF6     Look for label that starts with 'VMDU'
         <nothing> PF6     Repeat last PF6 argument; if no last
                             PF6 argument, scroll forward on line.
         top       PF6     Scroll to top of dsect
         bot       PF6     Scroll to bottom of dsect
         vmdu      PF6     Scroll to label that starts with 'VMDU'
         <nothing> PF6     Scroll to next label that start with 'VMDU'
         <nothing> PF6     Scroll to next label that start with 'VMDU'
```

HIDEFRM – FRM panel subcommand

The HIDEFRM command may only be issued from the FRM panel in TRACK.

See the FORMAT section on "HIDERM command for the FRM panel" for more information.

IUCV – Command and Panel

Syntax

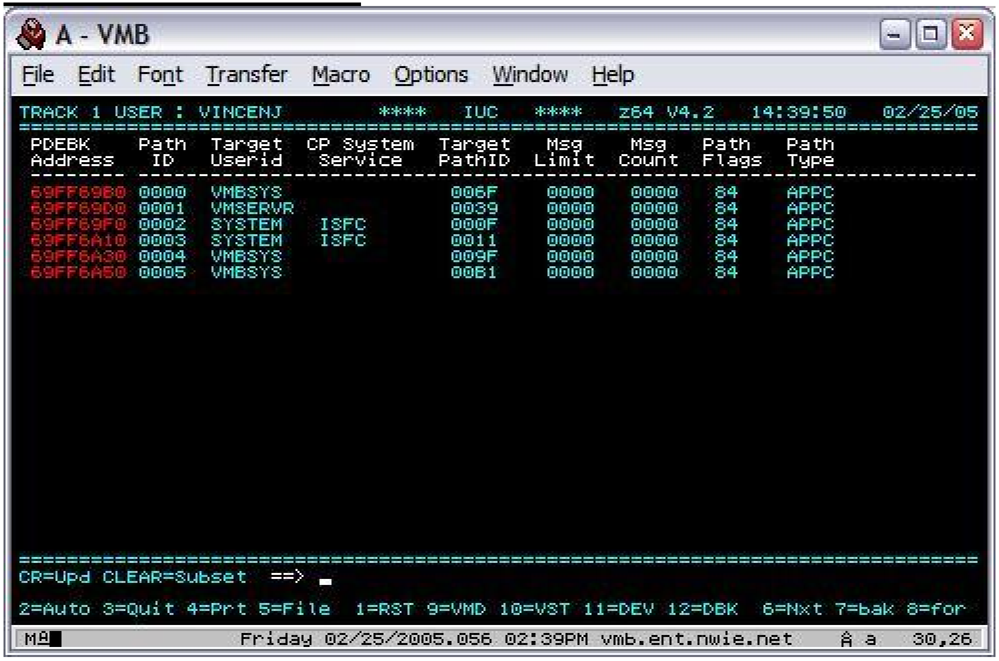
IUCV	[*cpservice]

Enter IUCV while on any panel. This will display the IUCV/APPC connections for the userid being tracked. If you are tracking the SYSTEM userid, then you can filter the CP Services on the IUCV panel by entering a *cpservice name after the IUCV command.

Example: TRACK SYSTEM IUCV *MSG

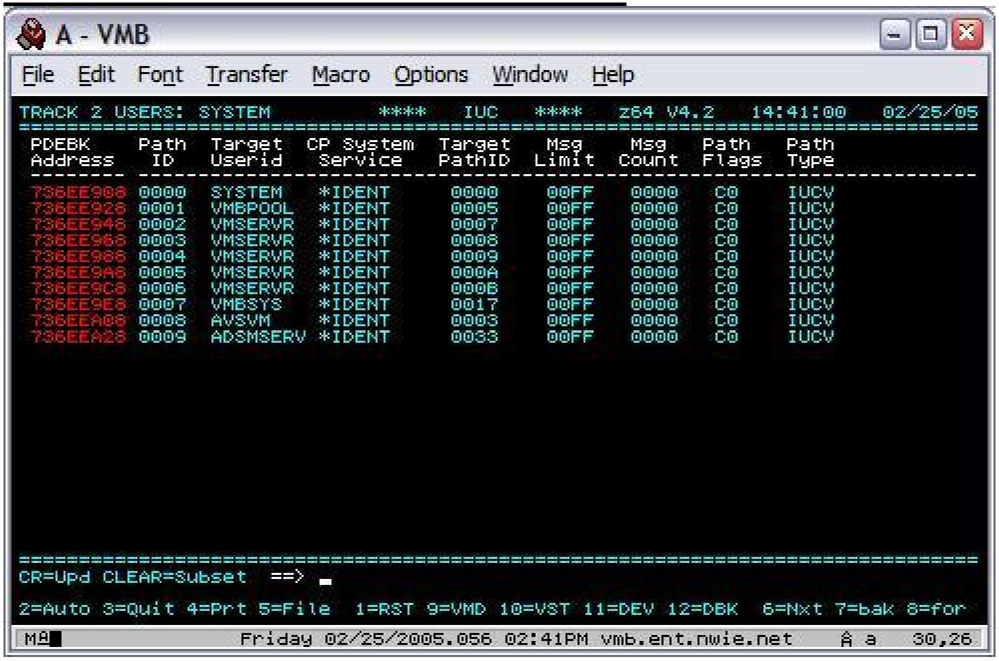
That will display only the *MSG services. To return to displaying all the services, simply enter **IUCV ALL** The CP Service filter keyword has no affect when used for any other userid than SYSTEM.

Screen Shot – Joe User



```
TRACK 1 USER : VINCENJ        ****   IUC   ****   z64 V4.2   14:39:50   02/25/05
===============================================================================
 PDEBK     Path  Target   CP System  Target   Msg    Msg    Path   Path
 Address    ID   Userid    Service   PathID   Limit  Count  Flags  Type
 -------- -----------------------------------------------------------------------
 69FF69B0 0000  VMBSYS                006F     0000   0000   84     APPC
 69FF69D0 0001  VMSERVR               0039     0000   0000   84     APPC
 69FF69F0 0002  SYSTEM     ISFC       000F     0000   0000   84     APPC
 69FF6A10 0003  SYSTEM     ISFC       0011     0000   0000   84     APPC
 69FF6A30 0004  VMBSYS                009F     0000   0000   84     APPC
 69FF6A50 0005  VMBSYS                00B1     0000   0000   84     APPC

===============================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=Nxt 7=bak 8=for
 MA█              Friday 02/25/2005.056 02:39PM vmb.ent.nwie.net    Â a   30,26
```

Screen Shot – SYSTEM with *IDENT filter



```
TRACK 2 USERS: SYSTEM         ****   IUC   ****   z64 V4.2   14:41:00   02/25/05
================================================================================
 PDEBK    Path  Target   CP System  Target   Msg    Msg    Path   Path
 Address   ID   Userid    Service   PathID  Limit  Count  Flags  Type
 -------- -----------------------------------------------------------------------
 736EE908 0000  SYSTEM    *IDENT     0000    00FF   0000    C0    IUCV
 736EE928 0001  VMBPOOL   *IDENT     0005    00FF   0000    C0    IUCV
 736EE948 0002  VMSERVR   *IDENT     0007    00FF   0000    C0    IUCV
 736EE968 0003  VMSERVR   *IDENT     0008    00FF   0000    C0    IUCV
 736EE988 0004  VMSERVR   *IDENT     0009    00FF   0000    C0    IUCV
 736EE9A8 0005  VMSERVR   *IDENT     000A    00FF   0000    C0    IUCV
 736EE9C8 0006  VMSERVR   *IDENT     000B    00FF   0000    C0    IUCV
 736EE9E8 0007  VMBSYS    *IDENT     0017    00FF   0000    C0    IUCV
 736EEA08 0008  AVSVM     *IDENT     0003    00FF   0000    C0    IUCV
 736EEA28 0009  ADSMSERV *IDENT      0033    00FF   0000    C0    IUCV

================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=Nxt 7=bak 8=for
 MA            Friday 02/25/2005.056 02:41PM vmb.ent.nwie.net    A a   30,26
```

MACRO facility

Overview

TRACK's MACRO facility allows you invoke a REXX exec containing READSTRG and NOTE subcommands (like the corresponding DUMPSCAN macro subcommands in IBM's (DVF) Dump Viewing Facility) that will be processed by TRACK. This can be used to read another user's (or your own) virtual storage, logically extending the information provided by TRACK's CMS panel.

In addition to the READSTRG and NOTE subcommands, TRACK also supports TRACKEE, REGS and CMSG subcommands. These subcommands are explained in more detail below.

Macros can be run from within TRACK or on the initial TRACK command. They can either cause execution of the TRACK module to end when they finish or not. Any command that is not recognized as a TRACK command will be assumed to be a TRACK macro, and an attempt will be made to execute commandname TRACK.

Syntax

You can invoke a TRACK macro in 2 different ways. One way is to use the MACRO keyword, which will cause execution of the TRACK command to end when the macro finishes. The syntax of this command is:

TRACK userid MACRO	[macroname [macro_parameters [(macro_options]]]

For example, the command **TRACK * MACRO TEST** will cause TRACK to invoke a REXX exec named TEST TRACK after it has set up the TRACK environment. When the macro has completed the TRACK program ends. You'll never see the inside of the TRACK environment. It's meant to be used as a way for a REXX exec to read a user's virtual storage.

The other way to invoke a TRACK macro is to omit the MACRO keyword. The syntax of this command is:

TRACK userid	[macroname [macro_parameters [(macro_options]]]

For example, the command **TRACK * TEST** will cause TRACK to invoke a REXX exec named TEST TRACK after it has set up the TRACK environment. However, when the macro has completed, the TRACK program does not end, and you will be left at the VMD screen of the user you tracked.

So, the MACRO keyword governs whether or not TRACK terminates when the macro finishes executing. The same holds true when a macro is run as an internal TRACK command. If you don't use the MACRO keyword (for example, typing **TEST** on the command line from within TRACK), you'll end up right where you were when you issued the command to invoke the TRACK macro

If you do specify the MACRO keyword (for example, typing **MACRO TEST** on the command line from within TRACK), when the exec completes, so will TRACK. If you're tracking more than 1 user, it will be as if you quit tracking that user, and you'll be left on another tracked user.

Note that any parameters or options used by the macro must be limited to 8 characters. Actually, if the macro is invoked from within TRACK, it can handle longer parameters and options, but they will be truncated to 8 characters when it's included on the initial TRACK command.

Macro Subcommands

READSTRG macro subcommand

While TRACK macros can contain any REXX exec statements, they can be written to use the READSTRG subcommand, which can read virtual or real storage and save it in a REXX variable. The syntax for READSTRG, which is exactly the same as for DUMPSCAN's READSTRG subcommand, except for the <H> enhancement, is:

READSTRG	[<H>address [flag [length [(VAR variable]]]]

```
H        - get CP real (host) storage
address  - the (real or virtual) storage address

flag     - "%" indicates 24-bit indirect address
         - "?" indicates 31-bit indirect address

length   - a hexadecimal value from 1 to 1000  (defaults to 4)

variable - the name of a REXX variable to hold the result
           (defaults to "RESULT")
```

For example, 'READSTRG E5D 1 (VAR CMSPROG' would read the byte of virtual storage at address x'00000E5D' for the user being tracked, and save the results in variable CMSPROG. 'READSTRG H2A20 4 (VAR CPVAL' would read the 4 bytes of CP real storage at x'00002A20' and save the results in the CPVAL variable.

NOTE macro subcommand

TRACK macros also support the NOTE subcommand. The syntax for this is:

NOTE	'string'

string - a 1 to 80 character string, enclosed in quotes

The output of the NOTE subcommand goes to your console, like the REXX Say command. A separate NOTE panel for the output of NOTE subcommands might be written in the future. In the meantime, you may want to spool your console, if you aren't already doing it.

TRACKEE macro subcommand

In addition to the READSTRG and NOTE subcommands, TRACK also supports TRACKEE and REGS subcommands. TRACKEE returns the userid and/or the VMDBK address of the user being TRACKed when the macro is invoked. By default, it returns both, separated by a blank. The syntax for the TRACKEE subcommand is:

TRACKEE	[USER|VMDBK [(VAR variable]]

```
USER      - return only the TRACKed user's userid
VMDBK     - return only the TRACKed user's VMDBK addr

variable  - the name of a REXX variable to hold the
            result (defaults to "RESULT")
```

REGS macro subcommand

The REGS subcommand returns all or one of the TRACKed user's general, control, access or floating point registers. By default, it returns the TRACKed user's 16 general registers, separated by blanks, in the variable RESULT. The syntax for the REGS subcommand is:

REGS	[Gn|Rn|Cn|An|Fn [(VAR variable]]

```
G or R    - return user's 16 general registers
Gn or Rn  - return user's 4 byte general register Gn,
            where n is 0 thru F <0123456789ABCDEF>

C         - return user's 16 control registers
Cn        - return user's 4 byte control register Cn,
            where n is 0 thru F <0123456789ABCDEF>

A         - return user's 16 access registers
An        - return user's 4 byte access register An,
            where n is 0 thru F <0123456789ABCDEF>

F         - return user's 4 floating point registers
Fn        - return user's 8 byte floating pt reg Fn,
            where n is 0, 2, 4 or 6 <0246>
```

```
variable  - the name of a REXX variable to hold the
            result (defaults to "RESULT")
```

CMSG macro subcommand

TRACK also supports a CMSG subcommand, the output of which is written to TRACK's command line. You can then redisplay this command by using the '?' TRACK command. The syntax for this is:

CMSG	string

string - a 1 to 134 byte character string

One handy use of this subcommand might be for a macro to locate a control block and write a FORMAT command to the command line with the address filled in. If a macro uses this subcommand more than once, only the last one will be available in the command line.

Use

A TRACK subcommand environment (SUBCOM TRACK) is created when the TRACK module is invoked, and it is removed when the module terminates. If you want to halt execution of a macro that you've invoked, the proper way to do so would be to use the CMS immediate command HI (Halt Interpretation), not the CMS immediate command HX (Halt Execution). Your TRACK macros could have code such as the following included in them to exit cleanly upon being ordered to halt interpretation:

```
Signal on halt
  . . .
Halt:    /* Exit cleanly if halted */
Say 'Macro terminated by your HI command'
Exit
```

If you use the HX command to stop a TRACK macro, you will be left with a SUBCOM TRACK that will not be removed, as HX will also halt execution of the TRACK MODULE itself, and it won't have had a chance to clean up.

In the past, filed TRACK screens were saved (by pressing PF5) as files with filetype TRACK. To avoid confusion with TRACK macros, this has been changed so that filed output will be saved with filetype TRACKFIL. You should rename any saved TRACK files you have to filetype TRACKFIL to continue to append to them and to avoid attempting to invoke them as macros.

In order to handle more parameters and options for macros, the command line had to be enlarged. This necessitated moving it up by 1 line and displaying messages on the separator line (line of = signs) instead of on a separate message line. It doesn't look as nice as it used to, but you can now enter longer commands from the TRACK command line.

I have supplied a few TRACK macros from Perry Ruiter of BC Systems to use and to be used as a guide for writing your own macros. To show you a simple one, here's Perry's PROGMAP TRACK macro. It displays programs currently loaded in storage (like the output you get from the PROGMAP command) for the user being tracked.

```
/* Display the program in storage chain */
  Say "Program in storage display"
  'READSTRG 2E8 4 (VAR NUCLDSET'
  If rc ^= 0 Then Exit rc

  do while nucldset ^= 0
    'READSTRG' nucldset '30 (VAR PRGBK'
    parse value (prgbk) with prgnext   9 prgname  25,
                             prgepa   33 prgladdr 41,
                             prglen   49 .
    Say nucldset x2c(prgname) prgepa prgladdr prglen
    nucldset = prgnext
  end

exit
```

NAME – Command and Panel

Syntax

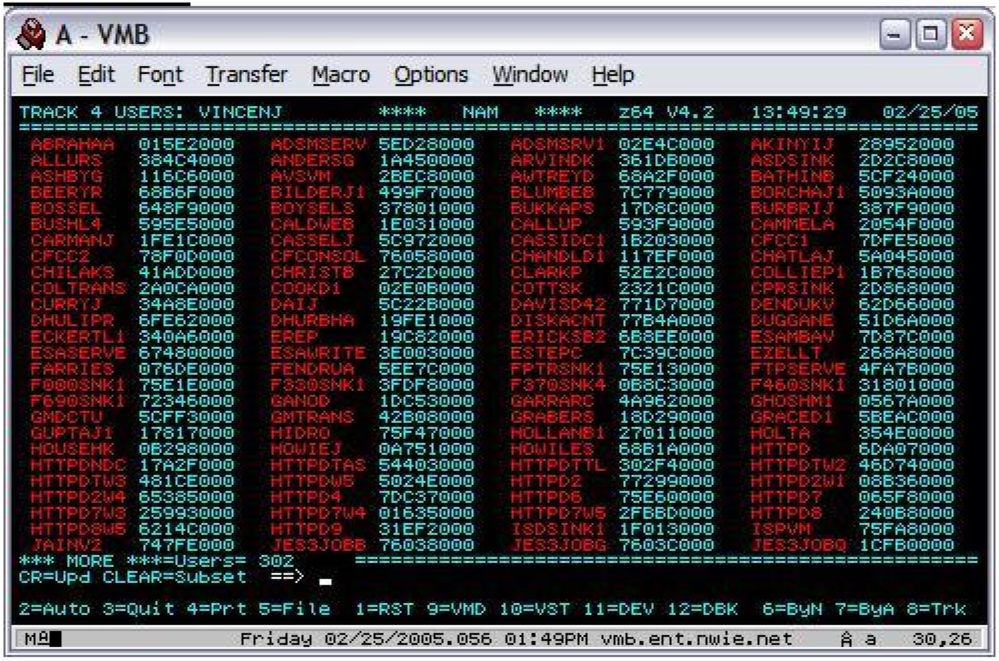
Type NAMes from ANY panel, then press ENTER. NAMes can be abbreviated to N.

You can start tracking a user in the NAM panel by appending the word Names to an initial or internal TRACK command.

For example, from CMS,   enter: Track system n
For example, from TRACK, enter: Track * n

Once in the NAM panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot



```
TRACK 4 USERS: VINCENJ       ****   NAM   ****   z64 V4.2   13:49:29   02/25/05
==============================================================================
 ABRAHAA   015E2000   ADSMSERV  5ED28000   ADSMSRV1  02E4C000   AKINYIJ   28952000
 ALLURS    384C4000   ANDERSG   1A450000   ARVINDK   361DB000   ASDSINK   2D2C8000
 ASHBYG    116C6000   AVSVM     2BEC8000   AWTREYD   68A2F000   BATHINB   5CF24000
 BEERYR    68B6F000   BILDERJ1  499F7000   BLUMBEB   7C779000   BORCHAJ1  5093A000
 BOSSEL    648F9000   BOYSELS   37801000   BUKKAPS   17D8C000   BURBRIJ   387F9000
 BUSHL4    595E5000   CALDWEB   1E031000   CALLUP    593F9000   CAMMELA   2054F000
 CARMANJ   1FE1C000   CASSELJ   5C972000   CASSIDC1  1B203000   CFCC1     7DFE5000
 CFCC2     78F0D000   CFCONSOL  76058000   CHANDLD1  117EF000   CHATLAJ   5A045000
 CHILAKS   41ADD000   CHRISTB   27C2D000   CLARKP    52E2C000   COLLIEP1  1B768000
 COLTRANS  2A0CA000   COOKD1    02E0B000   COTTSK    2321C000   CPRSINK   2D868000
 CURRYJ    34A8E000   DAIJ      5C22B000   DAVISD42  771D7000   DENDUKV   62D66000
 DHULIPR   6FE62000   DHURBHA   19FE1000   DISKACNT  77B4A000   DUGGANE   51D6A000
 ECKERTL1  340A6000   EREP      19C82000   ERICKSB2  6B8EE000   ESAMBAV   7D87C000
 ESASERVE  67480000   ESAWRITE  3E003000   ESTEPC    7C39C000   EZELLT    268A8000
 FARRIES   076DE000   FENDRWA   5EE7C000   FPTRSNK1  75E13000   FTPSERVE  4FA7B000
 F000SNK1  75E1E000   F330SNK1  3FDF8000   F370SNK4  0B8C3000   F460SNK1  31801000
 F690SNK1  72346000   GANOD     1DC53000   GARRARC   4A962000   GHOSHM1   0567A000
 GMDCTU    5CFF3000   GMTRANS   42B08000   GRABERS   18D29000   GRACED1   5BEAC000
 GUPTAJ1   17817000   HIDRO     75F47000   HOLLANB1  27011000   HOLTA     354E0000
 HOUSEHK   0B298000   HOWIEJ    0A751000   HOWILES   68B1A000   HTTPD     6DA07000
 HTTPDNDC  17A2F000   HTTPDTAS  54403000   HTTPDTTL  302F4000   HTTPDTW2  46D74000
 HTTPDTW3  481CE000   HTTPDW5   5024E000   HTTPD2    77299000   HTTPD2W1  08B36000
 HTTPD2W4  65385000   HTTPD4    7DC37000   HTTPD6    75E60000   HTTPD7    065F8000
 HTTPD7W3  25993000   HTTPD7W4  01635000   HTTPD7W5  2FBBD000   HTTPD8    240B8000
 HTTPD8W5  6214C000   HTTPD9    31EF2000   ISDSINK1  1F013000   ISPVM     75FA8000
 JAINV2    747FE000   JES3JOBB  76038000   JES3JOBG  7603C000   JES3JOBQ  1CFB0000
*** MORE ***=Users= 302    ===================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=ByN 7=ByA 8=Trk
 MA           Friday 02/25/2005.056 01:49PM vmb.ent.nwie.net     A a    30,26
```

Description

The NAM panel displays all logged-on users and their VMDBK addresses (users IN-LOGOFF are also displayed). The message line shows the total count of logged-on users.

Use

If the NAMes do not fit on one screen, TRACK puts up a MORE message. Press ENTER to scroll the display, or keep entering NAMes to update the panel without scrolling.

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - Sorts the display by USERID
PF7  - Sorts the display by VMBLOK address
PF8  - Tracks the user from cursor location
PF9  - Displays VMDBK (VMD panel)
PF10 - Display VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

PF6 and PF7 sort the display by VMDBK address and USERID respectively. You can TRACK any display user by TABbing to the userid and hitting PF8.

Commands

See the General Information section for a description of commands valid from ANY panel. There are no commands specific to the NAM panel.

PFX – Command and Panel

Syntax

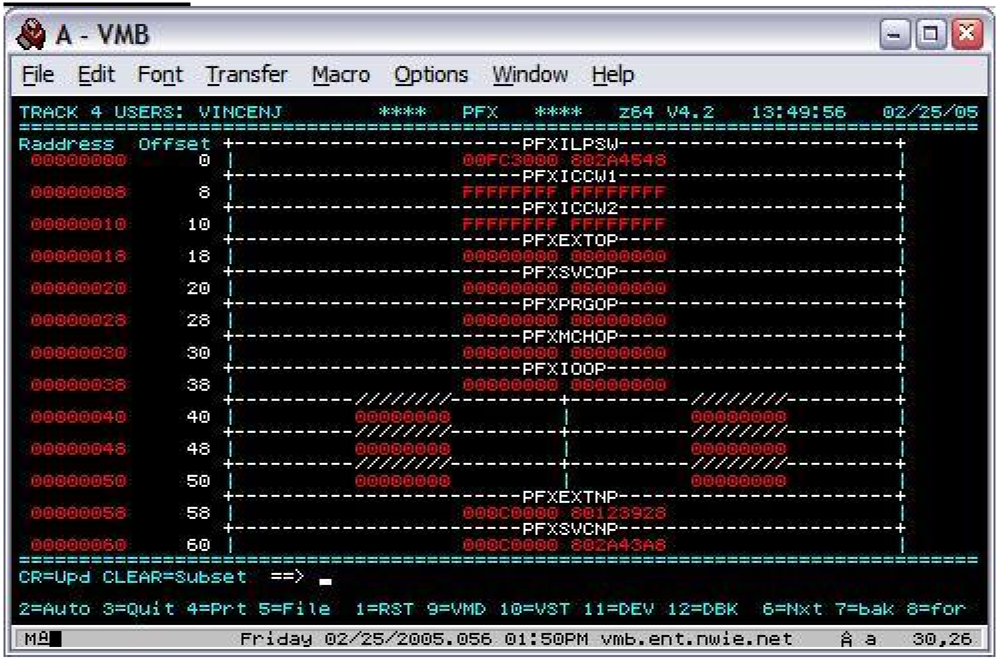
PFX	[I|M|A|0|1|2|3|4|5|6]

You can start tracking a user in the PFX panel by appending any of the above 'PFX' commands to an initial or internal TRACK command.

For example, from CMS, enter: Track system pfx
For example, from TRACK, enter: Track system pfx M

Once in the PFX panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot



```
TRACK 4 USERS: VINCENJ        ****   PFX   ****   z64 V4.2   13:49:56   02/25/05
================================================================================
Raddress  Offset +------------------------PFXILPSW------------------------+
 00000000       0 |                   00FC3000 802A4548                    |
                 +------------------------PFXICCW1------------------------+
 00000008       8 |                   FFFFFFFF FFFFFFFF                    |
                 +------------------------PFXICCW2------------------------+
 00000010      10 |                   FFFFFFFF FFFFFFFF                    |
                 +------------------------PFXEXTOP------------------------+
 00000018      18 |                   00000000 00000000                    |
                 +------------------------PFXSVCOP------------------------+
 00000020      20 |                   00000000 00000000                    |
                 +------------------------PFXPRGOP------------------------+
 00000028      28 |                   00000000 00000000                    |
                 +------------------------PFXMCHOP------------------------+
 00000030      30 |                   00000000 00000000                    |
                 +------------------------PFXIOOP-------------------------+
 00000038      38 |                   00000000 00000000                    |
                 +----------////////---------+----------////////----------+
 00000040      40 |          00000000         |          00000000          |
                 +----------////////---------+----------////////----------+
 00000048      48 |          00000000         |          00000000          |
                 +----------////////---------+----------////////----------+
 00000050      50 |          00000000         |          00000000          |
                 +------------------------PFXEXTNP------------------------+
 00000058      58 |                   000C0000 80123928                    |
                 +------------------------PFXSVCNP------------------------+
 00000060      60 |                   000C0000 802A43A8                    |
================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=Nxt 7=bak 8=for
 MA            Friday 02/25/2005.056 01:50PM vmb.ent.nwie.net    A a   30,26
```

Description

The PFX panel shows low core as per CP's PFXPG dsect. On all systems, "PFX" by itself shows the IPL processor's page 0, as does "PFX I". Entering "PFX M" will get you the Master processor prefix page. Entering "PFX A" will get you the (or an) Alternate processor prefix page, if there is one. If there's more than one, you'll get the next one in the chain. Entering "PFX n", where n = 1 thru 6, will get you a specific processor's prefix page, if that processor address exists. Entering "PFX 0" will show you the storage at the address identified as the processor's prefix value. Note that on VM/XA or VM/ESA systems, absolute page 0 (this page) contains very little useful information, at least as far as I (RJF) can tell.

Use

A number of the values displayed on the PFX panel are imbedded in unprotected fields. You can place your cursor under these values simply by pressing the TAB key. This is useful if you want to display a real or virtual address (TAB to the address, hit PF1 or PF10), display a device (TAB to the device address, hit PF12), or format the address
(type 'FORMAT dsect', TAB to the address, hit ENTER).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - Scrolls PFX display forward or back by lines
PF7  - Scrolls back    one screen's worth of PFX display
PF8  - Scrolls forward one screen's worth of PFX display
PF9  - Displays VMDBK (VMD panel)
PF10 - Display VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

PF7 and PF8 scroll the displayed lines BACK and FORWARD a screen-full at a time. PF6 scrolls forward or back in lines - type a number, or a number preceded by a '-' sign, before hitting PF6 (if you don't type anything, PF6 scrolls the display forward one line).

Commands

See the General Information section for a description of commands valid from ANY panel. You can enter any form of the PFX command from within the PFX panel (or from ANY panel) to switch between PFX's on an AP or MP generated system.

PRT – Command and Panel

Syntax

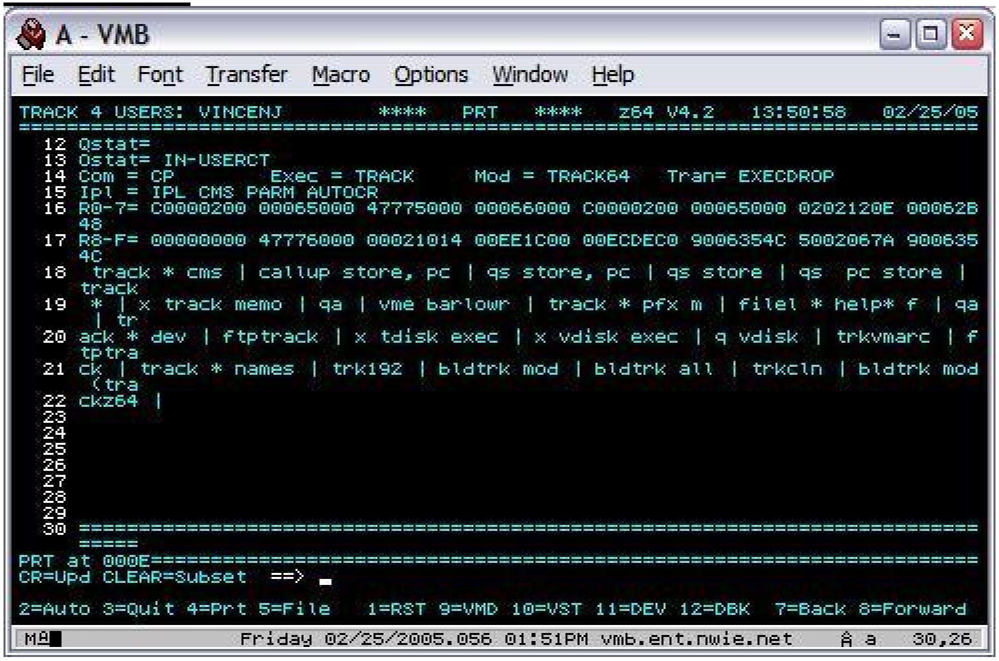
PRT	[address|Next|First]

You can start tracking a user in the PRT panel by appending any of the above 'prt' commands to an initial or internal TRACK command.

For example, from CMS,   enter: Track rscs prt
For example, from CMS,   enter: Track mvs prt 00E
For example, from TRACK, enter: Track mvs prt
For example, from TRACK, enter: Track mvs prt 00E

Once in the PRT panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot



```
TRACK 4 USERS: VINCENJ        ****   PRT   ****   z64 V4.2   13:50:58   02/25/05
================================================================================
   12 Qstat=
   13 Ostat= IN-USERCT
   14 Com = CP          Exec = TRACK     Mod = TRACK64   Tran= EXECDROP
   15 Ipl  = IPL CMS PARM AUTOCR
   16 R0-7= C0000200 00065000 47775000 00066000 C0000200 00065000 0202120E 00062B
      48
   17 R8-F= 00000000 47776000 00021014 00EE1C00 00ECDEC0 9006354C 5002067A 900635
      4C
   18  track * cms | callup store, pc | qs store, pc | qs store | qs  pc store |
      track
   19  * | x track memo | qa | vme barlowr | track * pfx m | filel * help* f | qa
       | tr
   20 ack * dev | ftptrack | x tdisk exec | x vdisk exec | q vdisk | trkvmarc | f
      tptra
   21 ck | track * names | trk192 | bldtrk mod | bldtrk all | trkcln | bldtrk mod
       (tra
   22 ckz64 |
   23
   24
   25
   26
   27
   28
   29
   30 ==========================================================================
      =====
PRT at 000E=====================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK  7=Back 8=Forward
 MA█              Friday 02/25/2005.056 01:51PM vmb.ent.nwie.net    Â a   30,26
```

Description

The PRT panel displays lines from an active spooled printer. By 'active' is meant a printer that is being accessed via the SIO/SIOF instruction.

Because of restrictions in the architecture of CP, TRACK cannot show lines being printed by a dedicated or attached printer.

If you enter 'PRT' by itself, lines from the FIRST active spooled printer are displayed (first in device address order). 'PRT N' skips to the next active spooled printer (if any). 'PRT F' returns to the first active spooled printer. Finally, 'PRT address' shows lines from a specific spooled printer (where 'address' is the virtual address of the printer). If the virtual address is 0F01, remember to specify the leading 0 to see this device, or the F will show the First active spooled printer instead.

Use

TRACK displays the lines in CP's current in-core spool buffer (if any). Each line is preceded by a line number, showing that line's offset from the top of the display. Every time you press ENTER, the entire CP spool buffer is re-read. Thus, the display may jump from showing many lines, to showing just a few (or none).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - No special action, updates the panel
PF7  - Scrolls back    one screen's worth of printer lines
PF8  - Scrolls forward one screen's worth of printer lines
PF9  - Displays VMDBK (VMD panel)
PF10 - Display VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

PF7 and PF8 scroll the displayed lines BACK and FORWARD a screen-full at a time.

Commands

See the General Information section for a description of commands valid from ANY panel. You can enter any form of the PRT command from within the PRT panel (or from ANY panel) to switch between active printers.

PUN – Command and Panel

Syntax

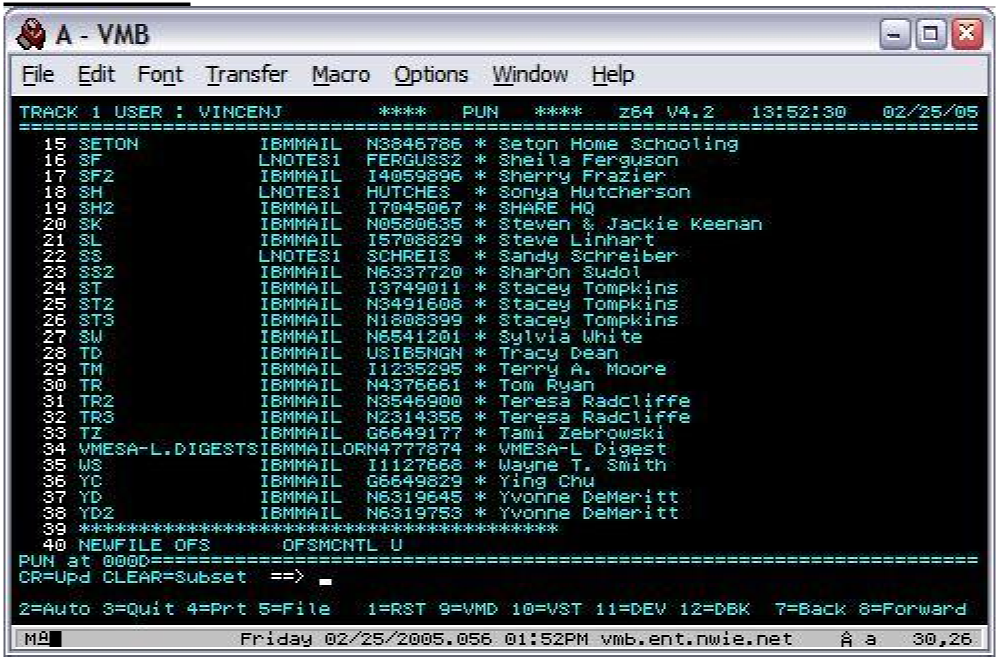
PUN	[address|Next|First]

You can start tracking a user in the PUN panel by appending any of the above 'pun' commands to an initial or internal TRACK command.

For example, from CMS,   enter: Track rscs pun
For example, from CMS,   enter: Track mvs pun 00D
For example, from TRACK, enter: Track mvs pun
For example, from TRACK, enter: Track mvs pun 00D

Once in the PUN panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot



```
TRACK 1 USER : VINCENJ        ****   PUN   ****   z64 V4.2   13:52:30   02/25/05
==================================================================================
  15 SETON          IBMMAIL  N3846786 * Seton Home Schooling
  16 SF             LNOTES1  FERGUSS2 * Sheila Ferguson
  17 SF2            IBMMAIL  I4059896 * Sherry Frazier
  18 SH             LNOTES1  HUTCHES  * Sonya Hutcherson
  19 SH2            IBMMAIL  I7045067 * SHARE HQ
  20 SK             IBMMAIL  N0580635 * Steven & Jackie Keenan
  21 SL             IBMMAIL  I5708829 * Steve Linhart
  22 SS             LNOTES1  SCHREIS  * Sandy Schreiber
  23 SS2            IBMMAIL  N6337720 * Sharon Sudol
  24 ST             IBMMAIL  I3749011 * Stacey Tompkins
  25 ST2            IBMMAIL  N3491608 * Stacey Tompkins
  26 ST3            IBMMAIL  N1808399 * Stacey Tompkins
  27 SW             IBMMAIL  N6541201 * Sylvia White
  28 TD             IBMMAIL  USIB5NGN * Tracy Dean
  29 TM             IBMMAIL  I1235295 * Terry A. Moore
  30 TR             IBMMAIL  N4376661 * Tom Ryan
  31 TR2            IBMMAIL  N3546900 * Teresa Radcliffe
  32 TR3            IBMMAIL  N2314356 * Teresa Radcliffe
  33 TZ             IBMMAIL  G6649177 * Tami Zebrowski
  34 VMESA-L.DIGESTSIBMMAILORN4777874 * VMESA-L Digest
  35 WS             IBMMAIL  I1127668 * Wayne T. Smith
  36 YC             IBMMAIL  G6649829 * Ying Chu
  37 YD             IBMMAIL  N6319645 * Yvonne DeMeritt
  38 YD2            IBMMAIL  N6319753 * Yvonne DeMeritt
  39 ****************************************
  40 NEWFILE OFS      OFSMCNTL U
PUN at 000D=======================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK  7=Back 8=Forward
```

Description

The PUN panel displays lines from an active spooled punch. By 'active' is meant a punch that is being accessed via the SIO/SIOF instruction.

Because of restrictions in the architecture of CP, TRACK cannot show lines being punched by a dedicated or attached punch.

If you enter 'PUN' by itself, lines from the FIRST active spooled punch are displayed (first in device address order). 'PUN N' skips to the next active spooled punch (if any). 'PUN F' returns to the first active spooled punch. Finally, 'PUN address' shows lines from a specific spooled punch (where 'address' is the virtual address of the punch). If the virtual address is 0F01, remember to specify the leading 0 to see this device, or the F will show the First active spooled punch instead.

Use

TRACK displays the lines in CP's current in-core spool buffer (if any). Each line is preceded by a line number, showing that line's offset from the top of the display. Every time you press ENTER, the entire CP spool buffer is re-read. Thus, the display may jump from showing many lines, to showing just a few (or none).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - No special action, updates the panel
PF7  - Scrolls back    one screen's worth of punch lines
PF8  - Scrolls forward one screen's worth of punch lines
PF9  - Displays VMDBK (VMD panel)
PF10 - Display VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

PF7 and PF8 scroll the displayed lines BACK and FORWARD a screen-full at a time.

Commands

See the General Information section for a description of commands valid from ANY panel. You can enter any form of the PUN command from within the PUN panel (or from ANY panel) to switch between active punches

QUIT or DROP - Commands

Syntax

Enter QUIT <userid> from ANY track panel
or
Enter DROP <userid> from ANY track panel

Description

If you are tracking multiple users, QUIT|DROP terminates tracking of the current user (if you just enter 'QUIT' or 'DROP'), or terminates tracking of any tracked user (if you enter 'QUIT userid' or 'DROP userid').

If you are tracking only one user, QUIT|DROP terminates TRACK (same action as hitting PF3).

Please note that PF3 terminates TRACK - it does not simply quit tracking the current tracked user.

RDR – Command and Panel

Syntax

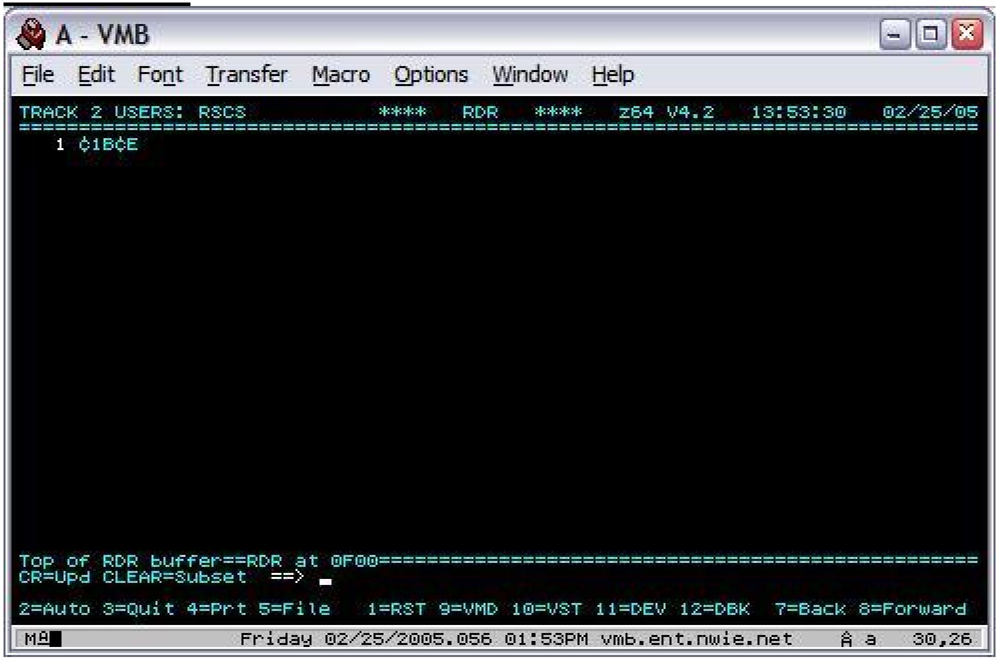
<table>
<tr><td>RDR</td><td>[address|Next|First]</td></tr>
</table>

You can start tracking a user in the RDR panel by appending any of the above 'rdr' commands to an initial or internal TRACK command.

For example, from CMS,   enter: Track rscs rdr
For example, from CMS,   enter: Track mvs rdr 00C
For example, from TRACK, enter: Track mvs rdr
For example, from TRACK, enter: Track mvs rdr 00C

Once in the RDR panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot



```
TRACK 2 USERS: RSCS           ****   RDR   ****   z64 V4.2   13:53:30   02/25/05
================================================================================
   1 ¢1B¢E











Top of RDR buffer==RDR at 0F00==================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK  7=Back 8=Forward
MA                 Friday 02/25/2005.056 01:53PM vmb.ent.nwie.net    Â a   30,26
```

(yeah, boring, but that was the best I could do!)

Description

The RDR panel displays lines from an active spooled reader. By 'active' is meant a reader that is being accessed via the SIO/SIOF instruction.

Because of restrictions in the architecture of CP, TRACK cannot show lines being read from a dedicated or attached reader, nor can it show lines being read via DIAGNOSE 14.

If you enter 'RDR' by itself, lines from the FIRST active spooled reader are displayed (first in device address order). 'RDR N' skips to the next active spooled reader (if any). 'RDR F' returns to the first active spooled reader. Finally, 'RDR address' shows lines from a specific spooled reader (where 'address' is the virtual address of the reader). If the virtual address is 0F01, remember to specify the leading 0 to see this device, or the F will show the First active spooled reader instead.

Use

TRACK displays the lines in CP's current in-core spool buffer (if any). Each line is preceded by a line number, showing that line's offset from the top of the display. Every time you press ENTER, the entire CP spool buffer is re-read. Thus, the display may jump from showing many lines, to showing just a few (or none).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - No special action, updates the panel
PF7  - Scrolls back    one screen's worth of reader lines
PF8  - Scrolls forward one screen's worth of reader lines
PF9  - Displays VMDBK (VMD panel)
PF10 - Display VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

PF7 and PF8 scroll the displayed lines BACK and FORWARD a screen-full at a time.

Commands

See the General Information section for a description of commands valid from ANY panel. You can enter any form of the RDR command from within the RDR panel (or from ANY panel) to switch between active readers.

RETRIEVE – VMD subcommand

Syntax

RETRIEVE	[ON | OFF | ?]

The RETrieve command toggles the display of the RETRIEVE buffer in the VMD panel. You may enter this command from any panel, but it will not take effect until you return to the VMD panel.

If you enter RETrieve by itself, or followed by a "?", the current status of the RETRIEVE display will be shown. The status of the RETRIEVE display is remembered by TRACK even if you leave and return to the VMD panel.

If you are tracking more than one user, the RETrieve display status is separately maintained for each user.

It is possible to generate a version of TRACK that will not display the RETRIEVE buffer, even if this command is used (see TRACK MEMO).

RST – Command and Panel

Description

The RST panel shows real storage in HEX, EBCDIC and ASSEMBLER format. The storage is displayed in 4 byte chunks, 16 bytes to a line, along with the real addresses and offsets.

Syntax

Type a real address on the command line, or place your cursor under a real address (use the TAB key to move your cursor), then press PF1 or PF13 from any track panel (including the RST panel itself).

You can also hit PF1/13 without typing a real address (or placing your cursor under a real address) -- in this case, the last real storage addresses are re-displayed in a cyclic fashion. If you are tracking multiple users, the RST storage addresses are remembered separately for each user.

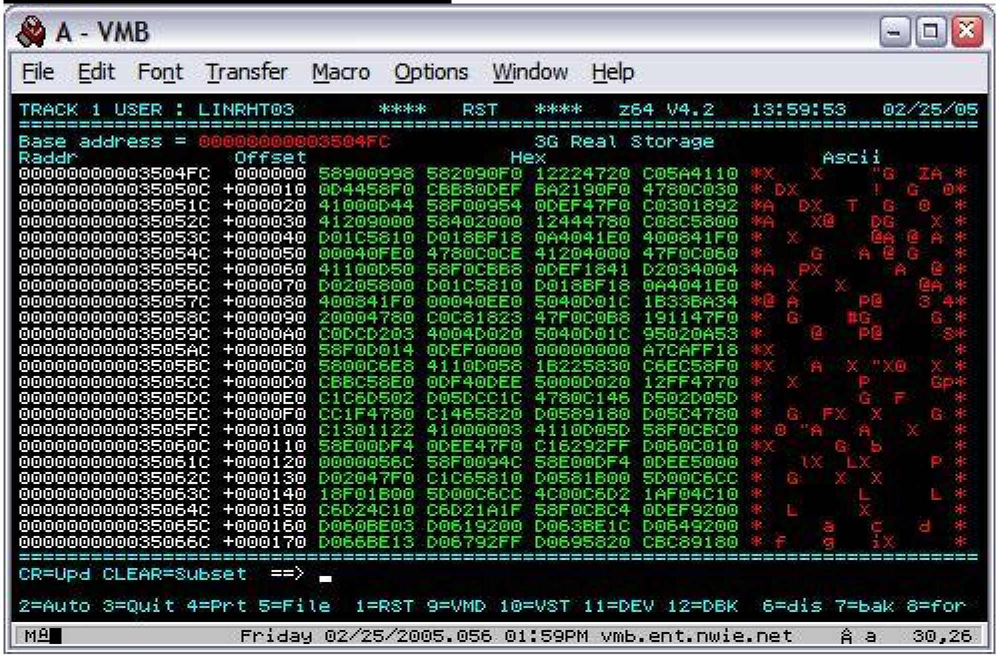
If you hit PF1 without specifying an address, and you have not previously used PF1, the display starts at real address 0.

You can start tracking a user in the RST panel by appending the word 'rst' and a real address to an initial or internal TRACK command.

For example, from CMS,   enter: Track system rst 0
For example, from CMS,   enter: Track system rst 2B28000
For example, from CMS,   enter: Track system rst 481000
For example, from TRACK, enter: Track pvm rst 1E7200

Once in the RST panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

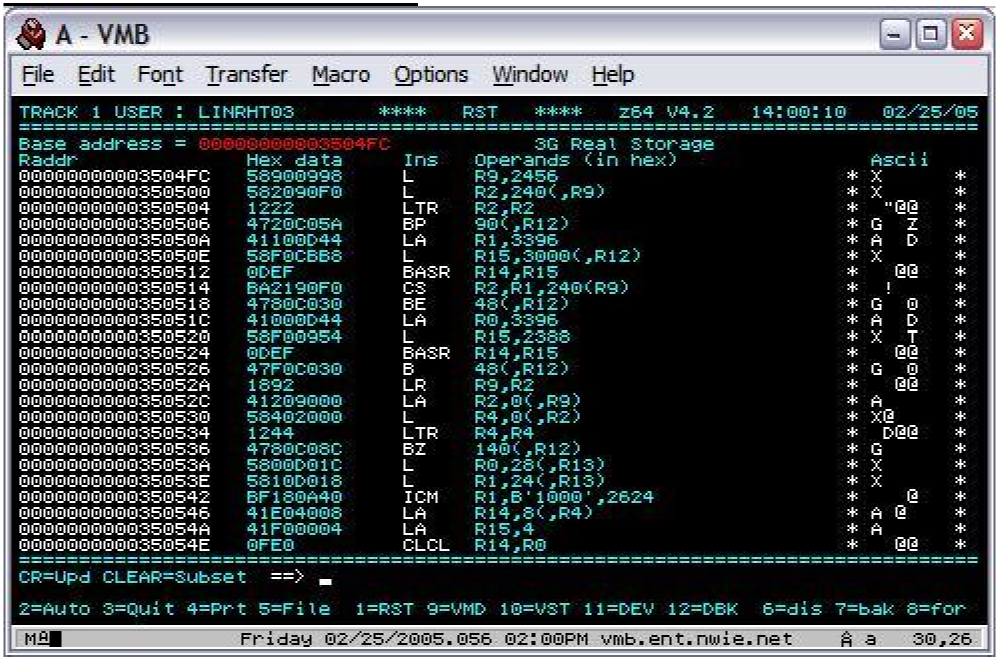
Screen Shot – normal display



```
TRACK 1 USER : LINRHT03       ****   RST   ****   z64 V4.2   13:59:53   02/25/05
================================================================================
Base address = 0000000000003504FC                3G Real Storage
Raddr              Offset                 Hex                         Ascii
0000000000003504FC  000000 58900998 582090F0 12224720 C05A4110 *X    X     "G  ZA *
00000000000035050C +000010 0D4458F0 CBB80DEF BA2190F0 4780C030 * DX        !  G  0*
00000000000035051C +000020 41000D44 58F00954 0DEF47F0 C0301892 *A   DX   T  G   @ *
00000000000035052C +000030 41209000 58402000 12444780 C08C5800 *A    X@    DG    X *
00000000000035053C +000040 D01C5810 D018BF18 0A4041E0 400841F0 *   X       @A @ A *
00000000000035054C +000050 00040FE0 4780C0CE 41204000 47F0C060 *     G    A @ G   *
00000000000035055C +000060 41100D50 58F0CBB8 0DEF1841 D2034004 *A   PX        A  @ *
00000000000035056C +000070 D0205800 D01C5810 D018BF18 0A4041E0 *   X     X      @A *
00000000000035057C +000080 400841F0 00040EE0 5040D01C 1B33BA34 *@  A        P@   3 4*
00000000000035058C +000090 20004780 C0C81823 47F0C0B8 191147F0 *   G      #G      G *
00000000000035059C +0000A0 C0DCD203 4004D020 5040D01C 95020A53 *     @     P@      S*
0000000000003505AC +0000B0 58F0D014 0DEF0000 00000000 A7CAFF18 *X                  *
0000000000003505BC +0000C0 5800C6E8 4110D058 1B225830 C6EC58F0 *X    A   X "X0   X *
0000000000003505CC +0000D0 CBBC58E0 0DF40DEE 5000D020 12FF4770 *   X        P     Gp*
0000000000003505DC +0000E0 C1C6D502 D05DCC1C 4780C146 D502D05D *          G  F     *
0000000000003505EC +0000F0 CC1F4780 C1465820 D0589180 D05C4780 *   G   FX    X    G *
0000000000003505FC +000100 C1301122 41000003 4110D05D 58F0CBC0 * 0 "A     A     X   *
00000000000035060C +000110 58E00DF4 0DEE47F0 C16292FF D060C010 *X        G  b      *
00000000000035061C +000120 0000056C 58F0094C 58E00DF4 0DEE5000 *      %X   LX     P *
00000000000035062C +000130 D02047F0 C1C65810 D0581B00 5D00C6CC *   G     X   X      *
00000000000035063C +000140 18F01B00 5D00C6CC 4C00C6D2 1AF04C10 *          L       L *
00000000000035064C +000150 C6D24C10 C6D21A1F 58F0CBC4 0DEF9200 *   L        X       *
00000000000035065C +000160 D060BE03 D0619200 D063BE1C D0649200 *       a    c    d  *
00000000000035066C +000170 D066BE13 D06792FF D0695820 CBC89180 * f     g     iX     *
================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=dis 7=bak 8=for
```

```
MA            Friday 02/25/2005.056 01:59PM vmb.ent.nwie.net     Å a   30,26
```

Screen Shot – disassemble



```
TRACK 1 USER : LINRHT03       ****   RST   ****   z64 V4.2   14:00:10   02/25/05
================================================================================
Base address = 0000000000003504FC                3G Real Storage
Raddr                Hex data    Ins    Operands (in hex)                Ascii
0000000000003504FC   58900998    L      R9,2456                        * X        *
000000000000350500   582090F0    L      R2,240(,R9)                    * X        *
000000000000350504   1222        LTR    R2,R2                          *   "@@    *
000000000000350506   4720C05A    BP     90(,R12)                       * G   Z    *
00000000000035050A   41100D44    LA     R1,3396                        * A   D    *
00000000000035050E   58F0CBB8    L      R15,3000(,R12)                 * X        *
000000000000350512   0DEF        BASR   R14,R15                        *    @@    *
000000000000350514   BA2190F0    CS     R2,R1,240(R9)                  *   !      *
000000000000350518   4780C030    BE     48(,R12)                       * G   0    *
00000000000035051C   41000D44    LA     R0,3396                        * A   D    *
000000000000350520   58F00954    L      R15,2388                       * X   T    *
000000000000350524   0DEF        BASR   R14,R15                        *    @@    *
000000000000350526   47F0C030    B      48(,R12)                       * G   0    *
00000000000035052A   1892        LR     R9,R2                          *    @@    *
00000000000035052C   41209000    LA     R2,0(,R9)                      * A        *
000000000000350530   58402000    L      R4,0(,R2)                      * X@       *
000000000000350534   1244        LTR    R4,R4                          *   D@@    *
000000000000350536   4780C08C    BZ     140(,R12)                      * G        *
00000000000035053A   5800D01C    L      R0,28(,R13)                    * X        *
00000000000035053E   5810D018    L      R1,24(,R13)                    * X        *
000000000000350542   BF180A40    ICM    R1,B'1000',2624                *     @    *
000000000000350546   41E04008    LA     R14,8(,R4)                     * A @      *
00000000000035054A   41F00004    LA     R15,4                          * A        *
00000000000035054E   0FE0        CLCL   R14,R0                         *    @@    *
================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=dis 7=bak 8=for
```

```
MA            Friday 02/25/2005.056 02:00PM vmb.ent.nwie.net     Å a   30,26
```

Use

All of the values displayed on the RST panel are imbedded in unprotected fields. You can place your cursor under these values simply by pressing the TAB key. This is useful if you want to display a new real or virtual address (TAB to the address, hit PF1 or PF10), or format the address (type 'FORMAT dsect', TAB to the address, hit ENTER).

You can type an address prefixed with a + or - sign, then hit PF1. This will scroll the display forward or back that many hex bytes -- e.g., you type +28 and hit PF1. The display will shift forward X'28' bytes.

PF7 and PF8 scroll the display a screen-full at a time.

PF6 will dis-assemble the displayed storage. Press PF6 again to return the display to it's standard mode. You can scroll forwards or back (via PF7 and PF8) while in dis-assembly mode.

You can search for a HEX or EBCDIC string in real storage. To search for a HEX string, type an X, immediately followed by the hex string, and hit PF1. To search for an EBCDIC string, type a /, immediately followed by a string, and hit PF1. TRACK will scroll the display to the location of the string (The string will always be found since, at a minimum, it will be contained in TRACK's own storage).

PFKeys

```
PF1  - Shifts to a new address if you type, or place cursor under, an
       address; else shifts to any previously displayed real address(es).
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - Toggles dis-assembly display mode
PF7  - Scrolls back    one screen's worth of real addresses
PF8  - Scrolls forward one screen's worth of real addresses
PF9  - Displays VMDBK (VMD panel)
PF10 - Displays VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

Commands

See the General Information section for a description of commands valid from ANY panel.

The only command specific to the RST panel is ASCII. See the *ASCII – Command* section for more information.

SAVEFRM – FRM panel subcommand

The SAVEFRM command may only be issued from the FRM panel in TRACK.

See the FORMAT section on "SAVERM command for the FRM panel" for more information.

TRACK command - External

Syntax

TRACK	userid[command [(options **Options:** FILE

Definitions:

userid

The userid of the user you wish to TRACK. TRACK recognizes the following special userids:

- ***** to track yourself
- **SYSTEM** to track the system VMDBK
- **OP** to track the system operator
- **X'*addr*'** to track the VMDBK at the specified address

command

Any command which is valid within TRACK. You can specify an initial command to go directly to some panel other than the VMD panel. For example:

- TRACK userid FORMAT NUCON V0
  enters TRACK in FRM panel

- TRACK userid CON
  enters TRACK in CON panel

Options:

FILE

Builds the TRACK panel contents and saves to a file called <userid> TRACKFIL A. Once the file is written, TRACK ends without displaying any panels. This option is useful when wrapping TRACK with a web CGI program or other service-machine type application.

TRACK command - Internal

The internal TRACK command is the TRACK command issued from within TRACK, as opposed to the TRACK command issued from CMS.

Syntax

Track	[userid|?[command [(options **Options:** NEW

'Track' can be abbreviated to 'T'.

```
Examples: Track operator
          Track pvm (new
          Track pvm format nucon v0
          Track pvm format nucon v0 (new
          Track pvm con (new
          Track pvm ret off
          T pvm track rscs
          Track pvm t rscs c
          Track pvm aup
          Track * (new
          Track = VMD
```

You can append an INTERNAL track command to your EXTERNAL track command (the one you issue from CMS) - this allows you to enter TRACK with multiple user's immediately tracked. You can also append any other TRACK command (including TRACK itself) to an internal TRACK command.

For example:

From CMS,   you enter 'Track userid t userid2'
From CMS,   you enter 'Track userid t userid2 con'
From CMS,   you enter 'Track userid t userid2 format pfxpg 0'

From TRACK, you enter 'T userid2 t userid3'
From TRACK, you enter 'T userid2 t userid3 con'

Description

You can track up to 8 users at a time via the internal TRACK command. TRACK displays the number of users you are currently tracking in the top left-hand corner of the screen.

Entering 'TRACK ?' will display a list of tracked users on the message line.

If you are tracking more than one user, entering 'Track' by itself will rotate you through the tracked users.

Entering 'Track userid' will start tracking a new user, or will rotate to that user if it is already being tracked.

You can track the same user multiple times by appending the 'NEW' option to the internal Track command. For example, if you are tracking OPERATOR, and you want to start another TRACK session for OPERATOR, you would enter 'TRACK OPERATOR (NEW'. Without the '(NEW' option, TRACK would simply rotate to the existing TRACK session for OPERATOR.

You can also append an initial command to the internal Track command (before any options). Any valid TRACK command can be used as an initial command. For example, to start tracking a new user (or rotate to an already-tracked user) in the CONsole panel, you would enter 'Track userid con'.

Use

It may be useful to track the same user multiple times if you wish to quickly jump between different panel displays for the same user. Instead of shifting between panels via PF keys and commands, you can create multiple TRACK sessions for one user, bring up a different panel in each session, then jump between the sessions (and hence between the panels) simply by entering 'T'.

Please note that you do NOT have to track the same user multiple times to display different panels for a user - from ONE track session, you can shift between the various TRACK panels simply by entering the appropriate TRACK command or hitting the appropriate PF key. Multi-user tracking may simply be a time-saver.

In particular, tracking the same user multiple times may be useful in connection with the FORMAT command. You can format and display up to 8 separate dsects at a time for one user by tracking the same user 8 times, and issuing a different FORMAT command in each TRACK session. Of course, you can also do this from ONE track session, by repeatedly entering a series of FORMAT commands. Doing it with multi-user tracking, however, may be a time-saver.

VMD – Command and Panel

Syntax

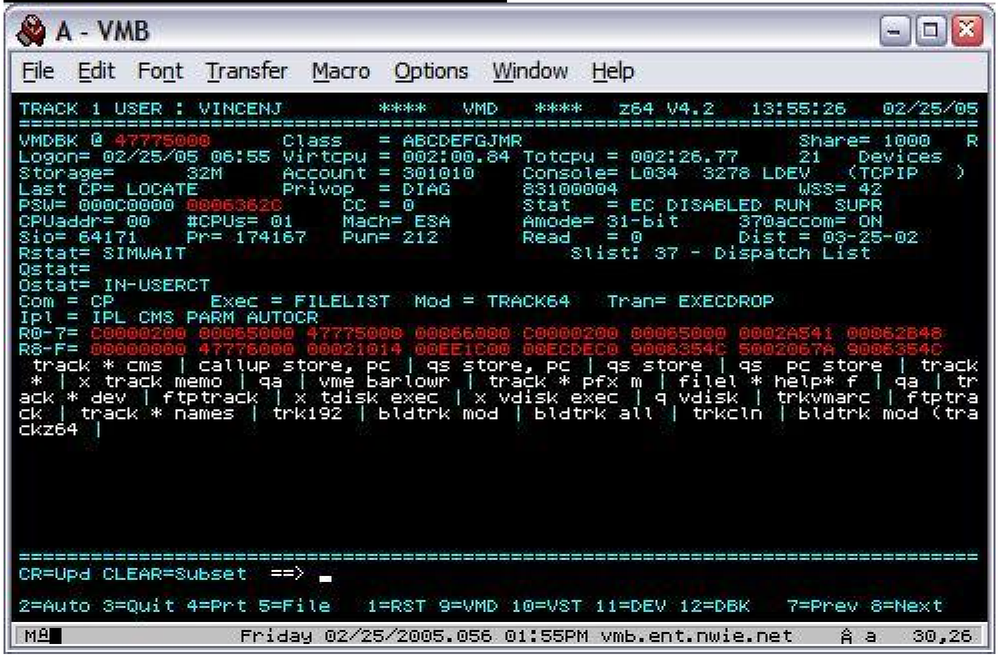
Hit PF9 or PF21 from ANY panel, or type VMD on the TRACK command line.

The VMD panel is the first panel displayed when you enter TRACK. It is also the first panel displayed when you start tracking a new user from within TRACK. You can get to the VMD panel at any time by pressing PF9 or PF21 or typing VMD.

If you want to enter TRACK with some panel OTHER than the VMD panel displayed, append the panel name to the TRACK command. For example, the command 'Track operator con' will enter TRACK in the CON panel. You can also append panel names to the internal form of the TRACK command.

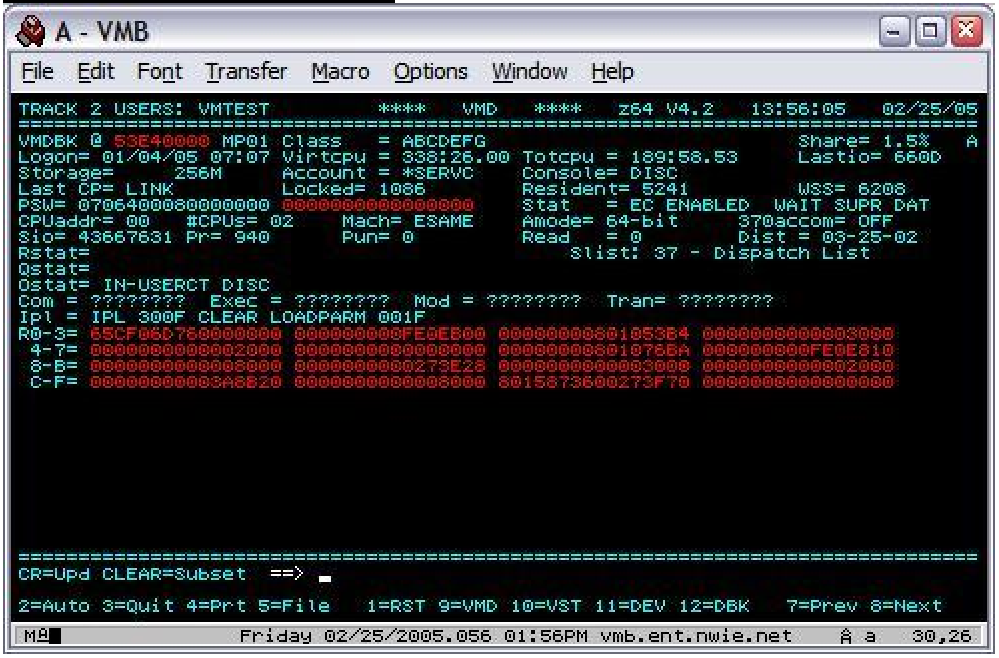
Once in the VMD panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot – 32bit (CMS) userid



```
TRACK 1 USER : VINCENJ        ****   VMD   ****   z64 V4.2   13:55:26   02/25/05
================================================================================
VMDBK @ 47775000       Class    = ABCDEFGJMR                   Share= 1000    R
Logon= 02/25/05 06:55 Virtcpu = 002:00.84 Totcpu = 002:26.77     21   Devices
Storage=      32M     Account = 301010    Console= L034  3278 LDEV   (TCPIP   )
Last CP= LOCATE       Privop  = DIAG      83100004               WSS= 42
PSW= 000C0000 0006362C     CC = 0         Stat    = EC DISABLED RUN  SUPR
CPUaddr= 00   #CPUs= 01    Mach= ESA      Amode= 31-bit      370accom= ON
Sio= 64171    Pr= 174167   Pun= 212       Read    = 0        Dist = 03-25-02
Rstat= SIMWAIT                                Slist: 37 - Dispatch List
Qstat=
Ostat= IN-USERCT
Com = CP          Exec = FILELIST  Mod = TRACK64   Tran= EXECDROP
Ipl = IPL CMS PARM AUTOCR
R0-7= C0000200 00065000 47775000 00066000 C0000200 00065000 0002A541 00062B48
R8-F= 00000000 47776000 00021014 00EE1C00 00ECDEC0 9006354C 5002067A 9006354C
 track * cms | callup store, pc | qs store, pc | qs store | qs  pc store | track
 * | x track memo | qa | vme barlowr | track * pfx m | filel * help* f | qa | tr
ack * dev | ftptrack | x tdisk exec | x vdisk exec | q vdisk | trkvmarc | ftptra
ck | track * names | trk192 | bldtrk mod | bldtrk all | trkcln | bldtrk mod (tra
ckz64 |







================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK   7=Prev 8=Next
 MA         Friday 02/25/2005.056 01:55PM vmb.ent.nwie.net     Â a    30,26
```

Screen Shot – 64bit guest



```
TRACK 2 USERS: VMTEST          ****   VMD   ****   z64 V4.2   13:56:05   02/25/05
================================================================================
VMDBK @ 53E40000 MP01 Class    = ABCDEFG                         Share= 1.5%    A
Logon= 01/04/05 07:07 Virtcpu = 338:26.00 Totcpu = 189:58.53     Lastio= 660D
Storage=     256M     Account = *SERVC    Console= DISC
Last CP= LINK         Locked= 1086        Resident= 5241         WSS= 6208
PSW= 0706400080000000 0000000000000000    Stat    = EC ENABLED  WAIT SUPR DAT
CPUaddr= 00   #CPUs= 02     Mach= ESAME   Amode= 64-bit     370accom= OFF
Sio= 43667631 Pr= 940       Pun= 0        Read    = 0       Dist = 03-25-02
Rstat=                                        Slist: 37 - Dispatch List
Qstat=
Ostat= IN-USERCT DISC
Com = ????????  Exec = ????????  Mod = ????????  Tran= ????????
Ipl = IPL 300F CLEAR LOADPARM 001F
R0-3= 65CF06D760000000 000000000FE0EB00 00000000801053B4 0000000000003000
 4-7= 0000000000002000 0000000000000000 00000000801076BA 000000000FE0E810
 8-B= 0000000000008000 0000000000273E28 0000000000003000 0000000000002000
 C-F= 00000000003A6B20 0000000000008000 8015873600273F70 0000000000000000





================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK   7=Prev 8=Next
 MA           Friday 02/25/2005.056 01:56PM vmb.ent.nwie.net     A a   30,26
```

Description

The VMD panel shows selected information from the tracked user's VMDBK and low core.

The following information is displayed on the VMD panel:

```
    VMDBK  @      :  Address of tracked user's VMDBK
    Class  =      :  Tracked user's CP privilege class(es)
    Share  =      :  Tracked user's share of the system
    Logon  =      :  Logon date and time
    Virtcpu=      :  Virtual CPU time used since LOGON
    Totcpu =      :  Total   CPU time used since LOGON

    Lastio =      :  Virtual address of device with outstanding i/o
Note: When this info is unavailable, the following is shown instead
    nnnn Devices  :  Count of defined virtual devices

    Storage=      :  Total virtual storage (size of virtual machine)
    Account=      :  Tracked user's CP account field
    Console=      :  Address and type of virtual machine console
    Last CP=      :  Last CP command issued

    Privop =      :  Last privileged operation issued
Note: When this info is unavailable, the following are shown instead
    Locked   =    :  Count of locked user pages
    Resident =    :  Count of resident pages

    WSS    =      :  Projected working set size
    PSW    =      :  PSW at last entry to CP
    CC     =      :  Condition Code of PSW
    STAT   =      :  Status from PSW (mode, state)
    CPUaddr=      :  CPU address of virtual CPU
    #CPUs  =      :  Number of guest CPUs
    Mach   =      :  Virtual machine mode type
    Amode  =      :  Addressing mode
    OrigVMD @     :  Origin VMDBK address
    Sio    =      :  Count of non-spooled i/o requests issued since LOGON
    Pr     =      :  Count of lines printed since LOGON
    Pun    =      :  Count of lines punched since LOGON
    Read   =      :  Count of lines read    since LOGON
    Dist   =      :  Distribution code
    R0-7   =      :  Registers 0 to 7  at last entry to CP
    R8-F   =      :  Registers 8 to 15 at last entry to CP
    Rstat  =      :  Bits from VMDBK flag field VMDRSTAT
    Slist:        :  Scheduling list definition status
    Qstat  =      :  Bits from VMDBK flag field VMDQSTAT
    Ostat  =      :  Bits from VMDBK flag field VMDOSTAT
```

The following five fields will be displayed as '???' if TRACK is unable to read the user's virtual page 0 (the fields will be meaningless if the tracked user is not running CMS).

```
    Ipl    =      :  The CMS NUCON field SYSNAME
    Com    =      :  The CMS NUCON field LASTCMND
    Exec   =      :  The CMS NUCON field LASTEXEC
    Mod    =      :  The CMS NUCON field LASTLMOD
    Tran   =      :  The CMS NUCON field LASTTMOD
```

The rest of the panel displays the user's RETRIEVE buffer, if any, unless the RET OFF command has been issued. If you have entered RET OFF, enter RET ON to turn the display of the RETRIEVE buffer back on (if you get a bad return code from the RET command, then the RETRIEVE buffer display has been disabled).

Use

A number of the values displayed on the VMD panel are imbedded in unprotected fields. You can place your cursor under these values simply by pressing the TAB key. This is useful if you want to display a real or virtual address (TAB to the address, hit PF1 or PF10), display a device (TAB to the device address, hit PF12), or format the address (type 'FORMAT dsect', TAB to the address, hit ENTER).

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - No special action, just updates the panel
PF7  - Shifts to PREVIOUS user in VMDBK chain (cyclic)
PF8  - Shifts to NEXT     user in VMDBK chain (cyclic)
PF9  - No special action, just updates the panel
PF10 - Displays VIRTUAL storage (VST panel)
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

Commands

See the General Information section for a description of commands valid from ANY panel. What follows are commands specific to the VMD panel.

RETRIEVE subcommand

Syntax: RETrieve <ON|OFF|?>

See the *RETRIEVE – VMD subcommand* section for more information.

VST – Command and Panel

Description

The VST panel shows virtual storage in HEX, EBCDIC and ASSEMBLER format. The storage is displayed in 4 byte chunks, 16 bytes to a line, along with the virtual addresses, offsets and corresponding real address (the real address is the location in CP real storage where the virtual storage currently resides). If the displayed storage crosses a page boundary, TRACK displays the real address of both real pages in which the virtual storage resides (they need not be contiguous).

TRACK attempts to read virtual storage by searching CP's segment and page tables. If the storage is resident or if the storage has been locked, this search will succeed, and the storage will be read and displayed.

If the search fails, TRACK attempts to issue a LOCK command, to force CP to page in the storage. If you do not have the privileges necessary to issue the LOCK command, TRACK will issue an error message to that effect. If the LOCK succeeds, TRACK will read the storage and immediately issue an UNLOCK command. Thus, TRACK will not be LOCKING pages for any longer than it takes to read them (generally under 5 hundredths of a second). Note further that TRACK will not UNLOCK pages that have been previously locked, because, in this case, TRACK will locate the storage via CP's segment and page tables.

Syntax

Type a virtual address on the command line, or place your cursor under a virtual address (use the TAB key to move your cursor), then press PF10 or PF22 from any track panel (including the VST panel itself).

You can also hit PF10/22 without typing a virtual address (or placing your cursor under a virtual address) -- in this case, the last virtual addresses are re-displayed in a cyclic fashion. If you are tracking multiple users, the VST storage addresses are remembered separately for each user.

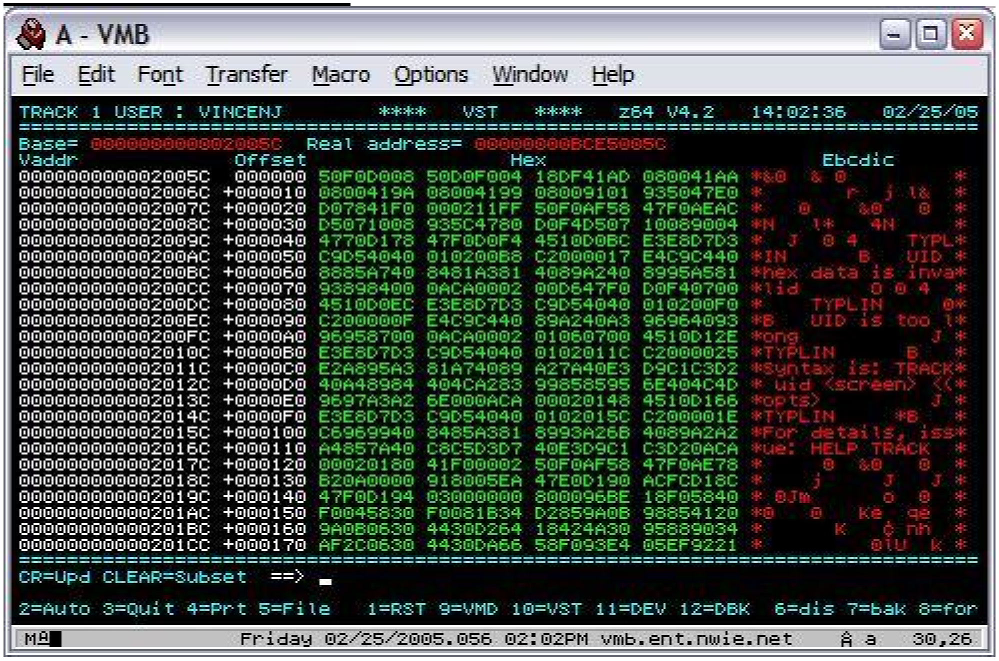
If you hit PF10 without specifying an address, and you have not previously used PF10, the display starts at virtual address 0.

You can start tracking a user in the VST panel by appending the word 'vst' and a virtual address to an initial or internal TRACK command.

For example, from CMS,   enter: Track operator vst 20000
For example, from CMS,   enter: Track pvm vst 30000
For example, from TRACK, enter: Track rscs vst 0

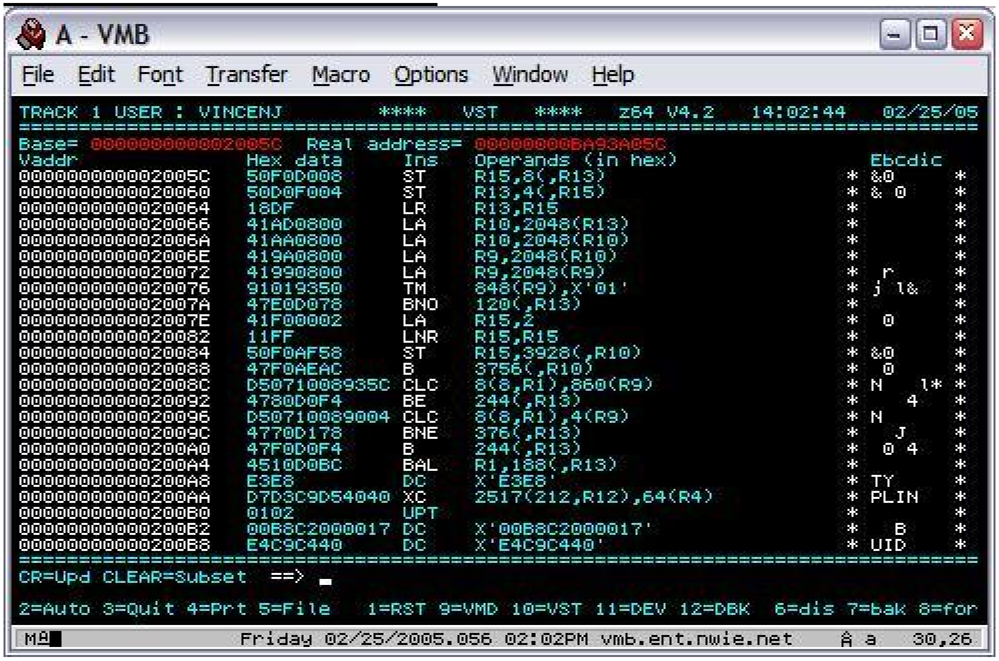
Once in the VST panel, simply keep pressing ENTER to have the displayed information updated, or press PF2 to have TRACK automatically update the displayed information (try HELP TRACK AUTO for more information on automatic panel updates).

Screen Shot - EBCIDIC



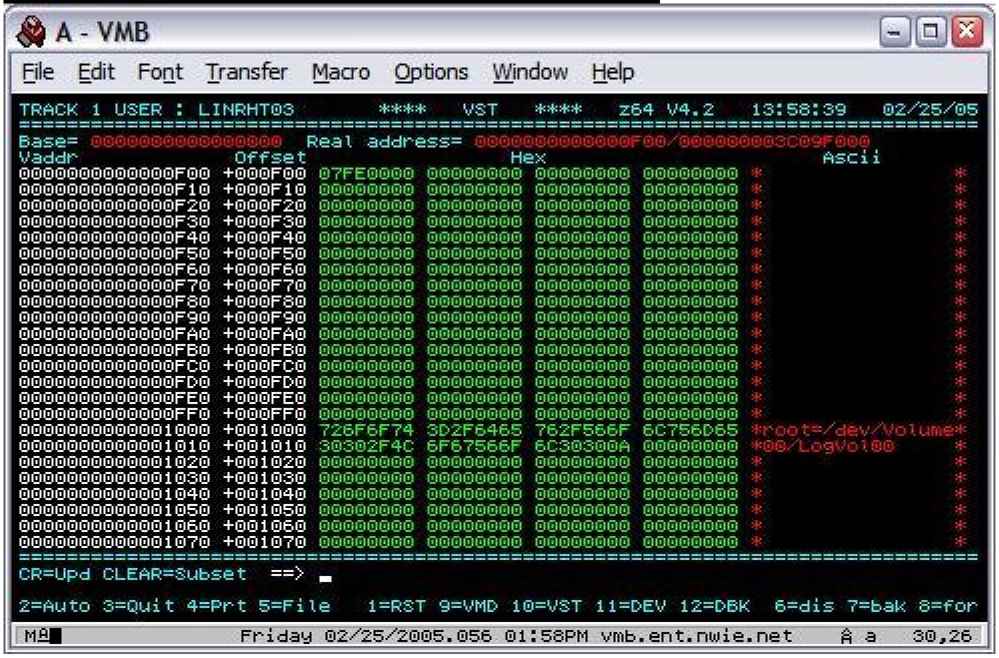
```
TRACK 1 USER : VINCENJ        ****   VST   ****   z64 V4.2   14:02:36   02/25/05
================================================================================
Base= 000000000002005C  Real address= 00000000BCE5005C
Vaddr             Offset                  Hex                         Ebcdic
000000000002005C  000000 50F0D008 50D0F004 18DF41AD 080041AA *&0  & 0        *
000000000002006C +000010 0800419A 08004199 08009101 935047E0 *       r  j l& *
000000000002007C +000020 D07841F0 000211FF 50F0AF58 47F0AEAC *   0    &0   0 *
000000000002008C +000030 D5071008 935C4780 D0F4D507 10089004 *N    l*   4N   *
000000000002009C +000040 4770D178 47F0D0F4 4510D0BC E3E8D7D3 *  J  0 4   TYPL*
00000000000200AC +000050 C9D54040 010200B8 C2000017 E4C9C440 *IN      B   UID *
00000000000200BC +000060 8885A740 8481A381 4089A240 8995A581 *hex data is inva*
00000000000200CC +000070 93898400 0ACA0002 00D647F0 D0F40700 *lid      O 0 4  *
00000000000200DC +000080 4510D0EC E3E8D7D3 C9D54040 010200F0 *    TYPLIN     0*
00000000000200EC +000090 C200000F E4C9C440 89A240A3 96964093 *B   UID is too l*
00000000000200FC +0000A0 96958700 0ACA0002 01060700 4510D12E *ong           J *
000000000002010C +0000B0 E3E8D7D3 C9D54040 0102011C C2000025 *TYPLIN      B   *
000000000002011C +0000C0 E2A895A3 81A74089 A27A40E3 D9C1C3D2 *Syntax is: TRACK*
000000000002012C +0000D0 40A48984 404CA283 99858595 6E404C4D * uid <screen> <(*
000000000002013C +0000E0 9697A3A2 6E000ACA 00020148 4510D166 *opts>         J *
000000000002014C +0000F0 E3E8D7D3 C9D54040 0102015C C200001E *TYPLIN     *B   *
000000000002015C +000100 C6969940 8485A381 8993A268 4089A2A2 *For details, iss*
000000000002016C +000110 A4857A40 C8C5D3D7 40E3D9C1 C3D20ACA *ue: HELP TRACK  *
000000000002017C +000120 00020180 41F00002 50F0AF58 47F0AE78 *     0  &0   0  *
000000000002018C +000130 B20A0000 918005EA 47E0D190 ACFCD18C *    j    J   J  *
000000000002019C +000140 47F0D194 03000000 800096BE 18F05840 * 0Jm      o  0  *
00000000000201AC +000150 F0045830 F0081B34 D2859A0B 98854120 *0    0   Ke  qe *
00000000000201BC +000160 9A0B0630 4430D264 18424A30 95889034 *       K   ¢ nh *
00000000000201CC +000170 AF2C0630 4430DA66 58F093E4 05EF9221 *        0lU   k *
================================================================================
CR=Upd CLEAR=Subset  ==>

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=dis 7=bak 8=for
MA              Friday 02/25/2005.056 02:02PM vmb.ent.nwie.net   Â a   30,26
```

Screen Shot – disassembled



```
TRACK 1 USER : VINCENJ        ****   VST   ****   z64 V4.2   14:02:44   02/25/05
================================================================================
Base= 000000000002005C  Real address= 00000000BA93A05C
Vaddr              Hex data     Ins   Operands (in hex)                 Ebcdic
000000000002005C   50F0D008     ST    R15,8(,R13)                     * &0      *
0000000000020060   50D0F004     ST    R13,4(,R15)                     * & 0     *
0000000000020064   18DF         LR    R13,R15                         *         *
0000000000020066   41AD0800     LA    R10,2048(R13)                   *         *
000000000002006A   41AA0800     LA    R10,2048(R10)                   *         *
000000000002006E   419A0800     LA    R9,2048(R10)                    *         *
0000000000020072   41990800     LA    R9,2048(R9)                     *  r      *
0000000000020076   91019350     TM    848(R9),X'01'                   * j l&    *
000000000002007A   47E0D078     BNO   120(,R13)                       *         *
000000000002007E   41F00002     LA    R15,2                           *  0      *
0000000000020082   11FF         LNR   R15,R15                         *         *
0000000000020084   50F0AF58     ST    R15,3928(,R10)                  * &0      *
0000000000020088   47F0AEAC     B     3756(,R10)                      *  0      *
000000000002008C   D5071008935C CLC   8(8,R1),860(R9)                 * N    l* *
0000000000020092   4780D0F4     BE    244(,R13)                       *    4    *
0000000000020096   D50710089004 CLC   8(8,R1),4(R9)                   * N       *
000000000002009C   4770D178     BNE   376(,R13)                       *   J     *
00000000000200A0   47F0D0F4     B     244(,R13)                       *  0 4    *
00000000000200A4   4510D0BC     BAL   R1,188(,R13)                    *         *
00000000000200A8   E3E8         DC    X'E3E8'                         * TY      *
00000000000200AA   D7D3C9D54040 XC    2517(212,R12),64(R4)            * PLIN    *
00000000000200B0   0102         UPT                                   *         *
00000000000200B2   00B8C2000017 DC    X'00B8C2000017'                 *   B     *
00000000000200B8   E4C9C440     DC    X'E4C9C440'                     * UID     *
================================================================================
CR=Upd CLEAR=Subset  ==>

2=Auto 3=Quit 4=Prt 5=File   1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=dis 7=bak 8=for
MA              Friday 02/25/2005.056 02:02PM vmb.ent.nwie.net   Â a   30,26
```

Screen Shot – with ASCII ON (Linux guest)



```
TRACK 1 USER : LINRHT03        ****   VST   ****   z64 V4.2   13:58:39   02/25/05
================================================================================
Base= 0000000000000000  Real address= 0000000000000F00/000000003C09F000
Vaddr                 Offset                   Hex                    Ascii
0000000000000F00 +000F00 07FE0000 00000000 00000000 00000000 *                *
0000000000000F10 +000F10 00000000 00000000 00000000 00000000 *                *
0000000000000F20 +000F20 00000000 00000000 00000000 00000000 *                *
0000000000000F30 +000F30 00000000 00000000 00000000 00000000 *                *
0000000000000F40 +000F40 00000000 00000000 00000000 00000000 *                *
0000000000000F50 +000F50 00000000 00000000 00000000 00000000 *                *
0000000000000F60 +000F60 00000000 00000000 00000000 00000000 *                *
0000000000000F70 +000F70 00000000 00000000 00000000 00000000 *                *
0000000000000F80 +000F80 00000000 00000000 00000000 00000000 *                *
0000000000000F90 +000F90 00000000 00000000 00000000 00000000 *                *
0000000000000FA0 +000FA0 00000000 00000000 00000000 00000000 *                *
0000000000000FB0 +000FB0 00000000 00000000 00000000 00000000 *                *
0000000000000FC0 +000FC0 00000000 00000000 00000000 00000000 *                *
0000000000000FD0 +000FD0 00000000 00000000 00000000 00000000 *                *
0000000000000FE0 +000FE0 00000000 00000000 00000000 00000000 *                *
0000000000000FF0 +000FF0 00000000 00000000 00000000 00000000 *                *
0000000000001000 +001000 726F6F74 3D2F6465 762F566F 6C756D65 *root=/dev/Volume*
0000000000001010 +001010 30302F4C 6F67566F 6C30300A 00000000 *00/LogVol00     *
0000000000001020 +001020 00000000 00000000 00000000 00000000 *                *
0000000000001030 +001030 00000000 00000000 00000000 00000000 *                *
0000000000001040 +001040 00000000 00000000 00000000 00000000 *                *
0000000000001050 +001050 00000000 00000000 00000000 00000000 *                *
0000000000001060 +001060 00000000 00000000 00000000 00000000 *                *
0000000000001070 +001070 00000000 00000000 00000000 00000000 *                *
================================================================================
CR=Upd CLEAR=Subset  ==> _

2=Auto 3=Quit 4=Prt 5=File  1=RST 9=VMD 10=VST 11=DEV 12=DBK  6=dis 7=bak 8=for
```

```
MA          Friday 02/25/2005.056 01:58PM vmb.ent.nwie.net    A a    30,26
```

Use

All of the values displayed on the VST panel are imbedded in unprotected fields. You can place your cursor under these values simply by pressing the TAB key. This is useful if you want to display a new virtual or real address (TAB to the address, hit PF10 or PF1), or format the address (type 'FORMAT dsect', TAB to the address, hit ENTER).

You can type an address prefixed with a + or - sign, then hit PF10. This will scroll the display forward or back that many hex bytes -- e.g., you type +28 and hit PF10. The display will shift forward X'28' bytes.

PF7 and PF8 scroll the display a screen-full at a time.

PF6 will dis-assemble the displayed storage. Press PF6 again to return the display to it's standard mode. You can scroll forwards or back (via PF7 and PF8) while in dis-assembly mode.

PFKeys

```
PF1  - Displays REAL storage (RST panel)
PF2  - Enters AUTO-UPDATE mode
PF3  - Quit
PF4  - Prints the displayed screen (does NOT close the printer)
PF5  - Files the displayed screen  (in file 'userid TRACKFIL A')
PF6  - Toggles dis-assembly display mode
PF7  - Scrolls back    one screen's worth of virtual addresses
PF8  - Scrolls forward one screen's worth of virtual addresses
PF9  - Displays VMDBK (VMD panel)
PF10 - Shifts to a new address if you type, or place cursor under, an
```

```
       address; else shifts to previously displayed virtual addresses.
PF11 - Displays DEVICES (DEV panel)
PF12 - Displays DEVICE BLOCKS (DBK panel)
```

Commands

See the General Information section for a description of commands valid from any panel.

The only command specific to the VST panel is ASCII. See the *ASCII – Command* section for more information.