

Advanced Technical Support
At the Washington Systems Center

**CP3KVMXT
User's Guide**

**Gretchen Frye
frye@us.ibm.com**

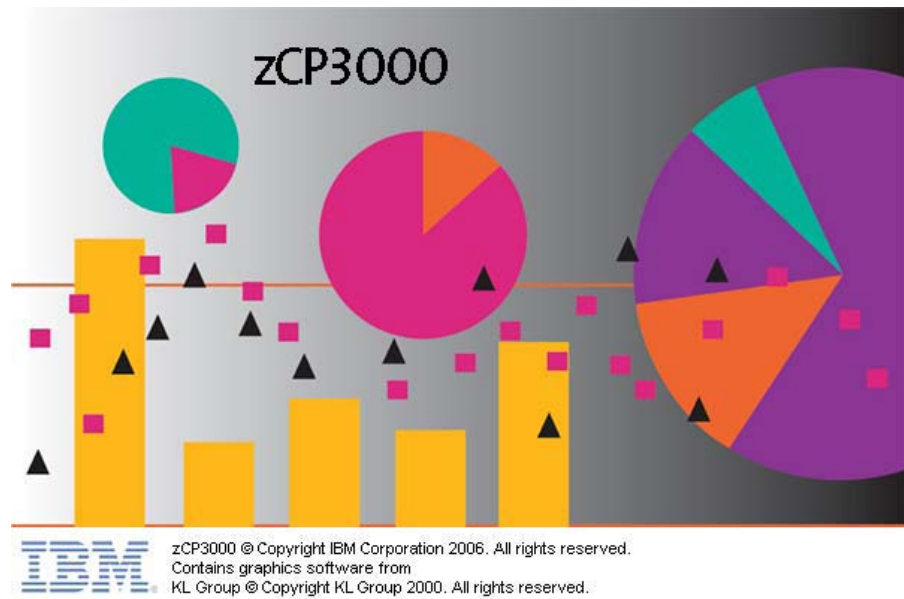


Table of Contents

OBJECTIVES	1
CHANGES	1
MODELING Z/VM AND Z/OS SYSTEMS WITH ZCP3000	2
VM EXTRACT UTILITY OVERVIEW:	2
ZCP3000 OVERVIEW:	2
INSTALLATION PLANNING	3
INPUT DATA REQUIREMENTS:	3
CP MONITOR REQUIREMENTS	3
USING MONWRITE TO	5
SAVE MONITOR DATA IN A FILE.....	5
ESTIMATING THE SPACE NEEDED FOR A MONITOR DATA FILE	2
CP3KVMXT INSTALLATION	3
SETTING UP A USERID TO RUN CP3KVMXT.....	3
<i>Userid requirements for saved monitor data</i>	3
<i>Userid requirements for realtime monitor data</i>	3
SOFTWARE PREREQUISITES	4
CP3KVMXT INSTALLATION.....	4
RUNNING CP3KVMXT	5
START WITH A SMALL TEST RUN	5
GROUPING USERS INTO WORKLOADS (THE UCLASS FILE):.....	5
<i>UCLASS Syntax</i>	6
COOKBOOK: USING REALTIME MONITOR DATA:.....	6
COOKBOOK: USING SAVED MONITOR DATA:	9
APPENDIX A: SYNTAX	12
<i>Examples</i>	13
APPENDIX B: PROBLEM DETERMINATION	14
APPENDIX C: SPEEDING UP CP3KVMXT:	16
APPENDIX D: EDFI FIELD DESCRIPTIONS	17
TABLE 1: HEAD MACRO – HEADER INFORMATION	17
TABLE 2: CEC MACRO – INFORMATION ABOUT THIS MAINFRAME	18
TABLE 3: SYS INFORMATION ABOUT THIS Z/VM IMAGE	19
TABLE 4: SAMP – INTERVAL TIMESTAMPS	21
TABLE 5: WORK – USER/WORKLOAD GROUP SAMPLES	21
TABLE 6: BCU DASD I/O INFORMATION	22
TABLE 7: BCU PATH INFORMATION	24
TABLE 8: ACTUATOR INFORMATION	25
APPENDIX E: RESOURCES	27
WEBSITES:	27
PUBLICATIONS:	27
CONTACTS:	27

Objectives

This program reads CP monitor data in order to produce a file which is directly useable by zCP3000. It is intended for use by customers to extract performance information which will then be used by IBM employees or IBM Business Partners to produce a capacity plan.

CP3KVMXT and its prerequisite packages, Monview and VMARC, may be obtained from the IBM VM Download Packages website at <http://www.vm.ibm.com/download/packages/>. Use of tools from this site is conditional on the "Agreement for Downloading from the IBM VM Download Library" at <http://www.vm.ibm.com/download/license.html> Please note that the zCP3000 modeling tool itself is available only to IBM employees and entitled IBM Business Partners.

IBM employees and IBM Business partners can access additional tools for capacity planning, including zCP3000, zPCR, and the CP2KEXTR z/OS data extract facility, can be found on the Capacity Planning Support website at

<http://w3.ibm.com/support/americas/wsc/cpsproducts.html>

or on the Business Partners website at

<http://partners.boulder.ibm.com/src/atmastr.nsf/WebIndex/PRS1762> .

Changes

Nov 2008 – Fields added to SYS and WORK to support new storage modeling functions:

- DPA_LO and DPA_HI fields added to SYS macro.
- The meaning of CSAVAILV and ESAVAILV has changed to mean pages in the DPA (<2GB and >2GB) that are in the available frame queue.
- WSSV and VSTORV added to WORK macro (projected working set size and virtual storage size)
- WCS, WES, WIO, WPAGE deleted from WORK (redundant; WCSV et al will still be included).
- The meaning of WCS and WES has changed to mean user-owned resident pages below and above the 2GB line.

Note: the CP3KVMXT will no longer use the same version number as the CP3KVMXT tool, since this has created the impression that the documentation would always be updated when the utility was updated. The utility is updated much more frequently than the documentation because of bug fixes.

Modeling z/VM and z/OS systems with zCP3000

zCP3000 is a PC-based JAVA productivity tool, designed to do performance analysis and capacity planning for IBM System z platforms running various SCP/workload environments. zCP3000 is designed to analyze logically partitioned processors running multiple images, FICON and ESCON DASD configurations, and cooperative Sysplex environments.

VM Extract Utility Overview:

The zCP3000 VM extract utility (hence referred to as CP3KVMXT) generates an Enterprise Data File (EDF) for input into zCP3000, which is used by IBMers and entitled Business Partners to generate a capacity plan. An EDF file is generated for each monitor file processed, and then merged into a single EDF file for each partition on each CEC being studied. The fields and macros generated by CP3KVMXT are documented in an [appendix](#) of this document. The utility is designed to be run on the same image that the monitor data was collected on. While it is possible to ship the monitor data elsewhere and run CP3KVMXT against it there, this is discouraged because of the sheer volume of data generated by the monitor and the sensitivity of the data to format changes introduced during file transfer.

zCP3000 Overview:

zCP3000 is a PC-based productivity tool designed to do performance analysis and capacity planning for IBM System z, and IBM z Series running various SCP/workload environments. zCP3000 is directly tied to the LSPR processor performance tables in the zPCR tool, and is designed to be able to model complex configurations, including multiple partitions, multiple operating systems, specialty engine usage, and the cooperative processing environment known as Sysplex. The Capacity Planning Support (CPS) team at the Washington Systems Center provides a comprehensive suite of tools designed to assist in capacity planning. Please visit our website at

<http://w3.ibm.com/support/americas/wsc/cps.html> (for IBM internal)

<http://partners.boulder.ibm.com/src/atmastr.nsf/WebIndex/PRS1762> (for IBM BPs)

or contact your IBM representative for more information about zCP3000 and other capacity planning tools.

Installation Planning

Input Data Requirements:

CP3KVMXT uses the monitor records created by the z/VM Monitor Service for input. These records must be in the format specified by the published z/VM Programming Interface. CP3KVMXT will *not* read the history or acum files, nor any sort of formatted performance report produced by the Performance Toolkit for z/VM, VMPRF, Tivoli Performance Monitor, or any OEM product.

Capacity planning requires a much longer period of measurement data than is typically used for performance tuning. Running it for one peak hour is not enough, particularly for systems running guests which may have a substantial scheduled workload that runs during off-hours. Collecting performance data should be considered a cost of doing business. Preferably, the capacity plan will contain an entire (business) week's worth of data.

CP3KVMXT is compatible with the data collection done by realtime performance monitors such as z/VM Performance Toolkit which also use z/VM Monitor Records. Multiple users can simultaneously access the monitor service. For information on the MONDCSS saved segment and running the monitor, please refer to "Chapter 9. Monitoring Performance Using CP Monitor" in SC24-5999 z/VM Performance Guide (see Appendix B for the url of the z/VM Publications library).

Please note that the default MONDCSS size is often too small, especially for guest environments when there are a lot of devices defined, and the result of this is lost data. If CP3KVMXT is using realtime data, it will shut down if the data is being dumped into the small MONDCSS at a faster rate than CP3KVMXT can process it. See the section entitled "[Speeding Up CP3KVMXT](#)", later in this document.

CP Monitor Requirements

For an excellent summary of the how to collect monitor data, please refer to <http://www.vm.ibm.com/perf/tips/collect.html> .

For use by CP3KVMXT, the CP Monitor must be enabled to collect event and sample data for the Processor, Storage, User, and I/O (dasd class) domains. If a realtime performance monitor is currently running, the monitor is already running and is probably ok like it is. The Monitor sample interval should be between 1 and 5 minutes long; real-time monitors tend to use a shorter interval, such as 1 to 2 minutes, which is fine, but there are tradeoffs when choosing the interval length:

- Shorter intervals increase the overhead of the monitor (because it is gathering data more frequently), and increase the CPU consumption of the userid running CP3KVMXT (because there are more samples to process).
- When writing to a monitor data file, in addition to more overhead, shorter intervals use a tremendous amount of disk space and lengthen the CP3KVMXT processing time.
- Longer intervals, on the other hand, run the risk of missing data. Internal counters of frequent events “wrap” and reset to zero once they’ve reached 65,535. If the number in the counter is less than the last time, CP3KVMXT logically adds 65,535 to get the difference. If it wraps more than once during a monitor interval, that information is lost and your system will look like it has a whole lot less I/O than it really does.
- Longer intervals are often incompatible with the requirements of realtime monitors such as IBM’s VM/Real Time Monitor or FCON, which need a short interval to gauge the effects of performance tuning.
- Monitor overhead can be lessened by monitoring only what you need. CP3KVMXT does not use data from the Scheduler, Seeks, or Application domains, and it uses only dasd class I/O. Here are sample commands to measure just these, sampling at 1 minute intervals:

```

CP MONITOR SAMPLE DISABLE ALL INT 1 MIN
CP MONITOR SAMPLE ENABLE PROC
CP MONITOR SAMPLE ENABLE STOR
CP MONITOR SAMPLE ENABLE USER ALL
CP MONITOR SAMPLE ENABLE I/O CLASS DASD
CP MONITOR EVENT DISABLE ALL
CP MONITOR EVENT ENABLE PROC
CP MONITOR EVENT ENABLE STOR
CP MONITOR EVENT ENABLE USER ALL
CP MONITOR EVENT ENABLE I/O CLASS DASD

```

Using Monwrite To Save Monitor Data in a File

If you want to save monitor information in a file, you will need a userid to connect to the monitor and use the MONWRITE utility to write out the data. This does not need to be the same userid that will, later, run the CP3KVMXT utility. It does not need anything higher than class G authority, but it will need special statements in its directory entry to be able to load the MONDCSS segment and connect to the monitor service. It needs a virtual storage size that will not conflict with MONDCSS, and it needs the following 2 statements to connect to the monitor:

```
IUCV *MONITOR MSGLIMIT 255  
NAMESAVE MONDCSS
```

Typically, the monitor is started and stopped from a scheduler program such as WAKEUP, running on an operations utility userid.

Once the monitor is enabled, start up the MONWRITE machine, so that it can write monitor data out from the monitor DCSS into a file. Typically, the monwrite machine is started and stopped by the same WAKEUP machine used to start and stop the monitor.

From the userid which will run CP3KVMXT, link to the Monwrite disk. Make sure the Monview and CP3KVMXT program files and the customer's UCLASS file are present (they can be on any accessed disk). Start the CP3KVMXT utility. Refer to [Command Syntax](#) later in this doc for syntax. In the parmfile or as the first argument to CP3KVMXT, you should specify the input monitor data fileid. This too can be automated by a WAKEUP type utility userid, to start up CP3KVMXT once the monitor has been stopped.

Progress messages are issued at interval and duration ends, and after every 5000 records processed. The utility will end with a message about where the new EDF file was written.

Using the same process as when running CP3KVMXT with realtime monitor data, you can create an EDF from the intermediate results if the final EDF was not created for some reason.

Send the EDF file to the IBMer or Business Partner performing the Capacity Analysis

For more information on setting up a Monwrite userid, please refer to “Chapter 9. Monitoring Performance Using CP Monitor” in SC24-5999 z/VM Performance Guide (see Appendix B for the URL of the z/VM Publications library).

Estimating the Space Needed for a Monitor Data File

The sheer volume of z/VM Monitor data generated on a standard system is surprising, and for a system with complex guests and farms of dasd, it becomes important to take steps to collect just the desired monitor data. Tuning the running environment of CP3KVMXT is also important when you are running with realtime monitor data because all monitor records are processed, whether they are useful or not. Please review the section entitled "Speeding up CP3KVMXT" in Appendix B if you will be using realtime monitor data.

The amount of space needed for the monitor data depends on many things, including the number of users, dasd devices, partitions, the complexity of the guests, and the length of the CP monitor interval. A table profiling 5 actual case studies is included below to help you, but the best way to estimate how much space the monitor data will be needed for your system is to collect data for 15 minutes and extrapolate from there.

	Lx/WAS/Plex	Bank	University	IBM 7 VMs	VM/Linux
	2084-303	9672-RA6	9672-RC6	9672-R96	2064-104
Total RDEV	785	616	128	2035	612
Max Logged Users	26	24	35	1035	25
Real CPs	12	1	3	9	8
LCPs this image	7	1	3	5	3
partitions	9	1	2	7	8
Guests w/ multiple engines	7	0	22	0	3
# cache dev	785	616	128	2035	612
CP Monitor Interval (min)	5	1	5	5	1
collection period (hrs)	0.5	0.2	23	1	0.1
MB per interval	0.7	0.4	0.2	1.7	0.6
MB per hour	8.8	22.0	2.4	20.2	35.3
4k blocks (actual)	4742	1220	13927	5172	1206

CP3KVMXT Installation

Setting up a userid to run CP3KVMXT

There are 2 basic ways to run CP3KVMXT: using Monitor Data which has been written to a file, or reading data directly from the Monitor DCSS in main storage. The requirements for the userid are different depending on how you want to run the utility.

Userid requirements for saved monitor data

Most sites set up a separate userid for CP3KVMXT, although it is possible to use an existing userid. Although you could use the same userid that collected the data for the saved monitor data file, generally it is better to have a different userid, so that the MONWRITE utility can be collecting current monitor data, while CP3KVMXT is concurrently processing the previous day's data.

This userid will require read access to the mdisk or SFS directory containing the saved monitor data. It will also require a reasonably large virtual storage size. A 64meg virtual machine should be enough, but 128 meg may be needed for a system with many DASD devices. The virtual storage size may have to be adjusted up or down so as not to conflict with the reserved addresses of saved systems such as CMS.

Please refer to the "Estimating the Space Needed" section for disk space requirements. For a quick estimate, divide the number of DASD rdevices in half to get the approximate number of 4k blocks of writable disk space required. For example, a system with 612 DASD addresses would need $306 * 4k = 1.2\text{meg}$ of writeable disk space for a single run of the extract utility. This does *not* include the space for the monitor data itself.

Userid requirements for realtime monitor data

A dedicated userid is required to process realtime monitor data. This userid needs to be authorized to connect to the monitor. Authorization is granted by statements in the userid's directory entry that look something like this:

```
IUCV *MONITOR MSGLIMIT 255  
NAMESAVE MONDCSS
```

It will also require a reasonably large virtual storage size. A 64meg virtual machine should be enough, but 128 meg may be needed for a system with many DASD devices. The virtual storage size may have to be adjusted up or down so as not to conflict with the address of the MONDCSS.

Software Prerequisites

For step-by-step instructions on how to collect performance data for a zCP3000 capacity planning study, please refer to

IBM: <http://w3.ibm.com/support/techdocs/atmastr.nsf/WebIndex/PRS2875>

IBM BP: <http://partners.boulder.ibm.com/src/atmastr.nsf/WebIndex/PRS2875>

The VMARC and MONVIEW tools are available from the VM Download package website. See Appendix B for URLs.

1. CMS Pipelines – this is part of the z/VM operating system, and should already be available to all VM users. If you can run this command, you already have access:
pipe q version
2. FTP or a file transfer program of 3270-emulator. You will need this to install the software prerequisites and to download the finished EDF file.
3. VMARC - VMARC is a data compaction and file archiving utility for VM/CMS. The attributes of files compacted with VMARC are protected during file transfer, which is critical to machine readable formats such as in z/VM Monitor records
4. MONVIEW provides tools for reading z/VM monitor data
5. CP3KVMXT to read monitor files and produce EDF files. See detail below.

CP3KVMXT Installation

Please see the url above for step-by-step instructions, or refer to installation instructions on the <http://www.vm.ibm.com/download> website and in the CP3KVMXT PACKAGE file for the most current information. But basically, here's what you do:

- Download VMarc: <http://www.vm.ibm.com/download/vmarc.module>
- Download Monview: <http://www.vm.ibm.com/download/packages/monview.vmarc>
- Download CP3KVMXT:
<ftp://ftp.software.ibm.com/software/mktsupport/techdocs/CP3KVMXT.vmarc>
- upload vmarc.module, monview.vmarc, and CP3KVMXT.vmarc in binary mode to the VM system where CP3KVMXT will be running.
- On the VM system, run the following commands:
PIPE < VMARC MODULE A | deblock cms | > VMARC MODULE A
VMARC UNPK MONVIEW VMARC fm = = outputFileMode (replace
VMARC UNPK CP3KVMXT VMARC fm = = outputFileMode (replace

Running CP3KVMXT

Because of the amount of z/VM Monitor data generated on a standard system, it is important to take steps to collect just the desired monitor data and to ensure that the process runs correctly. There are 2 basic ways to run CP3KVMXT: using Monitor Data which has been written to a file, or reading data directly from the Monitor DCSS in main storage as it's being written by the CP Monitor service. Whichever method you use, you should start with a small test run.

Start With A Small Test Run

A useful strategy is to plan to collect a 15 minute sample and send it all of the way through the process, including downloading it to the PC, and loading into zCP3000. It's much easier to find and fix process problems when there's not so much data involved, when you're not under pressure to process that day's data so that MONWRITE has room to write out the next day's data. The small test run will also generate a list of active users and their relative cpu utilization, which will help you to group into "workloads" those users who share a similar business function.

Grouping Users into Workloads (the UCLASS file):

In VM, "workloads" are groups of VM userids representing different business units, which will grow at different rates in the zCP3000 capacity planning model. By using the UCLASS file, you can group users into user-defined workload groups, like LINUXDB, MVS, NETWORK, LINUXWEB, VMSYS, etc

By default, the CP3KVMXT utility will look for a file with the same name as your output EDF file, but with the filetype of UCLASS. If it can't find this, it will use a file called GENERIC UCLASS to group user performance into workload groups. Users that don't fit any pattern found in the UCLASS file will form their own workload group, so you will potentially end up with many tiny workloads, more than your customer wants to see in a graph.

It is worth taking the time to tailor the groups so that they make sense to the people who will be seeing the results. When you tailor the UCLASS file, the extract utility has to be re-run in order to use the new workload grouping. Each time you run CP3KVMXT, a list of users active during that period, the UCLASS they were assigned to, and the CPU resource used by that individual userid, are logged in the ACTVUSRS file. You can use this to tailor your UCLASS grouping file into approximately even groups.

Alternatively, you can choose to merge workloads within zCP3000, but this can be tedious process if there are several dozen small userids that need to be merged into a "workload".

UCLASS Syntax

The UCLASS file can be on any accessible filemode. There are 3 columns in the UCLASS file.

1. The userid is in columns 1-8 and does not contain embedded blanks or any special characters other than _ and \$. A trailing * is allowed as a wildcard, i.e. LIN* in column 1 and LINUX in column 10 will accumulate work done by LINUPRD, LINTESTA, LINDBSVR, etc., userids into the LINUX workload.
2. Workload group in columns 10-17; no embedded blanks or special characters except _ and \$.
3. Columns 18 and up are ignored by CP3KVMXT and can be used for comments.

It may be useful to work with someone who is familiar with the work running on the system in order to group the userids into business unit based workloads.

Cookbook: Using Realtime Monitor Data:

The following procedure is for running CP3KVMXT realtime. The procedure for using saved monitor data follows this section.

Note: running against realtime monitor data can be the more complicated of the two methods, especially if the site does not already run the monitor. If tuning is needed, you will need to work closely with the site system programmer to size the dcsc and try the various suggestions for optimizing CP3KVMXT's performance.

1. Plan to run a test against 15 minutes of monitor data and reviewing the results before starting the real collection.
2. Review the setup requirements in the "Setting up CP3KVMXT" section and ensure that the CP3KVMXT userid has sufficient disk space, virtual storage, the prerequisite software packages installed, and has been granted authorization to connect to the MONDCSS segment.
3. Make sure that the userid which will run CP3KVMXT has been authorized to connect to the Monitor DCSS. Authorization is granted by these statements in the userid's directory entry:

```
IUCV *MONITOR MSGLIMIT 255  
NAMESAVE MONDCSS
```

4. If there is a virtual storage addressing conflict with the Monitor DCSS, you will have to adjust the virtual storage size of the userid running CP3KVMXT. You will know this right away when you start running CP3KVMXT if you get the message *DMSDCS343E Storage in range 01400000-02FFFFFF for MONDCSS in use*. If this happens, try increasing the size of your virtual machine (using CP Q V

STOR, then DEF STOR ___M). The max virtual storage setting in the directory entry may have to be increased to relieve the conflict.

5. If you will be using a parm file to supply runtime prompts, edit the CP3KVMXT PARMS file. The meaning of the various parms is explained in comments in the CP3KVMXT PARMS file. By default, CP3KVMXT will prompt for parameters. Parms supplied on the command line for input, output, and uclass fileid, will override the corresponding parm in the PARMS file.
6. If monitor data is not normally gathered at this site, then you must enable and start the CP monitor from an E class userid. Please refer to the [Monitor Requirements](#) for detail. The monitor should run during core business hours, including both peak and non-peak times. From the class E userid, you can CP Q MONITOR to see its status. You should see something like this:

```
MONITOR EVENT ACTIVE  BLOCK  4  PARTITION  3584
MONITOR DCSS NAME - MONDCSS
CONFIGURATION SIZE    68 LIMIT      1 MINUTES\
CONFIGURATION AREA IS FREE
USERS CONNECTED TO *MONITOR - MONWRITE
MONITOR  DOMAIN ENABLED
PROCESSOR DOMAIN ENABLED
STORAGE  DOMAIN ENABLED
SCHEDULER DOMAIN DISABLED
SEEKS    DOMAIN DISABLED
USER     DOMAIN ENABLED
  ALL USERS ENABLED
I/O      DOMAIN ENABLED
THE FOLLOWING DEVICES ARE ENABLED:
2000-203F 2300-247F 2540-257F 2600-263F 2C00-2EFF 4000-427F
4300-44FF 4800-4848 4880-48C8 4900-4948 4980-49C8 4A00-4BFF
5000-50FF 5300-53FF 6300-637F 6400-647F 7000-77FF 8420-842F
8440-844F AC00-AFFF B300-B3FF B700-B7FF B900-BAFF BD00-BDFF
BF00-C1FF D800-D853 D900-D953 E900-E93F
```

```
APPLDATA DOMAIN DISABLED
MONITOR SAMPLE ACTIVE
  INTERVAL  5 MINUTES
  RATE     1.00 SECONDS
MONITOR DCSS NAME - MONDCSS
CONFIGURATION SIZE    241 LIMIT      1 MINUTES
CONFIGURATION AREA IS FREE
USERS CONNECTED TO *MONITOR - MONWRITE
MONITOR  DOMAIN ENABLED
SYSTEM   DOMAIN ENABLED
PROCESSOR DOMAIN ENABLED
STORAGE  DOMAIN ENABLED
```

USER DOMAIN ENABLED
ALL USERS ENABLED
I/O DOMAIN ENABLED
THE FOLLOWING DEVICES ARE ENABLED:
2000-203F 2300-247F 2540-257F 2600-263F 2C00-2EFF 4000-427F
4300-44FF 4800-4848 4880-48C8 4900-4948 4980-49C8 4A00-4BFF
5000-50FF 5300-53FF 6300-637F 6400-647F 7000-77FF 8420-842F
8440-844F AC00-AFFF B300-B3FF B700-B7FF B900-BAFF BD00-BDFF
BF00-C1FF D800-D853 D900-D953 E900-E93F
APPLDATA DOMAIN DISABLED
Ready; T=0.01/0.04 20:26:08

7. Start the CP3KVMXT utility to create a 15 minute test EDF file, referring to [Command Syntax](#) later in this doc for syntax, and specifying MONDCSS as the input source. Do not use the CP3KVMXT PARMS file for the test run.
8. To stop data collection, any class E userid can issue the following command to manually stop the monitor with this command so that it can gracefully finish writing out interval data:
CP MONITOR STOP
Note: The monitor needs to be stopped only for an instant to complete the Monitor Data file. You can start the monitor back up again right away for other applications that use it (i.e., VMRTM or FCONX).
9. You may also have a scheduler/WAKEUP machine issue the CP MONITOR STOP after a certain interval.
10. After addressing any problems, start the CP3KVMXT utility to create the full EDF file, specifying MONDCSS as the input source.
 - If you specified that CP3KVMXT use the parameter file by using the PARMFILE option, it will pick up runtime parameters from there and override them with any command line options. Otherwise, you will see several prompts, the first of which is to enter a title for the study, including the customer name and perhaps a date.
 - If you did not specify the CP3KVMXT parm file, the utility will ask you for a DURATION. This is the period of time that a SAMPS record in zCP3000 will represent. You should pick a value which is a multiple of the CP Monitor interval and will generate a reasonable number of data points on a utilization graph (for example, 4 hours of data and a duration of 15 minutes would generate a graph with 16 data points). For an 8 hour measurement period, 30 minutes would be a good value for the DURATION.
11. Progress messages are issued at interval and duration ends, and after every 5000 records processed. The utility will end with a message about where the new EDF file was written. If you want to end CP3KVMXT earlier than the time you specified (via prompt or parmfile), you can do this by hitting the <enter> key once, letting it sit in VM READ for 2 minutes, and then hitting the <enter key> again. This will time out the MONDCSS connection. Entering HX will also work, but will likely abend CMS and require that the userid re-ipl.

12. If applicable, prepare the UCLASS file to group performance information for a set of userids into one workload. Refer to the section [Grouping Users into Workloads](#) for detail.
13. If the final EDF was not created for some reason (i.e., you HX'd it), you can create an EDF from the intermediate results file with the following command¹
`CP3KVMXT fn SAMPS A (restart`
14. Download the EDF file in ascii mode and send it to the IBMer or Business Partner performing the Capacity Analysis. Please make sure the record length formatting has been preserved, and that the lines do not "wrap".

Cookbook: Using Saved monitor data:

There are 2 basic ways to run CP3KVMXT: using Monitor Data which has been written to a file, or reading data directly from the Monitor DCSS in main storage. The following procedure is for using saved monitor data. The procedure for [running CP3KVMXT with realtime data](#) precedes this section.

Note: running against saved monitor data is usually the simpler of the two methods, as long as sufficient disk space for the monitor file is available.

1. Plan to run a test against 15 minutes of monitor data and reviewing the results before starting the real collection.
2. Review the setup requirements in the "Setting up CP3KVMXT" section and ensure that the CP3KVMXT userid has sufficient disk space, virtual storage, the prerequisite software packages installed, and has access to a monitor data file.
3. If you will be using a parm file to supply runtime prompts, edit the CP3KVMXT PARMS file. The meaning of the various parms is explained in comments in the CP3KVMXT PARMS file. By default, CP3KVMXT will prompt for parameters. Parms supplied on the command line for input, output, and uclass fileid, will override the corresponding parm in the PARMS file.
4. If monitor data is not normally gathered at this site, then you must enable and start the CP monitor from an E class userid. Please refer to the [Monitor Requirements](#) for detail. The monitor should run during core business hours, including both peak and non-peak times. From the class E userid, you can CP Q MONITOR to see its status. You should see something like this:

```
MONITOR EVENT ACTIVE  BLOCK 4  PARTITION 3584
MONITOR DCSS NAME - MONDCSS
CONFIGURATION SIZE 68 LIMIT 1 MINUTES\
CONFIGURATION AREA IS FREE
```

¹ Refer to the [CP3KVMXT Syntax](#) section for more detail.

USERS CONNECTED TO *MONITOR - MONWRITE
MONITOR DOMAIN ENABLED
PROCESSOR DOMAIN ENABLED
STORAGE DOMAIN ENABLED
SCHEDULER DOMAIN DISABLED
SEEKS DOMAIN DISABLED
USER DOMAIN ENABLED
ALL USERS ENABLED
I/O DOMAIN ENABLED
THE FOLLOWING DEVICES ARE ENABLED:
2000-203F 2300-247F 2540-257F 2600-263F 2C00-2EFF 4000-427F
4300-44FF 4800-4848 4880-48C8 4900-4948 4980-49C8 4A00-4BFF
5000-50FF 5300-53FF 6300-637F 6400-647F 7000-77FF 8420-842F
8440-844F AC00-AFFF B300-B3FF B700-B7FF B900-BAFF BD00-BDFF
BF00-C1FF D800-D853 D900-D953 E900-E93F

APPLDATA DOMAIN DISABLED
MONITOR SAMPLE ACTIVE
INTERVAL 5 MINUTES
RATE 1.00 SECONDS
MONITOR DCSS NAME - MONDCSS
CONFIGURATION SIZE 241 LIMIT 1 MINUTES
CONFIGURATION AREA IS FREE

USERS CONNECTED TO *MONITOR - MONWRITE
MONITOR DOMAIN ENABLED
SYSTEM DOMAIN ENABLED
PROCESSOR DOMAIN ENABLED
STORAGE DOMAIN ENABLED
USER DOMAIN ENABLED
ALL USERS ENABLED
I/O DOMAIN ENABLED
THE FOLLOWING DEVICES ARE ENABLED:
2000-203F 2300-247F 2540-257F 2600-263F 2C00-2EFF 4000-427F
4300-44FF 4800-4848 4880-48C8 4900-4948 4980-49C8 4A00-4BFF
5000-50FF 5300-53FF 6300-637F 6400-647F 7000-77FF 8420-842F
8440-844F AC00-AFFF B300-B3FF B700-B7FF B900-BAFF BD00-BDFF
BF00-C1FF D800-D853 D900-D953 E900-E93F
APPLDATA DOMAIN DISABLED
Ready; T=0.01/0.04 20:26:08

5. Start the CP3KVMXT utility to create a 15 minute test EDF file, referring to [Command Syntax](#) later in this doc for syntax, and specifying a 5 minute duration and a maximum 3 duration limit to be processed. Do not use the CP3KVMXT PARMS file for the test run.

6. If applicable, prepare the UCLASS file to group performance information for a set of usersids into one workload. Refer to the section [Grouping Users into Workloads](#) for detail.
7. After addressing any problems, start the CP3KVMXT utility to create the full EDF file, specifying MONDCSS as the input source.
 - If you specified that CP3KVMXT use the parameter file by using the PARMFILE option, it will pick up runtime parameters from there and override them with any command line options. Otherwise, you will see several prompts, the first of which is to enter a title for the study, including the customer name and perhaps a date.
 - If you did not specify the CP3KVMXT parm file, the utility will ask you for a DURATION. This is the period of time that a SAMPS record in zCP3000 will represent. You should pick a value which is a multiple of the CP Monitor interval and will generate a reasonable number of data points on a utilization graph (for example, 4 hours of data and a duration of 15 minutes would generate a graph with 16 data points). For an 8 hour measurement period, 30 minutes would be a good value for the DURATION.
8. Progress messages are issued at interval and duration ends, and after every 5000 records processed. The utility will end with a message about where the new EDF file was written. If you want to end CP3KVMXT earlier than the time you specified (via prompt or parmfile), you can do this by hitting the <enter> key once, letting it sit in VM READ for 2 minutes, and then hitting the <enter key> again. This will time out the MONDCSS connection. Entering HX will also work, but will likely abend CMS and require that the userid re-ipl.
9. If the final EDF was not created for some reason (i.e., you HX'd it), you can create an EDF from the intermediate results file with the following command²
CP3KVMXT fn SAMPS A (restart
10. Download the EDF file in ascii mode and send it to the IBMer or Business Partner performing the Capacity Analysis. Please make sure the record length formatting has been preserved, and that the lines do not “wrap”.

² Refer to the [CP3KVMXT Syntax](#) section for more detail.

Appendix A: Syntax

Only one argument is required: the input data source.

```

                                     '- outputFileid -'
>>--CP3KVMXT---+ monfileid ----+-----+-----+----->
                '- MONDCSS -----'
                '- SAMPSfileid -'

    .-(-- uclassFileid -)--.
>+-----+-----+-----+-----+-----+----->
  '-(----- RESTART  -)--'
  '-(-----  PARM    -)--'
```

mondataFileid : *filename filetype filemode* of a saved monitor data file. Monitor data files **always** have a record length of 4096. This may be on a filemode accessed in a read-only mode.

MONDCSS : says to connect directly to the monitor service to read realtime performance data from the MONDCSS saved segment.

SAMPSfileid : *filename SAMPS filemode* of a checkpoint file from an unsuccessful CP3KVMXT run, from which you want to create an EDF. There will be a BCUDATA file with the same filename, if i/o data was collected. The RESTART option must also be specified to create the EDF from a checkpoint.

outputFileid : *filename filetype filemode* of the file where you want the output to be written. The filemode, if specified, must be a filemode accessed in a read/write mode.

Default output filename : the filename of the monitor data file

Default output filetype : EDF

Default output filemode : the first filemode accessed in r/w mode

uclassFileid (optional): *filename filetype filemode* of the file containing the workload grouping relationships. to be written.

Default output filename : the filename of the monitor data file

Default output filetype : EDF

Default output filemode : the first filemode that contains this filename/filetype

RESTART: says that the CP3KVMXT run did not end successfully, but that you would like to create an EDF file from the checkpoint data collected during the run. The SAMPS fileid must be specified as the data input for this option.

PARM: says to use the CP3KVMXT PARMS file to supply runtime parameters.

Examples

CP3KVMXT *fn ft fm*

will run CP3KVMXT against the monitor data file indicated by *fn ft fm*, not use any uclass user classification, and will write the output into *fn* EDF A.

CP3KVMXT *mondataFileid outputFileid (uclassFileid*

will run CP3KVMXT against the monitor data file indicated by *mondataFileid*, will use the uclass user classification file indicated by *uclassFileid*, and will write the output into *outputFileid*.

CP3KVMXT MONDCSS *outputFileid (uclassFileid*

will run CP3KVMXT against the realtime monitor data, will use the uclass user classification file indicated by *uclassFileid*, and will write the output into *outputFileid*.

CP3KVMXT (PARM

will read the CP3KVMXT PARMS file and will run CP3KVMXT against the monitor data file identified in the parm file, and write the output to the file identified by the parm file. All options, including the uclass file, delayed start times, duration length, and a maximum number of durations to process, are specified in the CP3KVMXT PARMS file.

CP3KVMXT *SampsFileid outputFileid (restart*

Will build an EDF from an intermediate SAMPS checkpoint results file.

Appendix B: Problem Determination

Symptom: vmarc won't work.

Explanation: possibly something went wrong in the file transfer. Upload it again, and make sure it is being transferred in binary mode. It should end up with record length=80 (after the PIPE deblock).

Symptom: vmarc says something is wrong with monview.vmarc or CP3KVMXT.vmarc.

Explanation: possibly something went wrong in the file transfer. Upload them again, and make sure it is being transferred in binary mode. VMARC archived files should have a record length=80.

Symptom: *DMSDCS343E Storage in range 01400000-02FFFFFF for MONDCSS in use.*

Explanation: You must redefine the size of your virtual storage so that it does not conflict with the MONDCSS segment. I.e., in the above example, the user's virtual storage size was set to 32M, and MONDCSS tried to load into the area from 20-50M. DEF STOR 64M then re-ipling CMS fixed the problem.

Symptom: *FPLSMG319E Not authorized to communicate with *MONITOR or DMSDCS283E The MONDCSS saved segment could not be loaded; return code 449 from SEGMENT LOAD.*

Explanation: Your userid's directory entry needs 2 statements like this:

```
IUCV *MONITOR MSGLIMIT 255  
NAMESAVE MONDCSS3
```

Symptom: While running with realtime monitor data, CP3KVMXT stops with these messages: *HCPMOV6274I The sample data messages and corresponding records have been purged. FPLSMG313E IPRCODE Message was purged received on IUCV instruction.*

Explanation: CP3KVMXT could not keep up with the monitor. Refer to "Speeding up CP3KVMXT" following this section for suggestions of things you can do to speed things up.

Symptom: "Insufficient storage" or "Machine storage exhausted";

Explanation: there is not enough virtual storage defined for this userid to run the extract. Use the Q V STOR and DEF STOR xxM command to define a larger virtual storage size. If your userid is already at the max, you will need to have the system administrator authorize you for a larger virtual storage size.

³The NAMESAVE statement is needed if the MONDCSS was defined as a restricted segment.

Symptom: CP3KVMXT completes, but no EDF file is created.

Explanation: CP3KVMXT probably did not complete successfully. If a file with a filetype of DEBUG exists on the A disk, please send it to Gretchen Frye at frye@us.ibm.com, along with an explanation of the problem. Checkpoint data is kept as the utility runs, so it is possible to generate an EDF from the part of the monitor data that was successfully processed. Use the restart option to build an EDF from intermediate results (see above).

Symptom: CP3KVMXT completes, but no EDF file is created, and it seems to take a long time doing something after the last duration has been processed.

Explanation: It may have failed in the step that creates the EDF out of the checkpoint data. If the SAMPS file is very large, it may run out of virtual storage when processing the checkpoint data. If the SAMPS and BCUDATA files were created (check the timestamp), and you can see that the checkpoint is large (ie, larger than the DEBUG file), try the restart option:

```
CP3KVMXT fn SAMPS fm ( RESTART
```

If that doesn't work, try increasing the virtual storage of the machine and use the restart option again. If that doesn't work, then it may be necessary to re-run CP3KVMXT and use the start and end parameters to create multiple, smaller EDFs.

Symptom: CP3KVMXT completes, but no EDF file is created, and no DEBUG file was created.

Explanation: the buildrec rexx file may be flawed. A bad copy of CP3KVMXT.VMARC was on the VM download site for a while. To see if this is the problem, browse or edit the BUILDREC REXX file; if it looks like compiled code, then that is the problem. Download a new copy of the CP3KVMXT.vmarc and reinstall.

Symptom: CP3KVMXT gets a rexx error and does not complete.

Explanation: Something unexpected was seen in the data. Make sure the monitor file you are running against is actually a CP Monitor file. Monitor data files are in hex, in a FB4096 file. The records wrap, so there's not really anything you can look for, but it is supposed to look unreadable. If there is a rexx error that makes the utility end abnormally, a debug file with a filetype of DEBUG will be generated. Please attach that to an email and send to the author, Gretchen Frye, whose email address is listed in Appendix B.

Symptom: duration lengths in the SAMP records are not all the same, and are not what I specified for a duration;

Explanation: the duration has to be ended when the first CP monitor interval record (domain 1 record B) past duration end occurs, because that's when CP collects the sample data. If monitor samples are taken at 2 minute intervals, and you specified at 15 minute duration, the EDF will end up with a 16 minute duration.

Appendix C: Speeding up CP3KVMXT:

This section applies both to realtime MONDCSS input and saved monitor data files, but it is much more critical for realtime input because the connection to the monitor ends if CP3KVMXT can't keep up with the monitor. When processing realtime monitor input, CP3KVMXT, after reading each set of records, replies to the monitor service that it has read them, so that the monitor service can reuse that space when all of the connected users have responded. If it does not reply by the time the monitor service wants to reuse those pages, the monitor takes the pages and issues an HCPMOV6274I error message. The CMS Pipeline connection to the monitor ends the connection when this happens, issuing the FPLSMG313E error message.

1. Restrict what the monitor is enabled for.

- Unless a performance monitor is using it for some specific purpose, the scheduler and seeks domains should *not* be enabled. CP3KVMXT does not use these records. *CP Q MONITOR* and you should see the following both for event handling and sampling:

```
SCHEDULER DOMAIN DISABLED  
SEEKS   DOMAIN DISABLED
```

- Enable the monitor only for DASD class I/O:

```
CP MONITOR SAMPLE ENABLE I/O CLASS DASD
```

- Even better, enable only a specified device address range. Online but inactive dasd still gets sampled every minute, and even for offline dasd, configuration records are built and read for all devices in the IOCDs. These are all of the "Reading monitor record 10000 ... D1 system configuration" records that you see at the beginning of a CP3KVMXT run.

2. Ensure that the MONDCSS segment is the right size for your system. The default size of 3MB may not be big enough, especially for guest-hosting systems with lots of DASD. The smaller it is, the more quickly it has to round-robin and reuse the space, and the more quickly CP3KVMXT has to process it. Follow the guidelines for sizing MONDCSS in the z/VM Performance Guide on <http://www.vm.ibm.com/library/>
3. If your site has the Rexx compiler (REXXD EXEC is the compiler), try renaming and compiling BUILDREC REXX. The compiled BUILDREC must have a filetype of REXX in order to execute. It won't make any difference to compile CP3KVMXT, because it is mainly a front end and does not execute much code.
4. If you see any HCPMOV6274I messages, look carefully at those messages and note whether the problem occurred with "event" or "sample" data. Increase the size and limit of the config on the CP Monitor command for "event" or "sample" so that the records are retained a little longer.
5. Consider increasing the CP Monitor interval size. This will reduce the sampling frequency, and thereby reduce the amount of data collected.

Appendix D: EDFI Field Descriptions

The EDF file is a fixed block file with a record length of 80, with very restrictive formatting requirements. The z/OS equivalent field is provided as a matter of interest. Some fields are exactly the same in z/OS and z/VM (the LPAR related fields, for instance, are provided by the underlying PR/SM microcode). Others are similar in intent but not necessarily measuring the same thing (for instance, most workload related fields).

Table 1: HEAD macro – header information

Head	Header Section	Type (length)	Domain / Rec	VM Source Fieldname	MVS equivalent
ENT	Enterprise Name	char(50)	-	User	User
Source	Version which produced this file	Char	-	generated	

Table 2: CEC macro – information about this mainframe

CEC	CEC Section	Type (length)	Domain / Rec	VM Source Fieldname	MVS equivalent
CECID	CEC Identifier	Char	-	User	User
SUPVR	Supervisor	Char	-	generated	
CPUMOD	CPU Model	Char	D1R4	MTRSYS_SYSMTYPE, user prompt	SMF70MOD SMF70VER
VC	Version Code	Char	D1R5	MTRPRP_PFXIDVER	SMF70VER
PR	CP + IFL + ICF	Number	D0R17	SYTCUM_LCUPCPCT	SMF70BNP
SR	CPU serial #	Number	D1R5	MTRPRP_PFXIDSER	SMF70SER
PRV	Only if PR changes	Vector	D0R17	SYTCUM_LCUPCPCT	SMF70BNP
CPV	Traditional engines	Vector	D0R19	SYTSYG_CPUCFGCT	
IFLV	IFL engines	Vector	derived	PRV – CPV	
ICFV	Coupling Facility	Vector	n/a	Merge in from MVS EDF	
CPUMODV	CPU Model Names	Multiple text strings	D1R4	MTRSYS_SYSMTYPE, user prompt	SMF70MOD SMF70VER
CMIND	Index of current Vector CPC in CPUMODV	Vector			

Table 3: SYS information about this z/VM image

SYS	System Image	Type (length)	Domain / Rec	VM Source Fieldname	MVS equivalent
SYSID	System ID	Char	D1R4	MTRSYS_SYSTMID	SMF70SID
SCP	SCP	Char	D1R4	MTRSYS_HCPCPEPP	SMF70MVS
VERSION	SCP level	Char	D1R4	MTRSYS_HCPCPEID	SMF70RLS
NSAMPS	Number of samples	Number	-	generated	
WC	Wait Complete	1 or 0	D0R16	SYTCUP_LCUCWCPL	SMF70VPF SMF70WSA
BIT	64 bit mode indicator	1	D1R4	MTRSYS_CALESAME	SMF70EME
RMFINTL	CP monitor interval	Number (min)	D1R9	MTRSPR_INTERVAL	
GMTOF	offset GMT	+/- hh:mm	D1R4	MTRSYS_SYSZONE	
CS	LPAR defined CS	MB	D1R7	MTRMEM_RSAGSTOR	SMF71TFC +SMF71FIN
ES	LPAR defined ES	MB	D1R17	MTRXSG_SYSXTSIZ	SMF71OLE
SCPCS	Total Resident Nucleus	MB	D1R7	MTRMEM_HCPMM4	SMF71FIN+ ASR+ALP+ AVP
CSAVAILV	available frames below 2GB	Vector (MB)	D0R3	SYTRSG_RSAAVAIL	SMF71TFC +SMF71FIN
ESAVAILV	available frames above 2GB	Vector (MB)	D0R3	SYTRSG_RSA2GAV2	SMF71TFC +SMF71FIN
PGTOES	Total pages to ES	Vector	D0R5	SYTXSP_PFXPGIN + PLSPGIN	SMF71PES
PGFROMES	Pages from ES	Vector	D0R5	SYTXSP_PLSPGOUT	SMF71RES
PAGEV	Page	Vector	D0R1	SYTSYP_PLSIOPR +	SMF75SIO

	Operations			PLSIOPW	
DPA_LOV	Pageable Area <2GB	Vector (MB)	D0R3	SYTRSG_RSAPGABL - SYTRSG_RSANONPG	
DPA_HIV	Pageable Area >2GB	Vector (MB)	D0R3	SYTRSG_RSALGFRM	
DASDIOV	Dasd SSCH rate	Vector	D1R6 D6R3	MTRDEV_RDEVCLAS, IODDEV_SCMSSCH	SMF74SSC
LPARNO	index of this sysid in the LPAR array	Number	D0R16	SYTCUP_CALPTIS	SMF70PTN
LPAR	Partition names	Vector	D0R16	SYTCUP_LCUPNAME	SMF70LPM
LPPRCn	avg # CPs this partition	Vector	D0R16	SYTCUP_LCUPCPCT	SMF70ONT
LPPCTMn	seconds of CPU time this partition	Vector	D0R16	SYTCUP_LCUCACTM	SMF70PDT
LPPRIn	avg # ICFs this partition	Vector	D0R16	SYTCUP_LCUPCPCT	SMF70ONT
LPPITMn	seconds of ICF time this partition	Vector	D0R16	SYTCUP_LCUCACTM	SMF70PDT
LPPRLn	avg # IFLs this partition	Vector	D0R16	SYTCUP_LCUPCPCT	SMF70ONT
LPPLTMn	seconds of IFL time this partition	Vector	D0R16	SYTCUP_LCUCACTM	SMF70PDT
LPWGTn	avg LPAR wgts this partition	Vector	D0R16	SYTCUP_LCUCWGHT	SMF70MIS, SMF70BPS, SMF70MAS
LPCAPn	is partition capped? (1 or 0)	Vector	D0R16	SYTCUP_LCUCCAPP	SMF70VPF, SMF70CAP
CECUTILV	Utilization for the image	Vector	D0R16	SYTCUP_LCUCACTM	SMFWAT, SMF70PDT

Table 4: SAMP – interval timestamps

SAMP	System Samples	Type (length)	Domain / Rec	VM Source Fieldname	MVS equivalent
DUR	Duration (hh:mm)	Char	-	User	User
SIO	Sample I/O Rate	num	D0R1	SYTSYP_PLSCCTSS	
SPAGE	Sample Page Rate	num	D0R1	SYTSYP_PLSPIOPR +IOPW	

Table 5: WORK – user/workload group samples

WORK	Workload Samples	Type (length)	Domain / Rec	VM Source Fieldname	MVS equivalent
WDESC	Description	Char	-	from UCLASS file	user parm
WCPUTM	Wkld CPU seconds	vector	D4R3	USEACT_VMDTTIME	SMF72CTS+ SMF72STS+ R723Cnnn
VSTORV	Virtual Storage MB	Vector	D4R3	USEACT_ASCDEFSZ	n/a
WCSV	resident pages <2GB	vector	D4R3	USEACT_VMDCTPVR	SMF72FT1,2
WESV	resident pages >2GB	Vector	D4R3	USEACT_VMDCTPVR	SMF72ER1,2
WSSV	Projected working set size (pages)	Vector	D4R3	USEACT_VMDWSSPR	n/a
WPAGEV	Wkld Paging	Vector	D4R3	USEACT_CALCPPGR +CALCPPGW	SMF72PIN +R723CPIN
WIOV	SSCH count	vector	calc	IODDEV_SCMSSCH	SMF72IRC +R723CIRC
WEXCPV	Wkld Dasd IO Requests	Vector	D4R3	USEACT_VMDVDSCT	SMF72ITS +R723CIOC

Table 6: BCU dasd i/o information

BCU	Basic Configurable Unit for DASD	Type (length)	Domain / Rec	VM Source Fieldname	MVS equivalent
BCUID	BCU Identification	char	D6R4	IODCAD_CALDATA	User Parm
CTYPE	CU type	char	D1R6	MTRDEV_RDEVUID, MTRDEV_RDEVUM N	User Parm/ SMF74CU
BCUDASD1	Dasd type (online devices only)	char	D1R6	MTRDEV_RDEVDVID	User Parm/ SMF74DEV
BCUDASDN	# units this type (online devices only)	num	D6R3	counted	computed
CACHE	Available cache	num (MB)	D6R4	device hardware	UserParm/ CSCONF
NVS	Non-volatile storage	num (MB)	D6R4	device hardware	UserParm/ CSCONF
NOAD	# addresses under this BCU	num	D6R3	counted	From BCU MAP
BCUIO	Total I/O Rate	num4	D6R3	IODDEV_SCMSSCH	SMF74SSC
BCURESP	Average Response Time	num		computed	computed
BCUCONN	Average Connect Time	num	D6R3	IODDEV_SCMCNTIM	SMF74CNN
BCUDISC	Average Disconnect Time	num	D6R3	IODDEV_SCMDTIM	SMF74DIS
BCUPEND	Average Pend Time	num	D6R3	IODDEV_SCMFPTIM	SMF74PEN
BCUQUE	Average IOS Queue Time	num	D6R3	IODDEV_HFCTIO	SMF74QUE

⁴Time fields in the BCU are in seconds.

BCUSKEW	Maximum device busy to average .	num		computed	computed
BCUR	Read/Write Ratio ⁵	num	D6R4	computed	CRR:"Total (Cache) R/W Ratio"
BCUH	Ratio: Read Hits ⁶ / All Reads	0< num <1	D6R4	IODCAD_CALDATA	CRR:"Total Read H/R"
BCUW	Ratio: Fast Write ⁷ Hits / All Writes	0< num <1	D6R4	IODCAD_CALDATA	CRR:"Total F/W H/R"
BCUG	Ratio: Sequential DASD to Cache / All I/O	0< num <1	D6R4	IODCAD_CALDATA	CRR: "DASD to Cache Transfers - Sequential " divided by "Total I/O Requests"
BCUIOV	I/O Rate per sample.	vector	D6R3	IODDEV_SCMSSCH	SMF74SSC
BCURESPV	Avg Response Time per sample	vector	D6R3	computed	computed
BCUCONN	Avg Connect Time per sample	vector	D6R3	IODDEV_SCMCNTIM	SMF74CNN
BCUDISCV	Avg Disconnect Time per sample	vector	D6R3	IODDEV_SCMDTIM	SMF74DIS

⁵BCU time fields include all online devices under that BCU, whether they are cached or not.

⁶Cache Reads include Search/Read requests for Normal, Sequential, and Fast Write I/Os.

⁷Cache Writes include Write requests for Normal, Sequential, and Fast Write I/Os

BCUPEND V	Avg Pend Time per sample	vector	D6R3	IODDEV_SCMFPTIM	SMF74PE N
BCUQUEV	Avg IOS Queue Time per sample	vector	D6R3	IODDEV_HFCTIO	SMF74QU E
BCURV	Read Write Ratio per sample	vector	D6R4	computed	CRR:"Tot al (Cache) R/W Ratio"
BCUHV	Read Hit Ratio per sample	vector	D6R4	IODCAD_CALDATA	CRR:"Tot al Read H/R"
BCUWV	Fast Write Hit Ratio per sample	vector	D6R4	IODCAD_CALDATA	CRR:"Tot al F/W H/R"
BCUGV	Sequential Stage Ratio per sample.	vector	D6R4	IODCAD_CALDATA	CRR: "DASD to Cache Transfers - Sequential " divided by "Total I/O Requests"

Table 7: BCU Path information

PATH	BCU Path Data	Type (length)	Domain / Rec	VM Source Fieldname	MVS equivalent
PID	Path ID	Char chpid	D6R3	IODDEV_RDEVLPM	User
PTYPE	Channel Type	P parallel E ESCON F FICON		class B CP command Q CHPID chpid TYPE	SMF73ACR
PBUSYV	Path Busy	vector	D0R9	SYTCPC_HFCHBUSY	SMF73BSY

Table 8: Actuator information

ACT	Actuator Data	Type (length)	Domain / Rec	VM Source Fieldname	MVS equivalent
SID	Subchannel ID	num (hex)	D6R3	IODDEV_RDEVLPM	User
V	Volser8	Char	D6R3	IODDEV_RDEVSER	SMF74SER
A	Address	num (hex)	D1R6	MTRDEV_RDEVDEV	SMF74NUM
T	DASD Type	char	D1R6	MTRDEV_RDEVUID, MTRDEV_RDEVUCUMN	SMF74DEV
R	I/O Rate	num	D6R3	IODDEV_SCMSSCH	SMF74SSC
SDS	standard deviation for service	num	D6R3	computed	computed
Q	IOSQ	num	D6R3	IODDEV_HFCTIO	computed
P	Pend Time	num	D6R3	IODDEV_RDEVFPTIM	SMF74PEN
D	Disconnect Time	num	D6R3	IODDEV_RDEVDDTIM	SMF74DIS
C	Connect Time	num	D6R3	IODDEV_RDEVCONTIM	SMF74CNN
DS	Minidisks Defined	num	D6R3	IODDEV_RDEVLCNT	SMF74NDA
SDR	standard deviation for response	num	D6R3	computed	computed
RWR	Read Write Ratio	num	D6R3	computed	CRR:"Total (Cache)□ R/W Ratio"
RDHT	Read Hits / All Reads	num	D6R4	IODCAD_CALDATA	CRR:"Total Read H/R"
FWHT	Fast Write Hits / All Writes	num	D6R4	IODCAD_CALDATA	CRR:"Total F/W H/R"
SQST	Sequential	num	D6R4	IODCAD_CALDATA	CRR:

⁸Volser is not unique, since a device attached to a guest may have any volser (or null).

G	DASD to Cache / All I/O				DASD to Cache Transfers - Sequential divided by Total I/O Requests
PC	Percent Cacheable	num	D6R3	IODCAD_CALDATA	CRR: Total Cacheable I/Os divided by Total I/O Requests
ST	Actuator Status	N - Caching Activated, DASD FW Allowed, D - Caching Deactivatd DASD FW Deactivatd C - Caching Activated, DASD FW Deactivatd F - Caching Deactivatd DASD FW Allowed	D6R4	IODCAD_CALDATA	CRR: "Device Status".

Appendix E: Resources

Websites:

IBM VM Download Packages

<http://www.vm.ibm.com/download/packages/>

How to download a VMARC type package from VM Downloads

<http://www.vm.ibm.com/download/#downvmarc>

IBM WSC - Capacity Planning Support Tools menu

IBM Internal: <http://w3.ibm.com/support/americas/wsc/cpsproducts.html>

IBM BP: <http://partners.boulder.ibm.com/src/atmastr.nsf/WebIndex/PRS1762>

Publications:

The CP3KVMXT user's guide (that you are reading right now) can be found at:

ftp://ftp.software.ibm.com/software/mktsupport/techdocs/cp2kvmxt_ug.pdf

z/VM Internet Library

<http://www.vm.ibm.com/library/>

VMPRF V1R2.2 User's Guide and Reference

<http://www.vm.ibm.com/pubs/pdf/fpra1a04.pdf>

z/VM V4R3: CP Command and Utility Reference

<http://www.vm.ibm.com/pubs/pdf/hcsg0a00.pdf>

Contacts:

For questions and general support: Bernice Riley

Internet email address: cpstools@us.ibm.com

Normal US business hours (EST): 301-240-2645

For technical issues: Gretchen Frye

Internet email address: frye@us.ibm.com

IBM VM address: frye at wscvm

Note: Gretchen's VM address is for debug files and data ONLY. If sending data through email, please VMARC or PKZIP it first.

[Return to the top](#)