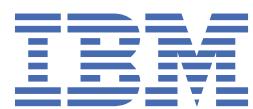z/VM
7.4

*Performance Toolkit Guide*

IBM

**Note:**

Before you use this information and the product it supports, read the information in "Notices" on page 189.

# Contents

# Figures

# About This Document

This document, with the *z/VM: Performance Toolkit Reference*, contains all information that is required for using Performance Toolkit for z/VM. Performance Toolkit for z/VM was previously known as the VM/ESA Full Screen Operator Console and Graphical Real Time Monitor, FCON/ESA, or FCON.

The suggested topics, in the order that is listed, are recommended for new users:

1. Chapter 1, "An Overview of Performance Toolkit," on page 3
2. Chapter 2, "Preparing Performance Toolkit," on page 9 to prepare Performance Toolkit
3. Part 2, "Using the Performance Toolkit," on page 61 for specific information about using Performance Toolkit, including Chapter 3, "A Guided Tour," on page 63 for a short tour of the program's features.

z/VM: Performance Toolkit Guide provides detailed information about Performance Toolkit subcommands, screen layouts, and messages. For more information, see *z/VM: Performance Toolkit Reference*.

**Note:** The topic collections in *z/VM: Performance Toolkit Reference* and *z/VM: Performance Toolkit Guide* do not contain information about z/VM Performance Data Pump. z/VM Performance Data Pump is licensed with Performance Toolkit for z/VM but does not support, depend upon, or interact with Performance Toolkit for z/VM in any way.

z/VM Performance Data Pump (Data Pump) converts machine-readable z/VM monitor and SFS data into a generic text-based data stream. Modern tools can use the data stream to display real-time performance dashboards, aggregate real-time data for long-term usage analysis, or integrate with existing enterprise observability solutions.

For more information, see z/VM Performance Data Pump in *z/VM: Performance* and Data Pump Messages in *z/VM: Other Components Messages and Codes*.

## Intended Audience

This document is intended for use by system programmers and operators for performance analysis and general system operation. People using the document are expected to have a good general knowledge of z/VM.

## Where to Find More Information

All Performance Toolkit reference information, including subcommands, screen layouts, and messages, can be found in the *z/VM: Performance Toolkit Reference*

For more information about z/VM functions, see the documents listed in the "Bibliography" on page 193.

Other manuals of the z/VM system reference library may be needed, especially for performance monitoring. Please refer to the bibliography of your operating system for locating the appropriate documentation.

### Links to Other Documents and Websites

The PDF version of this document contains links to other documents and websites. A link from this document to another document works only when both documents are in the same directory or database, and a link to a website works only if you have access to the Internet. A document link is to a specific edition. If a new edition of a linked document has been published since the publication of this document, the linked document might not be the latest edition.

# How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. See How to send feedback to IBM for additional information.

# Summary of Changes for z/VM: Performance Toolkit Guide

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line (**I**) to the left of the change.

## SC24-6302-74, z/VM 7.4 (July 2025)

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.3 and 7.4.

The following topic is updated:

- "[7.4 VM66826, 7.3 VM66825] z/VM Performance Data Pump enhancements" on page xv

## SC24-6302-74, z/VM 7.4 (June 2025)

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.3 and 7.4.

### [7.4 VM66826, 7.3 VM66825] z/VM Performance Data Pump enhancements

> **Performance Toolkit statement of support:** The Performance Toolkit Data Pump dashboards have been updated to provide information that is specific to the new architectures delivered with the IBM z17 family of servers. The 3270 and web interfaces of the Performance Toolkit are stabilized at the IBM z16 architecture level and will not reflect new hardware and facilities delivered with the IBM z17 or any future families of servers.

With the PTFs for APARs VM66826 (7.4) and VM66825 (7.3), z/VM 7.4 and 7.3 provide support for a new z/VM Performance Data Pump dashboard. The dashboard provides a graphical representation of IBM z17 family power consumption metrics from the z/VM monitor stream. The display includes information for the following components:

- CPC
- LPAR
- CPU
- I/O
- Memory

The information can be used to calculate or approximate guest level apportionment.

Information about setting up Data Pump is available. See z/VM Performance Data Pump in *z/VM: Performance*.

Information is available about other services that process the Data Pump output stream, including links to Grafana sample dashboards. See z/VM Performance Data Pump (https://www.vm.ibm.com/related/perfkit/datapump/).

## SC24-6302-74, z/VM 7.4 (September 2024)

This edition supports the general availability of z/VM 7.4. Note that the publication number suffix (-74) indicates the z/VM release to which this edition applies.

# SC24-6302-73, z/VM 7.3 (September 2023)

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.3.

### [VM66678, VM66709] Warning Track Interruption Facility

With the PTFs for APARs VM66678 (CP) and VM66709 (Performance Toolkit), z/VM 7.3 exploits a feature of Processor Resource/Systems Manager (PR/SM) called the *warning-track-interruption facility.* z/VM's exploitation of this facility helps improve guest response time and overall performance of workloads that are run on vertical-low or vertical-medium logical processors.

# SC24-6302-73, z/VM 7.3 (August 2023)

This edition includes changes to support product changes that are provided or announced after the general availability of z/VM 7.3.

### [VM66687] z/VM Performance Data Pump

With the PTF for APAR VM66687, z/VM 7.3 supports z/VM Performance Data Pump. z/VM Performance Data Pump (Data Pump) converts machine-readable z/VM monitor and SFS data into a generic text-based data stream. Modern tools can use the data stream to display real-time performance dashboards, aggregate real-time data for long-term usage analysis, or integrate with existing enterprise observability solutions.

z/VM Performance Data Pump is licensed with Performance Toolkit for z/VM but does not support, depend upon, or interact with Performance Toolkit for z/VM in any way.

For more information, see z/VM Performance Data Pump in *z/VM: Performance.*

# SC24-6302-73, z/VM 7.3 (September 2022)

This edition supports the general availability of z/VM 7.3. Note that the publication number suffix (-73) indicates the z/VM release to which this edition applies.

### Change in location and size of the MONDCSS and PERFOUT saved segments

The default CP MONITOR MONDCSS saved segment starting virtual address has been changed to 1 GB and the size has been increased to 96 MB. The IBM Performance Toolkit PERFOUT saved segment starting virtual address has been changed to 1120 MB, immediately following the new MONDCSS. The reasons for these changes include:

- Moving the segment starting locations addresses a problem with the Linux kdump configuration, which failed in a virtual machine with 1 GB of virtual memory that had attached one or both current versions of these segments.
- Increasing the size of the MONDCSS segment accommodates the increased volume of CP Monitor data produced and provides room for its future growth.

These changes will not affect existing versions of these segments; they must be re-created for the new defaults to take effect.

The following topic is updated:

- "Defining and Building the DCSS" on page 183

# Part 1. Overview and Preparation

The topics in this section introduce you to Performance Toolkit for z/VM.

# Chapter 1. An Overview of Performance Toolkit

**Performance Toolkit for z/VM**

Performance Toolkit for z/VM (also called Performance Toolkit) is designed to assist operators and system programmers or analysts in the following areas:

- **System console operation in full screen mode.** The features that are provided are designed to facilitate the operation of z/VM systems, which can improve operator efficiency and productivity.

- **Performance monitoring on z/VM systems.** An enhanced real-time performance monitor allows system programmers to monitor system performance and to analyze bottlenecks. Features included in the monitor, and the manner in which data is displayed, are designed to improve the system programmer's productivity when analyzing the system and to allow a new user to work efficiently with the tool.

  The Performance Toolkit can help system programmers to make more efficient use of system resources, increase system productivity and improve user satisfaction.

In this document, the abbreviation FL refers to the function level of the Performance Toolkit, which corresponds to the associated level of z/VM on which Performance Toolkit is running. For example, FL740 refers to Performance Toolkit function level 740 running on z/VM.

**z/VM Performance Data Pump**

z/VM Performance Data Pump (Data Pump) converts machine-readable z/VM monitor and SFS data into a generic text-based data stream. Modern tools can use the data stream to display real-time performance dashboards, aggregate real-time data for long-term usage analysis, or integrate with existing enterprise observability solutions.

z/VM Performance Data Pump is licensed with Performance Toolkit for z/VM but does not support, depend upon, or interact with Performance Toolkit for z/VM in any way. The topic collections in *z/VM: Performance Toolkit Reference* and *z/VM: Performance Toolkit Guide* do not contain information about z/VM Performance Data Pump.

For more information, see z/VM Performance Data Pump in *z/VM: Performance* and Data Pump Messages in *z/VM: Other Components Messages and Codes*.

# System Console Operation in Full-Screen Mode

Features include:

- General system output (informational messages and replies to commands entered) can automatically be scrolled, however additional privilege classes may be needed depending on implemented functions.

- Messages from other virtual machines are numbered and left pending at the top of the screen until explicitly deleted, even if automatic scrolling is active.

- The last few important 'action' messages (number can be specified) can also be left pending at the top of the screen until explicitly deleted.

- Additional processing of output lines which meet certain user specifications (optional). Selected lines can be:

  - Suppressed (not displayed, written to log only)

  - Displayed with different color and highlighting and/or made non-scrollable

  - Modified (for example, long RSCS message header information removed) to make the text more easily readable

  - Re-routed to other machines on the same system

- – Used to run EXEC procedures to help in automating system operation
- A redisplay facility allows browsing through the day's accumulated console log, or through previous days' logs.

# Performance Monitoring on z/VM Systems

The performance monitoring functions can be used in machines with privilege class E, and which have been authorized to access CP monitor data:

- Automatic threshold monitoring of many key performance indicators with operator notification if user defined limits are exceeded
- Special performance monitoring mode with displays for monitoring:
  - General CPU performance
  - System and user storage utilization and management
  - Channel and I/O device performance, including cache data
  - Detailed I/O device performance, including information on the I/O load caused by specific minidisks on a real disk pack
  - General user data: resource consumption, paging information, IUCV and VMCF communications, wait states, response times
  - Detailed user performance, including status and load of virtual devices
  - Summary and detailed information on Shared File System servers
  - Configuration and performance information for TCP/IP servers
  - Linux® performance data
  - System execution space (SXS) performance data
  - Virtual networking configuration and performance data

  The collected performance data is also used to provide:

  - A re-display facility for many of the key figures shown on the general CPU performance and storage utilization screens which allows browsing through the last measurements (up to twelve hours)
  - Graphical history plots with a selection of up to four re-display variables, either as simple plots, or, if the Graphical Data Display Manager program (GDDM®, 5684-007 or 5684-168) is available, also in the form of GDDM graphics
  - Graphical variable correlation plots (simple plots or GDDM graphics) which show how the values of specific variables are correlated to the values of another variable

    **Note:** GDDM is required for generating graphics on the console of a virtual machine, however no additional software is required when generating graphics with web browsers using the WWW interface.
- For later reference, the accumulated performance data can also be used for creating:
  - Printed performance reports
  - Simple performance history files
  - Extended trend files
- For capacity planning, the history files on disk can be used for:
  - Performance trend graphics
  - Numerical history data analysis
- A remote performance monitoring facility allows:
  - Central performance monitoring for many VM systems, with automatic display of summary exception data
  - Multiple concurrent access to the central monitor machine's data

&ndash; Retrieval and display of additional detailed performance data from these remote systems if problems are suspected

- A monitor data scan mode allows browsing through CP monitor data files written to disk by the standard MONWRITE utility. Most of the information available during real time monitoring can be viewed in this data scan mode.

Although performance monitoring mode must be entered for actually displaying performance data, the data collection process itself can go on even while working in any of the other console modes. The screen will typically be used in basic mode most of the time, though it can be switched to operate in monitoring mode when a performance problem is suspected.

## Online HELP Facility

HELP information available for online display includes a short description of the different operating modes, and full reference information on subcommands, messages, and field descriptions for all performance monitor displays.

## Time Zone Changes

Performance Toolkit tracks CP performance for specific time intervals. It is not able to handle time that is changed dynamically by such commands as SET TIMEZONE. IBM recommends that you stop Performance Toolkit before changing the time zone and start it up again after the time zone has been changed.

## Modes of Operation

The modes of operation for Performance Toolkit include:

- Basic Command Mode, usually referred to as 'basic mode' is the normal operating mode which allows entering CP and CMS commands. This mode immediately displays any messages received from CP.
- Re-Display Mode. The console log created while operating in basic mode can be browsed in re-display mode and specific strings (for example, USERIDs) can be searched using a 'LOCATE' facility.
- Performance Monitoring Mode displays data on CPU, I/O and user performance collected by *this* machine. It allows real time performance analysis of z/VM systems.
- Remote Performance Monitoring Mode displays performance data collected by the performance monitors from another virtual machine on the same or other VM systems.
- The Internet interface allows retrieval and displays the same performance data via a standard Web Browser.
- Monitor Data Scan Mode displays performance data from a MONWRITE disk file, created by z/VM systems.
- Trend File Scan Mode displays performance history data from a trend file (FCXTREND disk file).

## Minimum System Requirements

### Hardware Prerequisites

Any processor supported by one of the supported VM versions.

- Supported terminals: any display terminal supported as a z/VM logon console. Extended color, highlighting, or both can be used when supported by the hardware. Graphic performance data displays can be shown only on terminals with graphics capability which are also supported by GDDM.
- Disk space requirements: a total of approximately 131 cylinders 3390, or the equivalent space on any other device type supported by CMS, is required for the installation of the Performance Toolkit module, initialization profile, and HELP files. Additional space is required on each user's A-disk, depending on the size of the log files to be written.

- Virtual storage requirements:

  1. For collectors of Performance Toolkit data: a minimum storage size of 64 MB
  2. For observers of Performance Toolkit data: a minimum storage size of 32 MB.

  Starting with Performance Toolkit FL640, IBM recommends the following storage definition:

  ```
  CP DEFINE STORAGE CONFIG 0.512M 2G.2G
  ```

  The storage above the 2G line is called *high memory area (HMA)*. The discontiguous virtual storage definition is used to minimize the backing paging amount for the PERFSVM virtual machine. It is also possible to define a single memory extent as follows. For example:

  ```
  CP DEFINE STORAGE 4G
  ```

  If the virtual storage size is less than 2G, Performance Toolkit will continue to operate as it did previously, allocating conventional memory below 2G.

  For information about configuring HMA, refer to the FCONTROL HMA PAGE subcommand described in *z/VM: Performance Toolkit Reference*.

- The Performance Toolkit is installed when the z/VM system is installed.

## Software Requirements

Full use of all functions requires:

- z/VM 7.4.
- z/CMS Level 28, which is shipped with z/VM 6.4. The z/Architecture® mode provided by z/CMS is required by Performance Toolkit to exploit the 64-bit addressing mode. The latter is used to effectively eliminate the virtual storage shortage issue.
- A supported level of the Graphical Data Display Manager program (GDDM®, 5684-007 or 5684-168)[1]

# Restrictions for Basic Command Mode

PERFKIT is a CMS module and operation is based on the IUCV *MSG service, a standard facility of z/VM, and it also intercepts I/O, SVC and external interrupts. While all standard CP and CMS commands (except the "programmable operator" PROP) will run normally under Performance Toolkit, the program imposes some restrictions on other CMS applications that are to be run under its control. These are documented in the sections "Reply to CP and CMS Prompts (CP and VM READ state)" on page 80 and in "Restrictions for Operation in Basic Command Mode" on page 81.

Due to documented restrictions of the IUCV *MSG service:

- CP messages, warnings, and noheader messages will not be written to the CP console log of the machine where Performance Toolkit is active.
- Noheader messages will be treated as normal messages.
- CMS programs which use the CMS Application Multitasking functions will cause Performance Toolkit to ABEND when the application is run from Performance Toolkit Basic Command Mode. Examples of such applications are PING and FTP.

Because z/CMS is a prerequisite (starting with z/VM 6.4), Performance Toolkit can no longer run on CMS in ESA/390 mode or in XC mode. These and other restrictions and limitations are described in detail in the z/Architecture CMS section of *z/VM: CMS Planning and Administration*.

---

[1] GDDM libraries are a prerequisite only for the special graphics displays of the performance monitor part of Performance Toolkit; they are not required for any of the other functions. Note that graphics can be displayed using the Performance Toolkit's WWW capabilities without the need for GDDM.

# Performance Toolkit Overhead

**System console operation in full screen mode:** The CPU time required for running a command under control of Performance Toolkit is higher than when running the same command under normal CMS. The additional load imposed on the system when working with Performance Toolkit as the operator console is negligible.

Performance Toolkit runs a certain number of disk I/O's while writing or reading (for the 're-display' function) the console log. The number of I/Os is, in general, very low and will not cause performance problems.

# Installation Verification

After you have completed the installation as described in the Program Directory, follow the steps outlined in Chapter 3, "A Guided Tour," on page 63 to verify correct operation. If the Performance Toolkit is new to you, the Guided Tour will also help you learn how to use the program.

# Performance Monitoring

Performance monitoring may require more resource. The additional load generated during automatic data collection (that is, with a measuring interval of 60 seconds) is typically less than 1% of total CPU capacity (depends on system configuration, number of users on the system, and on the enabled monitor domains, see "Performance Considerations" on page 7).

# Customizing Performance Toolkit

The sample initialization file FCONX $PROFILE is intended to provide a starting point only. While it will allow you to use all functions, you will probably want to adapt it to your personal needs. Refer to the following:

- "FCONX $PROFILE" on page 92 for general information on the initialization profile
- "Setup for Threshold Monitoring" on page 126 and "Setup for User Monitoring" on page 128 for guidelines on how to set up the profile for performance monitoring
- "Saving Performance Data" on page 108 for information on how to start permanent data collection in history files (pre-requisite for trend analysis)
- "Monitoring Linux Internal Performance Data" on page 136 for information on how to set up the interface for retrieving Linux internal performance data
- "Implementing Remote Performance" on page 10 for information on how to implement the 'remote performance data retrieval and display facility'
- "APPC/VM Interface for Performance Data Retrieval" on page 32 for information on how to prepare for the use of the APPC/VM interface for performance data retrieval
- "Internet Interface" on page 45 for information on how to retrieve data via the Internet

Because specification of your own initialization commands requires previous knowledge of these commands and their effects, we recommend you start with the sample initialization profile, and adapt it at a later date when you have become more familiar with the product.

# Performance Considerations

## Full-Screen Operation Mode

Performance Toolkit causes additional paging and I/O operations compared to a standard VM system console with TERMINAL MODE set to CP. Response times should be similar to those experienced in an ordinary CMS machine. They can be improved by giving the system operator virtual machine a better

dispatching priority and, if necessary, by using the 'SET RESERVED' command to reduce paging activity for that specific machine.

You may also have to give the Performance Toolkit operator machine a better dispatching priority if it has to handle many incoming messages in quick succession: the CP IUCV *MSG service sets a limit of 2048 messages in z/VM. If Performance Toolkit is too slow in receiving them, the remainder will be directly displayed on the operator's terminal. Although this should not normally happen, the situation might occur if another user generates messages very fast. You may face this situation when running a CP command which produces more than the maximum of output lines which *MSG can handle in your own machine (for example, a 'Q RDR ALL' command on a large system), and even the best dispatching priority is not going to change that.

## Performance Monitoring

The CPU load for normal data collection with the default interval of 60 seconds will usually be considerably less than 1% of total CPU capacity (typically .1% to .2% on a system with 1000 users) and should not cause any problems. The CPU load will vary depending on the amount of data to be collected on the following:

- The system environment being monitored (for example, size of I/O configuration, and especially the number of users)
- The monitor domains which have been enabled
- The frequency of data collection
- The type of analysis to be performed.

You can influence only the last three points.

**Monitor domains:** If CP monitor data collection has been enabled, Performance Toolkit will collect all the data it can use. This can cause considerable load if you have enabled EVENT data for all users on a large system: records will be generated by CP, and collected by Performance Toolkit, for each single user transaction. Care should be taken.

The same is true when enabling EVENT SEEKS data for many disks. A large number of monitor records will be generated (one record for each disk I/O operation), and all these records will have to be processed by Performance Toolkit.

**Frequency of data collection:** The monitor SAMPLE interval is used for all the permanently collected data (the Performance Toolkit uses this interval also when obtaining data for any of the general performance displays directly from CP control blocks). The default interval is 60 seconds, but you can change it to a shorter or longer period. Setting a longer interval will reduce the data collection overhead for both CP and Performance Toolkit, while setting a shorter interval will increase it.

While looking at one of the detailed user, I/O device, or storage displays you can force a calculation of new performance data simply by pressing the ENTER-key. Forcing a calculation in short intervals may cause a proportional increase in the data collection overhead for the selected display.

**Type of analysis:** Use of some performance displays may cause additional overhead while that particular resource is monitored.

# Chapter 2. Preparing Performance Toolkit

## Installation

Performance Toolkit for z/VM is pre-installed on your system. Instructions for enabling, customizing, and initializing it are located in the Program Directory.

### Installation Verification

Follow the steps outlined in Chapter 3, "A Guided Tour," on page 63 to make sure everything works as expected. You can learn more about this program as you review this section.

## Remote Performance Monitoring Facility

The remote performance monitoring facility is intended to allow:

- Efficient central performance monitoring for many remote systems, by displaying the information on any exceptions found on these systems on a single screen
- Concurrent multiple access to the central monitor machine's data
- Performance data retrieval from local and remote Performance Toolkit monitoring machines, and display of the data in a manner very similar to native performance monitoring mode in the respective monitoring machine

The performance data retrieval and display function will usually be invoked when the central monitoring facility has indicated exceptions which need closer investigation. The facility is based on data transfer from remote systems using APPC/VM. This method has the advantage of using the same data transfer method for multiple z/VM systems while requiring only a small amount of data to be transferred. Once a central collector is set up, it will establish communication with other systems at the FL520 and above function levels and "pull" data from them. Note that earlier releases of Performance Toolkit used RSCS links and that these earlier releases will continue to "push" data to the central collector. See Figure 1 on page 10.
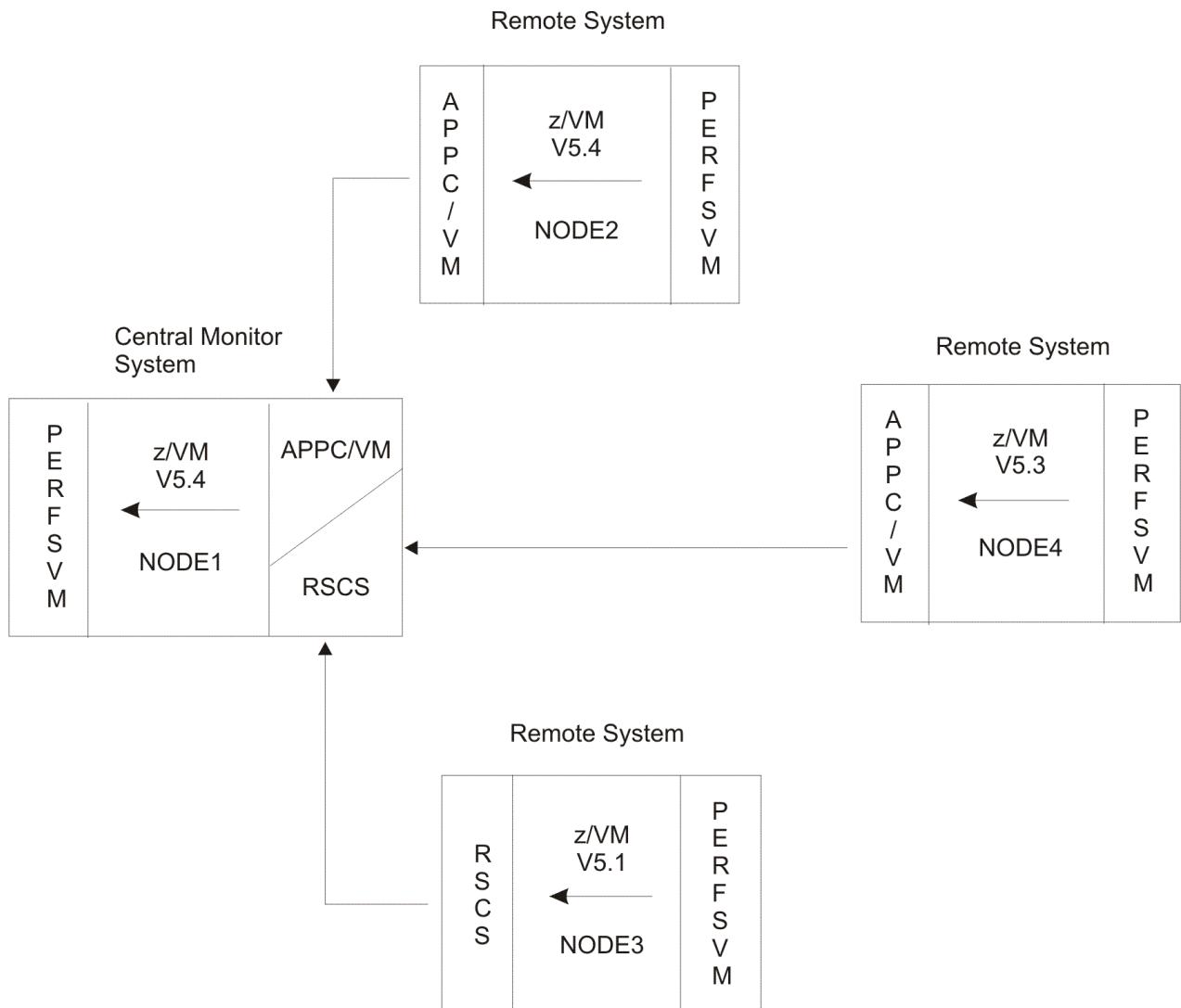
*Figure 1. Data Flow for Central Performance Monitoring of Remote Systems*

Multiple central collector machines can be defined (for example, for backup). In the remainder of this chapter you will find a detailed description of the remote performance monitoring facility with the following sections:

- Implementation
- Using remote performance monitoring
- Operation and data flow

Refer to if you are a general user.

## Implementing Remote Performance

To implement remote performance, do the following:

1. Define the virtual machine which is to act as the central monitor machine. It does not need any CP command privilege classes other than 'G', but it may require a considerable amount of A-disk space. Three generations are kept of the exception log files '*node-ID* EXCPLOG*n* A' and of the performance data log '*node-ID* RMONLOG*n* A' for each of the monitored systems. Provide at least the equivalent of additional 3 cylinders 3390 per monitored system.
2. Create the file FCONRMT SYSTEMS A in the central monitor machine with a list of all the remote systems which are to be monitored.

Each record of the file must contain the definitions for one system, in the sequence

```
node-ID   user-ID   type  append  <appcnick>
```

in free format, separated by at least one blank, where

**node-ID**
is the RSCS node-ID of the system

**Note:** The RSCS node-IDs or system-IDs to be used in the FCONRMT SYSTEMS file must not be SYSLOAD, SYSMENU, or any abbreviations of the 'RETurn' command. These are reserved strings; unpredictable results may occur when using them. The node-ID can contain only the following characters: letters A-Z, a-z, digits 0-9, and the special characters +, -, $, @, and _ .

**user-ID**
is the user-ID of the Performance Toolkit data collector machine for the system

**type**
is the system type

**append**
is the append specification ('Y' or 'N'). Set it to 'Y' if you want all performance data which you retrieve from the system to be saved in a history file for future reference

**appcnick**
is the nickname of an entry in a communications directory which describes the APPC/VM connection to the remote system. This entry is required to establish a connection to a remote system using Performance Toolkit at function level FL520 and above. Remote systems using earlier supported levels of Performance Toolkit will continue to work using RSCS links if no *appcnick* is specified.

See section "Central Monitoring Function" on page 14 for a more detailed description of these specifications.

The following is an example how the systems shown in the figure on the preceding page could be defined as follows:

```
*----------------------------------------------------------------------*
* System Definition File for Remote Monitoring                         *
*----------------------------------------------------------------------*
*Node-ID  PERFKIT-ID  VM-Type  Append  Nickname
*|        |           |        |       |
NODE1     PERFSVM     z/VM5.4  N       FCXRES0
NODE2     PERFSVM     z/VM5.4  N       FCXRES2
NODE3     PERFSVM     z/VM5.1  N
NODE4     PERFSVM     z/VM5.3  N       FCXRES4
```

Such a file should be accessible for the central monitor machine on system NODE1, and this file, or similar files, must also be available to all users who want to use the remote monitoring function from their own machines, on *any* of the systems.

3. Set up a file FCONRMT AUTHORIZ in each remote data collector machine from which you intend to retrieve data. This is the authorization file for the remote data retrieval facility. It must contain one record per authorized user. When an APPC/VM conversation is started, the remote user appears as a local userid to the server machine – that is, relevant entries in the FCONRMT AUTHORIZ file will have to be made with the SYSTEMID of the target system. The authorization for the collector at NODE1 in the FCONRMT AUTHORIZ on NODE2 should look like this:

```
Nodeid1 userid1 S&FSERV DATA
Nodeid2 userid2 DATA CMD
  ...      ...
Nodeidn useridn EXCPMSG
```

Where *userid1* is the collector userid on NODE1.

For our example, the first line would be:

```
NODE2 PERFSVM S&FSERV DATA
```

The authorization specifications can be a combination of:

**DATA**
> to allow performance data retrieval by the specified user

**CMD**
> to allow submission of CP and CMS commands to be run in the PERFKIT performance data collector machine. One of the CMS commands which can be run is the Performance Toolkit command interface module FCONCMD. Giving somebody CMD authority will, therefore, give that user also the authority to run all Performance Toolkit subcommands.
>
> *Any* CP/CMS command can be run, restricted only by the machine's privilege classes. A CMD specification also implies DATA authority.

**CPQRY**
> to allow submission of CP QUERY and INDICATE commands only to be run in the PERFKIT performance data collector machine, that is, this is for a small subset of the commands that would be authorized with the CMD argument described above. CPQRY authorization may be all that most users need, and the CP commands that can be used cannot directly endanger your system's integrity, so it is not necessary to be too restrictive with this argument. Because of the relatively harmless nature of the commands, CPQRY authorization may also be given to WWW users. However, in order to prevent accidental automatic WWW access to CPQRY authorizations intended for APPC/VM sessions as soon as the WWW interface is activated, an additional special entry

```
nodeid  ::WWW::  CPQRY
```

> is required in the remote server's authorization file to indicate that WWW users connected to a specific WWW interface may be given their defined CPQRY authorization. This indicates that the logon method implemented for the WWW interface at system *nodeid* is considered sufficiently safe to allow users their normal CPQRY privileges.

**EXCPMSG**
> to allow submission of user-generated exception messages to the data collector machine, for inclusion in the system's consolidated exception log file.

The file must contain at least the local system ID and the user-ID of the central monitor machine you set up with DATA authority to allow retrieval of performance data by this machine. *It is your responsibility to keep all systems secure* by doing the following:

- Giving the CP/CMS command run authorization only to people with a real need
- Restricting the privileged CP command range to the minimum set required for performance analysis. This can best be achieved by using the MODIFY command or statement and a special privilege class for performance monitoring commands and CP diagnose operations.

An asterisk in column one will cause the record to be treated as a comment (that is, ignored). Comment lines starting with double asterisks will be included in the machine's CONLOG file, as will be all of the actual data records. Asterisks may also be used for node-ID and/or user-ID to indicate that a specific user-ID should be authorized on ALL nodes, or that ALL users on a specific node are to be authorized, but the first asterisk must then *not* be on column one. Place records with asterisk specifications at the bottom of the file: The records are scanned from top to bottom, and the first record with an acceptable nodeid/userid specification will be used for authorization checking, that is, more specific nodeid/userid specifications will never be looked at if a previous general record with asterisks also fits. ***Use of asterisks can present a security exposure and should be carefully evaluated.***

This file must be accessed by the machine running the performance monitor, preferably on its A-disk. Restrict write access to this disk and LOGON to the collector machine to prevent unauthorized changes to the file.

All remotely run commands, whether successful or not, are inserted into the server machine's basic mode screen and console log, thus providing an audit trail of remote user's actions. A special action message

```
   FCXAUT410A Unauth. cccc cccc request by userid at nodeid
```

will also be inserted when an unauthorized retrieval attempt is detected.

The following is an example of an FCONRMT AUTHORIZ file as it could be used for the PERFSVM machine of the system NODE1 (see Figure 1 on page 10):

```
**-------------------------------------------------------------------**
** Authorization file for local and remote performance data        **
** retrieval and command execution.                                **
**-------------------------------------------------------------------**
*Node-ID  User-ID   Authorized for ..
*|        |         |
NODE1     PERFSVM   S&FSERV DATA
NODE2     PERFSVM   CMD DATA
NODE3     PERFSVM   CMD DATA
NODE4     FCON      CMD DATA
 *        PERFUSER  DATA
NODE1     *         DATA
```

The statements mean that:

- All the performance data collector machines which submit data to the central collector machine on NODE1 are also authorized to retrieve data from it and to run commands on that node.
- Users PERFUSER on *any* node are allowed to retrieve data. Be careful if you cannot control all the systems in the network. If other (foreign) systems are connected, a user with the identification of PERFUSER would also be authorized. *It is strongly recommended to always specify both the node-ID and the user-ID if command processing is authorized by the statement.*
- *All* the users on NODE1 are authorized to retrieve performance data for that node (their own), but not to run commands.

Similar FCONRMT AUTHORIZ files would have to be made available to the other data collector machines, too.

In large centers where many different z/VM systems have to be managed, you can also use the additional specifications described under "Extended Input File Processing" on page 93 in addition to the actual authorization statements. They give users more flexibility for setting up common files, and for separating system-specific and generic definitions.

Note that the definitions in file FCONRMT AUTHORIZ are read only once during initialization. Use the command FC RELOAD AUTHORIZ if you need to refresh the in-storage copy of the authorization data without restarting Performance Toolkit (see the "FCONTROL RELOAD" subcommand in the *z/VM: Performance Toolkit Reference*).

4. Activate threshold monitoring and user monitoring on all remote systems by doing the following:

- Setting suitable thresholds for the performance variables you want to monitor (see the "FCONTROL LIMIT" subcommand in the *z/VM: Performance Toolkit Reference*)
- Activating the user monitoring function, so that users in a CPU or I/O loop are detected (see the "FCONTROL FORCEUSR" subcommand in the *z/VM: Performance Toolkit Reference*).

Include these commands in file FCONX $PROFILE of the remote data collector machines.

5. Activate the communications interface by uncommenting the 'FC MONCOLL VMCF ON' entry in the FCONX $PROFILE. See "Retrieving Performance Data Using the VMCF Interface" on page 27 for more information.

Remember to restart PERFKIT to pick up these changes.

**Note:** Before restarting PERFKIT, first log off and then log back on the PERFSVM userid to ensure a clean environment.

# Using Remote Performance Monitoring

## Central Monitoring Function

Enter the subcommand FCONRMT in Performance Toolkit basic mode to invoke the remote performance monitoring facility. You will see an initial menu similar to the following:

```
FCX198          Performance Toolkit Remote Monitoring Facility      Remote Data

Sel Node-ID   Time   ------------- Exceptions & CPU Load -------------> AvExcp
 _  EHNIVM0A  15:12  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> :      .00
 _  WALTERVM  15:12  >>>>>>>>>>>>>>>>>>====                          :      .04
 _  BREUVM    15:12  %SPSL>>>>>>>>>>>>>>>>>>====                     :      .08
 _  DUFAVM1   15:12  >>>>>>>>>>>>>                                   :        0
 _  EHNIVM01  15:12  Cache-Sub>>>>>>>>>>>>>>>>>>>>                   :     1.05
 _  STUTVM1   15:12  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>           :      .35
 _  SDFVM003  15:12  Usrlimit>>>>>>>>>>>>>>>>>>>>>>>>>>>             :      .44
 _  CHVM1     15:12  CP-Read>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>          :     1.74
 _  VIEVMA    15:12  C1ES,_Usrlimit>>>>>>>>>>>>>>>>>>>>>>>          :     3.52
 _  GERMAZ01  15:12  Usrlimit>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>   :      .40
 _  GERMAZ06  15:12  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>                 :      .47
 _  BOEVM3    15:12  >>>>>>>>>>>>>>>>>>>>>>>>======                 :      .14
 _  SDFVM2    15:12  WSS-Loop>>>>>>>>>>>>>>>>>>>>>                   :      .56
 _  HERVM1    15:12  >>>>>>>>>>>>>>>>>>>>>>>>>>>                     :      .00
 _  BOEVMN    03:50  Not updated for  683 min.                      :      ...
 _  STUTVM4   15:12  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>          :      .89
 _  GERMAZ05  15:12  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>    :      .53

Select  e: exception log  h: history data  m: monitor
Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 2. System Load Overview Screen*

The contents of the initial menu depend on the system definitions you prepared in the file FCONRMT SYSTEMS: one line will be shown for each of the defined systems, where

**Sel**
is the column where you can enter further selection commands. Enter

**e**
to display the accumulated exception log file for the selected system

**h**
to display the history data overview for the selected system

**m**
to display the accumulated summary performance data for the selected system (same data as on performance REDISP screen of the system). The performance monitor MENU screen will be shown initially when an APPC/VM session is established to the target system (controlled by an APPC/VM nickname entry appended to the system's FCONRMT SYSTEMS entry).

Pressing ENTER without inserting one of the above characters in the `Sel` column corresponds to either an

- 'e'-selection when exceptions are currently indicated in the load bar of the system,
- 'm'-selection when no exceptions are indicated at the moment.

**Node-ID**
is the RSCS node-ID of the VM system

The remaining fields are intended for the remote monitoring function only. They will not provide useful data unless the remote systems have been set up to continuously send performance information to this central monitor machine (command 'FC MONCOLL REMSEND ...', see "Implementing Remote Performance" on page 10 for details).

**Time**

is a time stamp in the format *hh:mm* which indicates the time, in hours and minutes, when the last summary exception data record was received from the remote system. Note that the time stamp used is generated on *this* system so you can easily see whether an entry is being continuously updated. See the initial summary performance data display (selection 'm') for time stamps with the actual remote system's time.

**Exceptions & CPU Load**

will show:

- The CPU load on the system, indicated by a bar of greater than (>) signs (displayed with reverse video on displays which support extended highlighting). The length of the bar is proportional to the CPU load: 100% CPU load are indicated by a bar extending up to the end of the arrowhead (--->) in the header line. The number of greater than (>) signs is an indication of the real CPU load of the system. For VM systems which are running in an LPAR or second level, the 'logical' load (which includes suspended time) may be higher: The difference will then be indicated by a corresponding number of equal (=) signs on the right of the load bar.

- The names of performance variables which have exceeded the limits set on the remote system.

The CPU load bar will be overlaid by exception information. The inserted exception names can be

- Any redisplay variable name used with the FC LIMIT command
- The following expressions:

    **I/O cont**
    indicating at least one I/O device has exceeded the limit set for I/O contention

    **I/O resp**
    indicating the response time for at least one I/O device has exceeded the set limit

    **CH cont**
    indicating at least one channel of a VM system has exceeded the set contention limit

    **CH busy**
    indicating at least one channel of a VM system has exceeded the set 'channel busy' threshold

    **INT miss**
    indicates a missing interrupt for at least one I/O device

    **RSRV pend**
    indicates a RESERVE is pending for a device which has not run any I/O operations during the last measuring interval

    **CPU-loop**
    indicates at least one user is in a CPU loop

    **I/O-loop**
    indicates at least one user is in a I/O loop

    **WSS-loop**
    indicates at least one user appears to be in a general loop, with a practically constant working set size

    **Usrlimit**
    indicates at least one user has exceeded one of the thresholds set for user resource consumption (CPU, I/O rate, or UR I/O rate).

- The string `'none'` if no exceptions have been found, and if the CPU load is lower than the minimum that can be shown
- The string `'Collect Error'` if a data collect problem occurred on the remote system
- The string `'no data received'` if no data have been received so far from the remote system
- The string `'not updated for nnn min.'` when a remote system has ceased transmitting updates for a period of at least 10 minutes

**AvExcp**

is the average exception severity code since the last RESET. The summary counters are automatically reset at midnight. Other reset times can be specified using the 'FC MONCOLL RESET ...' command.

Format: Averages will be shown as a single '0' if no exceptions have been logged since the last RESET. They will be shown as a number with one or two decimals if at least one severity code > 0 has been found, that is, decimals always indicate that exception log data are available for display, even if the value shown is '.00'.

The following colors will be used for the data in the `Exceptions & CPU Load` and `AvExcp` fields:

**green**

for an exception severity code of 0

**red**

for an exception severity code of 1 or 2

**pink**

for an exception severity code of 3 or 4

**yellow**

for an exception severity code of 5 or 6

**wpresse**

for an exception severity code of more than 6

The severity codes are determined by dividing the sum of the 'weight' factors of all exceptions by 10. It is the system programmer's responsibility to set these weight factors so that color changes match the importance that your installation places on the occurrence of specific exceptions. See the "FCONTROL FORCEUSR", "FCONTROL LIMIT", and "FCONTROL USRLIMIT" subcommands in the *z/VM: Performance Toolkit Reference* for more information on how to specify 'weights' for the different exception messages.

**Usage notes:** This is essentially a centralized exception monitoring facility. It will automatically show exceptions detected by any of the remote systems: New system exception data will be inserted into the initial screen in 1-minute intervals, or when the screen is updated following a command entered, whichever occurs first.

Different colors will be used for displaying the `Exceptions & CPU Load` data fields, depending on the exception code found. So, provided that you have suitably set up threshold and user monitoring in your remote systems, you need now only watch the initial systems overview screen for highlighted status field information to be alerted in case of problems.

For any of the systems where exceptions have been logged, you can then:

• View the exception log for the system. Insert an 'e'' in front of the corresponding system and press ENTER. You will be shown the system's *Exception Log Display* with the day's exception messages for the system.

• View the collected summary performance data for the system: insert an 'm' in front of the corresponding system on the initial menu. The *Remote Performance Log Display* for the system will then be shown.

• View previously collected performance data for systems where the append flag has been set to 'Y': insert an 'h' in front of the corresponding system on the initial menu. The *History Data Selection Display* will be displayed.

See the description on the following pages for more detailed information on each of these displays.

Once you have selected a specific system for which to display data, by either the 'e', 'h', or 'm' selection characters, you need not return to the initial menu for selecting another type of data for the same system: use the commands

**EXCept**

to display the system's exception log

**HIStory**

for displaying performance history data (previously retrieved performance data displays)

**MONitor**
　　　or any valid data retrieval command to display either the system's performance log, or the retrieved performance data display

Entering a QUIT command or pressing the corresponding PF-key, will bring you back to the initial systems overview display, where you can select any of the other systems for displaying data. You can also switch directly between different systems by just entering another system's node-ID.

### Concurrent Multiple Access to Central Data

Additional images of the system overview display created in the central data collector machine can be reproduced concurrently in multiple other machines, provided that direct access to the central machine's A-disk is possible. Users which have been authorized to link to the central monitor machine's work disk can invoke the FCONRMT function in their own machines so as to give an exact copy of the central monitor machine's display:

FCONRMT *user-ID <addr>*

(See the "FCONRMT" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.) The central machine's disk with device number '*addr*' will be linked and automatically re-accessed. The display is updated, in one-minute intervals, and you can use all other displays as if you were working directly in the central monitor machine. Note however that each of the additional users will require authorization to do the following:

- LINK to the central monitor machines A-disk (for example, RACF PERMIT)

- Actively retrieve additional data from any of the systems (FCONRMT AUTHORIZ files on the remote systems. The central collector machine's authorization for data retrieval is not automatically extended to other users which access its data).
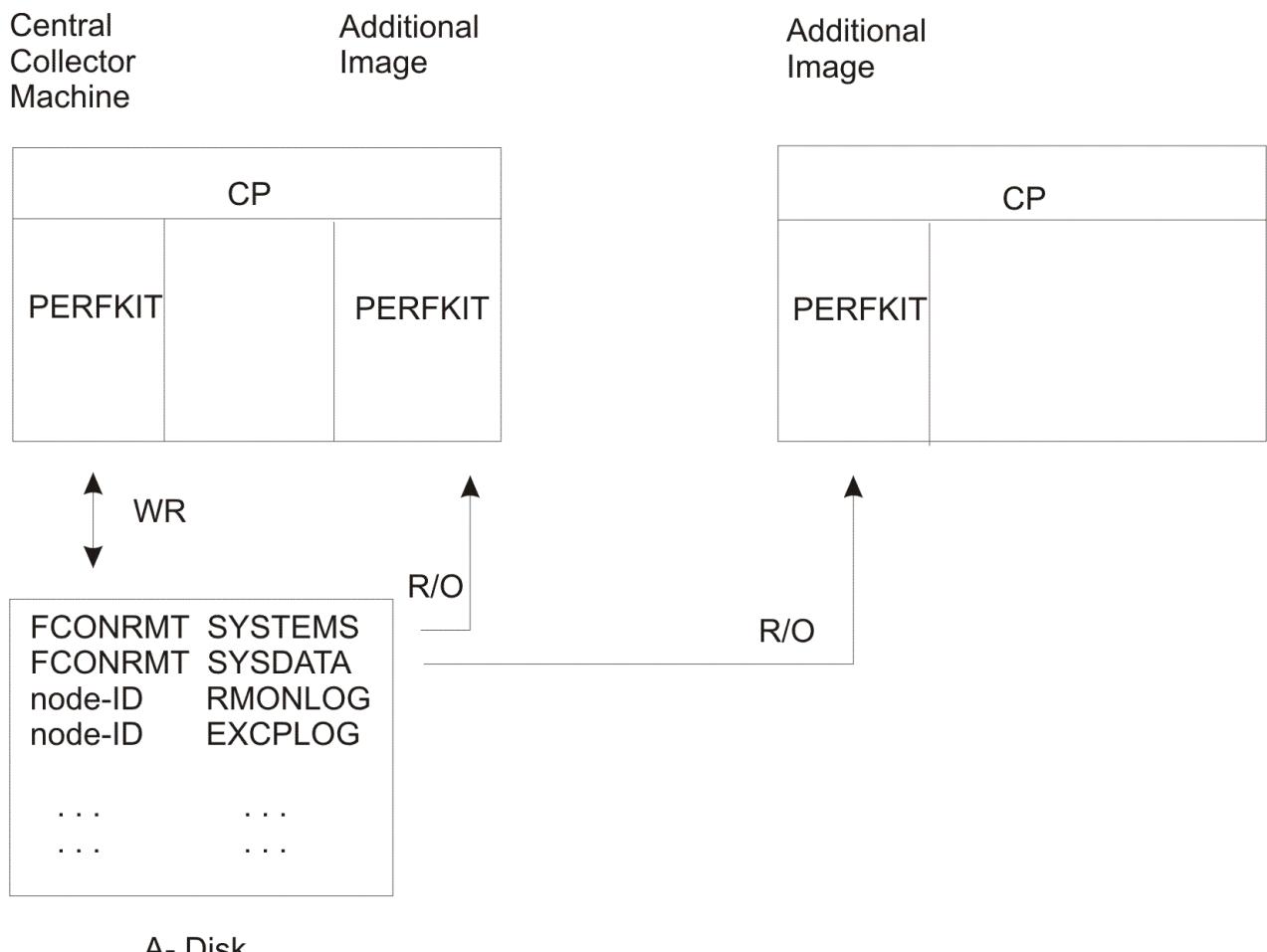


Figure 3. Central Collector Machine and Additional Images

You can control which systems are to be shown by using the FCONRMT SYSTEMS file.

- If you have no FCONRMT SYSTEMS file, all the systems will be shown, similar to the original display in the central data collector machine.
- If the FCONRMT SYSTEMS file is found when remote monitoring mode is initially entered, only data for the systems defined in the file will be shown .

Although the central monitor data disk will be re-accessed before data read attempts, errors may occur if the source data have been modified between the access and read operations (for example, history source data deleted while you are looking at data from the same system).

Availability of an access control program (for example, RACF®) for controlling minidisk access is assumed: no provision has been made to allow entering of MDISK read passwords when linking to the central collector machine's disk. On systems without such a facility a possible bypass is to do one of the following:

- Include a LINK statement to the central data collector machine's A-disk in the directory of the authorized users (for example, LINK PERFSVM 191 1A1 RR)
- Let the authorized users then specify their own ID, and the virtual device number defined in the directory LINK statement, when activating FCONRMT (for example, FCONRMT MYID 1A1)

LINK commands to one's own minidisks do not require passwords.

You could, of course, also specify a READ password of ALL for the central collector machine's A-disk, but this is not recommended because you could no longer control access to the disk.
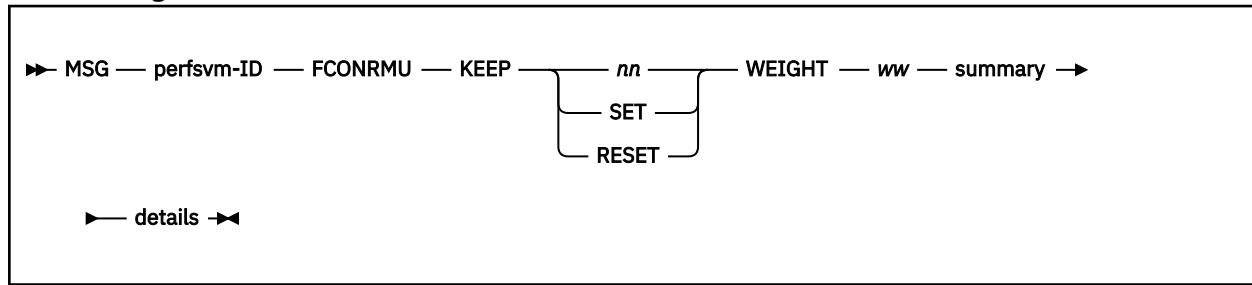
### Submitting User-generated Exceptions

Exception monitoring is not restricted to exception information generated by the exception, threshold, and user monitoring functions of Performance Toolkit. An interface is provided which allows the integration of exception messages generated by other monitoring tools (here called 'user exceptions') together with any Performance Toolkit generated data for display on the following displays:

- System overview display (exception summary)
- Exception log display (detailed exception messages)

Note the following:

- 'User exception' data must be submitted to the Performance Toolkit performance data collector machine *on the same system* where the local monitoring tool detected the exception.
- The exception data must be sent to the Performance Toolkit data collector machine as CP messages, in the following format:



**perfsvm-ID**
    is the user-ID of the performance data collector machine

**FCONRMU**
    tells Performance Toolkit that the message contains user exception data

**KEEP**
    is a keyword which allows you to tell Performance Toolkit for how long the summary exception information '*excp-summary*' is to be shown in the exception and CPU load field:

    **nn**
        indicates the time, in minutes, for which the summary exception information is to be shown

**SET**

indicates that the summary exception information is to be shown until a further KEEP specification for the same summary data specifies something different

**RESET**

indicates the summary exception information *excp-summary* is to be removed from the display.

Note that only the most recent specification for a specific type of user exceptions *excp-summary* will remain active: new specifications supersede all previous ones of the same type.

**WEIGHT ww**

allows specifying the 'weight' to be given to the exception, where *ww* is the weight factor. The weights of all exceptions are added together and the result divided by 10 to yield a severity code which is used in setting the color of the load bar with the exception summary on the system overview screen.

**summary**

is the summary exception information to be appended to the exception and CPU load field. Up to 16 characters of the first word found after the FCONRMU identifier will be treated as exception summary (cannot contain blanks, will be truncated if longer than 16 characters). Keep this word as short as possible: a total data length of 30 characters is available for *all* exception summaries in the 'Exceptions & CPU Load' field, and the remainder will be truncated.

**details**

is the detailed exception description which is to be appended to the exception log of the system. Up to 69 characters of the description can be displayed, prefixed by an 8-byte time stamp, the remainder will be ignored.

- The submitting user must previously have been authorized for submitting user exception data by a corresponding *node-ID user-ID* EXCPMSG entry in the FCONRMT AUTHORIZ file of the data collector machine (see also "Implementing Remote Performance" on page 10)

The local PERFKIT data collector machine will collect all user exceptions, append the exception summary information to any other summary exception data it had generated itself, and submit the detailed exception text to the central monitor machine, for inclusion in the common exception log file of the system. The consolidated exception summary and system load data will also be sent to the central machine in one-minute intervals, for display on the system overview screen. The average exception count fields on this screen include all user exceptions submitted for the respective systems.

Note that the data refresh cycle is one minute. Exception information which you expect to be shown for longer periods must be submitted using the 'KEEP xxx' arguments.

## Exception Log Display

For all systems which have been set up to submit summary performance data and exception information by the command 'FC MONCOLL REMSEND ...', the received exception messages will be collected in file '*node-ID* EXCPLOG  A'.

Enter the 'e' selection character on the initial FCONRMT menu or the command EXCEPT from a history or monitor data screen, to display the day's exception log for the system:

```
FCX153              Performance Toolkit Exception Log              CHVM1
10:50:12 FCXDEV320A Device 0853: resp. time 83ms exceeds limit 40ms (9.4 IO/s)
11:55:15 FCXPEU317A User RVS %CPU 48.0 exceeded threshold 40.0 for 2 min.
12:00:38 FCXPEU317A User RVS %CPU 47.3 exceeded threshold 40.0 for 7 min.
13:23:26 FCXPEU317A User GJA %CPU 61.7 exceeded threshold 40.0 for 2 min.
14:03:59 FCXPEU317A User PBO %CPU 57.3 exceeded threshold 40.0 for 2 min.
14:08:08 FCXPEU317A User PBO %CPU 82.1 exceeded threshold 40.0 for 2 min.
14:12:17 FCXPEU317A User PBO %CPU 72.9 exceeded threshold 40.0 for 2 min.
14:39:09 FCXPEU317A User PBO %CPU 90.1 exceeded threshold 40.0 for 2 min.
14:50:33 FCXPEU317A User PCHA %CPU 45.3 exceeded threshold 40.0 for 2 min.
15:00:12 FCXDEV319A Device 0847: queue length .66 exceeds limit of .40
15:00:12 FCXDEV320A Device 0847: resp. time 65ms exceeds limit 40ms (6.1 IO/s)
15:00:12 FCXDEV319A Device 0853: queue length .64 exceeds limit of .40
15:00:12 FCXDEV320A Device 0853: resp. time 91ms exceeds limit 40ms (10 IO/s)
16:11:21 FCXPEU338A User ZO in WSS loop during last 10 minutes
16:27:01 FCXPEU338A User ZO in WSS loop during last 10 minutes
16:46:42 FCXPEU317A User HMO %CPU 85.6 exceeded threshold 40.0 for 2 min.
16:50:49 FCXPEU338A User HMO in CPU loop during last 5 minutes
16:51:50 FCXPEU317A User HMO %CPU 82.7 exceeded threshold 40.0 for 7 min.
16:53:54 FCXPEU317A User PBO %CPU 65.2 exceeded threshold 40.0 for 2 min.
17:36:07 FCXPEU317A User CORADS %CPU 68.5 exceeded threshold 40.0 for 2 min.


Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 4. Exception Log Display*

Provided that suitable thresholds have been defined on the remote system ('FC SET LIMIT ...' command) the display will tell you:

• Which performance variables exceeded the set thresholds, and by how much

• How long the exceptions have persisted (messages with time stamps)

The last (most recent) part of the system's exception log file will be shown first. Use the scrolling PF-keys or commands to browse through the remainder of the day's exception messages for the day.

Error message

FCXxxx421E No exception data available for node *node-ID*

will be shown instead when the 'e' selection is entered for a system which has not submitted exception messages so far.

Three generations of the EXCPLOG file are kept for each system: '*node-ID* EXCPLOG A', '*node-ID* EXCPLOG1 A', and '*node-ID* EXCPLOG2 A', for the current day and two previous days respectively. The files are automatically renamed at midnight, and the oldest one erased; you can copy them onto another disk if you want to keep more than these three generations.

To some extent, you can control the insertion of exception messages into the central exception log files.

• Exception messages from a remote system can be suppressed by entering the appropriate 'FC PROCESS ...' commands with an output display type other than ERRMSG or CPAMSG, or with the NODISP argument *on the remote system*.

• Because the remote system will transmit *all* alert messages, both of the ERRMSG types created by Performance Toolkit's exception monitoring function, *and* any CP 'alert' messages of the type CPAMSG, you can cause other output lines on the remote system to be transmitted to the central monitor facility by changing their output display type to CPAMSG.

• User-generated exception messages of any kind can also be included in the central EXCPLOG file. See "Submitting User-generated Exceptions" on page 18 for details.

**Working with the Exception Log Display:** When exception messages appear in the exception log, you will usually want to proceed with some kind of analysis in order to determine where the problem is and whether it still persists. Code has been included in the program to help you in this task in the case of alert messages which have been generated by the threshold and user monitoring functions of Performance Toolkit. To begin analysis, do the following:

1. Place the cursor on the time stamp field of the message for the problem you want to analyze

2. Press the ENTER key

A retrieve command will automatically be generated for the performance data display which is most likely to help you in analyzing that particular problem. Examples:

- The general I/O device display will be retrieved when the message indicates that the total I/O rate threshold, or the percentage of users in I/O wait, were exceeded
- The CP owned device display will be retrieved when the message indicates that either the SPOOL or PAGE space utilization threshold was exceeded
- The detailed user display will be retrieved when the user monitoring function indicates that a specific user was either looping or had exceeded its thresholds

The general CPU screen, which will be shown for many of the 'threshold exceeded' messages, will also be shown for all non-Performance Toolkit alert messages, for which the program cannot determine the course of action to take.

Note that the time stamp fields are not intended for entering commands (all data entered will be ignored) and you must not enter a command on the command line if you intend to use the automatic features. If a command is found on the command line, *only* that command will be run and the cursor position will be ignored.

## Remote Performance Log Display

For all systems which have been set up to submit summary performance data and exception information by the command 'FC MONCOLL REMSEND ...', the received performance data will be collected in the file '*node-ID* RMONLOG  A'.

To display the day's performance log for the system, do either of the following:

- Enter the 'm' selection character on the initial FCONRMT menu or the subcommand MONITOR on an exception log or history data display, of the same system
- Enter the node-ID of the system on the command line of any FCONRMT screen

```
FCX101              Performance Toolkit Remote Monitoring Facility        CHVM1

TIME    CPU   %CP   %EM   %WT   %SY  -   -    IO/S  VIO/S  PG/S  XPG/S  DIAG  PRIV  LOGN  ACT
14:09   163   53    110   237   11   .   .    308    270   194    522   1199  2117  1519  226
14:10   153   51    102   247   11   .   .    322    269   242    571   1157  1957  1520  260
14:11   222   57    165   178   12   .   .    349    323   207    600   1166  2072  1517  257
14:12   146   51     95   254   11   .   .    307    296   141    418   1110  1981  1515  214
14:13   176   49    127   224   11   .   .    284    328   180    422   1084  1832  1510  228
14:14   167   54    113   233   11   .   .    354    323   177    429   1208  2069  1514  232
14:15   225   67    158   175   15   .   .    441    345   218    517   1542  2672  1513  234
14:16   212   66    146   188   14   .   .    403    332   223    622   1572  2847  1516  236
14:17   187   57    130   213   12   .   .    376    364   195    527   1193  2269  1514  229
14:18   164   52    112   236   11   .   .    340    302   203    561   1155  1953  1516  228
14:19   275   59    216   125   14   .   .    406    397   209    521   1261  2293  1518  230
14:20   177   51    126   223   11   .   .    319    274   203    477   1128  1939  1513  218
14:21   180   54    126   220   11   .   .    360    313   170    492   1292  2294  1516  210
14:23   144   42    102   256   10   .   .    269    240   133    442    938  1600  1520  213
14:24   212   53    159   188   12   .   .    363    315   123    456   1188  2171  1522  223
14:25   160   51    109   240   11   .   .    397    322   199    470   1165  2023  1528  266
14:26   149   50     99   251   11   .   .    320    222   190    437   1036  1873  1526  247
14:27   235   57    178   165   13   .   .    374    483   204    535   1359  2354  1524  238


Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F10=Left  F11=Right  F12=Return
```

*Figure 5. Remote Performance Log Display*

This display corresponds exactly to the performance 'redisplay' screen of the remote system. It will help you to get a clearer picture of the situation, because it shows how other performance values changed when the selected thresholds were exceeded. The initial display will show the bottom (most recent) part of the performance log. Use the scrolling PF-keys to browse through the remainder of the accumulated performance data of the selected system. Note that the performance data also extends to the right and use the LEFT/RIGHT shift keys or commands for viewing performance variables which are not visible on the initial screen.

The following screen will be shown when you enter the 'm' selection for a system which has not submitted performance data so far.

```
FCX200              Performance Toolkit Remote Perf. Monitoring Facility         Node-ID

          No automatically collected data available for this node.
          Retrieve the desired performance data by entering the
          corresponding selection command.
```

From either of these displays you can go on and request additional performance data. Enter the command which you would have used in performance monitor mode on the remote system for selecting a specific display. This is described in more detail in the following section.

## Remote Performance Data Retrieval and Display

Once you have started a remote performance monitoring session with the 'm' selection on the initial FCONRMT menu, performance data can be retrieved from the remote system and displayed simply by entering the performance monitor subcommand which you would use for displaying the data when in Performance Toolkit performance monitor mode. However, some restrictions apply:

- While most performance displays can be selected for direct retrieval, some others cannot (for example, GDDM graphics, see separate paragraph). The list below describes the subset of valid commands.
- Only some very basic command checking is done before sending the command to the remote system. Erroneous commands may be accepted and submitted to the target machine, where the error will eventually be detected. The error message generated on the target system will then be returned instead of the desired data. Be careful to avoid typing errors when keying in your commands.

*Data retrieval commands:* Most of the performance screen selection commands available in native monitoring mode are also accepted for remote performance data retrieval. See the "Performance Toolkit Subcommands and Screen Layouts" chapter in the *z/VM: Performance Toolkit Reference* for a detailed description of the monitor command format but be aware that the remote system may use a different level of Performance Toolkit if it is a different release of z/VM.

A special option '(nnn' is available for all of the performance data retrieval commands in the above list, in the format

*command ... ... (nnn*

where:

**nnn**
    specifies the number of output records to be returned from the remote system. The default value is *all* records, but specifying a smaller value may considerably speed up data transfer over highly loaded RSCS links (RSCS is often set up to select the smallest files first).

You can also set a new default number of lines to be retrieved by means of the FC RMTLINES command (see the "FCONTROL RMTLINES" subcommand in the *z/VM: Performance Toolkit Reference*). This default number will then take effect if you have not specified a number explicitly with the retrieval command, instead of returning *all* data, as with the initial setting.

Data will be transmitted from the remote systems using RSCS link, so response time will depend on the speed and current load of the links used. The status message

```
    Request pending
```

will appear in the message line at the bottom of the display while you are waiting for the remote data to arrive. However, the keyboard is not locked, and you may continue to browse through the performance data on the current display until the new data have been received and can be displayed.

Note that you will never receive a reply in the following instances:

- If an invalid target user address was specified

- If the target user is logged off (the RSCS message which indicates the target user does not exist, or is not logged on, is not intercepted; you will only see it when you leave remote monitoring mode and return to the basic mode screen)
- While some CMS program (for example, HELP) is active in the target Performance Toolkit data collector machine

Note that Performance Toolkit keeps only one common set of performance data. The retrieval of the following will interfere with the display of performance data in the target machine if the same display type is being viewed while data are also retrieved.

- Detailed device data
- Detailed SFS server data
- Detailed SPOOL data for a single user
- Detailed user data
- Detailed user storage data
- Performance plots

The display will then be changed as selected by the remote retrieval command.

Submitting the following commands will similarly switch the remote Performance Toolkit performance monitor to display average or current performance values, force collection of new CP control block data, or change the sorting sequence, not only for the user who requested the switch, but also for all others who are retrieving data from the same system.:

- AVERAGE
- CURRENT
- RECOMP and
- SORT

The layout of the remote performance data display is very similar to the one you would see if you were directly logged on to the remote system.

```
  Identifier                  Interval start and end times      Origin
      |                                    |                       |

   FCXnnn        CPU nnnn   SER nnnnn   Interval HH:MM:SS - HH:MM:SS      Node-ID
 ..... ..... ..... .....  ..... .....  ..... ..... ..... ..... ..... ..... .....
 ..... ..... ..... .....  ..... .....  ..... ........... ..... ..... ..... .....
 .....                                                                      .....
 .....                          performance data                           .....
 .....                                                                      .....
```

*Figure 6. Layout of Remote Performance Data Screens*

The only difference is the RSCS node-ID of the originating system which is inserted in the top right hand corner of the display. You can perform the following:

- *Browse* through the returned data using the corresponding PF-keys, as shown in the bottom line
- *Select other data* from the same system simply by entering the necessary command on the command line
- *Return* to the initial menu by pressing the QUIT PF-key

To select another system, you can either return to the initial menu or you can enter the other system's node-ID on the command line.

Support for GDDM graphics is as follows:

**APPC/VM:** Supports both the GRAPH*xxx* commands and the USTATG command for remote use, just as if the commands were entered for your own machine.

*RSCS:* GDDM graphics can be created using the GRAPHDET, GRAPHSUM, and GRAPHVAR commands. They cannot be retrieved directly from another system, but are based exclusively on previously retrieved data. To produce performance graphics you will first have to retrieve the necessary performance data from the remote system and then generate the graphics display locally. The following are possible sources for creating graphics output:

- REDISP screens with retrieved performance redisplay data
- The automatically collected summary data in the '*node-ID* RMONLOG*n*' files on the central data collector machine's A-disk
- The automatically collected '*mmddyy* PERFLOG*n*' files
- The accumulated performance history files 'ACUM PERFHIST'.

With the exception of the first example, these are all CMS files which reside on another machine's A-disk.

- You need either direct access to the corresponding disk (that is, you must be authorized to LINK to it), or
- You must be authorized to run CP and CMS commands in the machine which owns the files. You can then send the command 'CMS SENDFILE ...' to the owning machine and let the complete file be transmitted to you.

The following paragraphs give additional information on the source data which can be used for the different performance graphics types:

**GRAPHdet** displays are based on current performance REDISP data. They can be generated only while a REDISP screen is displayed (for example, the initial monitor screen which you see after entering the 'm' selection for a system for which a '*node-ID* RMONLOG' file is available or a REDISP display which has been directly retrieved from another machine).

**GRAPHSum** and **GRAPHVar** displays can be generated either from current REDISP data which are being viewed, like the GRAPHDET type of display, or from data in an RMONLOG*n*, PERFLOG*n*, or PERFHIST file when the 'FILE *fn ft fm*' arguments of these commands are used. The string 'FILE *node-ID* RMONLOG *'* will automatically be appended to GRAPHSUM and GRAPHVAR commands when they are entered from a non-REDISP screen and without the 'FILE ...' argument.

The variable names specified with the GRAPH*xxx* command must be valid for the system which supplied the source data.

**FCONTROL Commands:** FCONTROL commands entered while in remote data retrieval mode are assumed to be intended for the remote system and will be passed on to that system to be run. You will have to return to basic mode if you want to enter an FCONTROL command to take effect in your own machine.

Note that, because running of FCONTROL commands may change the setup of the remote Performance Toolkit server machine, the command will be accepted to be run only if the sender has been authorized for general command running on the remote system.

## Performance History Data Selection Display

For all remote systems where you have specified Y for the 'append' information, the received performance data will be collected in the file '*node-ID* FCONHIST  A'.

Enter the 'h' selection character on the initial FCONRMT menu to display performance history data for the system.

```
FCX199          Performance Toolkit Remote History Data Display      Node-ID

Sel Date       Interval                Type of Display
 _  2002/08/24  10:14:28 - 10:15:28 :  CPU
 _  2002/08/24  10:14:28 - 10:15:28 :  USER
 _  2002/08/24  10:15:28 - 10:16:29 :  DEV
 _  2002/08/24  10:15:28 - 10:16:29 :  DEV S
 _  2002/08/24  10:16:29 - 10:17:30 :  UPAGE
 _  2002/08/24  14:31:05 - 14:32:06 :  CPU
 _  2002/08/24  14:31:05 - 14:32:06 :  DEV
 _  2002/08/24  14:32:06 - 14:33:08 :  USER
 _  2002/08/24  13:30:15 - 14:33:08 :  UST
 _  2002/08/24  Initial    14:35:23 :  DEV 260
 _  2002/08/24  14:35:23 - 14:37:17 :  DEV 260
 _  2002/08/24  Initial    14:40:52 :  USER PROFCAL
 _  2002/08/24  14:40:52 - 14:42:25 :  USER PROFCAL
 _  2002/08/24  14:44:29 - 14:45:32 :  RED
 _  2002/08/24  14:37:17 - 14:51:45 :  DEV 260
 _  2002/08/24  14:58:28 - 14:59:30 :  USER
 _  2002/08/24  14:42:25 - 15:02:38 :  USER PROFCAL

   17 perf. displays saved; insert 's' to display or 'd' to delete
 Command ===>
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 7. History Data Selection Display*

The screen shows what type of performance data are available in the accumulated history file for the
selected VM system, together with the date and the measurement interval for the data.

Error message FCXxxx420E No history data available for node *node-ID* will be shown
instead when the 'h' selection is entered for a system for which no performance data have been saved in a
'*node-ID* FCONHIST' file.

### *Selecting Performance History Data*

Select history performance data by inserting an 's' to the left of the selected display and press the ENTER
key. The performance data will be displayed in the same way as when they were originally retrieved.
Use the scrolling PF-keys for viewing all the data available for the selected display. Further (current)
performance data can be retrieved simply by entering the corresponding data retrieval command: this will
cause an immediate switch to monitor mode, where the selected data will eventually be displayed once it
has arrived.

### *Deleting Performance History Data*

You can delete specific history data displays by inserting a 'd' on the left of the selected display. Multiple
'd' selections are allowed.

## Operation and Data Flow

The remote performance monitoring facility can run in any virtual machine (no special command privilege
classes required). Its operation is described in the following sections, referring to .

For central performance monitoring:

1. Activate the remote monitoring facility in the collector machine USER1 on system NODE*XX* by
   including the following entry in the FCONRMT SYSTEMS file:

   ```
   NODEYY   USER2  z/VM5.2 Y    FCXRES00
   ```
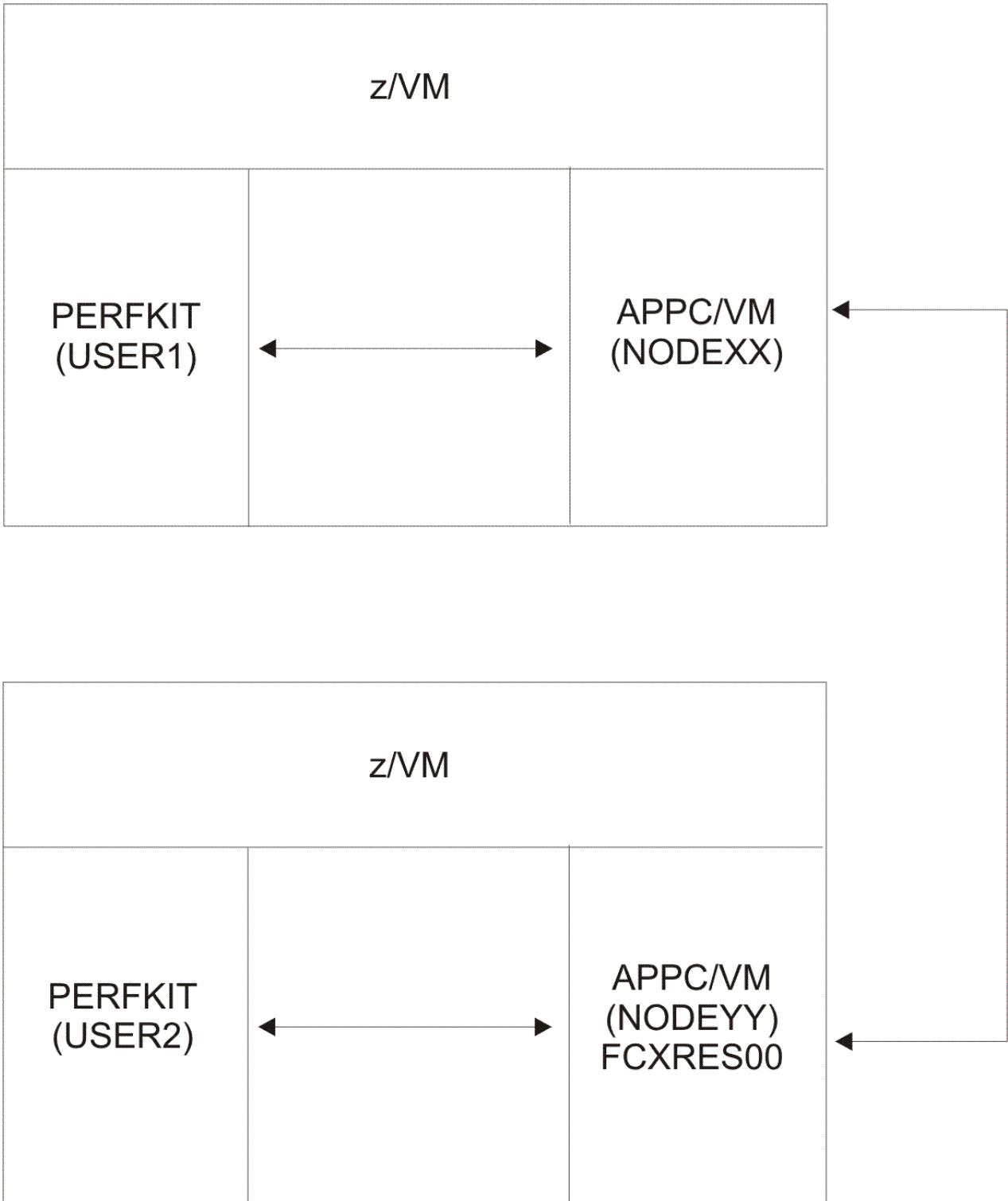
   When PERFKIT starts in the USER1 machine, it will try to establish an APPC/VM connection to the
   system resource FCXRES00 and request data from the PERFKIT running in the USER2 machine.

2. View the remote systems menu by entering the FCONRMT command from the USER1 userid. This
   menu will be refreshed in one minute intervals, or when the next command has been entered, and will
   display the updated system status.

**Note:** Note: The 'h' choice for history is available only for remote systems running earlier releases of PERFKIT. You can currently view the history data by entering the 'm' for monitor mode and then using option '32'.

For remote performance data retrieval:

1. Let us assume that you have invoked FCONRMT in virtual machine USER1 on system node NODE*XX* (this can be the same machine that also acts as central monitor machine, or any other user).
2. You must then select the system NODE*YY* you wish to retrieve data from by placing an 'm' next to the corresponding system entry. This will start a data retrieval session for system NODE*YY*.
3. Now you can enter Performance Toolkit monitor mode subcommands for the data you wish to view.
4. When you press enter, PERFKIT establishes an APPC/VM connection to resource FCXRES00 (as specified in the FCONRMT SYSTEMS file).
5. In the target virtual machine USER2 on node NODE*YY*, where the PERFKIT performance monitor is active, PERFKIT will:
   a. Check the FCONRMT AUTHORIZ file to verify that the requester has been authorized and either return an error message if the user is not authorized or else act on the request
   b. Return the data requested over the APPC/VM connection.

*Figure 8. Command and Data Flow for Remote Performance Monitoring and Data Retrieval*

## Retrieving Performance Data Using the VMCF Interface

The VMCF interface of Performance Toolkit is intended for retrieval of selected performance data by other virtual machines *on the same system*. For information about performance data retrieval from other z/VM systems, see "Remote Performance Monitoring Facility" on page 9.

The following methods of data retrieval using the VMCF protocol are supported:

- VMC compatibility mode
- The VMCX interface, which displays the retrieved performance data with field attributes for highlighting of selected data fields.

These methods are based on the SEND/RECEIVE protocol of VMCF.

The VMCF interface cannot be used for *retrieving* data from within a machine running Performance Toolkit if it has been enabled for *returning* data using the VMCF interface. These two functions (retrieving and returning) are mutually exclusive.

## Enabling the VMCF Interface

Initially, the VMCF interface is disabled. Before any VMCF data retrieval requests can be handled, the VMCF interface must be enabled using the FC MONCOLL VMCF ON command in the monitor machine. See the FCONTROL MONCOLL subcommand in *z/VM: Performance Toolkit Reference* for more information.

## Authorizing Data Retrieval

Data retrieval using the VMCF interface is controlled in the same way as remote data retrieval from other systems. The FCONRMT AUTHORIZ file must have been set up previously with the corresponding entries for retrieving data, running commands, or both.

## VMC Compatibility Mode

VMC compatibility mode is intended mainly to allow performance data retrieval for further processing in a REXX procedure. The performance data is returned as follows:

- As a block of several consecutive 80-byte records, one record per original output line
- With the block size depending on the available space in the calling program's buffer, but limited to a maximum of 8kB
- Without any special field attributes.

The preferred interface for this mode is CMS Pipelines. For more information about CMS Pipelines, see *z/VM: CMS Pipelines User's Guide and Reference*.

To retrieve performance data, use a vmc stage. A vmc stage places retrieved data in an internal buffer. You can then specify where you want the data to be placed.

### Syntax

The general syntax for vmc follows:

```
pipe vmc userid command ...   { ( nnn  }
```

where:

**userid**
is the user identification of the target machine where the Performance Toolkit performance monitor program is active (PERFSVM, for example).

**command ...**
specifies one or more data retrieval commands. You can specify:

- Any or all of the performance screen selection commands (except for the GRAPH*xxx* commands). See also the list in "Remote Performance Data Retrieval and Display" on page 22.
- Any or all of the scroll commands, to view different sections of the selected performance data.
- The control commands AVERAGE and CURRENT — for switching between displays for "current" data (that is, data for the last measuring interval) and "average" data — and SORT.
- The special command READCHEK *userid,* to retrieve information about the state of a selected virtual machine.

**(***nnn***

    specifies the maximum number of lines to be returned. Note that fewer lines than specified might be returned due to buffer size restrictions in the calling program or in Performance Toolkit.

**Note:** Special cases are as follows:

1. The REDISP subcommand supports some special arguments when used with the VMCF interface. Instead of the arguments that are described in the REDISP subcommand in *z/VM: Performance Toolkit Reference*, you can also select up to 15 of the REDISP performance variable names for display *on the same screen*, as long as they all fit on one line. This feature could prove particularly useful when retrieving data for further processing by your own programs. When specifying any of the ''MAX.' fields with user extremes, the field name must be specified in two arguments, for example: 'MAX. CPU' or 'MAX. RESIDENT'. Only the numerical part of the output fields is inserted and one of the following abbreviated headers is used to save space:

   **M_CPU**
      for the MAXIMUM CPU field

   **M_VEC**
      for the MAXIMUM VECTOR field

   **M_I/O**
      for the MAXIMUM I/O field

   **M_PAG**
      for the MAXIMUM PAGING field

   **M_RS**
      for the MAXIMUM RESIDENT field

   **M_MDC**
      for the MAXIMUM MDC field

   **M_XS**
      for the MAXIMUM XSTORE field

2. The READCHEK command allows information about a virtual machine's status to be retrieved. The command format is:

```
READCHEK userid
```

where:

***userid***
    is the user identification of the virtual machine that has status to be checked.

The returned information consists of a character string that contains a return code specifying the machine's status and some explanatory text. One of the following strings is returned:

```
RC   5: Missing argument: User-ID not specified
RC  22: Not authorized for 'LOCATE userid' cmd.
RC  45: User is logged off
RC   0: User is not disconnected
RC   1: User set to log off (read state or disabled wait state)
RC   4: User is in disabled wait state
RC   2: User is in CP read state
RC 145: User is in LOGOFF pending state
RC  57: User is disconnected and running
RC 512: Internal Perf. toolkit logic error
```

Where more than one of the conditions could apply, conditions are checked for in the preceding sequence (that is, you will not see a return code of 2 or 4 if the machine has already been flagged by CP to be logged off after 15 minutes; you will see return code 1 instead).

The function is intended to allow the monitoring of important service machines by another utility machine (VMUTIL with WAKEUP, for example).

### Examples

1. To place the retrieved *n* lines of I/O device performance data in the REXX variables `fconout.1` to `fconout.n`, use this host command (issued from REXX):

   ```
   'pipe vmc perfsvm device | stem fconout.'
   ```

   You can then process these variables using additional host commands issued from REXX. You might want to extract specific fields from some of the lines and use them as a basis for deciding on further actions.

2. To display the retrieved data directly on the console of your virtual machine, specify `console` as the output device:

   ```
   'pipe vmc perfsvm device | console'
   ```

   This host command (issued from REXX) lets all of the retrieved data be displayed in line mode. The data might fill several screens. The data transfer buffer that is used by CMS Pipelines has a size of 8kB, allowing transfer of up to 102 lines of 80 characters.

## The VMCX Interface

The VMCX interface is provided to allow the display of the retrieved performance data on your screen. The performance data is returned as follows:

- As one large block, *not* of 80-byte records
- With the block size adapted to fit the screen size of the caller's terminal
- With field attributes for highlighting selected fields.

This interface will show a screen of performance data very similar to the original performance display, with some of the fields highlighted. Note, however, that only 'normal' highlighting is possible, extended color and highlighting are not supported.

The screens presented by the VMCX interface are not interactive screens. There are no active input fields available within the displayed data. When viewing data using the VMCX interface, each report must be initiated by entering the appropriate Performance Toolkit command on the command line.

### Syntax

The general syntax for VMCX follows:

```
VMCX userid command ... { ( AUTO }
```

where:

**userid**
    is the user identification of the target machine where the Performance Toolkit performance monitor program is active (PERFSVM, for example).

**command ...**
    specifies one or more data retrieval commands. You can specify:

- Any or all of the performance screen selection commands (except for the GRAPH*xxx* commands). See also the list in .
- Any or all of the scroll commands, to view different sections of the selected performance data.
- The control commands AVERAGE and CURRENT — for switching between displays for "current" data (that is, data for the last measuring interval) and "average" data — and SORT.
- The special command READCHEK *userid*, to retrieve information about the state of a selected virtual machine.

**(AUTO**
> specifies that the VMCX interface should try to automatically refresh the data display in 60-second intervals by retrieving the same screen again. This option is valid only if the VMCX interface is invoked directly in native CMS, not under Performance Toolkit.

**Note:** Special cases are as follows:

1. The REDISP subcommand supports some special arguments when used with the VMCF interface. Instead of the arguments that are described in the REDISP subcommand in *z/VM: Performance Toolkit Reference*, you can also select up to 15 of the REDISP performance variable names for display *on the same screen*, as long as they all fit on one line. This feature could prove particularly useful when retrieving data for further processing by your own programs. When specifying any of the ''MAX.' fields with user extremes, the field name must be specified in two arguments, for example: 'MAX. CPU' or 'MAX. RESIDENT'. Only the numerical part of the output fields is inserted and one of the following abbreviated headers is used to save space:

   **M_CPU**
   > for the MAXIMUM CPU field

   **M_VEC**
   > for the MAXIMUM VECTOR field

   **M_I/O**
   > for the MAXIMUM I/O field

   **M_PAG**
   > for the MAXIMUM PAGING field

   **M_RS**
   > for the MAXIMUM RESIDENT field

   **M_MDC**
   > for the MAXIMUM MDC field

   **M_XS**
   > for the MAXIMUM XSTORE field

2. The READCHEK command allows information about a virtual machine's status to be retrieved. The command format is:

```
READCHEK userid
```

   where:

   **userid**
   > is the user identification of the virtual machine that has status to be checked.

   The returned information consists of a character string that contains a return code specifying the machine's status and some explanatory text. One of the following strings is returned:

```
RC   5: Missing argument: User-ID not specified
RC  22: Not authorized for 'LOCATE userid' cmd.
RC  45: User is logged off
RC   0: User is not disconnected
RC   1: User set to log off (read state or disabled wait state)
RC   4: User is in disabled wait state
RC   2: User is in CP read state
RC 145: User is in LOGOFF pending state
RC  57: User is disconnected and running
RC 512: Internal Perf. toolkit logic error
```

   Where more than one of the conditions could apply, conditions are checked for in the preceding sequence (that is, you will not see a return code of 2 or 4 if the machine has already been flagged by CP to be logged off after 15 minutes; you will see return code 1 instead).

   The function is intended to allow the monitoring of important service machines by another utility machine (VMUTIL with WAKEUP, for example).

## Usage Note

The VMCX interface works in two different modes, depending on whether it is invoked from basic CMS or from within Performance Toolkit.

1. Invoking VMCX from basic CMS starts a monitoring session. After the initially-requested performance data have been received and displayed, you will be prompted by this message to enter another data retrieval command:

```
 >>>>>>>>>> Enter next command, or 'QUIT' to exit <<<<<<<<<<
```

   The screen remains in `'VM READ'` status while waiting for the next command. You have to enter QUIT to exit from the VMCX module and to return to CMS. Pressing the ENTER key without any data while in such a VMCX session will cause the screen to be refreshed with new data, but it will not force re-calculation of data in the target machine.

2. When VMCX is invoked under Performance Toolkit, only the selected data is displayed and the screen is placed in `'MORE...'` status. Control returns immediately to Performance Toolkit when the `MORE...` status is cleared.

## Example

To display the CHANNEL performance screen, enter this command:

```
VMCX PERFSVM CHANNEL
```

The output usually contains highlighted fields, similar to the original Performance Toolkit performance screen. Note, however, that fields are highlighted only when the original default 'SCREEN' setting for VMOUT (VM output) is still active. If you have changed the setting, *all* of the output is displayed in a different color, highlighting, or both. You can reset it to the original default by entering this command:

```
SCREEN VMOUT DEFAULT
```

# APPC/VM Interface for Performance Data Retrieval

A special APPC/VM mode of remote performance data retrieval has been implemented to provide a more efficient means of data transmission between VM systems than (RSCS) file transfer. Two operating modes are possible:

- Retrieval from a PERFKIT machine on the same VM system
- Retrieval from a PERFKIT machine on another system either in the same TSAF collection created by TSAF or in the same Communication Services (CS) collection created by ISFC.

This section describes implementation and use of the APPC/VM interface in the following sections:

- "Implementing the APPC/VM Interface for a Local System" on page 32
- "Implementing the APPC/VM Interface for Remote Systems" on page 37
- "Using the APPC/VM Interface for Performance Data Retrieval" on page 43

## Implementing the APPC/VM Interface for a Local System

The interface is based on the use of global resources, that is, the resource names used must be unique. The default global resource name used when trying to establish an APPC/VM connection is FCXRES00 (server and requestor machines). You can choose between two options:

- Use the default resource name FCXRES00. This is the recommended method if you plan to use APPC/VM data retrieval only on one single local system. Make sure the default global resource name is not already used by some other service.

- Use any other resource name. You will be forced to choose this method if you plan to establish APPC/VM connections to other VM systems, too.

Connections between systems which are part of a local TSAF collection are defined in the same way as connections on a single system.

## Using Default Resource Name FCXRES00

The advantage of working with the default resource name of FCXRES00 is that no communications directory entries have to be created and activated. The only things you have to prepare for this setup are:

1. Add the following statements to the directory entry of the userid used for data collection (usually PERFSVM):

   ```
   IUCV *IDENT FCXRES00 GLOBAL
   IUCV ALLOW
   ```

   The first of these statements authorizes Performance Toolkit to make itself known to the *IDENT service as manager of the global resource FCXRES00, and the second statement authorizes any user to establish an APPC/VM connection to the resource(s) managed by the data collector machine.

2. Let the APPC/VM interface be activated automatically during startup of Performance Toolkit by including the statement

   ```
   FC MONCOLL VMCF ON
   ```

   in the FCONX $PROFILE of the data collector machine if you have not already done so before (activation of the VMCF and APPC/VM interfaces is controlled by the same command).

3. Create an FCONRMT AUTHORIZ file with entries for all users which are to be authorized for performance data retrieval, and place it on the PERFSVM machine's A-disk (see "Implementing Remote Performance" on page 10 for details).

4. Activate the changed directory and logoff/logon the data collector machine to let the new statements become effective. Start Performance Toolkit again. The message FCXAPP530I Connected to *IDENT for resource FCXRES00 should be written to the basic mode screen during startup. The installation has been successful if you see this message.

   If you receive an error message pertaining to resource FCXRES00 you will have to check all of the above steps again and correct the error. You can ignore error messages pertaining to resource FCXSYSTM; that resource is intended for use with remote systems only.

   If you receive no message at all, you must have forgotten to include the FC MONCOLL VMCF ON command.

Note that you will have to either restart Performance Toolkit after any additional changes to the FCONRMT AUTHORIZE file, or use the command FC RELOAD AUTHORIZ to activate them.

**Note:** Before restarting Performance Toolkit, first log off and then log back on the PERFSVM userid to ensure a clean environment.

The following picture gives an overview of this setup:

PERFSVM DIRECT:

IUCV *IDENT FCXRES00 GLOBAL
IUCV  ALLOW

```
USER1
```

```
USER2
```

```
USERn
```

```
PERFKIT
Data
Collector
```

FCONRMT AUTHORIZ:

| *Node-ID | User-ID | Authorized |
|----------|---------|------------|
| *        |         |            |
| locnode  | USER1   | DATA       |
| locnode  | USER2   | DATA       |
| . . .    | . . .   | . . .      |
| locnode  | USERn   | DATA       |
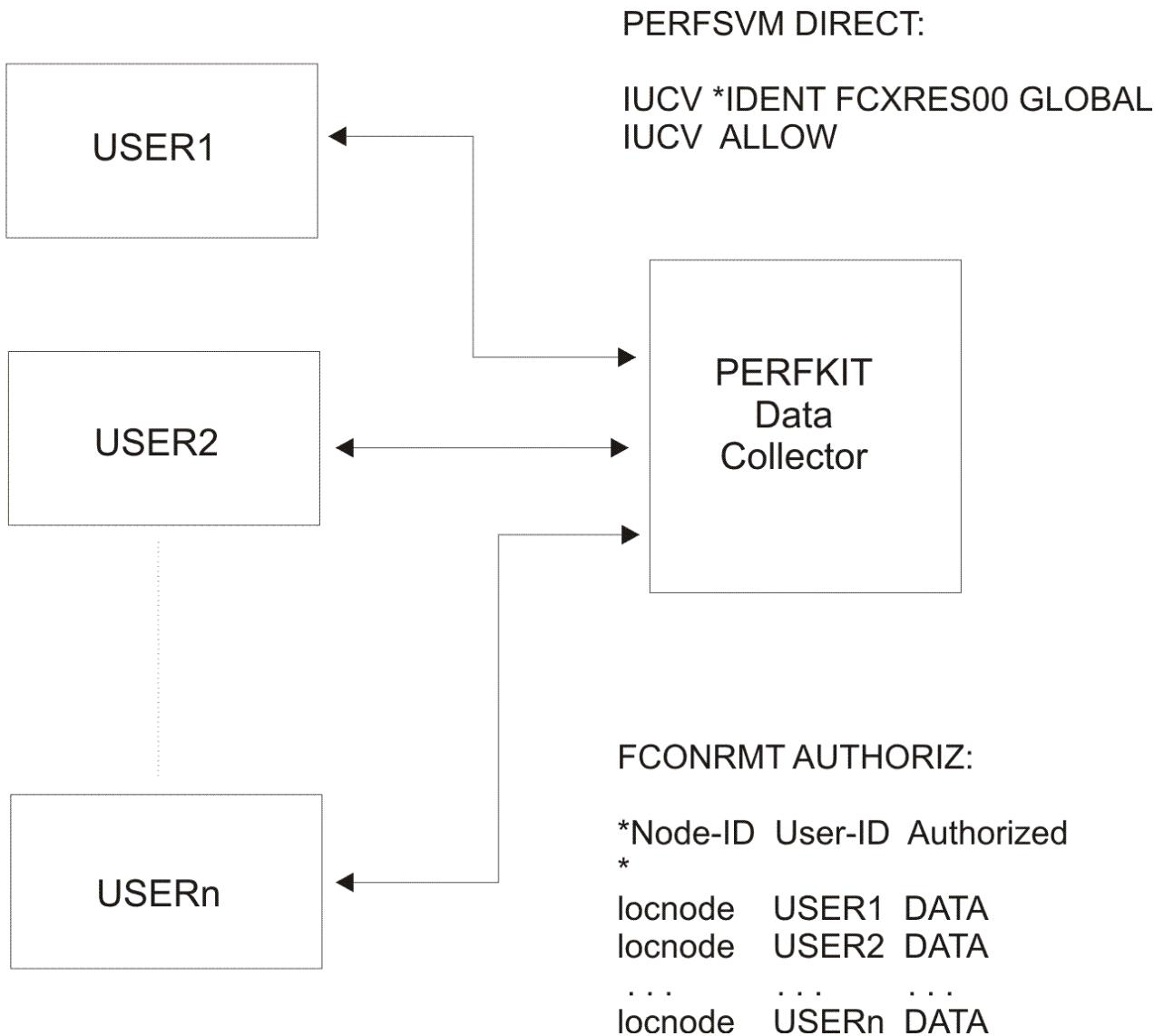
*Figure 9. Local APPC/VM Setup Using Default Resource Name FCXRES00*

No changes are required in the virtual machines of any of the users which want to work with the APPC/VM interface: Entering the command

```
FCONAPPC
```

in basic mode without further arguments will automatically establish the connection to the default global resource FCXRES00.

## Using Another Resource Name

Because the APPC/VM implementation for Performance Toolkit is based on the use of global resources, different resource names must be used on each VM system if you intend to establish APPC/VM connections for performance data retrieval with Performance Toolkit between two or more VM systems.

This section deals with the definitions required to allow Performance Toolkit performance data retrieval for users on the same system where the data collector machine is located. See the next section "Implementing the APPC/VM Interface for Remote Systems" on page 37 for instructions on how to prepare for remote connections from other VM systems.

When using a resource name other than the default one known to Performance Toolkit, you also have to set up a communications directory entry which lets the default name of FCXRES00 be translated to

whatever actual resource name you've chosen for the system. So the required implementation steps are now:

1. Select a resource name for the system. The name must be unique in all the connected systems.

   A resource name of FCXRES*xx* is recommended, where any combination of characters or digits can be inserted for *xx*, and such a scheme is assumed in the remaining samples in this section. You can also change the first six characters of the resource name, of course, but using FCXRES will help you to remember this is a Performance Toolkit resource.

2. Add the following statements to the directory entry of the userid used for data collection (usually PERFSVM):

   ```
   IUCV *IDENT FCXRESxx GLOBAL
   IUCV ALLOW
   ```

   where FCXRES*xx* is the resource name selected in the previous step.

   The first of these statements authorizes Performance Toolkit to make itself known to the *IDENT service as manager of the global resource FCXRESxx, and the second statement authorizes any user to establish an APPC/VM connection to the resource(s) managed by the data collector machine.

3. Let the APPC/VM interface be activated automatically during startup of Performance Toolkit by including the statement

   ```
   FC MONCOLL VMCF ON
   ```

   in the FCONX $PROFILE of the data collector machine if you have not already done so before (activation of the VMCF and APPC/VM interfaces is controlled by the same command).

4. Create an FCONRMT AUTHORIZ file with entries for all users which are to be authorized for performance data retrieval and place it on the PERFSVM machine's A-disk.

5. Add the following entry to a communications directory file which is accessible by all users who are to use the APPC/VM connection to Performance Toolkit:

   ```
   :nick.FCXRES00 :luname.*IDENT
                  :tpn.FCXRESxx
                  :security.SAME
   ```

   Use the common SCOMDIR NAMES file if a majority of users on the system will be using this as a global file for all users; otherwise add the entry to the individual private communications directory files in the virtual machine(s) which will use the interface.

   Make sure the relevant communications directory file is automatically activated after IPL of z/CMS by the appropriate SET COMDIR command in one of the automatically run procedures (for example, SYSPROF EXEC or PROFILE EXEC).

   The following figure shows again what has to be specified, and where:

PERFSVM DIRECT:

IUCV *IDENT FCXRESxx GLOBAL
IUCV  ALLOW

FCONRMT AUTHORIZ:

*Node-ID  User-ID  Authorized
*
locnode    USER1  DATA
locnode    USER2  DATA

locnode    USERn  DATA

XCOMDIR NAMES:

:nick.FCXRES00  :luname.*IDENT
                :tpn.FCXRESxx
                :security.SAME

XCOMDIR NAMES:

:nick.FCXRES00  :luname.*IDENT
                :tpn.FCXRESxx
                :security.SAME
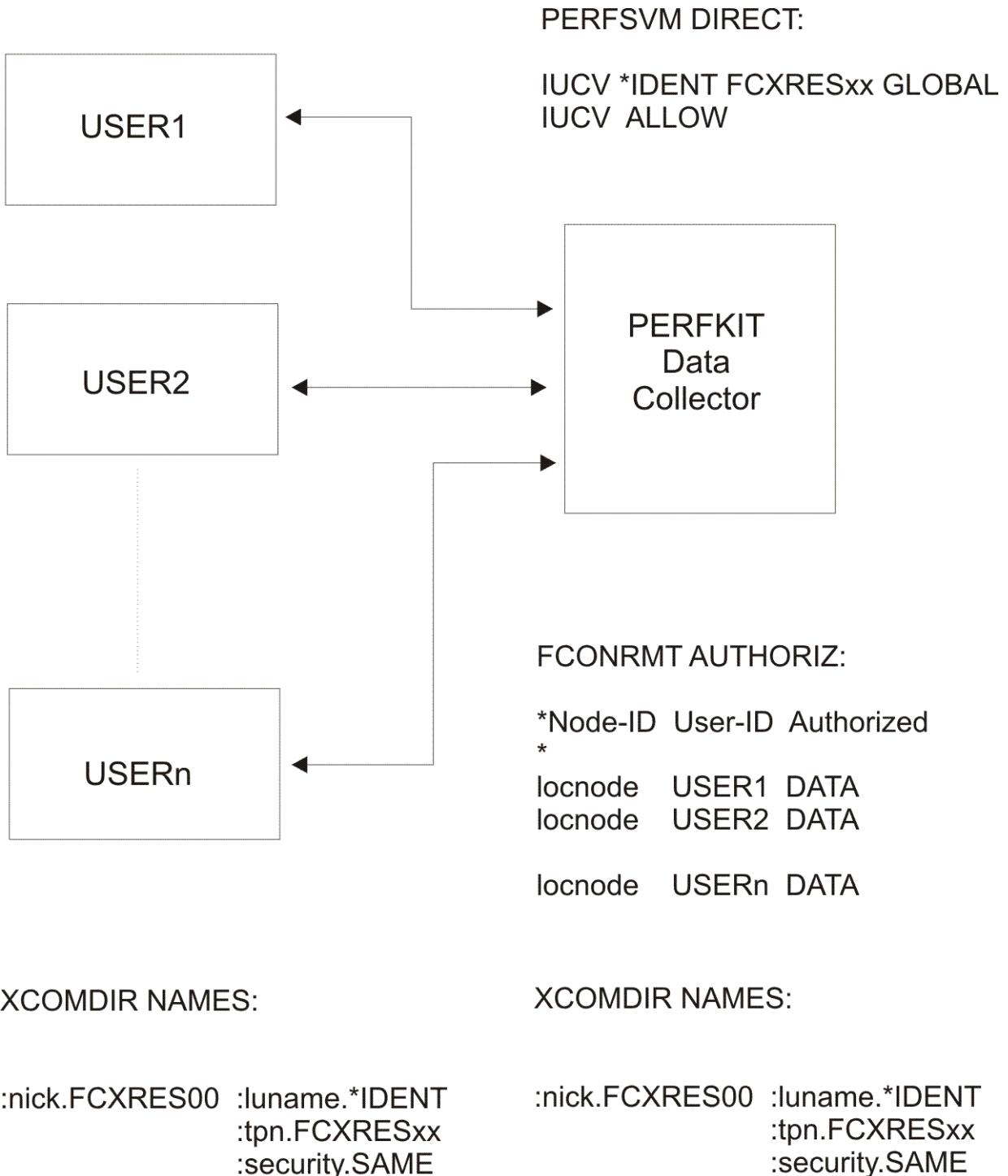
*Figure 10. Local APPC/VM Setup Using a Special Global Resource Name*

Note that the two communications directory entries shown above, as required by the Performance Toolkit data collector machine (the resource manager) and the users, are identical. You have to make two entries only if they are to be included in private communications directory files in each machine, otherwise one entry in a common communications directory (for example, SCOMDIR NAMES) will serve the same purpose.

6. Activate the changed directory and logoff/logon the data collector machine to let the new statements become effective. Then start Performance Toolkit again: The message FCXAPP530I `Connected to` `*IDENT for resource FCXRESxx` should be written to the basic mode screen during startup, where FCXRES*xx* is the global resource name you defined in the previous steps. The installation has been successful if you see this message.

If you receive an error message pertaining to the selected resource FCXRESxx you will have to check all of the above steps again and correct the error. You can ignore error messages pertaining to resource FCXSYSTM; that resource is intended for use with remote systems only.

If you receive no message at all, you must have forgotten to include the FC MONCOLL VMCF ON command.

Because the new communications directory entry will translate connect requests to the default resource FCXRES00 to a request for connecting to resource FCXRES*xx*, users will automatically be connected to the new global resource FCXRES*xx* on this system which is managed by our local PERFKIT machine when they enter the command FCONAPPC without further arguments.

As a last check, test the connection to the PERFKIT data collector machine from the virtual machine of one of the authorized users:

- IPL ZCMS to let the changed communications directory be activated.
- Start Performance Toolkit
- Enter the command FCONAPPC without further argument in basic mode. You should briefly see the message FCXCMD521I Connecting to server and then the initial performance data selection menu.

The installation has been successful if you see this message. An error message indicates a problem somewhere. If the connection to the same resource worked correctly when tested in step 6 above, the probable cause is either a problem with the communications directory or a missing IUCV ALLOW statement in the collector machine's directory entry.

## Implementing the APPC/VM Interface for Remote Systems

The establishment of APPC/VM connections over the SNA network requires that AVS machines be implemented, and global gateways defined, on all the affected VM systems. We will assume here that this has already been done and will concentrate only on the Performance Toolkit specific items; please refer to your VM system reference library for information on how to set up such AVS connections if they do not already exist.

### General Concept

While any 'local' APPC/VM application can be accessed remotely too, once the appropriate setup has been prepared in communication directories and AVS machines, the administrative work needed for each additional user-server connection is relatively complex, and it can take quite some time, especially for less experienced users.

To simplify this setup, and to avoid repetitive changes in AVS machines and communication directories, a 'Store and Forward' (S&F) concept has been implemented in Performance Toolkit, as illustrated by the following figure.
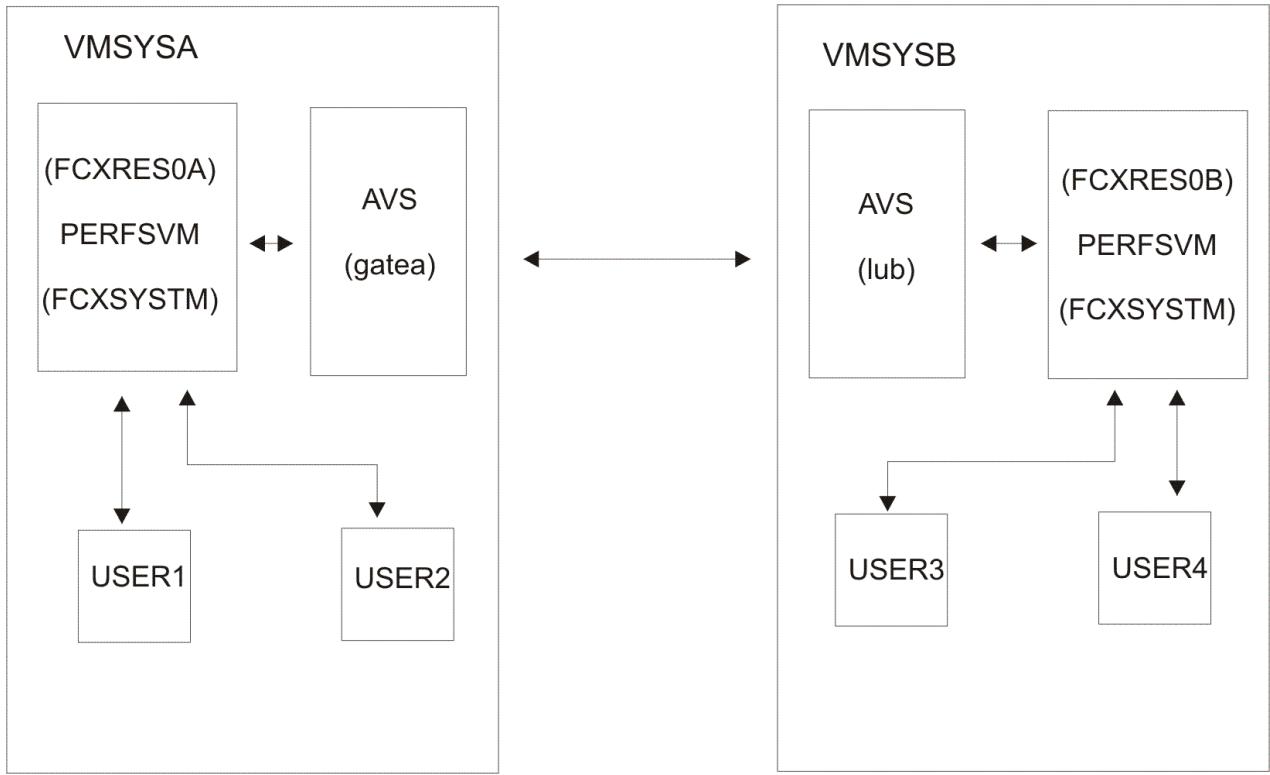
*Figure 11. APPC/VM Setup for 'Store & Forward' Logic of Remote Data Retrieval*

The setup for direct remote connections from user to remote server as they were used in the PERFKIT predecessor program is no longer described, but still supported. Where still needed, you can refer to the UCOMDIR NAMES entries and AVS `'AGW ADD USERID ...'` commands described for setting up the S&F server machine and use the same logic for any other user, too. Be aware that the user will then appear as a *local* userid to the server machine; that is, the relevant entries in file FCONRMT AUTHORIZ will have to be made with the RSCS node name of the *target* system.

**Operation:** In the Store and Forward concept:

- Remote connections to other VM systems exist only between PERFSVM server machines, that is, only one set of AVS specifications and communication directory entries has to be made per inter-system connection, to allow the two PERFSVM machines to communicate with each other. The end users do **not** have to know about this part of the setup.

- Users have direct connections only to the FCXSYSTM resource of their local PERFSVM machine.
  So if USER1 on VMSYSA wants a data retrieval session with VMSYSB, he/she will first connect to resource FCXSYSTM on the local system by entering the command FCONAPPC FCXSYSTM. (If users want FCXSYSTM to be selected by default, they will have to adapt their communications directory to translate the automatically generated request for connection to FCXRES00 to the new resource name FCXSYSTM.)

  The APPC/VM interface managing resource FCXSYSTM will then present a menu with a list of all systems to which remote connections are possible, based on the information contained in the local PERFSVM machine's FCONRMT SYSTEMS file.

- By selecting VMSYSB from the menu, a special 'Store & Forward' connect request to the appropriate remote global resource (in our example FCXRES0B) is generated by the PERFSVM machine at VMSYSA.

- The PERFSVM machine on VMSYSB first validates this request and checks whether user PERFSVM has been authorized to act as an S&F server by looking for an S&FSERV specification for user PERFSVM in the FCONRMT AUTHORIZ file on VMSYSB. (Due to an `'AGW ADD USERID gatea PERFKIT PERFKIT'` specification in the AVS machine on VMSYSB, the request is presented to the target PERFSVM machine like a request from local user PERFSVM on VMSYSB.)

It also checks whether USER1 on VMSYSA has been authorized for performance data retrieval, and, if so, generates the initial performance data menu for VMSYSB.

- Data buffers are passed back the same way. First from the PERFSVM machine on the target system VMSYSB then back to the PERFSVM machine on VMSYSA where this machine again acts as an S&F server and passes the data on to the original requestor, that is, USER1 in our example.
- User USER1 can now go on working just as in a direct APPC/VM connection. The store-and-forward logic is completely transparent to the end user.

Any PERFSVM data collection machine can also act as an S&F server, provided that:

- An `IUCV *IDENT FCXSYSTM GLOBAL` statement has been added to its directory
- An FCONRMT SYSTEMS file with the required remote resource names has been placed on its A-disk, and the necessary definitions added to its UCOMDIR NAMES file
- Suitable `'AGW ADD USERID ...'` statements in the AVS machines of all target VM systems allow it to connect to the remote systems' APPC/VM global resources
- It has been given S&FSERV authorization in all of its remote partners' FCONRMT AUTHORIZ files

So you will probably want to define such server machines on all systems where the performance people are likely to be logged on anyway. These servers could then each allow connections to all other systems, or to a subset only, as defined by the FCONRMT SYSTEMS and FCONRMT AUTHORIZ files.

## Implementation

The steps to be performed are:

1. **Prepare PERFSVM collector machines** on all systems, as described in "Using Another Resource Name" on page 34, in steps 1, 2, and 3. Also, include the additional statement `IUCV *IDENT FCXSYSTM GLOBAL` in the directories of PERFSVM machines you want to set up as S&F servers on the systems from where users will want to initiate cross-system data retrieval requests.

2. **Prepare communications directories for the PERFSVM machines:**

   *Entries required for S&F server machines:* These also need entries for all the remote systems you want to reach from the local one. The following example describes the entries that would be required for the PERFSVM machine on VMSYSA of the preceding figure:

   ```
   :nick.FCXRES00 :luname.*IDENT
                  :tpn.FCXRES0A
                  :security.SAME
   :nick.VMSYSB   :luname.gatea lub
                  :tpn.FCXRES0B
                  :security.SAME
                  :modename.#INTER
   ```

   Because only the PERFSVM machine itself has to know about these specifications, you will probably want to include these entries in the PERFSVM machine's private UCOMDIR NAMES file.

   We will discuss here briefly the specifications made for PERFKIT on system VMSYSA in the above example:

   - **:nick.FCXRES00**: Because we do not use the default global resource name of FCXRES00 on VMSYSA, we must make sure that it gets translated to FCXRES0A when Performance Toolkit tries to connect to its global resource.
   - **:nick.VMSYSB**: VM system VMSYSB can only be reached using an SNA link, through AVS machines. The entry looks different, therefore, from the one for our local connection:
     **:luname.**
       Instead of passing directly to the *IDENT service, the connection must now be established through our AVS machines, and `gatea` is the global gateway name and `lub` the name of the remote LU (AVS machine on VMSYSB).
     **:tpn.**
       The transaction program name is the global resource name FCXRES0B used on system VMSYSB.

**:modename.**
> Because we need an SNA session for our connection, we must specify the logon mode name to be used. Check what mode name has been specified on your system for interactive sessions to the target system and use this name. In our example the logon mode name #INTER was used.

Entries for additional remote systems would look similar to the one for VMSYSB, but use different nicknames, remote LU names, and transaction program names.

*Sample entry for end users* ('client' machines):

Although not mandatory (the resource name FCXSYSTM can be specified as an argument with the FCONAPPC command), including the following entry in the end users' UCOMDIR NAMES files will help avoid some typing and make life easier.

```
:nick.FCXRES00 :luname.*IDENT
               :tpn.FCXSYSTM
               :security.SAME
```

It translates the default resource name FCXRES00 to the name of the resource for the S&F interface.

Similar communications directory entries will also have to be made on system VMSYSB and any other systems from where connect requests are to be issued for remote resources.

Make sure the relevant communications directory files are automatically activated after IPL of z/CMS by the appropriate SET COMDIR command in one of the automatically run startup procedures (for example, SYSPROF EXEC or PROFILE EXEC).

3. **Prepare FCONRMT SYSTEMS Files:** The initial system menu a user sees is built based on the contents of the local S&F server's FCONRMT SYSTEMS file: All system entries which contain an APPC/VM resource (nick)name are assumed to be for systems that can be reached using the S&F interface. The FCONRMT SYSTEMS file used by the PERFSVM machine on VMSYSA might look as follows.
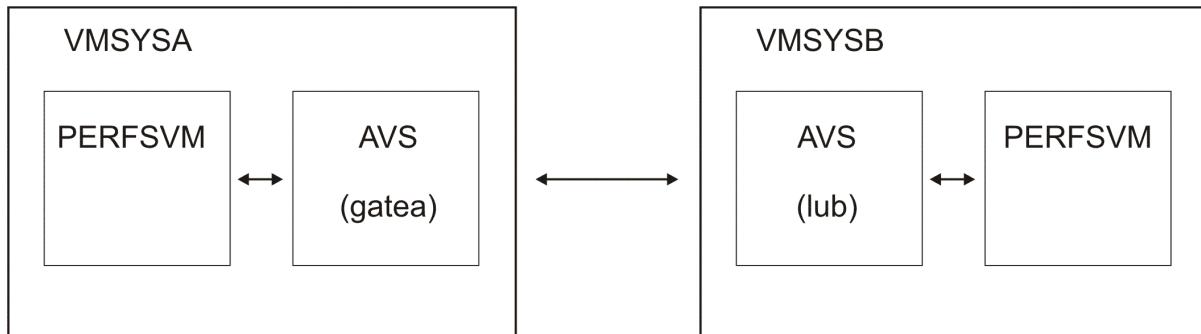
```
VMSYSA    PERFSVM    ESA   N    FCXRES0A
VMSYSB    PERFSVM    ESA   N    VMSYSB
```

Note that we just included the relevant APPC/VM resource names as defined in our UCOMDIR NAMES file above. Note that the *nicknames* must be used for all remote connections; you can specify the actual resource name only for the connection to the global resource on the local system. (see "Implementing Remote Performance" on page 10 for a detailed description of FCONRMT SYSTEMS entries). You will have noted that this extends the use of the FCONRMT SYSTEMS file: Originally intended for use by central performance data collector machines, it is now required for all S&F server machines too.

The file is still *not* required, on the other hand, for any PERFSVM machines that are neither central collector machines nor S&F servers.

4. **Prepare security definitions:** The APPC/VM implementation of Performance Toolkit has been based on SECURITY=SAME because of the administrative problems involved with SECURITY=PGM when the adopted security policy asks for frequent password changes. However, SECURITY=SAME also requires some administrative preparations for each server connection and for each user. These are described in the following paragraphs.

**Prepare AVS definitions:** We are not dealing here with the general definitions required for setting up global gateways and sessions (which we assume to exist already), but with the additional specifications required for supporting SECURITY=SAME.

```
                        Definitions required in VMSYSB:
                        AVS:        AGW ADD USERID GATEA PERFSVM PERFSVM
                        PERFSVM:   Entry in FCONRMT AUTHORIZ
                                    *Node-ID  User-ID  Authorized for . . .
                                    *
                                    VMSYSB  PERFSVM  S&FSERV


                        . . .         . . .         . . .
```

*Figure 12. APPC/VM Setup for Remote Performance Data Retrieval: Security Definitions*

An AGW ADD USERID command has to be included to be run by the local AVS machine for each remote user who is to be allowed connection with SECURITY=SAME to a resource managed on the local system, as follows:

```
  AGW ADD USERID remotelu remoteid localid
```

This is a mapping command which will translate the combination of 'remotelu remoteid' (the LU of the remote AVS machine and the user's ID on the remote VM system) to 'localid', the user identification which will be passed to Performance Toolkit for validation. One such statement is required for each remote PERFSVM machine that is to be allowed to use the AVS connection to the local system with SECURITY=SAME.

**Directory definitions and external security manager (ESM) definitions:** Where possible, let the 'localid' in the 'AGW ADD USERID ...' statement refer to a userid owned by the same user as the remote user-ID. If this can be done, no further directory and ESM definitions are necessary. In the above example we have assumed that all PERFSVM machines are controlled by the same group and therefore, we have used the PERFSVM userid again.

Otherwise, even though not absolutely necessary from a technical point of view, a dummy user-ID should be defined in the CP directory (and also in the external security manager program, if installed) in order to prevent a security exposure: After accidental definition of a virtual machine with the same name, but owned by a different user, such a user could take advantage of an entry in the local Performance Toolkit machine's FCONRMT AUTHORIZ file which was not originally meant for him.

**FCONRMT AUTHORIZ:** Two different kinds of entries have to be made:

- **For S&F Servers:** These require a special S&FSERV authorization, and they are treated as local users, with IDs generated by the AVS machine according to the `'AGW ADD USERID ...'` statements made.
- **Normal Users:** Remote APPC/VM connections made using S&F server will be presented to Performance Toolkit in a format that allows testing both the remote system identification and the requestor's userid. The entries made for remote APPC/VM users will, therefore, have to specify both the remote system identification and the user identification of the user to be authorized.

This is working now just as described for the original data retrieval solution using RSCS links in "Implementing Remote Performance" on page 10, *but it is different from the way remote data retrieval using APPC/VM worked with the PERFKIT predecessor program* where remotely connected users were always treated like local ones.

The following example illustrates the definitions required in the PERFSVM machine on VMSYSB to allow PERFSVM on VMSYSA APPC/VM connection with SECURITY=SAME and operation as a Store & Forward server for users on VMSYSA, and to authorize users on both systems for data retrieval.

```
*Node-ID User-ID   Authorized for ...
*|       |         |
VMSYSB   PERFSVM   S&FSERV
VMSYSA   USER1     DATA
VMSYSA   USER2     DATA CMD
VMSYSB   USER3     DATA
VMSYSB   USER4     DATA
```

Note that the S&F server can only pass on the RSCS node ID of its own system if it has access to a valid SYSTEM NETID file. Otherwise it will use the actual system identification (as specified in the system configuration file). The examples shown explain only what is required to allow the PERFSVM machine on VMSYSA working as an S&F server with VMSYSB. To fully implement the example layout shown in Figure 11 on page 38, equivalent definitions would have to be made to also allow the PERFSVM machine on VMSYSB to act as an S&F server.

5. **Activate the changed directory** and logoff/logon the data collector machine to let the new statements become effective. Then start PERFKIT again. The message FCXAPP530I Connected to *IDENT for resource FCXRESxx should be written to the basic mode screen during startup, where FCXRES*xx* is the global resource name you defined in step one.

   For S&F server machines only, you should also see the message FCXAPP530I Connected to *IDENT for resource FCXSYSTM at the same time. The installation has been successful if you see the expected message(s).

   If you receive an error message for resource FCXSYSTM in a PERFSVM machine that is *not* intended to run as S&F server, you can ignore it (Performance Toolkit will automatically try to connect to the FCXSYSTM resource on all systems where it finds a valid FCONRMT SYSTEMS file; it may thus generate the error message even if you did not want it to activate the S&F interface).

   For all other error messages you will have to check the relevant steps again and correct the error.

   If you receive no message at all, you must have forgotten to include the FC MONCOLL VMCF ON command.

6. **Test the connection** to each PERFSVM data collector machine using all defined S&F servers:

   • LOGON to a userid that has been authorized for data retrieval.

   • Start Performance Toolkit

   • Enter the command FCONAPPC FCXSYSTM (you can omit the FCXSYSTM argument if you have set up a communications directory entry for translating the default resource name FCXRES00 to FCXSYSTM, as suggested). You should briefly see the message

   ```
   FCXCMD521I Connecting to server
   ```

   and then the initial system selection menu.

   If you got this far, the local APPC/VM connection to the S&F server works as expected.

   • Select the first system on the list. You should see the initial performance data selection menu, and the status indication in the top right hand corner should tell you that you are now in a session with the selected system.

   If you got this far, the S&F connection to the selected system works as expected. Try selecting some other performance displays to make sure.

   Use the return key until you are back on the system selection menu and check the connections to all other systems too.

The installation has been successful as far as the current local system is concerned if you can connect to all the other systems on the system selection menu.

An error message indicates a problem somewhere in the communication chain, and you will have to analyze:

•

• Error messages returned by Performance Toolkit in your own 'client' machine

• Error messages generated in the local S&F server machine and in the remote target PERFSVM machine

• Messages generated by either of the involved AVS machines.

# Using the APPC/VM Interface for Performance Data Retrieval

## Connecting Directly to Target Server Machine

You will always connect directly to the target PERFSVM machine if you have not defined an intermediate S&F server for remote connections. Activate the APPC/VM interface by keying in the command FCONAPPC <sysname>> in basic mode or from the FCONRMT system overview menu, where *sysname* refers to an optional nickname in a communications directory entry which describes the target resource you want to connect to: The local or remote Performance Toolkit performance data collector machine which acts as manager of a specific global resource.

For ease of reference it is recommended that you use the VM systems' RSCS node-IDs or their system-IDs as nicknames when implementing the Performance Toolkit APPC/VM interface, and let the FCONAPPC command default to a connection to the PERFKIT machine on the local system when entered without an argument, but you will have to ask your systems administrator for the actual nicknames to be specified in your case (the easily memorized RSCS node-ID may already have been in use for defining some other application).

You can also customize your FCONRMT environment to let an APPC/VM connection be established automatically when selecting a system on the FCONRMT systems overview menu with the cursor by appending the matching communications directory entry's nickname to the system's entry in the FCONRMT SYSTEMS file. (See for instructions on how to create an FCONRMT SYSTEMS entry.)

After entering the FCONAPPC command (or selecting a system from a suitably prepared FCONRMT entry) you should briefly see the message FCXCMD521I Connecting to server in the message line, and you should then enter APPC/VM mode and be presented the initial performance data selection menu, or whatever other performance display you specified with the command. The RSCS node-ID of the system you connected to will be shown in the top righthand corner, similar to FCONRMT mode.

Failure to see the above, or an error message shown in the message line, indicates a problem somewhere in the communication chain. Contact your systems administrator for assistance.

Otherwise just proceed as usual. You can do the following as in a native monitor mode session:

• Enter commands

• Use cursor selection for obtaining more detailed performance data

• Use the cursor for sorting, or for context help selection

• Create graphics using the GRAPH*xxx* command, the GRAPHICS menu, or the USTATG command

## Connecting to an S&F Server for Remote Connections

Activate the APPC/VM interface by keying in the command

```
FCONAPPC FCXSYSTM <system>>
```

in basic mode or from the FCONRMT system overview menu, where *system* is the optional system name of the remote system you wish to connect to. When entered without the additional *system* argument, you should be presented with a system selection menu which can take two forms:

- SYSMENU, a simple system selection menu that shows system identifications and the matching APPC/VM resoure names
- SYSLOAD, a system load overview with CPU load bars and exception indications, similar to the one shown when accessing the centrally collected data with the FCONRMT command (see "Central Monitoring Function" on page 14).

The SYSLOAD display can be shown only when the S&F server machine also acts as central collector machine. Where available, it will be initially shown provided that all of the monitored systems fit onto the available number of screen output lines; the SYSMENU display is shown otherwise. You can also switch manually between the two displays by using the selection commands SYSMENU and SYSLOAD.

The following figure is an example for the SYSMENU display:

```
 FCX193        CPU nnnn   SER nnnnn     (Intermediate Server)           VMSYSA

                        Systems Accessible using APPC/VM

       System     Resource          System     Resource          System     Resource
       VMSYSA     FCXRES0A          VMSYSB     VMSYSB            TESTVM1    RESRC1
       TESTVM2    RESRC2            TESTVM3    RESRC3            TESTVM4    RESRC4
       TESTVM5    RESRC5            TESTVM6    RESRC6            TESTVM7    RESRC7
       TESTVM8    RESRC8            TESTVM9    RESRC9            TESTVM10   RESRC10









 Select the system to be monitored
 Command ===> _
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 13. Layout of System Selection Menu*

The field descriptions are the following:

**System**
 The system identification as defined in the FCONRMT SYSTEMS file. This will usually be either the actual VM system identification, as defined in the CONFIG file or the RSCS node name.

**Resource**
 The APPC/VM resource name of the system. It can be either the actual resource name on the local system or the nickname used to look up the resource in a communications directory file.

Select one of the systems in the menu and press ENTER to establish an APPC/VM session to the PERFSVM machine on that system. The initial performance data selection menu for the selected system should be returned, and the identification of the remote system shown in the top righthand corner.

Failure to see the above, or an error message shown in the message line, indicates a problem somewhere in the communication chain. Contact your systems administrator for assistance.

Otherwise proceed as usual. You can enter all commands, use cursor selection for obtaining more detailed performance data, for sorting, or for context help selection, just as in a native monitor mode session.

## Auto-Refresh Mode

Unlike a local monitor session where performance data displays are automatically refreshed whenever new data have been collected, the initial operating mode in an APPC/VM data retrieval session is to obtain data from the server, and refresh the client's display, only after manually entering a new command.

Originally implemented in this way to avoid long periods where the screen cannot accept a new command because data retrieval is active on a slow connection, it is obviously desirable to allow activating such an auto-refresh mode for sessions with good response times. A command has been implemented to give the user the free choice. Enter the command FC AUTOREFR ON to activate auto-refresh mode for your client session.

The resulting automatic refresh mode has a 'client pull' method of operation, that is, it is not directly synchronized with the data collect interval of the server machine. However, it *does* use the server's data collect interval to set its own refresh request cycle and so provides a display mode that is very similar to the feeling of a native monitor mode session.

The string (A) will be appended to the system identifier field in the top right hand corner of remote performance data displays while auto-refresh mode is active.

# Internet Interface

## General Concept

The Internet interface is intended to:

- Provide a graphical user interface based on standard Web Browsers, thus automatically allowing its use from any of the common supported platforms
- Allow the use of the interface with a minimum of additional prerequisites

To avoid some of the problems associated with the current 'stateless' HTTP 1.0 protocol, and also the requirement to license, install, and administer a full Web Server, a special interface was built into Performance Toolkit. It processes only the subset of HTTP requests expected for a Performance Toolkit performance data retrieval session. Anything else will generate an error message to the client, that is, it is *not* intended to become yet another general Web Server for VM.
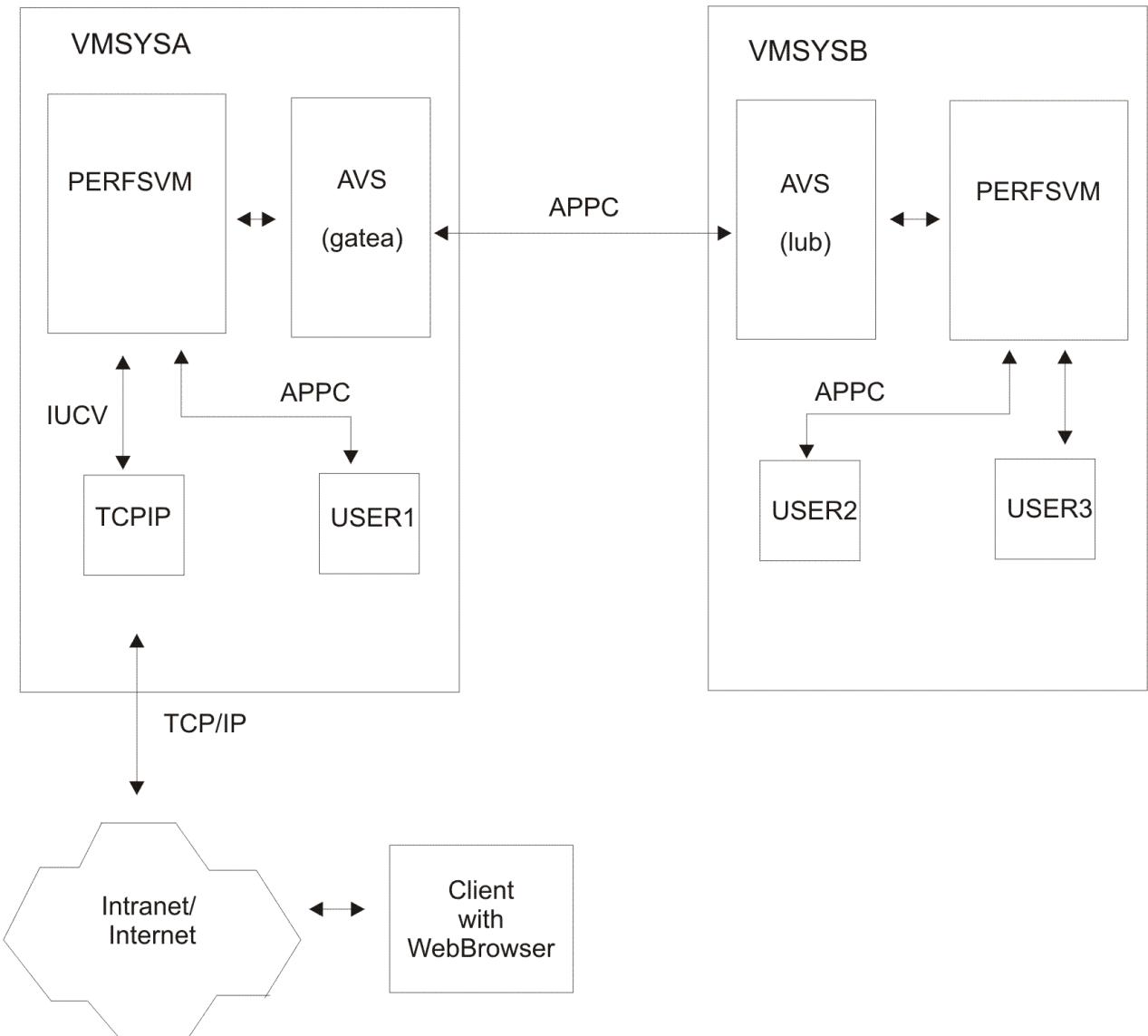
*Figure 14. Performance Toolkit Web Server Interface for Data Retrieval Using the Internet*

The interface has been built to work directly with the Store and Forward logic implemented for APPC/VM data retrieval sessions. Any PERFSVM machine set up as an S&F server can also be made to act as a Web Server for Performance Toolkit performance data, provided that it has access to a TCP/IP machine on the same system. Web clients can connect to such a Web Server machine and they will then receive output very similar to what a user would also see when connected using APPC/VM.

The following are restrictions:

- The standard HTTP/1.0 protocol is not session oriented. A connection exists only for the purpose of processing a single request. Once the request is processed, the connect is closed. The method used to allow simulating some sort of pseudo-session is the inclusion of additional fields in the URLs generated when selecting hyperlinks. This works well while selecting new data, but it would cause problems when trying to return to a previous display and selecting another link. Browsers are given an old expiration date, therefore, in order to prevent caching.

- While a userid/password authentication scheme has been implemented to provide some basic identification checking, you need to select secure (SSL based) operation in order to safeguard passwords and data across internet nodes outside your own area of administration. Only general performance data retrieval is allowed using this interface while general CP/CMS command running will be inhibited even if authorized by a corresponding FCONRMT AUTHORIZ file entry for a specific user. The only CP commands that may be allowed are QUERY and INDICATE commands; they can be

authorized using CPQRY authorization arguments, but will allow command running using a web browser only if the server machine itself has been authorized too by means of a special '::WWW:: CPQRY' statement. See "Implementing Remote Performance" on page 10 for more information on the required definitions.

- While a Java applet allows the creation of line graphics using GRAPHxxx commands, the SCATTER argument is not supported in this way, nor can user wait states be displayed graphically (USTATG command).

## Implementation

We assume here that the first prerequisite, a TCPIP server machine with access to your Intranet or to the Internet, is already available. If access is not available, refer to the relevant documentation for the implementation of such a TCP/IP interface machine.

1. **Implement S&F server:** The PERFSVM machine on the VM system with the TCP/IP server machine must be set up as an S&F server. Please refer to the description in section "Implementing the APPC/VM Interface for Remote Systems" on page 37 for instructions on how to do that. There are no additional considerations to worry about just because you intend to use the same machine as an Internet Web Server too.

2. **Implement IUCV connection from S&F server to TCPIP machine:**

    - Decide on the *TCP/IP port number* to be used. The default port number that Performance Toolkit will try to connect to for its interface is 81, the default is set in FCONX $PROFILE. In order to reserve this port for the PERFSVM machine's exclusive use and to prevent potential interference by other TCP/IP users, include the matching PORT definition in the PROFILE  TCPIP file of the TCPIP server machine:

    ```
    PORT
        ...
         81 TCP PERFSVM NOAUTOLOG; Performance Toolkit Internet Server
        ...
    ```

    **Note:** If the ASSORTEDPARMS statement FREELOWPORTS is enabled in PROFILE TCPIP, the above port statement is not necessary. However, it is not a problem if the statement is present.

    - Decide on the *use of SSL (TCP/IP Secure Socket Layer)*: use of SSL will make working with the Performance Toolkit Internet interface far more secure. It should be considered a MUST if you are concerned with Web clients using their normal VM userid/password combination at logon, but it can be activated only if a special prerequisite SSL server has been implemented for your TCP/IP stack. If SSL support is available on your TCP/IP, then the following additional actions are required to activate it for use with the Performance Toolkit Internet interface:

        – Tell the TCP/IP stack that requests received on the port dedicated to the Performance Toolkit (for example, Port 81) are to be handled as using the Secure Socket Layer protocol. This is done by appending the SECURE keyword to the PORT specifications and the certificate file name or label, as follows:

        ```
        PORT
           ...
        81 TCP PERFSVM NOAUTOLOG SECURE filename/label; Performance Toolkit
        Internet Server, using SSL
           ...
        ```

        – Tell the Performance Toolkit that any hyperlinks to itself that it passes back as part of a web page should be requesting a secure connection, by including the SSL keyword in the 'FC MONCOLL WEBSERV ON SSL ..' command for activating the interface. For an example of this, see Appendix A, "A Sample Initialization Profile (FCONX $PROFILE)," on page 173.

        – Inform your users to start their Web sessions by requesting a secure connection by using the **https://** prefix. (For more information see "Test the interface" in this section and "Using the Internet Interface for Data Retrieval" on page 49.)

Based on your decisions made above, set up the FC MONCOLL WEBSERV ON command required for activating the Internet interface and include it in the FCONX $PROFILE initialization file of the S&F server PERFSVM machine. You should now be able to use the interface.

3. **Implement user authentication:** Verify that the directory entry for the userid has OPTION DIAG88 specified to allow PERFKIT to verify the password entered from the web browser.

   Note that when your system is protected by an External Security Manager (ESM) such as RACF, you may need to provide additional authorization for PERFKIT to verify user passwords.

   Please consult your specific ESM's documentation for more information.

4. **Update the FCONX $PROFILE to automatically start the webserver:** Uncomment the FC MONCOLL WEBSERV statement for your security choice (SSL or not) and enter the TCP/IP port that you established in step 2. In order to control the resources used by the Performance Toolkit web interface, you can also change the default timeout and/or the limit of the active internet connections by using the FC MONCOLL WEBSERV TIMEOUT command or the FC MONCOLL WEBSERV MAXCONN command. For more information on using these commands, see *z/VM: Performance Toolkit Reference*.

5. **Activate the interface:** Restart the S&F PERFSVM machine and make sure the Internet interface is correctly brought up. You should see messages similar to the following during activation:

   ```
   FCXTCP571I Connected to TCP/IP server TCPIP on path 0004
   FCXTCP575I WebServer host IP address is 11.22.33.44:00081
   FCXTCP590I WebServer interface activated
   ```

   Failure to see the above messages, or any error messages, indicate a problem which you will have to analyze and fix.

6. **Test the interface:**

   a. Activate your preferred web browser and select the URL that describes the Performance Toolkit Web Server interface. For example:

      ```
              http://11.22.33.44:81
       or     https://11.22.33.44:81 (for SSL connections)
      ```

      You should see a Welcome page, followed after a few seconds by a Logon page. If you do not get through to the server, contact your network administration people. There is probably something in the TCP/IP setup that needs to be changed.

   b. On the Logon page, enter your userid/password combination, then click on the 'Submit' button. You should then be presented a web browser version of the system selection menu you also see when connecting to the S&F server using APPC.

   c. Select a system from the menu: You should receive the web browser version of the initial performance data selection menu from the selected system, and you can then go on selecting some other data from that system too.

   d. Select the 'Systems' button to return to the system selection menu and try some of the other systems, too.

   The installation has been successful if all of the above worked correctly. Otherwise, if any error messages were generated or if you could not connect to the Internet interface at all, you will have to check what could have gone wrong and fix the problem.

## Setting up a Customized Web Banner

If you wish to create your own initial "Web Banner" page that will be shown for five seconds after the Performance Toolkit introduction banner and before the logon page, follow these guidelines:

• Create an FCXBANNR HTML file, and optionally a FCXBANNR GIFBIN file, on any minidisk or SFS directory that can be accessed by the PERFKIT module. (A normal CMS search order will be applied when accessing these files.) The FCXBANNR HTML file must be an EBCDIC-coded page containing the customized screen. Here's a simple example:

```
<html><head>
<title>--- Web banner title ---</title>
</head><body>
<h1>Sample Web banner</h1>
<br>
<img alt="" src="FCXBANNR.GIFBIN">
<p>SAMPLE TEXT</p>
<br>
<%Button>
<br>
</body></html>
```

- The optional FCXBANNR GIFBIN is a binary file containing any bitmap image (GIF, JPG, PNG, etc.) that is supported by your Web browser. Note that the filetype must always be GIFBIN, regardless of its actual format. Both the FCXBANNR HTML file and the optional FCXBANNR GIFBIN image file may have any record format (fixed or variable) and any record length. Note, however, that the overall size of either of these two files must not exceed 11.5K.

- The FCXBANNR HTML file can be created in an XEDIT session, or with any HTML editor and then uploaded to the PERFKIT virtual machine as a text file. There are some limitations on the contents of this file, as follows:

  - The HTML header is generated automatically by PERFKIT, and all data outside of the <body> and </body> tags will be discarded, with the exception of the <title> contents.

  - The <title> contents length is limited to 80 characters.

  - For the first <img> tag found, the source file will always be assumed to be FCXBANNR GIFBIN. (So even if you code something other than that in the src= field, it will be replaced by FCXBANNR GIFBIN.) For any further <img> tags found, the source will be left unchanged.

  - There is one special "pseudo tag," <%Button> (as seen in the example above), that can be used anywhere within the HTML body. This tag will insert a 'Continue' button, which can be pressed to avoid waiting through the full five-second delay before entering the logon page.

- These files will be read and processed during execution of the FC MONCOLL WEBSERV ON command. If you need to modify either of them, make the changes and then restart the Web server interface (by issuing FC MONCOLL WEBSERV OFF and then FC MONCOLL WEBSERV ON) to use the new banner.

## Using the Internet Interface for Data Retrieval

- **Select the appropriate URL:** Activate your web browser and key in the URL for accessing the TCP/IP port assigned to the Performance Toolkit Web Server Interface, for example.

```
     http://11.22.33.44:81
 or  https://11.22.33.44:81 (for SSL connections)
```

Contact your systems programmer to obtain the correct IP address and port number if you do not know it. You should see a Welcome page, followed after a few seconds by a Logon page similar to the following:
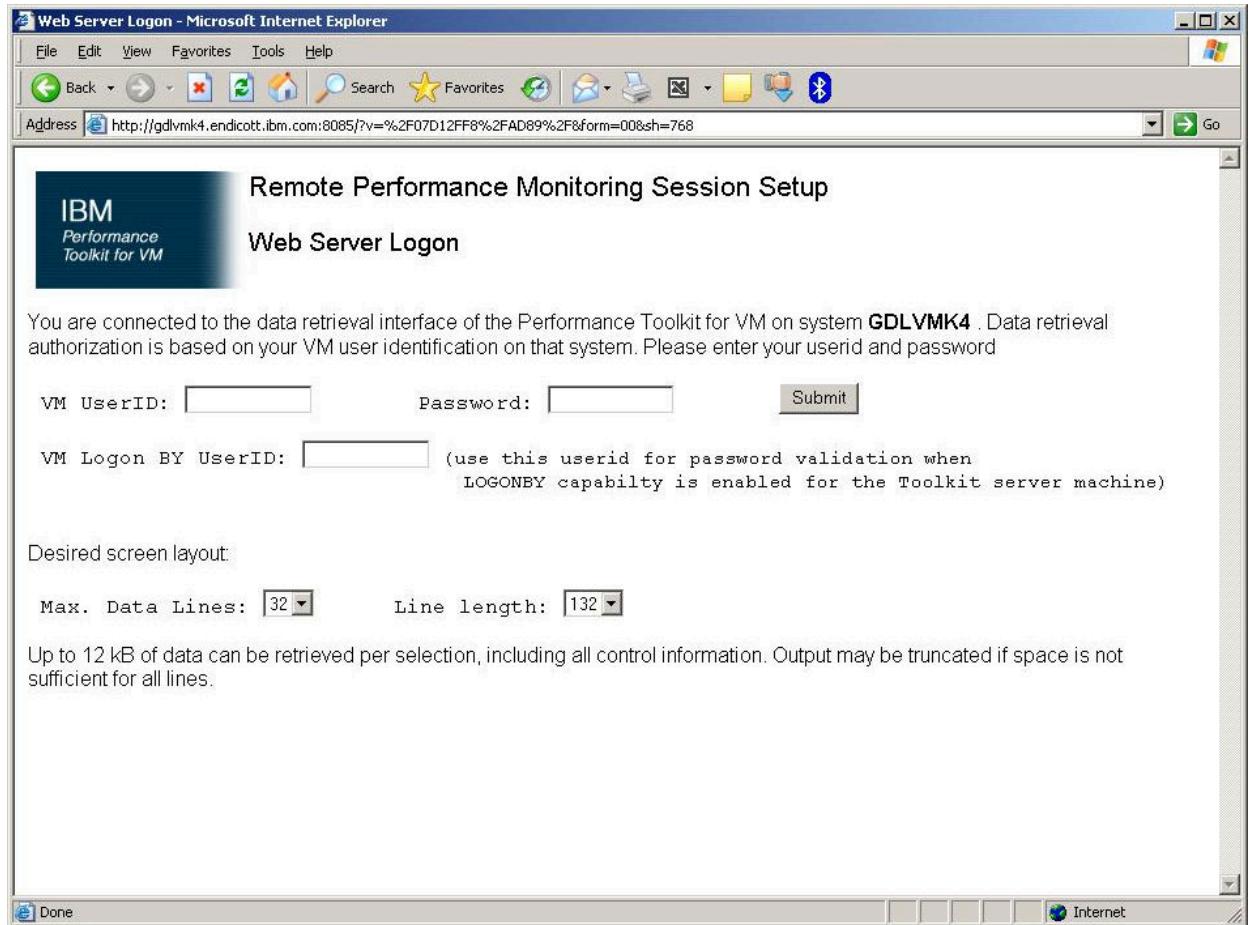
*Figure 15. Logon Page*

- **Identify yourself:** There are two choices for authorization: you can directly enter the user identification and password, or you can use the 'Logon by' method where the user identification, its BYUSER and a password must be entered.

  You can also set the format of the data returned. The initial default for normal browsers is to return up to 32 data lines with a maximum width of 132 characters. 32 data lines will usually fit into a PC window, allowing you to view all lines without touching the vertical scroll bar. Asking for more than 32 data lines probably means you will have to use the vertical scroll bar for viewing all the data. Please be aware that this will move the header lines too, and may leave you with data columns about whose meaning you are not quite sure.

  Mouse click on the 'Submit' button to transfer your specifications. You will then be presented a web browser version of the S&F server's system selection menu, described in detail in "Connecting to an S&F Server for Remote Connections" on page 43.

- **Select a system:** From the menu shown, select the system you want to monitor. You will receive a web browser version of the initial performance data selection menu for that system.

- **Select performance data:** You can go on selecting data by using mouse clicks on any of the defined hyperlink fields or on the push buttons shown.

  The following is an example for a performance data display in Web Browser format:
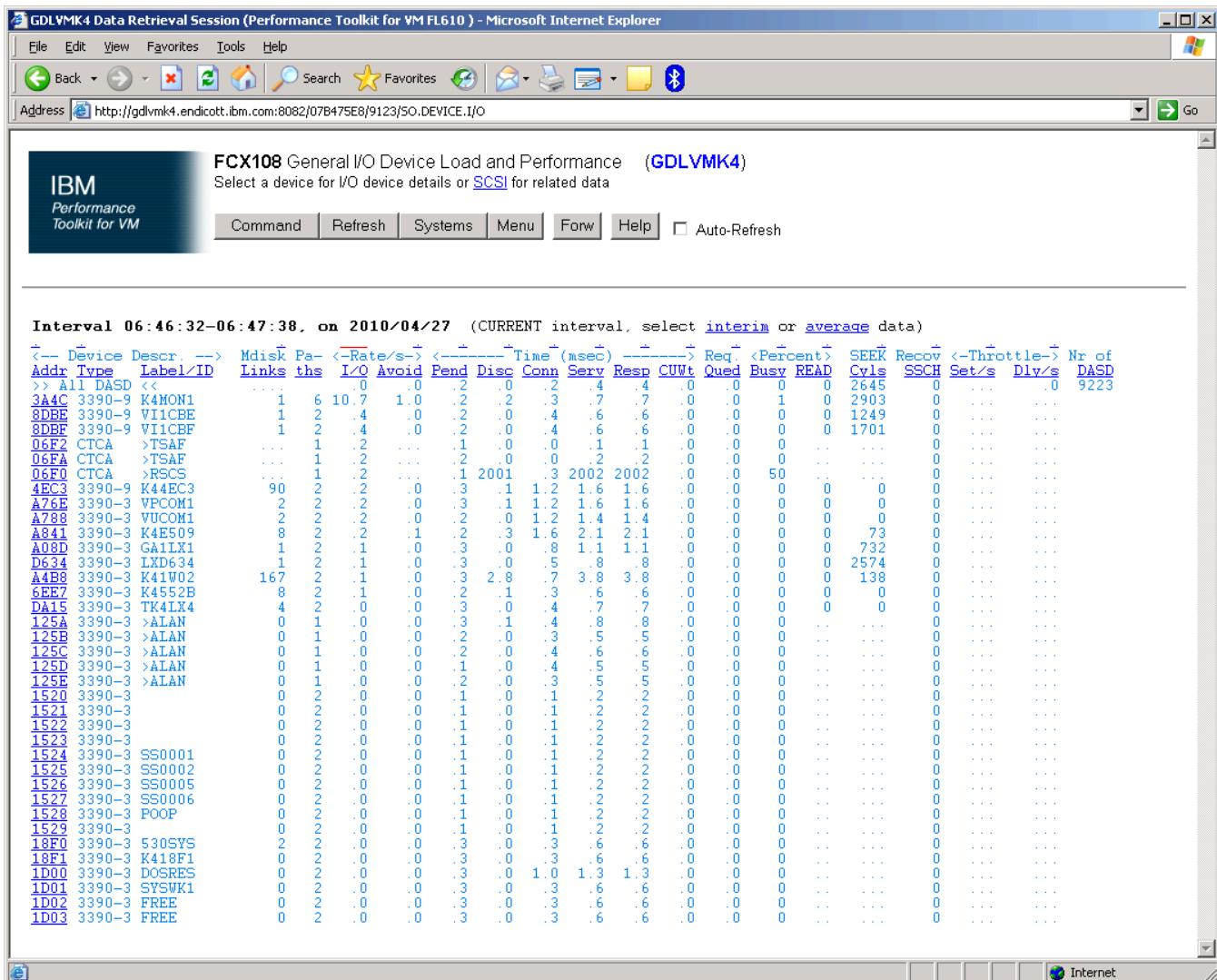
*Figure 16. Sample I/O Device Load Page*

Although the graphical appearance of the data might differ slightly from a 3270 based format, the layout of the actual performance data remains unchanged, so it should not be difficult to understand. You can:

– Select new performance displays from the initial menu

– Select more detailed user or I/O device data from the general user or I/O device displays

– Set different sort sequences by clicking on the sort indicators above the header lines

– Ask for context help text by clicking on a specific header field (the color of the field will tell you whether help text is available or not)

– Click on pushbuttons for scrolling to another part of the same data, for returning to the previous screen, the performance data selection menu, or all the way back to the system selection menu.

– Check the Auto-Refresh box to request that the screen be updated automatically at the rate that the monitor updates the data (with a minimum rate of 60 seconds – see note below). The screen must be refreshed using the refresh button before the Auto-Refresh will take effect. Note that in order to avoid unnecessary system load, the Performance Toolkit minimum refresh rate is 60 seconds, regardless if the monitor interval collection rate is less than this value.

– Log off from the Performance Toolkit Internet Interface by clicking the "Logoff" button. This will lead you to the Logon page and, in turn, release all resources used during your web session.

## Web Browser Setup/Dependencies:

**Fonts:** A size of 10 points for both the variable width and fixed width fonts should be set to obtain the intended layout for screens with low to medium resolution. Such fonts will be set automatically using <STYLE> tags (Helvetica or Arial for variable width font, and Courier as fixed width font) for browsers supporting that tag, such as Netscape Communicator V.4 or Internet Explorer V.3 and later. For other browsers (for example, Netscape Navigator V.2 or V.3) the font size must be adapted manually, where necessary. Larger fonts may have to be set on displays with a high resolution. Please note that you may have to deactivate the <style> support of the browser in order to activate your own default settings.

Note the following:

1. *Repetitive LOGONs:* If you have ended a data retrieval session by retrieving data from some other Web Server, you may have to deactivate and then re-initialize your web browser before attempting to sign on again to the Performance Toolkit interface. Failure to do so can result in your browser using a cached copy of a previous logon page; the logon attempt will then be aborted, and a new copy of the logon page returned.

2. You should use the navigation control buttons as provided on the Performance Toolkit WEB interface screens, not the Browser navigation buttons (that is, BACK and FORWARD) when making a selection from the Performance Toolkit WEB interface screens. This will help to ensure any performance and/or menu screen data is current and is obtained using a direct request from the Performance Toolkit.

## WWW Interface Recovery:

When activated, the interface is in a continuous IUCV session with the TCP/IP server machine and it will notice if this IUCV connection is severed due to a restart of the TCP/IP server, or if the TCP/IP server machine is logged off. It will then automatically attempt to restart the interface.

- Up to five initial restart attempts will be made in one minute intervals after detecting the severed connection.
- If these initial attempts all failed, the program will try to restart the interface in one hour intervals, on each full hour.

The interface cannot be restarted while the TCP/IP server machine is not active; it is the system programmer's responsibility to set up suitable recovery functions to keep the TCP/IP server up and running.

# Central Monitoring in a Single System Image Cluster

If you are running a single system image (SSI) cluster and you are using Performance Toolkit to monitor the performance of members of the cluster, it is recommended that you set up central monitoring of the Performance Toolkit instances.

This section provides an example of Performance Toolkit central monitoring in a SSI cluster.

In this example, the SSI cluster consists of four members: PTKSSIA1, PTKSSIA2, PTKSSIA3, and PTKSSIA4. The members are all connected via CTCA devices. The "connection scheme" is:

```
 PTKSSIA1 connects to PTKSSIA2 via X'912A'
                      PTKSSIA3 via X'913A'
                      PTKSSIA4 via X'914A'

 PTKSSIA2 connects to PTKSSIA1 via X'921A'
                      PTKSSIA3 via X'923A'
                      PTKSSIA4 via X'924A'

 PTKSSIA3 connects to PTKSSIA1 via X'931A'
                      PTKSSIA2 via X'932A'
                      PTKSSIA4 via X'934A'

 PTKSSIA4 connects to PTKSSIA1 via X'941A'
                      PTKSSIA2 via X'942A'
                      PTKSSIA3 via X'943A'
```

Each PERFSVM user ID on each PTKSSIA*n* member has its own IUCV global resource defined. These global resources are:

```
User ID    Member     Resource
-------    --------   -------------------
PERFSVM    PTKSSIA1   FCXC1R01 / FCXC1S01
PERFSVM    PTKSSIA2   FCXC1R02 / FCXC1S02
PERFSVM    PTKSSIA3   FCXC1R03 / FCXC1S03
PERFSVM    PTKSSIA4   FCXC1R04 / FCXC1S04
```

## CP Directory Entries for PERFSVM Machines in a Four-Member SSI Complex

In the CP directory entries shown in the following example, a unique pair of APPC resource names are created for each member's Performance Toolkit instance. These are assigned in the global IDENTITY statement so that the names are available across the SSI cluster.

The PERFSVM user IDs are defined as IDENTITY IDs, meaning there is a unique user ID on each member in the SSI cluster.

```
      IDENTITY PERFSVM  PASSWORD  512M  3G   ABDEG
      BUILD ON PTKSSIA1 USING SUBCONFIG PERFSV-1
      BUILD ON PTKSSIA2 USING SUBCONFIG PERFSV-2
      BUILD ON PTKSSIA3 USING SUBCONFIG PERFSV-3
      BUILD ON PTKSSIA4 USING SUBCONFIG PERFSV-4
      MACHINE ESA
      XAUTOLOG AUTOLOG1
      ACCOUNT xxxx
      NAMESAVE MONDCSS
      IUCV *MONITOR MSGLIMIT 255
      IUCV *IDENT FCXC1R01 GLOBAL
      IUCV *IDENT FCXC1S01 GLOBAL
      IUCV *IDENT FCXC1R02 GLOBAL
      IUCV *IDENT FCXC1S02 GLOBAL
      IUCV *IDENT FCXC1R03 GLOBAL
      IUCV *IDENT FCXC1S03 GLOBAL
      IUCV *IDENT FCXC1R04 GLOBAL
      IUCV *IDENT FCXC1S04 GLOBAL
      IUCV ALLOW
      SHARE ABS 3%
      IPL ZCMS PARM AUTOCR
      OPTION QUICKDSP DIAG88
      COMMAND DEFINE STORAGE CONFIG 0.512M 2G.2G
      CONSOLE 0009 3215
      SPOOL 000C 2540 READER *
      SPOOL 000D 2540 PUNCH A
      SPOOL 000E 1403 A
      *
      SUBCONFIG PERFSV-1
      LINK MAINT 190 190 RR
      LINK MAINT 19D 19D RR
      LINK MAINT 19E 19E RR
      LINK 7VMPTK10 200 200 RR
      LINK 7VMPTK10 29D 29D RR
      LINK 7VMPTK10 1CC 1CC RR
      MDISK 201 3390 600 20 A1E000 MR READ WRITE MULTI
      MDISK 1CC 3390 620  2 A1E000 MR READ WRITE MULTI
      MDISK 195 3390 622 60 A1E000 MR READ WRITE MULTI
      MDISK 191 3390 682 120 A1E000 MR READ WRITE MULTI
      *
      SUBCONFIG PERFSV-2
      LINK MAINT 190 190 RR
      LINK MAINT 19D 19D RR
      LINK MAINT 19E 19E RR
      LINK 7VMPTK10 200 200 RR
      LINK 7VMPTK10 29D 29D RR
      MDISK 201 3390 600 20 A2E000 MR READ WRITE MULTI
      MDISK 1CC 3390 620  2 A2E000 MR READ WRITE MULTI
      MDISK 195 3390 622 60 A2E000 MR READ WRITE MULTI
      MDISK 191 3390 682 120 A2E000 MR READ WRITE MULTI
      *
      SUBCONFIG PERFSV-3
      LINK MAINT 190 190 RR
      LINK MAINT 19D 19D RR
      LINK MAINT 19E 19E RR
      LINK 7VMPTK10 200 200 RR
      LINK 7VMPTK10 29D 29D RR
```

```
         MDISK 201 3390 600 20 A3E000 MR READ WRITE MULTI
         MDISK 1CC 3390 620  2 A3E000 MR READ WRITE MULTI
         MDISK 195 3390 622 60 A3E000 MR READ WRITE MULTI
         MDISK 191 3390 682 120 A3E000 MR READ WRITE MULTI
         *
         SUBCONFIG PERFSV-4
         LINK MAINT 190 190 RR
         LINK MAINT 19D 19D RR
         LINK MAINT 19E 19E RR
         LINK 7VMPTK10 200 200 RR
         LINK 7VMPTK10 29D 29D RR
         MDISK 201 3390 600 20 A4E000 MR READ WRITE MULTI
         MDISK 1CC 3390 620  2 A4E000 MR READ WRITE MULTI
         MDISK 195 3390 622 60 A4E000 MR READ WRITE MULTI
         MDISK 191 3390 682 120 A4E000 MR READ WRITE MULTI
```

# Customizing the PERFSVM User IDs

There is a small amount of set up work to perform on each PERFSVM ID on each member. Changes must be made to the FCONX $PROFILE, the FCONRMT AUTHORIZ, and the UCOMDIR NAMES files to complete the connections between the PERFSVM IDs. In general, you need to enable Performance Toolkit's VMCF interface, authorize the PERFSVM user IDs to issue commands and retrieve data, and establish the user-level communications directory for each PERFSVM instance.

For this example, the PERFSVM ID on member PTKSSIA1 is the central monitor. The following additional customization work must be done on the PTKSSIA1 member:

- You must tell Performance Toolkit to turn on its web server interface. You do this with the FC MONCOLL WEBSERV command in the FCONX $PROFILE on PERFSVM's 191 minidisk, as shown in .
- You must tell the TCP/IP virtual machine to define a port for to which Performance Toolkit can connect from a web browser. You do this by adding a port statement to the TCP/IP PROFILE used by your TCP/IP virtual machine.
- You must tell the PERFSVM user ID on member PTKSSIA1 to monitor the other members in the SSI cluster. You do this by adding the other PERFSVMs and members to the FCONRMT SYSTEMS file on PERFSVM's 191 minidisk. On the other three members, you must add a line in the FCONRMT AUTHORIZ file authorizing PERFSVM on PTKSSIA1 for DATA collection.

**Note:** This example assumes all web connectivity to Performance Toolkit is through only the PTKSSIA1 member. If you want web connectivity through any other member of the SSI cluster, you must enable, on that member, the three changes described previously.

The next several figures show the contents of these files on each of the PERFSVM IDs.

```
     PROFILE TCPIP  - added
          PORT
            8081 TCP PERFSVM NOAUTOLOG Performance Toolkit Internet Server


     FCONX $PROFILE - added
          *    Following command activates VMCF data retrieval interface
          FC MONCOLL VMCF ON
          *    Following command activates Internet interface
          FC MONCOLL WEBSERV ON TCPIP TCPIP 8081


     FCONRMT SYSTEMS
          *Node-id  Userid    VM-Type    Append    Nickname
          PTKSSIA1  PERFSVM   z/VM6.2    N         FCXC1R01
          PTKSSIA2  PERFSVM   z/VM6.2    N         FCXC1R02
          PTKSSIA3  PERFSVM   z/VM6.2    N         FCXC1R03
          PTKSSIA4  PERFSVM   z/VM6.2    N         FCXC1R04


     FCONRMT AUTHORIZ
          * Authorization file for local and remote
          * data retrieval and command execution
          *Node-ID  User-ID  Authorized for ..
          PTKSSIA1  PERFSVM  S&FSERV DATA

     UCOMDIR NAMES
          :nick.FCXRES00    :luname.*IDENT
                     :tpn.FCXC1R01
                     :security.SAME
          :nick.FCXSYSTM    :luname.*IDENT
                     :tpn.FCXC1S01
                     :security.SAME
```

*Figure 17. Performance Toolkit files for PERFSVM at PTKSSIA1*

```
     FCONX $PROFILE - added
          *    Following command activates VMCF data retrieval interface
          FC MONCOLL VMCF ON

     FCONRMT SYSTEMS
          *Node-id  Userid    VM-Type    Append    Nickname
          PTKSSIA1  PERFSVM   z/VM6.2    N         FCXC1R01
          PTKSSIA2  PERFSVM   z/VM6.2    N         FCXC1R02
          PTKSSIA3  PERFSVM   z/VM6.2    N         FCXC1R03
          PTKSSIA4  PERFSVM   z/VM6.2    N         FCXC1R04

     FCONRMT AUTHORIZ
          * Authorization file for local and remote
          * data retrieval and command execution
          *Node-ID  User-ID  Authorized for ..
          PTKSSIA1  PERFSVM  DATA
          PTKSSIA2  PERFSVM  S&FSERV DATA

     UCOMDIR NAMES
          :nick.FCXRES00    :luname.*IDENT
                     :tpn.FCXC1R02
                     :security.SAME
          :nick.FCXSYSTM    :luname.*IDENT
                     :tpn.FCXC1S02
                     :security.SAME
```

*Figure 18. Performance Toolkit files for PERFSVM at PTKSSIA2*

```
      FCONX $PROFILE - added
            *    Following command activates VMCF data retrieval interface
            FC MONCOLL VMCF ON

      FCONRMT SYSTEMS
            *Node-id  Userid    VM-Type    Append    Nickname
            PTKSSIA1  PERFSVM   z/VM6.2    N         FCXC1R01
            PTKSSIA2  PERFSVM   z/VM6.2    N         FCXC1R02
            PTKSSIA3  PERFSVM   z/VM6.2    N         FCXC1R03
            PTKSSIA4  PERFSVM   z/VM6.2    N         FCXC1R04

      FCONRMT AUTHORIZ
            * Authorization file for local and remote
            * data retrieval and command execution
            *Node-ID  User-ID  Authorized for ..
            PTKSSIA1  PERFSVM  DATA
            PTKSSIA3  PERFSVM  S&FSERV DATA

      UCOMDIR NAMES
            :nick.FCXRES00    :luname.*IDENT
                      :tpn.FCXC1R03
                      :security.SAME
            :nick.FCXSYSTM    :luname.*IDENT
                      :tpn.FCXC1S03
                      :security.SAME
```

*Figure 19. Performance Toolkit files for PERFSVM at PTKSSIA3*

```
      FCONX $PROFILE - added
            *    Following command activates VMCF data retrieval interface
            FC MONCOLL VMCF ON

      FCONRMT SYSTEMS
            *Node-id  Userid    VM-Type    Append    Nickname
            PTKSSIA1  PERFSVM   z/VM6.2    N         FCXC1R01
            PTKSSIA2  PERFSVM   z/VM6.2    N         FCXC1R02
            PTKSSIA3  PERFSVM   z/VM6.2    N         FCXC1R03
            PTKSSIA4  PERFSVM   z/VM6.2    N         FCXC1R04

      FCONRMT AUTHORIZ
            * Authorization file for local and remote
            * data retrieval and command execution
            *Node-ID  User-ID  Authorized for ..
            PTKSSIA1  PERFSVM  DATA
            PTKSSIA4  PERFSVM  S&FSERV DATA

      UCOMDIR NAMES
            :nick.FCXRES00    :luname.*IDENT
                      :tpn.FCXC1R04
                      :security.SAME
            :nick.FCXSYSTM    :luname.*IDENT
                      :tpn.FCXC1S04
                      :security.SAME
```

*Figure 20. Performance Toolkit files for PERFSVM at PTKSSIA4*

## Starting Performance Toolkit Central Monitoring

As each member is IPLed, it joins the SSI cluster and messages appear on each member's operator's console describing the state of the SSI cluster until the cluster becomes STABLE.

After each member completes its IPL, issue LOGON PERFSVM for each member and have Performance Toolkit start on each one. As Performance Toolkit starts up, each member establishes its APPC resources and connects to the other Performance Toolkit instances in the SSI cluster.

Figure 21 on page 57 shows the Performance Toolkit console when the PERFSVM at PTKSSIA1 starts running. You can see that the APPC resource (FCXC1R01) for PTKSSIA1 becomes active and the other APPC resources are not available.

```
 FCX001           Performance Toolkit for VM        Autoscroll

 10:37:04 FCXAPP530I Connected to *IDENT for resource FCXC1R01
 10:37:04 FCXAPF530I Connected to *IDENT for resource FCXC1S01
 10:37:04 FCXAPC523E Resource 'FCXC1R03' not available
 10:37:04 FCXAPC523E Resource 'FCXC1R04' not available
 10:37:04 FCXAPP527I User PERFSVM connected on path 0006
 10:37:04 FCXAPC535I Connected to resource FCXC1R01 on path 0005, for S&F-Coll
 10:37:04 HCPMOF6229E Monitor event collection is already active.
 10:37:04 HCPMOG6229E Monitor sample collection is already active.
```

*Figure 21. PERFSVM on PTKSSIA1 Initialization*

Figure 22 on page 57 shows the Performance Toolkit console when the PERFSVM at PTKSSIA2 starts running. This is the view from the PTKSSIA2 side. You can see that the APPC resource (FCXC1R02) for PTKSSIA2 becomes active, the APPC resource for PTKSSIA1 is connected, and the other two resources are not available.

```
 FCX001           Performance Toolkit for VM        Autoscroll

 10:40:52 FCXAPP530I Connected to *IDENT for resource FCXC1R02
 10:40:52 FCXAPF530I Connected to *IDENT for resource FCXC1S02
 10:40:52 FCXAPC523E Resource 'FCXC1R04' not available
 10:40:52 FCXAPP527I User PERFSVM connected on path 0007
 10:40:52 FCXAPC535I Connected to resource FCXC1R01 on path 0005, for S&F-Coll
 10:40:52 FCXAPC535I Connected to resource FCXC1R02 on path 0006, for S&F-Coll
 10:40:52 HCPMOF6229E Monitor event collection is already active.
 10:40:52 HCPMOG6229E Monitor sample collection is already active.
 10:41:00 FCXAPP527I User PERFSVM connected on path 0006
 10:41:00 FCXAPP527I User PERFSVM connected on path 0007
```

*Figure 22. PERFSVM on PTKSSIA2 Initialization*

Figure 23 on page 57 shows the view from the PTKSSIA1 side, when the PTKSSIA2 side is started and the connections are made.

```
 FCX001           Performance Toolkit for VM        Autoscroll

 10:39:48 FCXAPP527I User PERFSVM connected on path 0007
 10:40:00 FCXAPC523E Resource 'FCXC1R03' not available
 10:40:00 FCXAPC523E Resource 'FCXC1R04' not available
 10:40:22 FCXAPP527I User PERFSVM connected on path 0008
 10:40:26 FCXAPP528I User PERFSVM has severed path 0008
 10:40:52 FCXAPP527I User PERFSVM connected on path 0008
 10:41:00 FCXAPC523E Resource 'FCXC1R03' not available
 10:41:00 FCXAPC523E Resource 'FCXC1R04' not available
 10:41:00 FCXAPC535I Connected to resource FCXC1R02 on path 0009, for S&F-Coll
 10:42:00 FCXAPC523E Resource 'FCXC1R04' not available
```

*Figure 23. PERFSVM on PTKSSIA1 After PTKSSIA2 is started*

Figure 24 on page 57 and Figure 25 on page 58 show the available APPC resources as seen by the PTKSSIA1 and PTKSSIA2 members, respectively.

```
 FCX001           Performance Toolkit for VM        Autoscroll

 10:45:27 q resource
 10:45:27 Resource: FCXC1R01  Type: Global  Owning Userid: PERFSVM
 10:45:27 Resource: FCXC1S01  Type: System  Owning Userid: PERFSVM
 10:45:27 Resource: FCXC1R02  Type: Global  Owning Nodeid: PTKSSIA2
```

*Figure 24. Available Resources as seen from PTKSSIA1*

```
    FCX001              Performance Toolkit for VM         Autoscroll

    10:45:57 q resource
    10:45:57 Resource: FCXC1R02  Type: Global   Owning Userid: PERFSVM
    10:45:57 Resource: FCXC1S02  Type: System   Owning Userid: PERFSVM
    10:45:57 Resource: FCXC1R01  Type: Global   Owning Nodeid: PTKSSIA1
```

*Figure 25. Available Resources as seen from PTKSSIA2*

As the other members of the SSI cluster are brought online, their PERFSVM user IDs start Performance Toolkit, and the connection messages for APPC resources FCXC1R03 and FCXC1R04 appear on the Performance Toolkit consoles for the first two members.

When all the members of the SSI cluster are active and connected and all your Performance Toolkit user IDs are running, you can point your web browser to the IP address of PTKSSIA1 and include port 8081 in the address. You will see the Performance Toolkit "Web Server Logon" page. Logon to the PERFSVM user ID. The next page to appear is shown in Figure 26 on page 58.

```
    Node-ID   Time   ------------ Exceptions & CPU Load ------------> AvExcp
    PTKSSIA1  10:46  none                                        :      0
    PTKSSIA2  10:46  none                                        :      0
    PTKSSIA3  ....   no data received                            :    ...
    PTKSSIA4  ....   no data received                            :    ...
```

*Figure 26. Central Monitoring System Load Overview Page*

The first two entries are active on the page and can be selected. Selecting one of them takes you to the Performance Toolkit panels for the selected system. You can return to the System Load page to select another system to review. As each member of the cluster and each Performance Toolkit instance becomes active, the status of that member will change on the System Load page

# Setting up Performance Toolkit for Formatted Output Collectors

To use Performance Toolkit to work with a formatted output collector (such as OMEGAMON XE on z/VM and Linux), you must set up Performance Toolkit to collect z/VM data and then write it to a discontiguous shared segment (DCSS). This DCSS must be defined to z/VM prior to use by Performance Toolkit or by the monitoring agent.

**Note:** If you are using OMEGAMON XE on z/VM and Linux as your formatted output collector, see the *OMEGAMON XE on z/VM and Linux Planning and Configuration Guide* (GC32-1957) for more information. Otherwise, see the documentation provided with your specific formatted output collector.

Use the FCONTROL MONCOLL subcommand to control data collection for performance monitoring. Use the SEGOUT option on that subcommand to specify the name of the DCSS that will be used to hold the data. Note that the default name of the DCSS for the SEGOUT option is PERFOUT – so if you decide to use the default name of the DCSS, specify PERFOUT. See the *z/VM: Performance Toolkit Reference* for more information on the FCONTROL MONCOLL subcommand.

For more information on defining the DCSS (including a tool to estimate its size), see Appendix B, "Building a DCSS for a Formatted Output Collector," on page 181.

The amount of data written to the DCSS is determined by the CP monitor domains that are enabled. If a domain is disabled, no data is written to the DCSS for that particular domain. You must enable the CP monitor domains for which you would like to collect data.

Command privilege class E, and a connection to the *MONITOR IUCV service on z/VM, are prerequisites for using the full capabilities of Performance Toolkit with a formatted output collector. The *MONITOR service is used by the z/VM control program to supply monitor data to virtual machines connected to the MONITOR IUCV service. Additional information on the use of this service is available in the *z/VM: CMS Application Development Guide for Assembler*.

Note that Performance Toolkit must be set to the CURRENT mode of data collection. The CURRENT mode displays the values calculated for the last measuring interval. If Performance Toolkit is set to any other mode, it will stop updating the DCSS associated with this monitoring agent. After three sampling intervals have passed, the monitoring agent will emit an alert to let you know that Performance Toolkit is not active. At that point, you will need to reset the Performance Toolkit mode to CURRENT.

# Part 2. Using the Performance Toolkit

The topics in this section contain details on using Performance Toolkit.

# Chapter 3. A Guided Tour

This section describes a sample session in Performance Toolkit. It shows how the program can be used in basic mode and redisplay mode for system operation, and it also gives you a short introduction to some of the features available in performance monitoring mode.

The idea is for you to enter the suggested commands on your terminal and then verify that you obtained the expected results.

## Starting Performance Toolkit

To start the program:

1. IPL ZCMS
2. Access the disks that contain the Performance Toolkit program code and help files
3. Enter this command in native CMS:

```
PERFKIT
```

**Note:** If you are *restarting* Performance Toolkit, first log off and then log back on the PERFSVM user ID to ensure a clean environment.

## Full-Screen Operation

After startup, you should see a screen similar to the following, with a header line at the top, an information message telling you about the maintenance level of the program, a command-input line, and a bottom line that shows PF-key usage.

```
FCX001          Performance Toolkit for VM       Autoscroll 12
FCXBAS500I Performance Toolkit for VM FL720




















 Command ===> _
 F1=Help  F2=Redisplay  F3=Quit  F12=Return
```

This initial screen mode is referred to as "Basic Mode". It allows you to work in a manner that is similar to working in native CMS. The next few steps show similarities and differences.

In machines that have been set up for performance monitoring, two more lines might appear, such as:

```
 MONITOR EVENT STARTED -- RECORDING IS ACTIVATED
 MONITOR SAMPLE STARTED -- RECORDING IS ACTIVATED
```

if CP monitor data collection had not been activated previously, or, if monitor data collection was already active:

```
HCP6229E Monitor event collection is already active.
HCP6229E Monitor sample collection is already active.
```

## Is Something Wrong?

If you see other lines with an FCX*cccnnn*E prefix, such as in the following example, one or more statements in the FCONX $PROFILE initialization file are not correct and should be adapted.

```
FCX001          Performance Toolkit for VM      Autoscroll 12
FCXCMF358E Command valid only if permanent data collection is active
(File: FCONX    $PROFILE *)
(  99: 'FC LIMIT   CPU   180  0/4  5/10')
FCXBAS500I Performance Toolkit for VM FL720
```

To display the message description, enter this command, substituting the message prefix of the error message:

```
help msg fcxcmf358e
```

Otherwise, look the message up in the "Performance Toolkit Messages" chapter in the *z/VM: Performance Toolkit Reference*. The message explanation should tell you what needs to be fixed. (You would see the above example if you had changed the provided FCONX $PROFILE initialization file to not activate the performance monitoring part of the program, but forgot to comment out or delete one of the FC LIMIT statements in the sample file.)

Adapt the incorrect statement in the initialization file, then press the F3 key to return to native CMS and start again. Press F3 twice to quit the program, xedit FCONX $PROFILE to adapt the statements, file your changes, and restart Performance Toolkit.

## Entering CP and CMS Commands

- Enter the CP command q  t (for 'Query Time') on the command line. The command entered should be shown again in the display area, followed by the output lines which it generated.

- Enter the CMS command l  perfkit * * (da. You should see the same output lines as would have been generated by a LISTFILE command entered in native CMS, with one exception: Where the output lines are long, as in this case, they will be split and the last few characters shown in a second line.

- Enter the CMS command q  search. Instead of the expected minidisk search order, you will see the error message HCPCQV003E Invalid option - SEARCH. This is the result of the default command search order of Performance Toolkit, which is:

  - a PERFKIT subcommand

  - a CP command

  - a CMS command.

  For example, the program will first try to execute a command as a CP command, and it will be executed as a CMS command only if the return code from CP indicated 'unknown command'. QUERY is also a CP command ...

  Enter the command again as cms  q  search (where the 'CMS' prefix tells the program that the standard search order is to be bypassed, and the command executed as a CMS command only). You should now see the expected output lines.

- Enter the command q  user (we are not interested in the output right now). Then using the cursor control keys, move the cursor over the screen. Note that it stops at all the command lines which are still visible on the screen. They are all unprotected and can be used for re-entering or modifying a command.

  Select the q  user command we entered above and modify it to q  userid then press the ENTER key. You will find that the command 'Q USERID' has been executed just as if it had been entered from the command line.

- Enter a question mark ?. The last executed command will be retrieved and inserted into the command line, i.e. You should now see the string `q  userid` on the command line where you can modify it and then execute it again.

  PF-key 11 has been set up to execute the same function in the supplied sample initialization file.

## Output Scrolling

By this time the screen will be near full. Operation of Performance Toolkit for a full screen is basically different from native CMS, as you will see now:

- **Automatic scrolling**: Enter a few `ind` commands until the screen becomes full. Once it is completely full, you will see the string

  ```
      --------------      5 sec. scroll wait      -----------
  ```

  in the center part of the top line, indicating that the screen contents will be scrolled after a delay of five seconds. The string `Autoscroll 12` in the top right hand corner of the screen shows what action will be taken. The screen contents will automatically be scrolled upwards by twelve lines after the five second delay. This operation will be repeated, if necessary, until at least one line remains free in the output area of the screen.

  You can change both the scroll wait time and the number of lines to be scrolled by means of the 'FC SCROLL ...' subcommand (see the "FCONTROL SCROLL" subcommand in the *z/VM: Performance Toolkit Reference*).

- **Locking the screen manually:** Enter a command which produces more output than can be shown on one screen, e.g. q  mon. Notice that all the remaining previous output has been moved out of the screen and that the 'q mon' command is shown in the top line, thus allowing a maximum of output for the last command to be displayed.

  Then press the ENTER key before the screen contents have been scrolled. Automatic scrolling is now inhibited, thus giving you all the time you need for checking the command output. The string

  ```
      -------------- Press 'ENTER' to continue ------------
  ```

  in the header line tells you how to continue when you are done. press the ENTER key once, and automatic scrolling will again be in effect.

- **Handling of messages/warnings:** Enter the command `m * test message` to send a message to yourself. Note that the message line in the output area has a '*nn' prefix (where *nn* is a consecutive number, starting from '00' up to '99') and that it is displayed with special color and highlighting attributes.

  Now enter the command q  mon again to force scrolling. The message line will be moved immediately up to the top of the screen after entering the command, but it will remain pending there even while the remainder of the screen contents with the 'Q MON' command output are scrolled.

  Enter the command d  0 (or whatever number is shown in the message prefix) to delete the message line. The asterisk in front of the line will disappear, and the line itself should now be shown with the color and highlighting attributes of normal CP output lines to indicate that it has become scrollable.

  Enter q  mon again. The message line will be scrolled now like all ordinary output lines, and it will disappear from the screen.

- You can clear the complete screen using the PA2 or CLEAR key. This will also clear any message lines on the screen when working with the sample FCONX $PROFILE initialization file, but you can control the action of the PA2 and CLEAR key on message lines with the 'FC MSGCLEAR' subcommand (see the "FCONTROL MSGCLEAR" subcommand in the *z/VM: Performance Toolkit Reference*).

See "Controlling the Scrolling Mechanism" on page 86 for a more detailed description of the scrolling logic.

# Additional Processing / Automation

Output lines shown on the basic mode screen can be subject to further processing and can be used to trigger the execution of CMS commands and procedures, i.e. You can use Performance Toolkit as a tool for automating certain tasks. These functions are useful especially when the full-screen operating functions of the program are used in the system operator's machine, but they come in handy for automating processes in a performance monitoring machine.

- **Suppressing output lines:** Examples for output lines which you would want to have suppressed are the many LOGON, DISCONNECT, RECONNECT and LOGOFF messages which are generated on the system operator console. We will use the output of the QUERY TIME command instead for our example just because we can reproduce it whenever we want which makes testing easier.

  Enter the command `q t`. You will see output lines similar to the following:

  ```
  q t
  TIME IS 13:11:42 CET SATURDAY 12/31/05
  CONNECT= 53:26:18 VIRTCPU= 007:06.64 TOTCPU= 009:07.07
  ```

  Now enter the command `fc process cpout * 'time is' nodisp` which tells Performance Toolkit to

  1. look for 'CPOUT' lines (normal CP command output)

  2. scan them for the string 'TIME IS', where the preceding asterisk '*' indicates that the word may appear in any position in the output line

  3. not to display such lines when found, as indicated by the 'NODISP' argument.

  Enter the command `q t` again, and the first output line should now be missing on the screen, i.e. you should see something similar to:

  ```
  q t
  CONNECT= 53:26:18 VIRTCPU= 007:06.64 TOTCPU= 009:07.07
  ```

  The suppressed line will still be written to the console log of Performance Toolkit, as you will see further on.

- **Changing color/highlighting:** Let us assume that you want to convert message lines from a specific user to the CPMSGN format ('noheader' messages) which can be scrolled like ordinary output lines. The typical example would be message lines from the RSCS machine; you can use those in the example if you have RSCS on your system (ask the machine for the status of a link to make it send you some messages), but the following description is for a message we will send ourselves.

  Enter the command `m * test message`. You will see the following output lines:

  ```
  m * test message
  *01 MSG FROM OURID   : test message
  ```

  i.e. the output line which is reflected back to us has a '*nn' prefix which prevents scrolling, and it is shown with the CPMSG color and highlighting attributes (green and reverse video if you use the initial settings). Now enter the command `fc process cpmsg * 'from ourid : ' rep '' disp cpmsgn` (substitute the userid of the virtual machine you are working in for *ourid*) which tells Performance Toolkit to:

  1. look for 'CPMSG' lines (CP message lines)

  2. scan them for the string 'FROM OURID : ' anywhere in the input line

  3. to replace the beginning of the line, up to the string specified in the command, by a null-string

  4. to display the remainder as noheader message, i.e. with the color and highlighting attributes specified for noheader messages, and also not to insert the '*nn' prefix which makes message lines non-scrollable.

Enter the command m * test message 2 now. You should see the following output lines:

```
m * test message 2
test message 2
```

The message header has disappeared, and by entering some more commands to fill up the screen you can prove that the line will indeed be scrolled.

- **Invoking a REXX procedure:** The possibility to have a CMS program or a REXX procedure started automatically is a basic requirement for automating the operation of a VM system. xedit autotest exec a and create the following test procedure:

```
/* Test automated operation capability of Performance Toolkit */
arg input
say 'Input received:' input
```

File it to your A-disk. The procedure should show two things:

1. Produce some output to show that it has been executed
2. Read and display the data string which caused it to be invoked, to show that the contents of this string could be used to control the actions such a procedure would have to perform in real life.

Now enter the command fc process cpmsg * 'special msg' nodisp exec autotest passarg which tells Performance Toolkit to:

1. look for 'CPMSG' lines (CP message lines)
2. scan them for the string 'SPECIAL MSG' anywhere in the input line
3. to suppress output on the screen when a matching message line has been found ('NODISP' argument)
4. to call procedure AUTOTEST ('EXEC AUTOTEST' arguments), and
5. to pass the complete output line to the procedure as arguments when it is executed, as specified by the 'PASSARG' argument.

Enter the command m * this is a special msg test. You should see the following output lines:

```
m * this is a special msg test
Input received: this is a special msg test
```

i.e. the message line we sent ourselves has been suppressed, but the 'Input received ...' line shows that our AUTOTEST procedure has successfully intercepted the line and could use it for further processing.

See section "Special Line Processing" on page 88 if you are interested in additional information on the processing of output lines.

## Initialization File FCONX $PROFILE

This should be a good time to introduce you to the initialization file, a CMS file named FCONX $PROFILE on any accessed disk. In the preceding section on additional processing and automation you had to enter many complicated commands, and you obviously would not want to repeat this process for each new session, but would rather include such commands in some list of commands which will automatically be executed whenever the program is started.

You can include in the initialization file any of the 'FCONTROL ...' subcommands which help you customizing your environment. They are executed in the sequence in which they are found.

The system profile is shipped as the file FCONX $PROFILE. To change this file you should use the VMSES/E local modification procedure. This procedure uses the XEDIT command with the CTL option, the VMFEXUPD command ,and the VMFBLD command to apply an update to the FCONX $PROFILE file to generate an updated FCONX $PROFILE. Note that the FCONX $PROFILE is located on the 5VMPTK40 1CC disk. (For more information on the local modification procedure, see *z/VM: Installation Guide* and *z/VM: Service Guide*.)

See "FCONX $PROFILE" on page 92 for more information on the initialization file.

# Re-displaying Previous Output

The possibility to review system messages and operator activity for some earlier period of the day is a very frequent requirement. Performance Toolkit saves all the output lines it received, including the ones which were suppressed using 'FC PROCESS ...' specifications, in a disk file which can be browsed.

Enter the command REDisp or press F2 to switch to 'redisplay' mode. The new mode is indicated by the string Redisp. Mode in the top right hand corner of the initial screen you should now see, which displays the last (most recent) part of the log.

```
FCX002              Viewing Log File 'yyyymmdd CONLOG  A'           Redisp. Mode
                                                             Line nnnnn of nnnnn
 ...
 ...
```

- **Suppressed lines:** Scan the last few lines of the log, up to the part where you entered the command m * this is a special message test and you will see that the log contains *all* lines, even the ones which were suppressed by means of an 'FC PROCESS ... NODISP' statement. Suppressed lines are marked by a '>' indicator on the left.
- **Scrolling:** The usual method is to work with the scrolling PF-keys, whose definitions you will find in the bottom line of the screen, but you can also move upwards or downwards for a specific number of lines by using the FORWARD/BACKWARD commands or their synonyms DOWN(or NEXT)/UP with an argument specifying the number of lines to be skipped.
  - Press F7. You should see the log lines now which immediately precedes the ones you have seen on the previous screen.
  - Press F4. You will see the beginning of the log, with a header line which tells you when the log has been started.
  - Press F8. You will see the lines which immediately follow the ones from the previous screen (note that the default number of lines scrolled is 'available output lines - 1', i.e. the bottom line of the previous screen becomes the top line on the next screen while scrolling forward).
  - Enter the command n  10 (abbreviated for 'NEXT 10') to skip forward 10 lines.
- **Locating specific strings:** The log can become very long if you use Performance Toolkit on the system operator's console of a large system. Browsing through such a large log using the FORWARD/ BACKWARD commands or PF-keys is not efficient. It is usually much faster to look for a specific string (time stamp, or part of an output line which you remember).
  - Press F5 (bottom) to display the last few lines again.
  - Enter the command -/search. If you have done exactly what you were asked to in this script, the screen should now be positioned near the top of the log where you entered a 'CMS Q SEARCH' command.
  - Press the ENTER key again and the previous 'locate' command will be repeated, and it should have brought you to the first attempt with the 'Q SEARCH' command.

  Try to locate some other string if 'Q SEARCH' cannot be found because you did not enter this command during the current session.
- **Returning to basic mode:** Press F12 to return to basic mode.

  You will also be returned automatically to basic mode as soon as you enter a command which is not valid in redisplay mode (e.g. a CP command).

See also "Redisplay Mode" on page 81 for a general description of redisplay mode.

# Help Facility

The general help facility of Performance Toolkit is based on the CMS help facility.

- **General help:** Press F1. You should see a master help menu for Performance Toolkit.

- **More details:** Move the cursor to the line

```
'Sub-commands-General (all modes)'
```

and press ENTER. You will see a further selection menu for 'General Performance Toolkit sub-commands' .

Let us assume you are interested in more details on the 'FC PROCESS' command which we used in some of our examples. Enter the 'locate' command /`process` on the command line. This will position the menu to the entry for the command.

Now move the cursor to the 'FC PROCESS' line on the screen and press ENTER again. You should now see the command description we looked for.

Leave the HELP facility by pressing F3 until you are back to Performance Toolkit.

You can bypass all the menus if you know what to look for:

- **Fast path to command:** Enter the command `help fc process` on the command line. You should see the same text we found above going through some menus. press F3 once to return to Performance Toolkit.

- **Fast path to message:** Let us create an error message first ...

  – Enter the command `xyz`. This is unlikely to be a valid command on your system and should thus result in error message 170E being shown on the message line just above the command line.

  – Enter the command `help msg`. The help text for that message should now be shown (if an error message is shown in the message line, the help text for that message is selected by default unless you specify another message number).

    Press F3 once to leave HELP.

  You can also use the 'standard' fast path to messages, i.e. enter the command `help msg fcxbas170e` (or any other message prefix) to display the desired help text.

A special 'contextual help' facility is available for use in conjunction with performance monitor displays. You will learn about it while you work your way through the next section. You can place the cursor on any field and press F1 to display help information.

Note that the HELP command and the F1-key will bring you directly into the Performance Toolkit subset of the CMS help facility. Enter the command `cms help` to select the normal CMS help menu if you need information on standard CP and CMS commands and messages.

See also section "Performance Toolkit HELP Facility" on page 83 for more information on the help facility.

# Performance Monitoring Mode

The prerequisites are as follows:

1. Performance monitoring mode can be entered only when working in a virtual machine with the necessary privilege classes, and with the directory statements which allow it to connect to the *MONITOR system service.

2. CP monitor data collection must have been activated before any useful performance monitoring can be done.

Both prerequisites are met if you used the sample FCONX $PROFILE as it was supplied with the program.

Enter the command `MONitor` to start working in performance monitor mode.

Your screen should look similar to the following example:

```
FCX124                    Performance Screen Selection  (FL720        )    Perf. Monitor

General System Data         I/O Data                    History Data (by Time)
1. CPU data menu*           11. Channel load            31. Graphics selection
2. Storage utilization      12. Control units           32. History data files*
3. SSI data menu*           13. I/O device menu*        33. Benchmark displays*
4. Priv. operations         14. PCI Function menu*      34. Correlation coeff.
5. System counters          15. Cache extend. func.*    35. System summary*
6. CP IUCV services         16. Reserved                36. Auxiliary storage
7. SPOOL file display*      17. DASD seek distance*     37. CP communications*
8. LPAR data menu*          18. I/O prior. queueing*    38. DASD load
9. Shared segments          19. I/O configuration       39. Minidisk cache*
A. Shared data spaces       1A. I/O config. changes     3A. Storage mgmt. data*
B. Virt. disks in stor.                                 3B. Proc. load & config*
C. Transact. statistics     User Data                   3C. LPAR logs menu*
D. Monitor data             21. User resource usage*    3D. Response time (all)*
E. Monitor settings         22. User paging menu*       3E. RSK data menu*
F. System settings          23. User wait states*       3F. Scheduler queues
G. System configuration     24. User response time*     3G. Scheduler data
H. VM Resource Manager      25. Resources/transact.*    3H. SFS/BFS logs menu*
                            26. User communication*     3I. System log
I. Exceptions               27. Multitasking users*     3K. TCP/IP data menu*
                            28. User configuration*     3L. User communication
K. User defined data*       29. Linux systems*          3M. User wait states
                            2A. CPU Pools menu*
                            2B. User Processor Time


              Pointers to related or more detailed performance data
              can be found on displays marked with an asterisk (*).



 Command ===> _
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 27. Performance Data Selection Menu*

If you see error message 311E, your machine does not have the minimum CP privilege class requirements (class E).

If you see error message 310I, data collection from the CP monitor shared segment has not been activated.

**You cannot continue if either message is shown. Make sure the machine's directory is updated as required before trying to execute the remainder of this script. You will have to log off and then log back on after the directory changes to let them become effective.**

Performance screens with numbers that do not appear highlighted on the initial menu cannot be selected because CP monitor data has not been collected for them. Possible reasons are:

• The required monitor records are not available because the CP level of your VM system does not yet produce them

• The required monitor domains have not yet been enabled

• The first monitor sample interval has not ended and we are still waiting for the initial set of monitor records

• The display is intended for data which cannot be generated on the system (LPAR data on an unpartitioned system, for example).

## Selecting Performance Screens

The screens are grouped into 'general system data', 'I/O data', 'user data' which show data for either the last monitor sample interval or overall averages, and a series of 'by time' log files which show a separate set of values for each of the monitor sample intervals. They can be selected from the initial menu by either:

• entering the number of the selected performance data display on the command line, or

• moving the cursor to the selected display (note that it will move only to highlighted numbers if you use the tabulator keys), or

- entering the 'short-path' command for the desired screen

and then pressing the ENTER key.

For example:

- **Selection by number:** Enter 1 on the command line. You should see the 'CPU load' screen with information on processor utilization, queue statistics, transactions, etc.

  Press F12 to return to the initial menu.

- **Selection with cursor:** `Move the cursor` to the number field for the 'Shared segments' screen (number 9), then press ENTER. The 'Shared segments' screen should be displayed.

  Note the PF-key allocation in the bottom line. Keys for right/left shifting are included in the list, indicating that the available information extends beyond the 80 columns of an ordinary display terminal. Press F11 to shift the display window to the right (the first data column with the segment names is not shifted so that you can still see to which segment the data fields apply). Press F10 to shift back to the initial display.

  Press F12 to return to the initial menu.

- **Selection with short-path command:** Enter the command `user` on the command line. You should see the general 'User resource usage' screen, with information on user resource consumption.

  Ask for the initial menu by command: Enter menu on the command line instead of pressing the F12 key to bring you back to the initial selection screen.

Screen selection by short-path command or by number is possible from any of the many other performance monitor displays, while the first two methods will only work from the initial menu.

## Selecting Detailed Data

Many of the available performance displays allow easy selection of more detailed data for one specific user, or one specific I/O device. An information message line near the bottom of the screen will inform you whenever this is the case.

For example:

- Select the 'User resource usage' screen again (by number, with the cursor, or using the short-path command `user` on the command line).

- `Move the cursor` to the userid of an active virtual machine (note that the entries for machines with some activity are highlighted) where you would like to see more detailed information and press ENTER.

  You should see the 'User resource details' screen with a lot of detailed information for the selected user. Note especially the data on the virtual I/O configuration of the selected machine. This part can be very helpful to tell you the target of all the I/O activity of a user with a suspected I/O problem.

  This screen will be updated automatically in 60-second intervals, but you can force more frequent updates by pressing the ENTER-key. Press ENTER in intervals of a few seconds and note how the load figures change.

- Assume that one specific minidisk is heavily loaded, and that you want more information on the physical disk pack where it is located: `Move the cursor` to that minidisk and press ENTER.

  You should see the 'I/O device details' screen for the disk pack where the selected minidisk is located.

- You can force updates manually for this screen. Press ENTER in intervals of a few seconds and note how the load values change.

- Assume that you find a heavily loaded minidisk on this pack (not necessarily owned by the user we started with), and that one user is shown to be responsible for most of the load. `Move the cursor` to that userid and press ENTER.

  You should see the 'User resource details' screen for the selected user.

# Sorting Data

Many displays allow sorting of the data shown in different sequences. This is indicated by the special SORT indicators above the column headings:

- A dot indicates that the screen can be sorted for that column
- Underscore characters above a column heading indicate the variable that is currently being used for sorting.

The sort sequence can be changed by command or by cursor selection and pressing the ENTER key.

For example:

- Enter the short-path command dev to select the I/O device screen.

    Note the sort indicators. The underscores will initially be above the Addr field (devices sorted to device numbers), but they may be somewhere else if you set another sort sequence in the initialization file FCONX $PROFILE.

- Enter the command sort disc. The output lines should immediately be re-organized in descending sequence of Disc (the disconnected time).

    The argument to be used with the SORT command is usually identical to the column description in the heading line. Where more than one heading line is shown it is the last part of the column description, in the lowest heading line.

- Move the cursor to the heading fields of another column, for example, Pend and press ENTER. The output lines should now be sorted in descending sequence of Pend values (function pending time).

# Displaying Average or Current Data

All of the general performance data displays show 'current' data initially, i.e. values based on the last monitor sample interval. But it is possible to display average data too, i.e. values based on the whole period where performance data were collected, or since the time when they were last reset.

For example:

You should still be looking at the I/O device screen now. Enter the command dev to bring it back.

- Note the time interval indicated in the header line. It should show the start and end time for the last CP monitor sample.
- Now enter the command average on the command line.

    The time interval in the header line should have changed to indicate the period from the beginning of the first monitor sample up to the end of the last one, and most of the values on the screen should have changed too since they will now be averages for the whole period.

- Enter the command current on the command line.

    The time interval in the header line should show the start and end time for the last monitor sample again, and all performance values should show values for this interval.

# Contextual Help

The meaning of a specific performance variable will not always be obvious, especially if it is only shown in abbreviated form as a column header. While the documentation will explain all these fields, it will take time to find the item even when you have the manual on your desk. The 'contextual help' facility is another method you can use for displaying relevant help text.

For example:

You should still be on the I/O device screen now. Enter the command dev if you have left it in the mean time.

- Let us assume you want a description for the variable

```
Disc
```

which we used in the preceding section for sorting. `Move the cursor` to the word `Disc` in the header line and press the F1 key.

You should now see a window with the help text for this variable overlaying part of the I/O device screen. Press F12 to return after reading the explanation.

- Repeat the same process for the `Avoid` variable.
- Now `move the cursor` to one of the I/O data lines (*not* a header line) and press F1 again.

  All the help text for *all* fields of the current performance screen should now be displayed in a large window. Forward and backward scrolling PF-keys allow you to browse through the help text. Press F12 to return when you are done.

The help text you have been shown is a duplicate of the variable descriptions in the documentation.

## Graphics

Enter the command `graphics` to activate the graphics selection menu. It allows you to specify what kind of graphics should be shown: source of data, period to be selected, and the variables which are to be displayed.

For example:

- The `Format` field shows GDDM initially, meaning that GDDM graphics would be generated. let us assume that GDDM is not installed on your system, and that you want to create a simple plot instead. There are two methods, and you just learned about the first one, context help.
  - `Move the cursor` to the `Format` line and press F1. You should see a help window which explains that the string PLOT must be inserted.
  - Press F12 to return.

  Knowing now that we should replace the string GDDM by the string PLOT you could just key in the new specification, but let us try another way first. The text below the selection fields tells you that instead of directly keying in your selection you can also enter a question mark (or any other invalid selection, actually) to be prompted with a more detailed selection menu:
  - Enter a question mark ? at the beginning of the input field for the `Format` selection and press ENTER. A window with a list of the valid selections and their explanation should be shown.
  - Move the cursor in front of the PLOT selection and press ENTER.

  Note that the field on the GRAPHICS selection menu has changed and that it now contains the string PLOT.
- Press ENTER to build graphics based on the current selection. You should see a plot for `'% total CPU'` the total CPU load, and `'Cl1 85% time'` the length of the (major) class 1 dispatch time slice.

  Press F12 to return to the selection menu.
- Let us see how the output looks if a single variable has been selected: Overtype the string C1ES which describes the second Y-variable with blanks (at least one blank required, an 'erase field' is not sufficient) and press ENTER. You should see a plot for the total CPU load only, where the CPU load is shown in the form of a bar chart, not just as single markers. This format is always used when displaying a single variable.

  Press F12 to return to the selection menu.
- Let us try to select a new second performance variable. `Move the cursor` to the field for the second Y-variable which is blank at the moment, key in a question mark ? and press ENTER.

  You will see a variable selection menu with a list of all the performance variables whose values can be plotted.
- Move the cursor down to the field `IO/S` and press ENTER.

Note that the second Y-variable is now set to IO/S i.e. it will show the system's real I/O activity.

- Press ENTER again. A plot for total CPU time (we have not touched the definitions for the 1st variable) and the 'SSCH rate/s' should be shown.

  Press F12 to return to the graphics menu.

- If GDDM is available on your system overtype the current format definition of PLOT with GDDM and press ENTER. You should now have GDDM graphics for the same data which you just plotted in the previous paragraph.

  Press F12 to return to the graphics menu.

- All the graphical displays we've seen so far were 'by time' graphics, i.e. the X-axis was used as a time scale, as specified by the 'DETailed history' definition in the Graphics type input field. Enter a question mark ? on the Graphics type input field and press ENTER.

  You will see a selection menu which offers DETailed history, SUMmary history and VARiable correlation as possible choices. (Note that the selection PROfile history is not highlighted, i.e. is not valid at the moment. It can be specified only when a condensed history file of filetype PERFHIST is used as input.)

- Move the cursor to the VARiable correlation selection and press ENTER. The graphics type field should now have changed to VARiable correlation and the field describing the X-variable should have been set for the default X-variable 'ACT' (number of active users in a sample interval).

- Press ENTER. You should see a variable correlation plot where the same Y-variables we had used before are now plotted as a function of the number of active users. This kind of plot can be very useful on systems with a large number of CMS users, where it can show probable trends in resource utilization which allow extrapolation for different numbers of active users.

  Press F12 to return to the graphics menu.
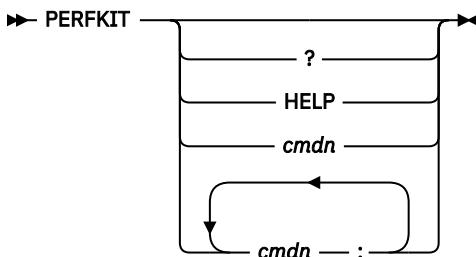
See the "Graphical Displays of Performance History Data" section in the *z/VM: Performance Toolkit Reference* for a detailed description of all the different kinds of graphics which can be generated.

Enter the command basic. This should bring you back to basic mode.

This is the end of our short tour through the performance monitoring part of Performance Toolkit. See Chapter 5, "Operation in Performance Monitor Mode," on page 99 for a more detailed introduction to performance monitoring.

# Chapter 4. General System Operation

To start Performance Toolkit in interactive mode, enter:

```
►►── PERFKIT ──┬────────────────┬──►◄
               ├─────── ? ───────┤
               ├────── HELP ──────┤
               ├────── cmdn ──────┤
               │   ◄───────────   │
               └──┬─ cmdn ── ; ─┬─┘
```

For information about starting Performance Toolkit in batch mode, see Chapter 6, "Operation in Batch Mode," on page 139.

Performance Toolkit also supports using CMS Pipelines as monitor data input for MONSCAN and BATCH operation. See "Using CMS Pipelines as Monitor Data Source" on page 133 for more information.

When entered without a parameter, the command 'PERFKIT' puts the screen into basic mode (see "Basic Mode" on page 76). If '?' or 'HELP' is entered, Performance Toolkit will directly enter 'HELP' mode and an exit from help mode will, in this case, return control directly to CMS.

Other arguments will be stacked and run after initialization completes. A semicolon (;) is interpreted as a command separator. You can let several Performance Toolkit sub-commands be stacked in this way by separating them with a semicolon when you enter them. They will be run first-in first-out, but *after* any commands found in the initialization file FCONX $PROFILE.

**Note:**

1. For a user with privilege class B authorization, the FCONCMD FCSTACK command may be needed to stack commands (instead of using a semicolon as a command separator). See the "Entering Performance Toolkit Subcommands from EXECs or CMS Programs" section of the "Performance Toolkit Subcommands and Screen Layouts" chapter in the *z/VM: Performance Toolkit Reference* for more information.

2. If you are *restarting* Performance Toolkit, first log off and then log back on the PERFSVM user ID to ensure a clean environment.

The operating modes used for general system operation are:

- Basic mode: the standard operating mode which is automatically entered after Performance Toolkit initialization
- Redisplay mode: allows browsing through the day's console log

See "Modes of Operation" on page 5 for a list of all operation modes. You can also use the HELP facility for information on subcommands or the meaning of messages generated by the program.

Once Performance Toolkit is started in basic mode, you can go on working in your machine as before. You can run CP and CMS commands and read their output without knowing any further details about Performance Toolkit. However, there are a few things (for example, setting PF-keys, command search order) that work differently.

The basic information required to efficiently use Performance Toolkit in its two general system operation modes can be found in the next three sections of this chapter. This is essential information which you should read before starting. The remaining sections of this chapter provide more detailed information on tailoring and special uses. Chapter 5, "Operation in Performance Monitor Mode," on page 99 describes how to use Performance Toolkit for performance monitoring and the "Performance Toolkit Subcommands and Screen Layouts" chapter in the *z/VM: Performance Toolkit Reference* describes the subcommands in detail.

Enter the QUIT command to end operation or press the corresponding PF-key. The interface to the IUCV *MSG service will be reset (MSG, WNG, EMSG, and IMSG all set ON; VMCONIO, CPCONIO and RUN set OFF), leaving the screen so normal CMS operation can continue.

# Basic Mode

This is the standard mode of operation. It allows entering CP and CMS commands as well as some subcommands to tailor Performance Toolkit operation to the user's needs.

## Screen Layout

```
Screen Identifier       Status Message Area        Scroll setting
     |                          |                        |


 FCX001                Performance Toolkit for VM         Autoscroll 10
 01
 02
 03 <----- Line numbers (optional)
 04
 05
 ..
 ..
 ..         Output area
 ..
 ..
 16
 17
 18
 19
 20
 ...  message area ...
 Command ===> _     Command Line
 F1=Help  F2=redisplay  F3=Quit  F12=Return
```
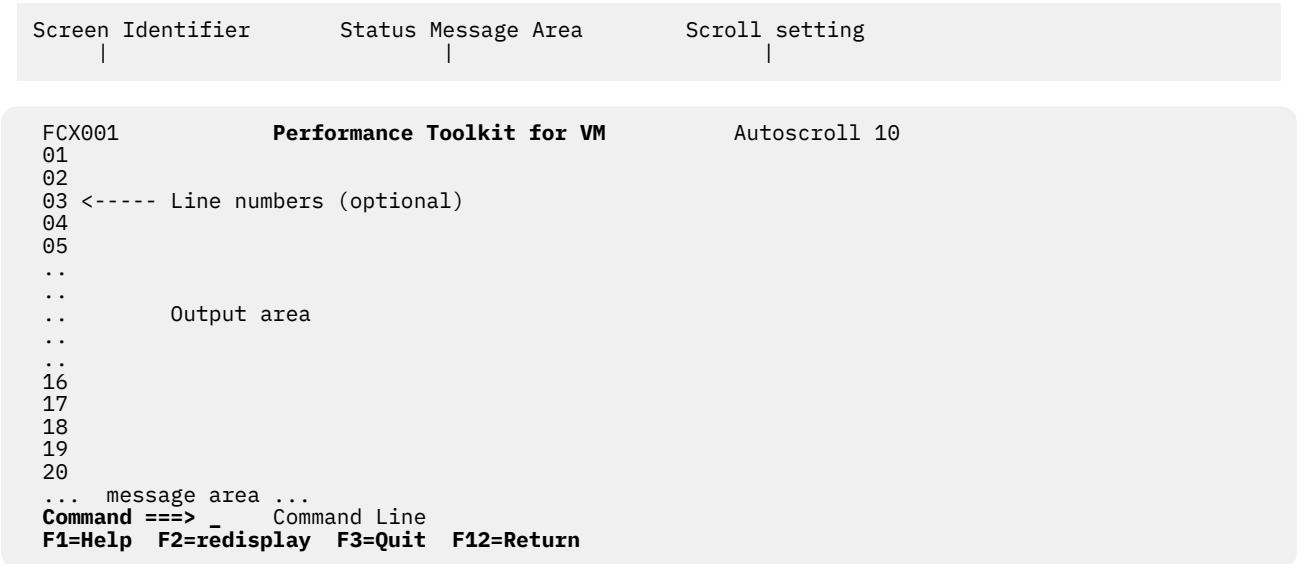
*Figure 28. Layout of Basic Mode Screen*

The PF-key settings are:

Fixed PF-key usage in basic mode:
**PA1**   switch to 'CP READ' mode
**PA2**   clear screen, leave cmd.  (also CANCEL-key on 3036 consoles)
**CLEAR**  run CMS CLEAR

Initial setting for PF-keys (can be re-defined by user):
**F1**   switch to 'help' mode
**F2**   switch to 'redisplay' mode
**F3**   end Performance Toolkit operation
**F12**   end Performance Toolkit operation
PF-keys 1-24 can be set to run any desired command.

## Entering Commands

**On the Command Line:** Performance Toolkit's command line is shorter than the maximum command length allowed by CP. Most commands will easily fit into this shortened line. Others (such as messages) may require more space. To accommodate such long commands, the bottom line with the PF-key allocations has been designed for use as a command line extension. Continue keying in your command even if it extends into the bottom line. The command will be correctly interpreted if the bottom line is just overtyped.

Please note that the command length is determined by comparing the original bottom line with the altered one. You can erase the remainder of the bottom line, but ***do not*** shift it (Delete key) or it will be interpreted as part of the command.

**In the Output Area:** While commands will usually be entered on the dedicated command line as described above (the cursor will always be positioned there after pressing the Enter key), you may also overtype any of the previously entered commands which are still visible on the screen, or any of the old commands shown in redisplay mode.

At least one character must be keyed on the line where the command is displayed to run it. The command must fit in one line. Command continuations or additional commands entered on another line will be ignored.

## Logical Line-End and Escape Characters

**Logical line-end character:** The # character is interpreted as a logical line-end character. A command line with an intervening # will be split at that location, and the two parts run separately.

**Logical escape character:** The " character is interpreted as a logical escape character. A # character following a " character will be accepted as an ordinary character instead of being treated as a logical line end.

**Note:**

1. Both of these logical line-editing characters, although identical to the defaults used by z/VM, are fixed and cannot be changed to other settings.

2. These characters *cannot* be used when accessing Performance Toolkit remotely.

## Command Search Order

Commands entered on the command line will usually be processed as follows (default search order as set by the FC SEARCH CPCMS command):

1. If the command is a Performance Toolkit subcommand which is valid in the current mode, the subcommand will be run.

2. The command will be passed to CP to be run as it would have been for the default system console mode (TERMINAL MODE set to CP).

3. If CP does not know the command it is assumed to be a CMS command or a CMS EXEC procedure. If an EXEC with the command name can be located on any accessed disk, the EXEC will be run. If no EXEC can be found, the command is run as a CMS command. An error message is issued if the command is still not successful.

This default search order will be bypassed if the command is preceded by a CP, CMS, or EXEC prefix, and it can also be changed so that commands that are not part of Performance Toolkit will be run as CMS commands. See the FCONTROL SEARCH subcommand in the *z/VM: Performance Toolkit Reference*.

The command output will be shown on the screen as usual. The following are exceptions:

- Due to the command search order some abbreviations of CP/CMS commands may be interpreted as Performance Toolkit subcommands. Entering 'D 20000.10', for instance, to display your virtual storage at address X'20000', will lead to error message 'FCXCMD216E PARAMETER '20000.10' CONTAINS NON-NUMERIC CHARACTERS' because Performance Toolkit assumes the command to be an abbreviation of the DELETE subcommand (where '20000.20' cannot be a valid message line number). By using the form 'CP D 20000.10' or 'DI 20000.10' instead of the minimum abbreviation the meaning of the command can be made clear.

  Similarly, a CMS command that can also be interpreted as a CP command (such as QUERY) will be run as a CP command with Performance Toolkit's default search order unless it is preceded by CMS, that is, you will have to enter the command 'CMS Q SEARCH' to display the CMS disk search order.

- You will find the message 'CMS command executing' highlighted in the center field at the top of the screen when a command has been passed to CMS for running.

- CMS commands can be interrupted using the HX command, but Performance Toolkit will also terminate with an ABEND to make sure that everything is cleaned up again. All other 'immediate' commands are not supported. They will be treated as ordinary CMS commands. They will be stacked and eventually run only once the currently active CMS program has ended.

- The short form of the CMS 'Ready' message will always be simulated by Performance Toolkit, even if RDYMSG has been set to 'LMSG'.
- The manner in which output lines are displayed while a CMS command is active, differs from standard CMS console mode and can be influenced with the FC UPDTCMS subcommand (see the "FCONTROL UPDTCMS" subcommand in the *z/VM: Performance Toolkit Reference*). In the initial display mode (UPDTCMS set to DELAYED), output lines are stacked and will be shown only when the command has ended or when we enter a CMS console read state. This is the preferred display mode, both for performance reasons and because it eliminates the potential problems associated with IMMED mode which are described below.

  With UPDTCMS set to IMMED, any output lines received while a non-full screen CMS command is active will be displayed immediately. The advantage of this mode is apparent when you have invoked a long running CMS program under Performance Toolkit that writes status messages from time to time to inform you of the progress made. However, you should be aware that undesirable results can occur in the case of full screen applications. If they also generate normal CMS output lines, or if CP output is received, Performance Toolkit will immediately show such output lines and overlay whatever data was on the screen before, unless the other application also used the CONSOLE macro for handling the full screen I/O.

  On large screens (model 3, 4, or 5) a 'PROG nnn' error might even be caused if the full screen application had previously reset the display to model 2 mode, so that the lower part of the screen is no longer addressable. Although Performance Toolkit can usually handle this situation and should recover automatically, your machine may already have been disconnected by CP, so that a RECONNECT or LOGON is required before you can go on.

  Performance Toolkit will be notified that another program took over control of the console if the other program also used the CONSOLE I/O interface of CMS, and it will then wait until the CMS program has ended, or until a console read is required, before it updates the screen.

  Use the SET CMSTYPE HT command or spool the console to 'NOTERM' in all full screen applications that might create unwanted output lines if you choose to work with UPDTCMS set to 'IMMED'.

## Scrolling

Command output will immediately be displayed as long as there are still free lines on the screen. When the screen becomes full, the message 'nn sec. scroll wait' will appear in the center field of the top line if 'auto scroll' mode is active (the default). If you want to freeze the current screen in order to have a closer look at the output, press ENTER while the output lines are still displayed. The screen will become locked and the message 'Press 'ENTER' to continue' will be displayed on the top line. The same message will also appear immediately if scrolling has been set to 'manual'.

If you pressed 'ENTER' too late and the output lines you were interested in have already scrolled from the screen, you can enter redisplay mode (see "Redisplay Mode" on page 81) to again view all output. See also "Controlling the Scrolling Mechanism" on page 86 and the "General Subcommands" section in the *z/VM: Performance Toolkit Reference* for more details on the scrolling mechanism and information on how to change the scroll mode and the scroll wait time.

## Special Handling of Message Lines

'Message lines' in the Performance Toolkit terminology can be CP messages or warnings (default) or any other output lines which you have redefined as Performance Toolkit message lines via the FCONTROL PROCESS subcommand (see "Special Line Processing" on page 88 and the "FCONTROL PROCESS" subcommand in the *z/VM: Performance Toolkit Reference*). These lines are designated as Performance Toolkit message lines by a leading asterisk which lets them be scrolled only up to the top of the screen, where they will be left pending until explicitly deleted.

This feature facilitates operating the system by letting the operator see at one glance all the pending messages that still require some action to be performed while allowing concurrent display of output from commands that should be run **before** the message(s) are deleted.

A typical example is the handling of requests for tape mounts. Assume that you receive the message

```
*03 MSG FROM CMSA   : PLEASE ATTACH TAPE UNIT AS 181 AND MOUNT TAPE 'MYTAPE'
```

from user CMSA. To handle this mount request you might use the following sequence:

```
1) 'QUERY TAPES'                    (To find a free tape unit.
                                     Message will stay on screen.)
2) mount the user's tape
3) 'ATTACH 380 CMSA 181'            (attach the tape unit to the user)
4) 'R 3 MSG CMSA TAPE IS NOW READY  (delete message and inform user)
```

Please note that messages should be deleted only when there is no further action required.

## Action Messages

CP inserts additional carriage return/line feed (CRLF) characters before some messages that are used, for example, to inform the operator on hardware problems or on the 'intervention required' status of some device. The intention is to attract the operator's attention so the necessary actions can be performed.

All general CP output lines (line type not CP message, MSGNOH, warning or SCIF) that are preceded by at least one additional CRLF character (X'15') or which contain at least two consecutive X'15' characters are treated as a special 'action message' line type CPAMSG. These action messages can be shown with special color or highlighting, and they will be preceded by a plus sign (+) and a consecutive message number. A certain number of these action messages will not be automatically scrolled but will be left pending at the top of the screen. You can specify this number using the FC ACTMSG subcommand, the initial setting is 10. See the "FCONTROL ACTMSG" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.

Please note that the Performance Toolkit action messages as defined above may not be identical with CP action messages (messages with suffix 'A'), and that only the first line of a two or three line message will usually be treated as an action message (second and consecutive lines are not preceded by an additional X'15'). However, you can let Performance Toolkit treat any other CP output line as an action message by just entering the necessary FC PROCESS command (see "Special Line Processing" on page 88 and the "FCONTROL PROCESS" subcommand in the *z/VM: Performance Toolkit Reference*).

## Error Messages

Performance Toolkit error messages will generally be shown in the special message area only, where they are automatically deleted whenever the ENTER-key or a PF-key has been pressed. When additional error messages should be shown before the last one has been deleted, they will be inserted into the general output area of the screen, except if the new message is identical to the one already being displayed in the message area.

## Subcommands

Subcommands are provided for:

- Deleting message lines: message lines will not be scrolled and have to be explicitly deleted. To delete the messages with numbers 01 and 04 enter

  ```
  'D 1 4'
  ```

  To indicate the command has been run, the messages will no longer be highlighted and when the screen becomes full again they will disappear.

  Messages that request an operator action can be answered and deleted at the same time using the REPLY command.

- Displaying HELP text: press PF1 or use the HELP subcommand (see the *z/VM: Performance Toolkit Reference*) if you need assistance. See also "Performance Toolkit HELP Facility" on page 83 for more information on the HELP facility.

- Setting up PF-keys: Use the FC PFKEY subcommand to define or re-define a PF-key or to display its current contents. **Only** the FC PFKEY subcommand can set PF-keys for use with Performance Toolkit,

the standard CP command SET PFK has no effect on the program function keys used while Performance Toolkit is active. See the "FCONTROL PFKEY" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.

- Leaving Performance Toolkit mode: Enter 'QUIT' or press PF3 to exit and return to CMS.
- Entering redisplay mode: Enter 'REDISP' or press PF2 to switch from Basic to redisplay mode.
- Rerunning commands: You can either just rerun the last command entered by keying in '=' or you can have it inserted into the command line by using the retrieve function '?' and then modifying it.
- Controlling many functions like scrolling, color, and highlighting, or special additional processing of selected output lines. The FCONTROL command with its many subcommands deals with functions you will probably want to set up only once when tailoring Performance Toolkit to your needs. FCONTROL subcommands will, therefore, usually be included in the initialization profile and not entered directly on the command line. See Appendix A, "A Sample Initialization Profile (FCONX $PROFILE)," on page 173 for details.

Please refer to the "General Subcommands" and "Basic Mode Subcommands" sections in the *z/VM: Performance Toolkit Reference* for a detailed description of subcommands valid in basic mode.

## Reply to CP and CMS Prompts (CP and VM READ state)

**CP READ:** CP will prompt you for a reply when running:

- A LINK command for a password-protected minidisk
- An AUTOLOG command
- A SET LOGMSG command

When run in the machine that also runs Performance Toolkit these commands may result in the screen being placed in CP READ status before the Performance Toolkit display that contains the prompting message (for example, 'ENTER PASSWORD') can be shown.

In the case of the LINK or AUTOLOG command this should not be a problem, because you know that a password has to be keyed in. After entering it, the screen will switch back to the Performance Toolkit display and show the CP prompt.

When using the SET LOGMSG command, however, it is probably easier to either key it in from another terminal or to quit Performance Toolkit mode for a few minutes so the new log message can be entered in standard console mode. You can switch from Performance Toolkit mode to CP mode by entering the command CP. Then, after entering the SET LOGMSG command and all the necessary data, you return to Performance Toolkit mode by keying in the CP BEGIN command.

**Do not use the PA1-key for switching temporarily to CP mode: although CP mode will successfully be entered, you will not be able to see any command output because all system output lines are still being converted to IUCV \*MSG service format.**

**VM READ:** Prompts generated by CMS programs and EXECs will be shown and the reply can be entered on Performance Toolkit's command line if the line is written by CMS I/O routines and when the reply is read by an EXEC or an assembler program using the CMS RDTERM macro. Performance Toolkit will indicate the next input is read from another program by replacing the arrow in front of the command line by the word 'ENTER''. Only explicit CP commands entered with the '#CP'' prefix can be executed under Performance Toolkit while the 'ENTER:' prompt is shown, all other input will be passed on to the CMS program.

To allow confidential data (passwords) to be entered in this way, a PF-key can be set for a special 'NODISP' function. When the PF-key is pressed, the command line is reset and the next input line will remain invisible while being entered. It will also not be redisplayed on the screen nor will it be written to the console log.

Attempts to read input from the console via other means may place the screen in VM READ status and are not recommended.

The restrictions described in this section apply only to commands run in the same machine that runs Performance Toolkit. When using Performance Toolkit as a secondary console for controlling another

machine, that is, when the commands that cause the CP READ and VM READ state are run in the other machine, there is no such restriction and you will not see the 'ENTER:' prompt.

## Restrictions for Operation in Basic Command Mode

CMS programs running under control of Performance Toolkit must not violate certain restrictions imposed by the manner in which Performance Toolkit operates. These restrictions include:

- Performance Toolkit needs the IUCV *MSG service interface for its own exclusive use
- Performance Toolkit intercepts all SVC, external, and attention interrupts from the console (via STAX macro)

While it will allow all standard CMS commands (except PROP) to be run, other CMS applications may not be able to operate normally if they violate one of the following restrictions:

- CMS programs which use IUCV *MSG service cannot run under Performance Toolkit.
  - Programs using other IUCV services can coexist with Performance Toolkit, but only if they use the HNDIUCV macro for working with IUCV.
- CMS programs must use the HNDEXT macro to trap external interrupts and must not change the CMS 'External New PSW'. The CPU timer must be left free for use by Performance Toolkit.
- Programs abending in the CMS machine where Performance Toolkit is active will also end Performance Toolkit operation.
- The BATLOAD flag in NUCON is set on by Performance Toolkit to allow interception of CMS console reads. This flag prevents proper running of the CMS LOAD command. While simple programs can usually be correctly loaded and run, the command is likely to result in an ABEND for more complex LOAD operations.
  - You can bypass this problem by running the LOAD command from within an EXEC procedure, and by resetting BATFLAGS in NUCON to X'00' just prior to the LOAD command. The flag will automatically be set on again after running of the EXEC procedure, but any CMS console reads run while the flag is reset will result in an empty screen in VM READ status.

Performance Toolkit is designed for simplifying system operation, not as a general full screen interface to CMS. Use the standard full screen mode of CMS for full screen operation in general purpose CMS machines.

When Performance Toolkit is used as secondary console for another virtual machine, there are no Performance Toolkit-imposed restrictions on programs running in the primary machine, which can then run any programs and operating systems supported by the SCIF facility.

## Redisplay Mode

You enter redisplay mode by pressing PF2 or entering 'REDISP' in basic mode.

## Screen Layout

Screen Identifier    File identification of log file
    |                       |

```
 FCX002                  Viewing Log File 'yyyymmdd CONLOG  A'        Redisp. Mode
                                                                Line nnnnn of nnnnn
 01                                                              A         A
 02                                                              |         |
 03 <----- Line numbers (optional)              Line number   ----\        |
 04                                             of top line                |
 05                                                                        |
 06                                             Nr. of lines  ------------\
 07                                             in CONLOG file
 ..
 ..
 ..       Output Area
 ..
 ..
 16
 17
 18
 19
 ...   message area ...
 Command ===> _     Command Line
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F10=Left  F11=Right  F12= Return
```

*Figure 29. Layout of Redisplay Screen*

PF-key settings for redisplay mode Initial PF-key setting in 'redisplay' mode:

**F1**
    switch to 'help' mode

**F4**
    display start of log

**F5**
    display end of log

**F7**
    scroll screen backward

**F8**
    scroll screen forward

**F10**
    display left side of log

**F11**
    display right side of log

**F12**
    return to 'basic' mode

Other PF-key numbers can be assigned to these functions by means of the FC PFKEY SET nn FUNCTION command (see the "FCONTROL PFKEY" subcommand in the *z/VM: Performance Toolkit Reference*).

## Working in Redisplay Mode

The initial screen shown in redisplay mode will display the last 'page' of the 'mmddyy CONLOG' file created by Performance Toolkit while operating in 'basic' mode. The remainder of the console log can be browsed using the following subcommands (see the "Redisplay Mode Subcommands" section in the *z/VM: Performance Toolkit Reference*):

    BACKWARD
    FORWARD
    TOP

```
    BOTTOM
```

You can also press the corresponding PF-keys (shown on bottom line of Redisplay screen, see ).

If only some specific incidents are of interest, not the complete log, use of the 'locate' function (see example below) may be a more efficient way for finding a certain area in the console log.

```
  /..text..
```

Pressing Enter repeatedly after entering a 'locate' command will rerun the command. Pressing Enter after entering any other redisplay subcommand will scroll the screen backward or forward, depending on which way it was scrolled last.

Refer to the "Redisplay Mode Subcommands" section in the *z/VM: Performance Toolkit Reference* for a detailed description of valid subcommands.

If you receive messages or action messages while the screen is in redisplay mode, the message

```
 'kkkk line(s) (nnnn message(s) / mmmm act. msgs) stacked'
```

will be inserted into the top line and inform you of the number of lines stacked (*kkkk* = total number of lines, *nnnn* = number of messages and *mmmm* = number of action messages that have been received).

**Scrolling into other days' logs:** When the end of the day's log is reached while scrolling forward or backward, scrolling will temporarily be suspended while either of the strings

```
  *** START OF LOG ***
   *** END OF LOG ***
```

is shown at the top or bottom of the screen to indicate no more lines are available for that specific day. Repeating the last scroll command will result in scrolling to continue into the previous or next day's log, provided that such a log can be found, which must still have the default file name format *yyyymmdd* and a filetype of CONLOG, either without suffix or with one of the suffixes 1, 2 or 3.

# Performance Toolkit HELP Facility

HELP operation is based on the standard CMS HELP facility. It provides

- A short introduction to each of the different operating modes
- Descriptions of all Performance Toolkit subcommands
- Descriptions of all Performance Toolkit messages
- A list of all performance monitor displays with the commands needed for selecting them
- Descriptions of all performance monitor fields

The descriptions of commands, messages, and the performance monitor fields contain the complete text which you can also find in this manual.

You will usually access the HELP text via the general PERFKIT HELP menu. In the case of messages or performance monitor field descriptions, a direct approach is also possible. Both ways are described in the following paragraphs.

## Using the General HELP Menu

You can invoke the initial menu from the Performance Toolkit operating mode in either of two ways:

- Pressing the PF1-key
- Entering the command HELP with or without further arguments

Pressing PF1, or entering the HELP command without other arguments, will cause the following initial HELP screen to be displayed.

```
 PERFKIT TASKS              Task Help Information                 line 1 of 27
 (c) Copyright IBM Corporation 1986, 2020

                          Performance Toolkit for z/VM

 Move the cursor to the selected item and press Enter or F1.


 Introduction  - General introduction for beginners
               - Use in basic command mode for system operation
               - Use in redisplay mode for re-viewing accumulated log
               - Use in perf. monitor mode (selecting performance screens)
               - Use in remote performance monitor mode
               - Use of VMCF data retrieval interface
               - Use of APPC/VM data retrieval interface
               - Use of the WWW data retrieval interface

 Sub-commands  - General (all modes)              (provides a detailed
               - Basic mode sub-commands           list of commands
               - Redisplay mode sub-commands       which are valid
               - Performance monitor commands      in the selected mode)

 Fields        - Selecting performance monitor field descriptions




 * * * End of File * * *
 PF1= Help      2= Top      3= Quit     4= Return     5= Clocate   6= ?
 PF7= Backward 8= Forward  9= PFkeys  10=            11=          12= Cursor

 ====>
                                                    Macro-read 1 File
```

*Figure 30. Layout of Initial HELP Screen*

This screen provides a list of specific areas for which HELP text is available. Place the cursor on the line which describes the area for which you need information, and then press PF1 or the Enter key. This will lead to one of the following:

- The actual HELP text
- To a more detailed list (for most of the items on the initial HELP screen) describing the HELP information which is available for the area selected on the previous HELP screen.

If another HELP menu is shown, continue with the selection of the particular area you need information for until the actual HELP text is displayed.

When found, you can scroll through the HELP information using the corresponding PF-keys or you can search for specific data with the Locate command

```
    /data
```

Use the QUIT command or the corresponding PF-key to return to previous HELP menus and eventually back to the Performance Toolkit operating mode where you called HELP.

## Direct Access to Specific HELP Text

HELP information can be accessed directly for the following by entering the HELP command with the appropriate argument(s).

- Performance Toolkit sub-commands
- messages
- performance monitor fields

**Displaying HELP Text for SubCommands:** Enter 'HELP' followed by the subcommand for which you need the HELP information. In the case of the FCONTROL subcommand you should also enter the specific FCONTROL argument you are interested in (for example, HELP  FC  PROCESS for displaying HELP text

for the FC PROCESS command). Otherwise a menu with a selection of all general Performance Toolkit subcommands will be shown.

If the same command can be entered in more than one mode, and if different HELP text is available for each mode, the HELP text corresponding to your current mode will usually be displayed. If the command is not valid in the current screen mode, the search order is:

1. Basic mode HELP text
2. Redisplay mode HELP text
3. Performance monitor mode HELP text

If you enter the command HELP  BACK in performance monitor mode, the HELP text shown will be for the BACKWARD command in performance monitor mode. If the same command is entered in basic mode (where the BACKWARD command is not valid), the HELP text for the BACKWARD command in redisplay mode will be shown instead.

Direct selection of the corresponding HELP text for performance monitor mode is not possible while you are in basic or redisplay mode. You can pass through the main HELP menu (by pressing PF1) and then select any command you want.

**Displaying Message Descriptions:** There are several different ways for displaying help text for messages:

- If you see an error message anywhere on the screen after entering a subcommand, such as

```
   FCXCMD218E Missing parameter
```

you can enter the command

```
   HELP MSG FCXCMD218E
```

to retrieve the HELP text for this message, that is, you enter the command 'HELP MSG' followed by the message prefix.

- If you want to see the description of the most recent message which appeared in the message area above the Performance Toolkit command line, you can omit the message prefix. A 'HELP MSG' command entered without message prefix while a Performance Toolkit message is being shown in the message area is assumed to pertain to that specific message, and the corresponding HELP text will automatically be selected.

- You can also move the cursor to the message you want the description for and press the PF1 key: This context sensitive help mode will generate a window with the message description text.

**Displaying Performance Monitor Field Descriptions:** Two different modes are available:

- Display of the complete description for all performance data fields on the selected display
- Context sensitive help text selection, for selecting the description for just one specific performance data field

**Display of the complete description for all fields:** HELP text with performance monitor field descriptions is selected by entering the command:

```
   HELP FIELDS
```

When entered in performance monitor mode, and from a performance monitor display for which a field description is available, the field description for that display will immediately be shown.

When entered in another mode, or from a performance monitor screen where no field description is available, a HELP menu will be shown instead with a list of all performance monitor displays for which HELP text can be shown. Place the cursor on the line which describes the monitor screen you need information for and press PF1 or the Enter key to display the HELP information.

**Context sensitive HELP text selection:** While you are viewing a performance data display, you can move the cursor to the heading of any of the performance fields on the screen and then press the HELP key

(usually PF1). For most performance screens a pop-up window will be shown with the help text for just the selected field.

Note the following:

- A pop-up window with the description of *all* fields on the selected performance screen will be shown if the cursor is positioned above the message line, but on a field for which no specific help information is available.
- A pop-up window with help text for more than one field (but still a subset of all fields on the screen) can be shown if such a description is thought to better explain the selected field.
- Selection of help text for a specific field is possible only on screens with a fixed layout. The full description for *all* fields will be shown instead for performance screens whose layout is dynamically adapted, depending on the data to be shown (for example, I/O device details, user details).

## Controlling the Scrolling Mechanism

The actions taken when the screen becomes full are different from standard VM console handling: instead of clearing the complete screen (automatically after 60 seconds in `'MORE ...'` status or by pressing the PA2 or CLEAR key) the screen contents are shifted upwards for a specified number of lines, but messages and the last action messages will remain on the screen.

Scrolling can be automatic after a preset time interval or it can be controlled manually. The scroll mode is set using the

```
FCONTROL SCROLL
```

command, and the currently set scroll mode is displayed in the top right hand corner of the display while in basic control mode.

## Automatic Scroll Mode

This is the initial default scroll mode. It is set using the command

```
FCONTROL SCROLL AUTO nn
```

where '*nn*' specifies by how many lines the screen contents should be shifted whenever the display becomes full.

The display lines will be shifted only after a certain time interval so the operator can lock the screen before it is scrolled when still interested in the current output. This interval can be set with the FCONTROL SCROLL WAIT nn command, where '*nn*' defines the delay in seconds from the moment when the last available display line has been filled until the screen contents are scrolled. During this time the message `'nn sec. scroll wait'` will be highlighted in the center of the top line. This is the equivalent of the `'MORE...'` status of standard VM console operation. Pressing the ENTER-key in this status will freeze the screen and cause the message `'Press "ENTER" to continue'` to be shown in the message area, the equivalent of the `'CONTINUE'` status on a VM console. The screen will be scrolled immediately when you press ENTER again.

## Manual Scroll Mode

You set manual scrolling with the command

```
FCONTROL SCROLL MANual nn
```

where '*nn*' is again the number of lines the screen contents should be shifted. Instead of being shifted automatically after a certain time interval the message

```
'Press "ENTER" to continue'
```

will appear immediately when the screen becomes full and the display will remain locked in this state until you press enter.

## Special Cases

There are some cases where scrolling works differently:

**Command entered:** When you enter a new CP or CMS command which can be run (that is, does not have to be stacked because there already is another active CMS command), the screen will be scrolled immediately, if necessary, so the command line and a maximum of output is visible on the screen. In other words, the output for the last command will be shown regardless of the number of other output lines that may still have been waiting to be displayed, and the command line will be moved up to the top of the display lines, if necessary, before the next scroll wait is activated.

The reason for designing the scrolling mechanism in this way was the assumption that you will probably enter a new command only after you are satisfied that the remaining output of the previous command has no more interest for you (useful if the last command was a 'NET Q ALL' or 'Q RDR ALL' that can fill many screens.).

Use the redisplay facility and browse through the 'old' output lines that were not shown on the screen if you still need them.

**Screen locked:** The default setting of the FCONTROL SCROLL MSGS command – used when FCONTROL SCROLL MSGS has not been issued – is FCONTROL SCROLL OFF, which disables the autoscrolling of the priority CP messages. If this setting is in effect and the screen fills up with CP messages, you might see this message in the center of the top line: `Screen locked - Delete some messages`. When this happens, you need to delete some messages so normal operation can continue.

You can use the FCONTROL SCROLL ON command to prevent this screen locking situation. FCONTROL SCROLL ON causes priority CP messages to be automatically shifted out of the screen when it becomes full.

For more information, see the "FCONTROL SCROLL" subcommand in the *z/VM: Performance Toolkit Reference*.

## Controlling Color and Highlighting

By default, Performance Toolkit will highlight all command lines entered at the bottom line of the screen. It will also highlight information which indicates unusual or non-standard conditions, such as:

- Pending scrolling actions
- Messages and action messages
- Redisplay or help mode active.

You can change this default highlighting and the color of any field on all displays which have extended color and/or extended highlighting support, and so take full advantage of the screen's facilities. On display terminals which do not support extended color or highlighting, you can still use 'high intensity' display for important data fields.

Both color and highlighting are controlled by means of the

```
FCONTROL COLOR ...
```

command (see the "FCONTROL COLOR" subcommand in the *z/VM: Performance Toolkit Reference* for details). Bear in mind that the objective is to assist operating and that 'eye-catcher' functions like reverse video or blinking are required to call the operator's attention to exceptions (messages and action messages). Overuse of these functions may detract from its use as an eye-catcher.

Once you have found suitable definitions, the necessary commands should be included in the initialization file 'FCONX $PROFILE'. The sample 'FCONX $PROFILE' delivered with the program already contains a set of suggested FCONTROL COLOR commands which you can adapt.

## Using Performance Toolkit as a Secondary Console

Reasons for using Performance Toolkit as a secondary console for another virtual machine could be:

- The machine to be controlled runs its own non-CMS operating system (for example, GCS for the RSCS machine and the VTAM® machine)
- The primary machine does use CMS but also uses another program that cannot be run under Performance Toolkit due to the restrictions mentioned in "Restrictions for Basic Command Mode" on page 6.

Any virtual machine can be defined as a secondary console for another virtual machine, and running Performance Toolkit in such a secondary machine presents no basic problems. However, there are a few things that might make operating a bit more complicated:

- Every command to the primary machine must be prefixed with the words 'SEND userid'
- Every line received from the primary machine is prefixed with the primary machine's user identification which is important information if the secondary machine is primarily used for other work, but definitely a nuisance if the machine has been set up only for operating the other system
- All lines are flagged as being originated by the Single Console Image Facility (SCIF), so that Performance Toolkit cannot easily tell whether a line should actually be handled as a message or action message.

The solution to the first two problems is the FCONTROL SECUSER userid subcommand (see the "FCONTROL SECUSER" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description). This command tells Performance Toolkit that we are primarily acting as secondary console for another machine, and once the command has been entered:

- The prefix 'SEND userid' will automatically be placed at the beginning of the command line (it can be overtyped or deleted for entering other commands)
- The prefix 'USERID :'' will be removed from all incoming message lines, thus giving better readability of the data received.

A good example for using this facility is the control of an RSCS machine via a secondary console with Performance Toolkit. This gives you the advantage of the automatic scrolling mechanism, and it allows simple browsing through the RSCS console log for answering the typical query to the networking machine 'what happened to my file xyz which I sent two hours ago?'.

See the example in "Special Line Processing" on page 88 which describes the use of Performance Toolkit for operating RSCS.

# Special Line Processing

Output lines will normally be displayed and scrolled according to the criteria described in previous sections, for example:

- CP message lines and warnings will not be scrolled
- Action messages will be scrolled only when their number exceeds the maximum number to be kept on the screen
- Output line color and highlighting depend on the line type and on the color and highlighting set with the FC COLOR command.

The following sections explain how to change this default line handling, if required.

## Handling of General Console Output Lines

All output lines can be selected according to user-defined criteria for special treatment. The lines can:

- Be re-defined as another line type. They will then be displayed with the new line type's color and highlighting definitions, and they will also use its scrolling characteristics (message lines and action message lines).
- Be rerouted to another virtual machine on the same VM system as CP message, warning, noheader message, or SMSG.

- Be suppressed for immediate display. The line will still be written to Performance Toolkit's console log with a greater than (>) flag, so it can always be redisplayed.

  Three different levels of suppression can be specified, and you can then select the class(es) that are still to be suppressed even when redisplaying such output. All of the originally suppressed output will be displayed in REDISP mode, by default.

- Cause an EXEC procedure or CMS module to be run, that is, the facility can be used for automating system operation. The output line can optionally be passed to the EXEC to allow further analysis of the line.

- Be modified to some extent. For example, leading text such as message headers can be removed from the received data string, and replaced by some other information.

All these functions are specified using the FCONTROL PROCESS ... command (see the "FCONTROL PROCESS" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description). Although some of its functions can, in fact, cause Performance Toolkit to perform actions in a way usually associated with the 'programmable operator' facility of z/VM, the purpose of the FCONTROL PROCESS command is not to replace existing PROP applications. Instead, its purpose is to extend Performance Toolkit's facilities in such a way that most requirements for operating a Performance Toolkit controlled virtual machine can be handled by a single tool.

Once defined, you can QUERY the PROCESS settings for a specific kind of output line, and you can also disable the selected settings.

## Example 1: Implementing a tape operator

In large installations the tape units and tape archive are usually located some distance away from the VM consoles. In many cases, especially where there is also a z/OS production system, there will even be a dedicated tape operator. In any case it is on a separate console near the tape units that we would like to receive all messages that are related to tapes and tape units.

Ways to handle this include:

- Make sure all 'TAPE' messages are re-routed from the system console to a new user ID 'TAPEOP' which we will set up. You do this by entering the commands

```
FC PROCESS CPMSG * 'TAPE' | * 'BAND' NODISP RER TAPEOP CPMSG
FC PROCESS CPAMSG * 'HCPERP2215A'    RER TAPEOP CPMSG
```

  This ensures that all messages which contain either of the words 'TAPE' or 'BAND' will be re-routed to user TAPEOP as a message ('BAND', in Germany, stands for the local name used for tapes). 'NODISP' means the message will not be shown on the VM master console, so you should use this option only if you are sure user TAPEOP is logged on.

  The second command also re-routes the 'INTERVENTION REQUIRED' messages for tape units to the tape operator, because the tape operator is the one who eventually will have to do something about such conditions. Both commands should be included in the FCONX $PROFILE of the system operator machine.

- Define a special machine TAPEOP for the tape operator and also run Performance Toolkit in it. Then remove the 'MSG FROM OPERATOR:' prefix from the messages to recreate the original message format for better readability. This is achieved by entering the command

```
FC PROCESS CPMSG * 'MSG FROM OPERATOR: ' REP ''
```

  The command should be included in the FCONX $PROFILE for the tape operator machine.

Note that Performance Toolkit cannot tell the difference between noheader messages and ordinary messages because the IUCV *MSG service does not provide that information, so any noheader messages will be shown as messages, and the normal CP message header will also be included.

You can use SMSG to achieve the same result as with MSGNOH by:

- Specifying 'CPSMSG' in the REROUTE option of the FC PROCESS command on the system console

- Including the CP command 'SET SMSG IUCV' in the FCONX $PROFILE so Performance Toolkit will use data passed in SMSG format
- Including the command 'FC PROCESS CPSMSG * * DISPLAY CPMSGN' in the FCONX $PROFILE to request display of SMSGs as messages without header.

## Example 2: Controlling an RSCS machine

Because RSCS is not a CMS application, there is no way to run Performance Toolkit in the RSCS machine, but we can activate it in a CMS machine that has been defined as secondary console for the disconnected RSCS machine and use the 'FC SECUSER userid' command to simplify operating (see "Using Performance Toolkit as a Secondary Console" on page 87).

Unfortunately, we now face the situation that *all* console output received from RSCS is just flagged as being originated by the Single Console Image Facility (SCIF), so that we can no longer tell the difference between RSCS and CP output and messages. But we can use the PROCESS command to re-establish the difference:

```
FC PROCESS CPSCIF 19 'DMT' DISPLAY VMOUT
FC PROCESS CPSCIF 19 'DMT' & 28 'E' DISPLAY CPEMSG
FC PROCESS CPSCIF 19 'MSG' DISPLAY CPMSG
```

This takes care of error messages and messages (the displacements to be entered refer to the start of the original line received. If you removed the 'userid:' prefix by entering the 'FCONTROL SECUSER' command, you have to add 10 to the displacement actually seen on the screen.).

In addition you might use the the different color and highlighting that can be assigned to other fields for displaying RSCS messages associated with certain links in different ways:

```
FC PROCESS CPSCIF * 'MVSLINK' DISPLAY CPIMSG
FC PROCESS CPSCIF * 'RPT21'   DISPLAY CPOUT
```

By letting these fields be displayed in different colors it is very easy to see which links are active, even without looking closely at the RSCS messages.

Note that the PROCESS definitions are run in the order in which they were entered. The commands defining messages and error messages should be run *after* the others if you expect error messages to be shown as error messages regardless of the link on which the error occurred.

## Example 3: Implementation of a logical operator

Systems working with PROP, the programmable operator facility of VM, usually have a 'logical operator' machine defined. This machine is set up to receive all messages (that need further action) from the PROP machine as well as informational messages. In large VM service centers with several VM systems these messages will be transferred via RSCS, resulting in a message format like:

```
hh:mm:ss MSG FROM RSCS: DMTRGX171I FROM node-ID(user-ID): actual text
```

You will see more message prolog than actual data, especially if RSCS does not use the MSGNOH format.

The REPLACE argument of the 'FC PROCESS ...' command can help you to achieve better readablility. Entering the command:

```
FC PROCESS CPMSG * 'ROM node-ID(user-ID)' REP 'PROFSA' DISP CPOUT
```

will cause the whole message prolog, up to and including the originator's address, to be removed and to be replaced by the string 'PROFSA', resulting in the following new message format:

```
PROFSA: actual text
```

which would be displayed with the color and scrolling attributes of ordinary CP command output. The example assumes the remote system is generally known under the name of 'PROFSA'. Similar commands for each of the other systems, using different nicknames and different output line types, will provide all the necessary originator information and remove any extraneous message headers for better readability.

# Example 4: VTAM command execution interface

Because it is not possible to logon to the VTAM machine from a VTAM controlled terminal for entering commands, one usually defines the OPERATOR machine as a secondary console for VTAM. Systems programmers with the necessary CP privilege class can then temporarily take over the console by means of the SET SECUSER command.

In many cases, however, one might wish to give some users the ability to run only a subset of VTAM commands without also giving them special CP privileges. Implementing a scenario similar to the following can bring this capability to selected end users:

1. Let VTAM commands be passed to the OPERATOR machine as CP messages, using a special keyword, such as 'VTAMCMD', for easy interception:

   ```
   MSG OPERATOR VTAMCMD d net,id=xxxxxxxx
   ```

2. In the OPERATOR machine, intercept all messages with the special VTAMCMD keyword and let the VTAMCMD EXEC be called for further analysis and actual command performance. To do this, include the following statement in the machine's FCONX $PROFILE:

   ```
   FC PROCESS CPMSG * 'VTAMCMD' DISP CPSMGN EXEC VTAMCMD PASSARG
   ```

   This ensures the message will be inserted into the basic mode screen as scrollable, and it passes the full message text to the VTAMCMD EXEC.

3. The VTAMCMD EXEC has to do all the rest:

   a. Validate the user's authorization for submitting specific commands

   b. Temporarily reroute all VTAM output to the calling user, using the command

      ```
      FC PROCESS CPSCIF 0 'VTAM ' RER userid CPMSGN
      ```

      The FCONCMD module is needed to run this subcommand from within a REXX™ procedure, and the return code from the module which corresponds to the number of the PROCESS entry generated (will be needed later for deleting the entry again) needs to be saved.

   c. Submit the VTAM command to the VTAM machine, using

      ```
      'SEND VTAM VTAM ...'
      ```

   d. Go to sleep for one second, in order to allow the VTAM machine sufficient time for generating its reply

   e. Do a CONWAIT to allow Performance Toolkit to receive any pending output

   f. Finally reset the rerouting by deleting the PROCESS entry created earlier, by means of the command

      ```
      FC PROCESS CPSCIF DELETE nn
      ```

      Note that this requires knowledge of the entry number, that is, the return code of the FCONCMD module which was saved in step b) above.

Any additional logic can be placed into the VTAMCMD procedure. The following listing shows a sample VTAMCMD EXEC for doing the above, which also contains some additional tests and comments. You can copy it and use it as a starting point, but **be sure to carefully evaluate and adapt the logic that does the actual VTAM command validation before activating it.**

```
/*------------------------------------------------------------------*/
/* Sample procedure for implementing VTAM command execution         */
/* by (selected) general users:                                     */
/* 1. Validate command; pass back error message if not allowed.     */
/* 2. Generate 'FC PROCESS' command for rerouting any output        */
/*    received from VTAM machine back to the calling user.          */
/* 3. Pass the VTAM command to the VTAM machine for execution.      */
/* 4. Wait for one second to allow for command execution time.      */
/* 5. Do a CONWAIT to make sure any pending output is received.     */
/* 6. De-activate the FC PROCESS entry set in step 2.               */
/*                                                                  */
/*------------------------------------------------------------------*/
```

```
parse arg . 'MSG FROM' sender':' vtamstr commandstr;
if vtamstr<>'VTAMCMD' then do;
   call sendmsg "Incorrect request format. Format must be:"
   call sendmsg "MSG OP VTAMCMD vtamcmd"
   exit 0;
end;

/* Validate command: Make sure user is authorized to execute     */
/* the command passed with the current CP message.               */
/* THIS IS JUST AN EXAMPLE: You'll have to adapt the code, and    */
/* you may want to add some more checking to allow adding further */
/* restrictions ...                                              */

parse upper var commandstr cmd argrest;

select;
   when sender = 'FCONMNT' then nop;    /* These userids belong to  */
   when sender = 'MAINT' then nop;      /* systems programmers who  */
   when sender = 'RSI' then nop;        /* can enter any command.   */
   otherwise if cmd<>'D' then do;       /* Anybody else is only     */
                                        /* allowed to enter 'display'*/
                                        /* commands, but we don't   */
                                        /* further check what kind  */
                                        /* of display.              */
      call sendmsg "You're only allowed to enter 'D' requests."
      call sendmsg "Command '"commandstr"' not passed on to VTAM."
      exit 0;
   end;
end;

/* Command found to be valid: Let any output from the VTAM machine  */
/* be returned to the user who submitted the command:              */

"FCONCMD FC PROCESS CPSCIF 0 'VTAM    :' RER" sender "CPMSGN";

select;                          /* Check return code for FCONCMD call: */
   when rc>0 then procno=rc;      /* Return code is PROCESS entry number. */
                                  /* Anything else indicates a problem:   */
   when rc=0 then do;             /* Must have old FCONCMD/FCONX modules: */
      say "Working with old FCONX/FCONCMD modules;"; /* Tell user.   */
      exit 99;                    /* Cannot proceed: Exit              */
   end;
   otherwise do;                  /* Rc<0: FCONCMD missing or native call */
      'STATE FCONCMD MODULE *'    /* FCONCMD module available?         */
      if rc<>0 then do;           /* No, not found; tell user..        */
         say "FCONCMD MODULE not available; please obtain latest version.";
         exit 99;                 /* Cannot proceed: Exit              */
      end;
      else do;                    /* Must be running natively, otherwise..*/
         exit 99;                 /* Cannot proceed: Exit              */
      end;
   end;
end;

"SEND VTAM VTAM" cmd argrest;     /* Submit the command              */

/* We also have to reset the FC PROCESS entry built above:         */
/* - First wait a bit to allow the VTAM machine to execute the cmd. */
/* - Then call CONWAIT to make sure any pending output is received  */
/* - Finally reset the FC PROCESS entry and exit.                  */

"SLEEP 1 SEC";                    /* Wait time may have to be adapted,  */
                                  /* depending on VTAM responsiveness.  */
"CONWAIT";
"FCONCMD FC PROCESS CPSCIF DELETE" procno;

exit 0;

/* Use Diag. 8 to generate mixed case text with CP MSGNOH command  */
sendmsg:
   parse arg msgstring;
   diag(08, "MSGNOH" sender msgstring );
return;
```

# FCONX $PROFILE

The 'FCONTROL ...' subcommand with its different arguments controls many general functions like the assignment of PF-keys to the performance of certain functions, the operation of the scrolling mechanism,

color definitions, or special processing to be performed for specific lines. These subcommands, and other commands, are used for adapting Performance Toolkit operation to a specific environment. Because this environment is always the same, the same commands would always be used for tailoring Performance Toolkit. In order to simplify the start-up process you can add all these 'customizing' commands to the initialization file 'FCONX $PROFILE'. Performance Toolkit will read the first initialization profile it finds on any accessed disk and automatically run the subcommands included whenever it is started.

## Basic Processing

The initialization file will be processed as follows:

- Lines starting with an asterisk (*) in column one will be ignored. You can use the asterisk to include comments. Comment lines starting with double asterisks will be included in the machine's CONLOG file, as will be all of the actual initialization commands.

- 'FCONTROL ...' commands will be run immediately during initialization.

  If an error is detected while trying to run such a command, the relevant error message will be shown on the basic mode screen, followed by another line which contains the record number of the incorrect statement and the failing command itself as follows:

  ```
  (nnnn: 'failing statement')
  ```

  This line is displayed with the VMOUT attributes, that is, like a CMS output line.

- Any other commands (other Performance Toolkit subcommands, CP commands, or CMS commands) found in the initialization file will be stacked during the initialization phase. They will be run in the order in which they have been found after Performance Toolkit has been completely initialized.

Make sure the commands are ordered so their sequential running makes sense. Be careful when including commands which cannot be run in basic mode, and make sure to include all commands to switch, for instance, to a specific performance monitor display and back.

The display will be updated for the first time in either of the following instances:

- When all stacked commands have been run
- Before the first command is passed to CMS for running

In this example, the following commands will change the initial I/O device sort sequence on the general I/O display so devices are inserted depending on the device busy percentage (the initial sort sequence is for ascending device numbers), and users will be sorted according to their CPU utilization on the general user display.

```
MONITOR
  SORT DEV    BUSY
  SORT USER   %CPU
QUIT
```

A sample 'FCONX $PROFILE' is supplied with the program and can serve as a starting point for preparing your own version. It is also included in this manual as Appendix A, "A Sample Initialization Profile (FCONX $PROFILE)," on page 173.

## Extended Input File Processing

In large centers where many different VM systems have to be centrally managed, a requirement exists for separating generic and individual definitions, and for automatically selecting the correct set of definitions for a system even though the same set of files may be distributed to all systems. Inclusion of additional control information allows specifying that a statement should be run only on a specific system, or a group of systems, and it also allows the imbedding of additional files. Such extended processing is currently supported for the following files:

- FCONX $PROFILE
- FCONX TRENDREC

- FCONRMT AUTHORIZ
- FCONX SUMREC

The special control strings are:

- `:SYSID:`

  which will be recognized in any context, in any record of the file, and will be translated to the system identification as defined in the system's configuration file before being passed on for processing.

- `::IMBED::` *fn ft {fm}*

  indicating the specified file is to imbedded, that is, processed before continuing reading the current file, where

  **fn**
  >  is the file name of the imbedded file

  **ft**
  >  is the filetype of the imbedded file

  **fm**
  >  is the optional filemode of the imbedded file

  The file name and/or filetype can also be specified as the string `:SYSID:` in which case it will be replaced by the system identification of the VM system.

- `::GRPDEF::` *grpname* {`::EXCL::`} *sysid1 ... sysidn*

  indicating a group of systems is to be defined, where

  **grpname**
  >  is the group's name

  **`::EXCL::`**
  >  indicates the listed systems are to be excluded from the group (all other systems will automatically be assumed to be included).
  >
  >  When specified, the argument must directly follow the group name; it must precede all system identifications.

  **sysid1 ...**
  >  are the system identifications that are to become part of the group, or, if an `::EXCL::` argument was included just after the group name, systems that are to be excluded from the group.

- *name*: *any-specification*

  This format indicates the record is to be processed only for a specific system (if *name* is a valid system identification), or for the group of systems defined with group name *name*.

  The string *any-specification* can be a simple Performance Toolkit specification (for example, an FCONTROL subcommand for the initialization file, or an authorization record for file FCONRMT AUTHORIZ), or an `::IMBED::` specification, instructing the input processor to imbed a certain file only for a specific or group of system(s).

Example: the following statements

```
 :OURVM1: FC LIMIT %SPSL 80 0/0 0/10

 :OURVM2: ::IMBED:: OURVM2 $PROFILE *
```

will set the threshold for the 'spool slots in use' percentage only for system OURVM1, and will imbed the additional initialization statements contained in file OURVM2 $PROFILE on system OURVM2 only.

## Handling of Performance Toolkit Console Logs

Performance Toolkit initializes a new console log for each day and keeps three generations of these logs. The file type is

CONLOG  for the current log
CONLOG1 for the previous log
CONLOG2 for the previous log minus one

All console logs have as file name the creation date of the log in the format '*yyyymmdd*' (*yyyy*=year, *mm*=month, *dd*=day), that is, the Performance Toolkit log file created on the 6th of July, 2003, would have the file identification '20030706 CONLOG A1'.

The file type is automatically renamed whenever a new log has to be opened because the date has changed, and the oldest file (with file type CONLOG2) is deleted.

All console logs, even the old ones, can be browsed in redisplay mode. If you want to keep more than the default number of log file generations, you should regularly rename the CONLOG2 file to CONLOG3. CONLOG3 files will not be renamed or erased by Performance Toolkit, but CONLOG3 is still one of the default file types that will automatically be searched for. If other file types are used, they must be explicitly specified when entering redisplay mode.

Note that the program will attempt to switch automatically to the previous or next day's log if you attempt to scroll beyond the top or bottom of a specific log's data. This is possible, however, only for:

- Logs with the original default file name format *yyyymmdd*
- Logs with one of the file types CONLOG, CONLOG1, CONLOG2 or CONLOG3

Please keep this in mind when renaming your console logs.

## Archiving Files

An exit has been coded in Performance Toolkit to call procedure FCXRENAM EXEC on any accessed disk immediately *after* doing its own renaming. You can use it for additional housekeeping. The following sample procedure is the minimum you could use for copying the oldest console log file and also the oldest detailed performance history file (of file type HISTLOG2) to another disk for later reference:

```
/* Sample procedure for saving CONLOG2 and HISTLOG2 files     */

   'ACC 195 Z'
   'COPY * CONLOG2 A  = CONLOG3  (PACK'
   'COPY * HISTLOG2 A = HISTLOG3 (PACK'
   'REL Z'
```

(You may want to use more elaborate code to erase saved files on the archive disk and to handle error conditions.)

Note the following:

- Packed files will need to be unpacked manually before they can be processed again as input files by Performance Toolkit.
- The restrictions documented above for the renaming of console logs also apply to all of the many different kinds of performance history files that allow reprocessing. Wherever a generation number is appended to a file type, you may change this generation number, but you should not alter the fixed part of the file type. When working with performance history files, the file type is assumed to indicate a specific record format, so unknown file types will not be recognized as valid history data files.

The advantage of using Performance Toolkit redisplay for looking at old log files instead of the BROWSE or XEDIT command is that Performance Toolkit will take the color and highlighting definitions as specified by the FCONTROL COLOR command when displaying the log, while BROWSE or XEDIT cannot use these definitions but will find some non-displayable characters instead at the beginning of each record.

## Co-existence of Performance Toolkit and PROP

Both PROP and Performance Toolkit use the IUCV *MSG service for trapping console output and they can, therefore, not be active in the same virtual machine.

Performance Toolkit's FCONTROL PROCESS command is designed to allow automation of certain operator activities by running pre-defined commands when specific messages are received. This function thus allows implementing automated actions similar to PROP, but it is not intended to replace PROP or existing PROP applications, especially if these applications involve the handling of messages generated on other (remote) VM systems.

Once you have decided to use Performance Toolkit for operating the system, there are basically three ways open to you for handling existing PROP applications:

1. Leave the PROP machine unmodified and just run Performance Toolkit in the 'logical operator' machine. This will enable you to scroll and redisplay the PROP generated messages together with your own commands and output, and it will require no additional work.

    Because scrolling becomes very easy now, you might decide to change the PROP specifications so you receive additional 'nice-to-have' information that had previously been suppressed. Such general information should be scrollable by Performance Toolkit: because the IUCV *MSG service doesn't distinguish between normal messages and noheader messages, you may want to remove the message headers, and make the messages scrollable by letting them be displayed as something else (for example, CP output), using the FC PROCESS command with the 'REPLACE' argument. Use caution when using this option and make sure you do not cause genuine messages to be scrolled as well.

2. If you used PROP only for a few simple applications, you may decide to handle these with the PROCESS function of Performance Toolkit and to de-activate PROP. Performance Toolkit can then be run directly in the system operator machine.

3. You could also run Performance Toolkit in the system operator machine previously used for PROP. It will then receive all system messages and you could pass a selection of these messages on to another machine, using Performance Toolkit's PROCESS function with the REROUTE parameter. PROP could then be run in this second machine and do any additional processing you do not want to (or cannot) handle directly in Performance Toolkit.

    Note that Performance Toolkit can re-route system messages only in MESSAGE, MSGNOH, WARNING, and SMSG format and that the PROP routing table will have to be adapted accordingly.

Choose your solution based on the importance and complexity of the current PROP applications.

# Error Handling

Any ABEND occurring in the virtual machine where Performance Toolkit is running could possibly impair its further operation and the machine should therefore be re-initialized by IPLing ZCMS before Performance Toolkit is restarted.

**Note:** Before restarting Performance Toolkit, first log off and then log back on the PERFSVM user ID to ensure a clean environment.

Performance Toolkit intercepts ABENDs using the CMS ABNEXIT macro and will then:

- Automatically spool a DUMP file to the virtual reader
- Reset the IUCV interface
- Pass control back to DMSABN, so normal ABEND analysis with the CMS DEBUG facility can be done

Where this is not possible because the ABEND could not be intercepted for some reason, or if the problem is a loop or wait state, you should perform the necessary functions of documenting the error and re-initializing z/CMS by entering the following commands:

```
PA1                                 (to place the machine in CP MODE)
SET RUN OFF
SET CPCONIO OFF                     (partial reset of IUCV interface)
SET EMSG ON                         (so that you can see messages)
SP  PRT *                           (to create the DUMP file)
VMDUMP 0-END DCSS
IPL ZCMS                             (to reinitialize the machine)
```

An IPL of z/CMS completely resets the IUCV *MSG service interface.

Save the CP and Performance Toolkit console log and check it for the ABEND reason. Make sure there was no abending CMS program under Performance Toolkit. If the message in the CP console log indicates a Performance Toolkit problem, contact IBM using your z/VM support channel.

It may be helpful to supply other information to help locate the problem. For example:

- Command sequence which led to the error
- Whether the problem be reproduced
- If the problem was first found after specific software or hardware changes
- The current software level of CP, CMS, and Performance Toolkit.

# Emergency Safeguard Feature

Performance Toolkit's emergency safeguard feature (ESF), when enabled, will automatically save any unsaved data collected during normal monitoring in the event of an emergency, such as a z/VM shutdown or a sudden shortage in virtual memory. Performance Toolkit will also call any user-defined EXEC procedures you've specified when an emergency event occurs. To enable the emergency safeguard feature, to specify which actions to take in the event of an emergency (and in which order), and to specify the additional EXEC procedures to be called, use the FC EMERGENC command, as described in the *z/VM: Performance Toolkit Reference*.

Data that can be saved includes printed reports (produced via PRINT/INTERIM resets or by PRINTRMT commands scheduled by FC SETEVENT), as well as trend and summary files. The data will either be saved to local files or printed to local or remote nodes, as appropriate, so that it will be available upon system restart. If ESF is enabled and the appropriate actions have been defined, you can stop Performance Toolkit monitoring at any time by issuing a CP SIGNAL SHUTDOWN command to the Performance Toolkit virtual machine, without losing any gathered data.

By specifying the user-defined EXEC procedures, you can provide a common set of system-wide tasks that will be performed at the time of a z/VM shutdown. You can send alerts, for instance, via CP MSG or SMSG to other virtual machines, notifying them of the shutdown so that they can take appropriate actions such as closing all work.

# Chapter 5. Operation in Performance Monitor Mode

This chapter will guide you in using the performance monitor mode of Performance Toolkit. Command privilege class 'E', and a connection to the *MONITOR IUCV service, are prerequisites for using the full capabilities of the performance monitor.

The examples used may or may not be similar to what is actually displayed on your system.

## Displaying Performance Data

Performance monitor mode is activated by entering the MONITOR subcommand in basic mode (see the MONITOR subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description). The following performance data selection menu is shown initially:

```
FCX124                  Performance Screen Selection  (FL720       )    Perf. Monitor

General System Data        I/O Data                    History Data (by Time)
1. CPU data menu*          11. Channel load            31. Graphics selection
2. Storage utilization     12. Control units           32. History data files*
3. SSI data menu*          13. I/O device menu*        33. Benchmark displays*
4. Priv. operations        14. PCI Function menu*      34. Correlation coeff.
5. System counters         15. Cache extend. func.*    35. System summary*
6. CP IUCV services        16. Reserved                36. Auxiliary storage
7. SPOOL file display*     17. DASD seek distance*     37. CP communications*
8. LPAR data menu*         18. I/O prior. queueing*    38. DASD load
9. Shared segments         19. I/O configuration       39. Minidisk cache*
A. Shared data spaces      1A. I/O config. changes     3A. Storage mgmt. data*
B. Virt. disks in stor.                                3B. Proc. load & config*
C. Transact. statistics    User Data                   3C. LPAR logs menu*
D. Monitor data            21. User resource usage*    3D. Response time (all)*
E. Monitor settings        22. User paging menu*       3E. RSK data menu*
F. System settings         23. User wait states*       3F. Scheduler queues
G. System configuration    24. User response time*     3G. Scheduler data
H. VM Resource Manager     25. Resources/transact.*    3H. SFS/BFS logs menu*
                           26. User communication*     3I. System log
I. Exceptions              27. Multitasking users*     3K. TCP/IP data menu*
                           28. User configuration*     3L. User communication
K. User defined data*      29. Linux systems*          3M. User wait states
                           2A. CPU Pools menu*
                           2B. User Processor Time


         Pointers to related or more detailed performance data
         can be found on displays marked with an asterisk (*).




 Command ===> _
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 31. Performance Data Selection Menu*

The performance data selection menu gives an overview of the main performance data displays. You can select one of them by entering the appropriate number, directly entering a subcommand, or by placing the cursor on the desired selection and pressing Enter.

Use Quit or the appropriate function key to return to previously-displayed screens.

Note, however, before these displays will be shown, that most of the displays depend on CP monitor data. The relevant data domains must have been enabled, the monitor started, and monitor data by Performance Toolkit activated.

Performance screens whose selection numbers are not highlighted on the selection menu cannot be selected because the prerequisite monitor data is not available.

**Note:** The Exceptions report selection will be highlighted when an exception from the current session occurs. The Exceptions report might still be selectable if there is only data from previous sessions available.

The following is a short summary describing the type of information shown on each of the available displays and the commands used to select them.

*General System Data:*

**CPU data menu**
CPU activity reports.

Command: CPUMenu

The following CPU activity reports can be selected:

```
CPU        CPU Load and Transactions
DSVBKact   Dispatch Vector Activity
LOCKact    Spin Lock Collision Activity
MTCORElg   SMT Metrics Per Core Log
MTCTYPlg   SMT Metrics Per Core Type Log
```

**Storage utilization**
An overview of current storage utilization, including information about expanded storage and minidisk cache usage.

Command: STOrage

**SSI data menu**
SSI and ISFC performance reports.

Command: SSIMenu

**Priv. operations**
The number and rate of privileged operation simulations and execution of virtual diagnose instructions. Intended for experienced performance analysts.

Command: PRIVop

**System counters**
A number of important system count fields, with information about the system's management of storage, paging, and the minidisk cache. Intended for experienced performance analysts.

Command: SYStem

**CP IUCV Services**
The number and rate of IUCV CP service calls, and VMCF communications.

Command: CPServ

**SPOOL file display**
An overview of current SPOOL utilization (SPOOL space used by the 100 largest files, or by each user).

Commands: SPOOL, SPOOL User

**LPAR data menu**
Reports about the logical partitions defined for the processor complex.

Command: LPARMenu

The following logical partition reports can be selected:

```
LPAr       LPAR Data
LSHaract   Logical Partition Share
```

**Shared segments**
Information about named saved systems and discontiguous saved segments.

Command: NSS

**Shared data spaces**
Information about the paging activity of shared data spaces.

Command: DSPACESh

**Virt. disks in stor.**
Information about the use of VDISKs (virtual I/O activity, data space information).

Command: VDIsks

**Transact. statistics**
System-wide transaction statistics (response time, resources used).

Command: SYSTRans

**Monitor data**
A summary of monitor data records.

Command: MONData

**Monitor settings**
Initial monitor settings and MONITOR commands entered that affect these settings.

Command: MONSet

**System settings**
Initial SRM settings and later changes of these settings.

Command: SYSSet

**System configuration**
System configuration data.

Command: SYSConf

**VM Resource Manager**
VM Resource Manager data.

Command: VMRM

**Exceptions**
The accumulated exception messages log.

Command: EXCept

**User defined data**
User-defined performance displays.

Command: UDEFMenu

*I/O Data:*

**Channel load**
Channel load data.

Command: CHannel

**Control units**
Overall load and efficiency of cached control units.

Command: CTLunit

**I/O device menu**
I/O device reports.

Command: DEVMenu

The following I/O device reports can be selected:

```
VOLUME          DASD Volume Performance
DEVICE          I/O device load
DEVICE DASD     General DASD load and performance
DEVICE CPOWNED   CPOWNED DASD load and performance
DEVICE HPF      HPF DASD load and performance
SCSI            SCSI EDEV load and performance
HPALIAS         HyperPAV Alias Activity
HPSHARE         HyperPAV Share Activity
```

**PCI Function menu**

PCI function performance reports.

Command: PCIMenu

The following PCI function reports can be selected:

```
PCIConf    PCI function configuration
PCIAct     PCI function activity, Format 0
PCIAct ROCE   PCI function activity, Format 1
PCIAct ZEDC   PCI function activity, Format 2
PCIAct ISM    PCI function activity, Format 3
```

**Cache extend. func.**

Information about the load and efficiency of extended function caching for individual disks.

Command: CACHExt

**DASD seek distance**

The distribution of seek distance per disk (the number of cylinders skipped per seek).

Command: SEEKdist

**I/O prior. queueing**

System and user I/O priority queueing settings.

Command: IOPQUEUE

**I/O configuration**

I/O device configuration data (status, control units, and available paths).

Command: DEVConf

**I/O config. changes**

Changes to the initial I/O configuration and throttle settings.

Command: IOChange

*User Data:*

**User resource usage**

General information about resource usage by individual users.

Command: USEr

**User paging menu**

User paging reports.

Command: UPGMenu

The following user paging reports can be selected:

```
UPAge    User paging load
UPGAct   User page activity
UPGUtl   User page utilization
```

**User wait states**

General user status (user wait states).

Command: USTat

**User response time**
Information about users' transaction rates and response times.

Command: UREsp

**Resources/transact.**
Resources used per transaction (CPU, I/O, and spooling, for example).

Command: UTRans

**User communication**
Information about users' IUCV and VMCF communication activity.

Command: UCOmm

**Multitasking users**
Multitasking activity data by user.

Command: MTUsers

**User configuration**
Information about users' virtual machine configurations.

Command: UCONf

**Linux systems**
Allows the selecting of Linux internal performance data based on detailed RMF PM Linux data gatherer information or on summary CP monitor Linux APPLDATA.

Command: LINUXMen

The following Linux data displays can be selected:

- RMF PM Linux systems selection menu, allowing selection of specific Linux systems for displaying detailed Linux internal performance information based on RMF PM Linux data gatherer.

  Command: LINux

The following system-specific Linux performance data displays are available:

```
LXCPU       Linux CPU utilization
LXMEM       Linux memory utilization and activity
LXNETwrk    Linux network activity
LXFILsys    Linux filesystem usage
```

**CPU Pools menu**
CPU pool activity reports.

Command: CPLMenu

The following CPU pool activity reports can be selected:

```
CPLConf     CPU pool configuration
CPLAct      CPU pool activity
```

**User Processor Time**
User processor time reports.

Command: USRPrctm

*History Data (by Time):*

**Graphics selection**
Helps you select graphics displays based on the system overview history data, in storage or from disk files (graphics can also be selected directly by command). The following types of graphic displays are available:

- Detailed history graphics (periods up to 2 hours).

  Command: GRAPHdet

- A summary of history graphics (periods up to 12 hours).

  Command: GRAPHSum

- Trend graphics for periods of several months or years, based on history files ACUM HISTSUM (new) or ACUM PERFHIST (old).

  Command: GRAPHSum

- A performance profile that shows load variations during an average day, based on history files ACUM HISTSUM (new) or ACUM PERFHIST (old).

  Command: GRAPHSum

- Correlation graphics that show how selected variables depend on another performance variable.

  Command: GRAPHVar

**History data files**

All Performance Toolkit history data files on any accessed minidisk.

Command: HISTdata

**Benchmark displays**

Benchmark logs of all currently-benchmarked objects, such as users and I/O devices.

Command: BENchmrk

The following benchmark log screens can be created and selected when benchmarking has been activated:

**For I/O devices:**

```
CACHDlog    Control unit cache log (per device)
CACHELog    Cache extended function log (per device)
CPOWNLog    CPOWNED disk log (per device)
DEVLog      General I/O device performance log (per device)
HPFLOG      HPF I/O device activity log (per device)
QDIOLog     QDIO device activity log (per device)
QEBSMLog    QEBSM device activity log (per device)
SCSILog     SCSI device activity log (per device)
SEEKDLog    Seeks distances log (per device)
SEEKLLog    Seeks locations log (per device)
VOLLog      DASD volume performance log (per device)
```

**For users:**

```
LXCPULog    Summary Linux CPU data log (per user)
LXMEMLog    Summary Linux memory utilization log (per user)
LXNETLog    Summary Linux network activity log (per user)
MTUSRLog    User multitasking log (per user)
UCOMMLog    User communication log (per user)
UPAGELog    User paging activity log (per user)
UPGACTLg    User page activity log (per user)
UPGUTLLg    User page utilization log (per user)
UQDIOLog    User QDIO activity log (per user)
URESPLog    User response time log (per user)
USERLog     User resource utilization log (per user)
USRMPLog    Multiprocessor user activity log (per user)
USRTMLOG    User processor time log (per user)
USTATLog    User wait states log (per user)
USTMPLog    Multiprocessor user wait states log (per user)
UTRANLog    User resources per transaction log (per user)
```

**For logical links:**

```
ISFLLog     ISFC logical link activity log (per partner/node)
```

**For endpoints:**

```
ISFELog     ISFC end point activity log (per ISFC endpoint)
```

**For PCI functions:**

```
PCILog      PCI function activity log, Format 0, Format 1, Format 2, and Format 3
            (per PCI function)
```

**Correlation coeff.**
Correlation coefficients for all variables in the HISTLOG and HISTSUM files, based on a selectable base variable.

Command: CORrel

**System summary**
A condensed summary of processor, I/O, paging, and user activity and performance.

Command: SYSSUmlg

**Auxiliary storage**
Information about auxiliary storage utilization.

Command: AUXlog

**CP communications**
Information about CP IUCV communication service, for all types of data transfers.

Command: CPSAll

**DASD load**
Overall DASD performance data by time for the entire system.

Command: DASDlog

**Minidisk cache**
Data related to minidisk cache utilization and activity.

Command: MDCache

**Storage mgmt. data**
Storage management logs.

Command: STORMenu

The following storage management logs can be selected:

```
AVLB2glg   Below 2G Available list management log
AVLA2glg   Above 2G Available list management log
STEALlog   Frame steal details log
AGELlog    Age List details log
AVAILlog   Available list management log
DEMNDlog   Demand scan details log
PAGelog    Paging activity log
STORLog    Storage utilization log
PINLog     Pinned storage statistics log
SUBPlog    Subpool storage usage log
SXSAvail   System execution space available page queues management log
SXSDefer   System execution space deferred tasks management log
SXSPage    System execution space page management log
SXSUtil    System execution space storage utilization log
```

**Proc. load & config**
Processor load and configuration logs.

Command: PROCmenu

The following processor-related logs can be selected:

```
PROCSum     Processor summary log by time
PRCLog,     Processor load by processor and by time
PROCLog
IOPRoclg    I/O processor activity by processor and by time
DSVSLog     DSVBK Steals per logical CPU by time
LPARConf    Logical processor configuration by time
PROCConf    Real and virtual CPU configuration by time
TOPOlog     System topology machine organization event log
PUOrglog    Logical Core Organization by time
PUCfglog    Logical Core Configuration event log
DSVClog     Dispatch Vector Configuration log
LOCKAct,    Processor spin locks by time
LOCKLog
```

**Notes:**

- The PROCLog subcommand supports the collecting of monitor data when running Performance Toolkit against z/VM 6.2 or earlier data. When running Performance Toolkit against z/VM 6.3 or later data, the PROCLog subcommand is available for compatibility purposes only. Use the PRCLog subcommand instead.

- The LOCKLog subcommand supports the collecting of monitor data when running Performance Toolkit against z/VM 6.3 or earlier data. When running Performance Toolkit against z/VM 6.4 data, the PROCLog subcommand is not available. Use the LOCKAct subcommand instead.

**LPAR logs menu**

Logical partition logs.

Command: LPARLOGM

The following logical partition logs can be selected:

```
LPARLog    LPAR load log
PHYSLog    Real core utilization by time
```

**Response time (all)**

Response time and transaction data for UP and MP users.

Command: RESPall

**RSK data menu**

Reusable server kernel performance data.

Command: RSKmenu

The following reusable server kernel performance displays can be selected:

```
RSKDET     Reusable Server Kernel performance details
RSKDISK    Reusable Server Kernel disk activity log
RSKLOG     General RSK activity log
RSKSTOR    RSK storage utilization log
```

**Scheduler queues**

Information about scheduler queues.

Command: SCHedlog

**Scheduler data**

Data fields used by scheduling algorithms.

Command: SCHData

**SFS/BFS logs menu**

Shared file system and byte file system performance data.

Command: SFSMenu

The following shared file system and byte file system logs can be selected:

```
SFSIolog   Shared file system I/O activity log
SFSLog     General shared file system performance log
SFSReq     Shared file system requests log
BFSFiles   Byte file system file activity log
BFSLocks   Byte file system locking activity log
BFSPipes   BFS pipes and linking activity log
```

**System log**

System facilities activity data.

Command: SYSLog

**TCP/IP data menu**

TCP/IP-related data.

Command: TCPMenu

**General data:**

```
HIPSock    Hipersocket activity display
QDIO       QDIO device activity
VNIC       Virtual Network Device activity
VSWITch    Virtual Switch activity display
```

The following series of TCP/IP displays can be selected for servers that supplied APPLDATA monitor records:

```
TCPActlg   General TCP/IP activity log
TCPBplog   TCP/IP buffer pools log
TCPConf    TCP/IP server configuration
TCPDatlg   General TCP/IP data transfer log
TCPDOslg   TCP/IP denial of service log
TCPIcmp    TCP/IP ICMP messages log
TCPIOlog   TCP/IP I/O activity log
TCPLinks   TCP/IP links activity log
TCPSess    TCP/IP TCP and UDP sessions log
TCPUsers   TCP/IP users activity log
TFTPLOG    TCP/IP TFTP activity log
```

**User communication**

Overall user communication data (IUCV and VMCF) for the entire system.

Command: UCOMMLog

**User wait states**

Overall user wait state information for the system.

Command: USTLog

See the "Performance Toolkit Subcommands and Screen Layouts" topic in the *z/VM: Performance Toolkit Reference* for a complete description of all of the different performance displays and the commands that are required for activating them. After you have selected the desired display, you might want to:

- Sort the I/O devices or users in a different sequence. See "Sorting" on page 108 for more information.
- Shift the selected output in order to view another area. You would usually do this by pressing the appropriate function key (the key numbers and functions are displayed at the bottom of all performance monitor displays) or by entering the equivalent command.
- Locate a specific character string (disk label or user ID, for example). Use the locate function to do this:

```
/xxxx
```

See the "Performance Toolkit Subcommands and Screen Layouts" topic in the *z/VM: Performance Toolkit Reference* for a description of these commands.

CP and CMS commands are accepted only when entered with the CP or CMS prefix while you are in performance monitor mode. The display will then be switched back to basic mode.

## Displaying Current, Interim or Average Data

Current data, that is, the values calculated for the last measuring interval, will be shown initially for most of the displays that do not show 'by time' data.

Average data for two more periods can usually be selected as follows:

- Use the AVERAGE command to cause calculation of average data for the overall reporting period that is also used for automated printing. Most of the performance values shown will then be averages for the whole period since the last 'PRINT' reset.
- Use the INTERIM command to cause calculation of average data for the current 'interim' period, i.e. the period since the last 'INTERIM' reset.

See <u>"Resetting Performance Data/Defining Periods" on page 121</u> for a detailed explanation of the different periods, and how to set them.

Note that in both the AVERAGE and INTERIM cases some fields may still show current information if calculating averages for them is not appropriate. More information on the operation of the AVERAGE and INTERIM commands for a specific performance display is given at the end of each display's description in the "Performance Toolkit Subcommands and Screen Layouts" chapter in the _z/VM: Performance Toolkit Reference_.

The CURRENT command will cause the initial display mode of current data to be resumed.

The switch between the display of average, interim and current data will occur immediately on all screens that are based on the analysis of CP monitor data.

The time interval for which data have been calculated will always be inserted in the top line.

Exceptions: The AVERAGE, INTERIM and CURRENT commands have no meaning for the 'history data' log displays, and only current data can be shown for the detailed I/O device and user displays.

## Sorting

Many performance displays show a number of rows with similar layout, but for different entities (I/O devices and users, for example). Most of these displays have been designed to allow sorting of the rows according to different criteria. The performance fields which can be sorted are then marked with a dot above the heading with the description for that column, and the column whose data were used as the base for the current sort is designated by a number of underscore characters above the heading line.

To select a different performance field for sorting, do one of the following:

- Use the SORT subcommand, specifying the field name from the (lowest) heading line as a sort argument. (See the SORT subcommand in the _z/VM: Performance Toolkit Reference_ for a detailed description.)
- Move the cursor to the description of the field in one of the heading lines or to the sort indicator and then press the ENTER key.

The screen will be re-sorted immediately if it is based on CP monitor data, it will be sorted at the end of the current data collection interval if it is based on data extracted from CP control blocks using Diagnose X'04'.

## Saving Performance Data

The PERFKIT performance monitor is a real time monitor whose primary goal is to provide instant information on the systems _current_ performance. However, some kind of _history data_ is often required too, and the program also contains functions for collecting and analyzing history performance information.

Data collection is controlled by means of the FC MONCOLL subcommand. See the "FCONTROL MONCOLL" subcommand in the _z/VM: Performance Toolkit Reference_ for a detailed description.

The performance monitor collects data from CP control blocks in real storage and also from CP MONITOR records. It then uses this information for computing all the data which you can display on your terminal.

The following figure illustrates the data collecting mechanism:
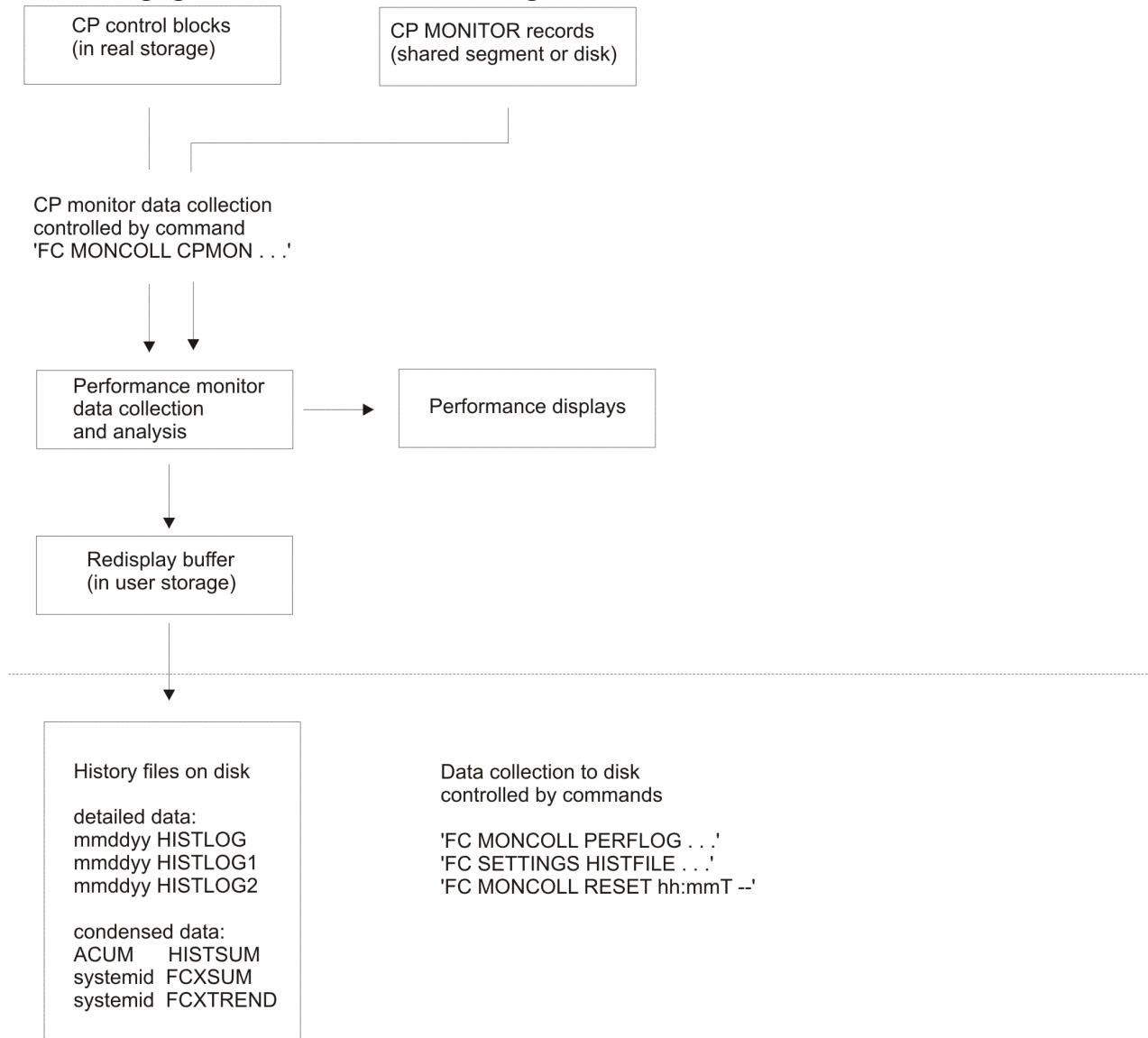


*Figure 32. Performance Data Collection*

The basic options are:

- Data collection only into temporary work buffer
- Data collection into temporary work buffer *and* to disk in history files

The first choice would allow you to analyze data in real time only, while the second method also provides history data which allows trend analysis and capacity planning, but at the expense of additional disk space requirements. Read the following paragraphs carefully and then decide on the method you want to use.

## Data Collection into Work Buffer Only

Activate permanent data collection in the machine which will be used for performance monitoring, using the command FC MONCOLL CPMON ... ON. This will start permanent performance data collection from both CP control blocks and CP monitor records, even while the machine is used in basic mode for other work, and data will be saved in the 'redisplay buffer' so you can redisplay some of the most recent performance information when a problem has to be analyzed.

Permanent data collection causes some overhead, so you should use it only in machines where the collected data will actually be used (usually only one machine per system). Permanent data collection is a prerequisite, on the other hand, if any serious analysis is to be done, because you will then need the system's performance data for the *whole* time the system was active, and not just the periods when a performance problem is suspected.

## Data Collection in History Files on Disk

Performance Toolkit can produce two basically different kinds of history files.

- A set of simple history files containing only overall system data:

  - Detailed files *yyyymmdd* HISTLOG*n*, with one record per monitor sample interval

  - Summary file ACUM HISTSUM with the same set of data fields, but containing only one record per hour

- Extended summary and trend files not only contain overall system data but can also include performance information for single users, channels, and I/O devices. The period covered by each set of trend records depends on the user's settings ('FC MONCOLL RESET ...' command).

Both kinds of history files are described in more detail in the following paragraphs.

# Creating Simple History Files

All the performance data prepared for viewing on the performance redisplay screen, and most of the other 'by time' logs' data can optionally be written to disk for archiving or as input for the history graphics programs, or for numerical analysis by the FCONXREP procedure provided with Performance Toolkit (or any additional programs you may care to write). This function is activated by the command FC MONCOLL PERFLOG ON which will cause the data to be written to disk files.

Performance Toolkit currently supports two different sets of simple history files:

- An extended history data format stored in files *yyyymmdd* HISTLOG*n* (detailed log) and ACUM HISTSUM (summary data)

- The old history data format stored in files *mmddyy* PERFLOG*n* (detailed log) and ACUM PERFHIST (summary data)

While the old PERFLOG and PERFHIST files were based exclusively on the performance data shown in the REDISP display, the new HISTLOG and HISTSUM files contain all of this information and most of the data provided with the remainder of the 'by time' logs available in realtime mode. It is recommended that you use the new files. Support for the PERFLOG and PERFHIST files was kept only to ease migration. You may want to go on creating the old files too, however, in order to have a complete basis for trend analysis, until performance data for a sufficient period is also available in the new format.

A further command, 'FC SETTING HISTFILE ...', allows specifying what kind of history data to write to disk:

**NEW**
    will let *only* the new history files be created. This is also the initial setting that will take effect if users do not specify something else.

**OLD**
    will let *only* the old history files be created.

**BOTH**
    will let both the old and new set of files be created.

The current date is used to build the file names of the detailed history data log files HISTLOG and PERFLOG. Up to three generations of these detailed performance data files will be kept on your A-disk, with the filetypes HISTLOG, HISTLOG1, and HISTLOG2 and/or PERFLOG, PERFLOG1, and PERFLOG2 respectively for the current file and for the two previous files. The filetypes will automatically be renamed, and the oldest file (PERFLOG2 and HISTLOG2) erased when a file for a new date is created.

An exit has been coded in Performance Toolkit to call procedure FCXRENAM EXEC on any accessed disk immediately *after* doing its own renaming. You can use it for additional housekeeping (see "Handling of Performance Toolkit Console Logs" on page 94 for more information).

The detailed HISTLOG*n* and PERFLOG*n* files contain the same kind of detailed data as the realtime 'by time' logs from which they stem, that is, one record per measurement interval. Each day, when a new generation of log files begins, the contents of the most recent one will be appended to the performance history file 'ACUM HISTSUM A' and/or 'ACUM PERFHIST A'. These files contain performance data in a more condensed format. One record is generated per hour with the averages of the values found in the original detailed log file for the same period. These history files are intended to provide data for performance trend analysis in capacity planning, and also as a reference where you can easily see what the average load was during a certain period.

Further arguments of the 'FC MONCOLL PERFLOG ON' command allow you to specify:

- Start and stop times for data collection to disk. Using this, you can collect information from specific periods in history files, for instance, only from prime shift.

- Whether data collection in history files is to be active each day, or only on specific days (for example, Monday to Friday). History data collected during weekends will usually have undesirable effects when creating trend graphics.

Once collected, you can analyze the data contained in these history files using one of the following.

- The performance redisplay screen for displaying the actual performance figures

- One of the 'GRAPH...' or 'PLOT...' commands for displaying the values of selected performance variables graphically

- The FCONXREP procedure for producing a table with average load values for selected performance variables on a daily, monthly, and yearly basis

# Creating Extended Summary and Trend Files

Extended summary and trend files contain data records with general system load information, but can also contain records with more detailed data, such as:

- Specific processors
- Specific users
- Specific disks
- Specific channels

While the simple HISTLOG and HISTSUM history files will usually provide sufficient data for overall trend analysis and capacity planning purposes, you will need extended trend file data to see more detailed information, for example on the behavior of a specific user, or on the performance of a specific disk.

You can control both the contents, the kind of file records to be created, and the data collection interval for which a set of trend or summary file records should be created.

**Contents:**
The kind of records to be created, and any additional restrictions to specific users or I/O devices, are defined in a TRENDREC file for the trend files, and in a SUMREC file for summary files. The default file names are:

- FCONX  TRENDREC for controlling trend file contents
- FCONX  SUMREC for controlling summary file contents

Different file identifications can be set by the TRENDREC and SUMREC arguments on the 'FC SETTINGS ..' command. See the "FCONTROL SETTINGS" subcommand in the *z/VM: Performance Toolkit Reference*.

**Period:**
The time at which a new set of trend or summary data records should be created is specified by the command FC  MONCOLL  RESET

The performance fields saved in the trend and summary file records correspond to the values accumulated since the last reset, that is, both the start and the end time of the period for which data are to be saved are determined by a pair of reset time specifications. See the "FCONTROL MONCOLL" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.

Note that the system profile is shipped as the file FCONX $PROFILE. See "Initialization File FCONX $PROFILE" on page 67 for more information.

## Output Files Built

**Summary Files** contain data for one single 'summary' period – that is, a new file will be created for each new period. The *initial* default file ID used is `systemid` FCXSUM A1 where the VM system's system identification is used as the file name. A different *initial* file ID can be set using the SUMFILID argument of the FC SETTINGS command. See the "FCONTROL SETTINGS" subcommand in the *z/VM: Performance Toolkit Reference*.

If there is an existing file with the same file ID, consecutive numbers from '01' to '99' will be either appended to the filetype if it is up to 6 characters long, or to the first 6 characters of the filetype if it is longer, to create unique file *identifiers* .

**Trend Files** are cumulative files. Data from each new 'trend' period will be appended to an already existing file, thus creating a history file that can contain data for many days or months, and that provides a base for capacity planning and trend analysis. The default file ID is `systemid` FCXTREND A1 where the VM system's system identification is used as the file name. A different file ID can be set using the TRDFILID argument of the FC SETTINGS command. See the "FCONTROL SETTINGS" subcommand in the *z/VM: Performance Toolkit Reference*.

## Summary and Trend Record Selection

SUMREC and TRENDREC control files can be defined to control record creation. All possible records will be included in a summary or trend file if no SUMREC or TRENDREC control file has been defined – that is, these control files are intended to be used for restricting record creation as required to prevent building unnecessarily large and detailed files.

Summary and trend files can contain the same kind of data, so the same logic applies for both trend and summary file record selection. The only difference is that trend record selection is controlled by entries in a TRENDREC control file, while summary record selection is controlled by a SUMREC control file.

The following records deal with general system performance data. They are always created (provided the prerequisite monitor domains were enabled) and need not be explicitly specified.

| Number | Data |
|---|---|
| X'FC00' | System configuration data |
| X'FC01' | General system load data |
| X'FC02' | Processor load data |
| X'FC03' | LPAR load data |
| X'FC04' | Minidisk/fulltrack cache data |
| X'FC05' | CP services activity data |
| X'FC09' | System spin lock data |
| X'FC3A' | Overall user transaction and response time data |
| X'FC42' | User class resource usage and wait state data |
| X'FC43' | Overall user resource usage and wait states data |
| X'FC45' | User class transaction and response time data |
| X'FC46' | Overall user transaction and response time data |
| X'FC47' | User CPU pool resource usage data |
| X'FC51' | I/O processor activity data |

If an FCONX TRENDREC or FCONX SUMREC file (or corresponding control file defined using the TRENDREC or SUMREC arguments of the 'FC SETTINGS ..' command) is found on any accessed disk

all of the remaining record types are created only on request – that is, if the control file contains a corresponding RECORDS specification. The format for the record selection specification is:

```
RECORDS          spec1  {spec2} {spec3} ... ...
```

**spec*n***
> can be any of the following record selection names:

**CHANNEL**
> for selecting channel data records:
>
>> X'FC06' with channel high frequency (HF) sampling interval
>> X'FC07' with channel measurement facility data
>> X'FC08' with extended channel measurement facility data

**PRIVOPS**
> Record X'FC0A' with privileged instructions counters

**DIAGS**
> Record X'FC0B' with DIAGNOSE instructions counters

**ISFC**
> for selecting ISFC data records:
>
>> X'FC0C' with ISFC End Point configuration
>> X'FC0D' with ISFC Logical Link configuration
>> X'FC0E' with ISFC End Point activity
>> X'FC0F' with ISFC Logical Link activity

**NSS**
> Record X'FC3C' with shared segment data

**DSPACES**
> Record X'FC3E' with shared data spaces data

**USER**
> for selecting user data records:
>
>> X'FC41' user resource usage and wait state data
>> X'FC44' user transactions and response time data (created only when USER domain EVENT data is collected)

**VSWITCH**
> for selecting vswitch data records:
>
>> Record X'FC54' with global networking object data
>> Record X'FC55' with virtual switch data

**VNIC**
> Record X'FC56' for virtual network interface cards (VNIC) data

**PCIFUNC**
> for selecting PCI Function data records:
>
>> X'FC57' with PCI Function configuration data
>> X'FC58' with PCI Function activity data

**DASD**
> for selecting general DASD data records:
>
>> X'FC61' with general DASD performance data
>> X'FC65' with DASD cache utilization data
>> X'FC68' with CP owned DASD data (system areas)

**QDIO**
> Record X'FC6D' with QDIO device data

**SCSI**
> Record X'FC6F' with SCSI device data (must specify with DASD to generate complete SCSI DASD information)

**SEEKS**
> Record X'FC71' with DASD seeks data

**SFS**
> Record X'FCA2' with SFS and BFS server data

**MTUSER**
> Record X'FCA4' with multitasking users data

**TCPIP**
> for selecting TCP/IP data records:
>
>> X'FCA6' TCP/IP server data
>> X'FCA7' TCP/IP links data

**RSK**
> Record X'FCA8' with RSK server data

**LINUX**
> Record X'FCA9' with Linux internal performance data

**SSI**
> for selecting SSI data records:
>
>> X'FCB1' with SSI configuration
>> X'FCB2' with SSI State Change Synchronization activity and State/Mode information

## Selection of Specific User IDs

For the user related record types (DSPACES, USER, SFS, MTUSER, TCPIP, RSK, and LINUX) record creation can be further restricted with the USERID keyword, by defining that records should be built only for the specified user IDs:

```
USERID          userid1  userid2  ...  ...
```

**userid*n***
> defines the user identification of a virtual machine for which trend or summary records should be created.

## Selection of Specific I/O devices

For all the I/O device related record types (DASD, QDIO, SCSI, and SEEKS) record creation can similarly be restricted with the DEVNO keyword. DASD, QDIO, SCSI, and SEEKS trend or summary records will then be created only for the devices whose device numbers are explicitly specified:

```
DEVNO           devno1  devno2  ...  ...
```

**devno*n***
> defines the device number of an I/O device for which records should be created.

Multiple RECORDS, USERID, and DEVNO selection records are accepted.

# FCONX SUMREC and FCONX TRENDREC File

The following figure shows the sample TRENDREC file delivered with the program (the FCONX TRENDREC and FCONX SUMREC sample files are identical). Its logic follows that of other Performance Toolkit control files in treating any record with an asterisk in column one as a comment, ignoring the remainder of the record's contents.

```
*----------------------------------------------------------------*
*----------------------------------------------------------------*

* General Trend Record Selection
RECORDS    CHANNEL NSS DSPACES USER DASD SEEKS SCSI VSWITCH VNIC QDIO
RECORDS    SFS MTUSER TCPIP RSK LINUX
RECORDS    PRIVOPS DIAGS
RECORDS    ISFC SSI
RECORDS    PCIFUNC


* Specific User Selection (DSPACES, USER, SFS, MTUSER, TCPIP, RSK, LINUX)
* USERID    userid1  userid2  userid3  userid4  ...
* USERID    OPERATOR RSCS     TCPIP     VTAM

* Specific I/O Device Selection (for DASD, QDIO, SCSI, and SEEKS records)
* DEVNO     devno1  devno2  devno3  devno4  ...
* DEVNO     0781 0782 0784 0786 0788
* DEVNO     0798 0799
```

In this sample, *all* possible trend and summary records displayed will be created. The sample will include data for *all* users and disks because the USERID and DEVNO specifications are commented out. *On large systems with hundreds of users and disks this will result in fast-growing trend files, so you might need to change the specifications to your needs.* (See the "Layout of Extended Trend File Records" appendix in the *z/VM: Performance Toolkit Reference* for information on trend record layout and size.)

In large centers where many different z/VM systems are managed, you can use the additional specifications described under "Extended Input File Processing" on page 93 in addition to the actual authorization statements. This allows users more flexibility when setting up common files, and when separating system-specific and generic definitions.

You can choose to work alternately with several different SUMREC or TRENDREC files. Use the SUMREC and TRENDREC arguments of the FC SETTINGS command to request use of control files with file IDs other than the default names (see the FCONTROL SETTINGS subcommand in the *z/VM: Performance Toolkit Reference*).

# Data Collection Interval

**Data collection for the CP monitor based screens** is obviously linked to the monitor sample interval. You can set the interval using the CP command 'MONITOR SAMPLE INTERVAL ...'; the initial default sample interval is 60 seconds.

**Performance data for the CP control block based screens** are updated in one-minute intervals. Data extraction and recalculation in shorter intervals can be forced for these displays simply by pressing the ENTER key without entering any data. However, be aware that this process requires a certain amount of CPU time which, although usually negligible for the standard interval of 60 seconds, may cause a significant load if often repeated within a short period. After forcing recalculation by pressing the ENTER key, the next one-minute interval timer interrupt will be ignored, that is, up to two minutes may elapse before the next automatic recalculation.

While the data collect interval is the smallest unit of time for which averages of any activity can be calculated and displayed, there are a number of other (longer) periods that can be set to allow calculating smoothed averages. See "Resetting Performance Data/Defining Periods" on page 121 for a description of all these periods, and of their uses.

# Printing Performance Data

All the data generated for viewing on the different performance monitor displays can also be printed on a virtual line printer. Three different modes are available:

## Printing Concurrent with Display

After entering the command PRINT ON, all the data viewed on the display will also be printed at the same time. This command ensures all the data you have seen on the screen will also be available for

re-checking at a later time. Concurrent printing is stopped and the print file created so far closed, when the command `PRINT OFF` is entered.

## Printing all Data for Selected Displays

Only the data shown on the screen are included in the print file for concurrent printing. This is only a small part of all the data available at any specific moment, and it may be desirable to print more data to allow later in-depth analysis. The PRINT command allows you to specify what data you require and it will provide a print file complete with the necessary carriage control characters for separating the data. Average or current data will be printed, depending on the setting used for displaying the data.

See the "PRINT" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.

## The FCONX REPORTS File

The FCONX REPORTS file allows you to define which part of the collected performance data should be printed when:

- A PRINT ALL command is entered.
- Average data are printed automatically prior to RESET.
- Interim reports are to be printed at the end of an INTERIM interval.

A sample FCONX REPORTS file is supplied with the program. This file must be fixed or variable format, with a record length that is less than or equal to 80. While processing this file, if Performance Toolkit finds a record that exceeds this limit, it will issue error message FCXPRT120E with return code 8 and use only the first 80 bytes of that record for processing. You can adapt this file to fit your needs by doing the following:

- Insert an asterisk in column 1 of records that describe reports you do not need or remove the asterisk for reports that should be printed.
- Limit the number of data lines to be printed for reports that would become too large if all data were printed - all of the user-related reports on large systems with hundreds of users, for example.
- Change the sorting sequence to be used for printing.
- Specify the PAGESIZE to be used for printing.

In the following sample FCONX REPORTS file, we have limited the number of print lines for all user reports to a maximum of 100 lines, and we have asked for the user resource usage report to be sorted in descending sequence of total CPU used, for a page size of 29 lines for the user paging activity report.

```
*-------------------------------------------------------------*
* Performance Toolkit for VM Report Selection File            *
*                                                             *
* File format:                                                *
* - Asterisks in column 1 indicate comments                   *
*-------------------------------------------------------------*

*-General System Data----------------------------------------*
  CPSERV
  INTERIM CPSERV
  CPU
  DSPACESH
  INTERIM DSPACESH
  DSVBKACT
  INTERIM DSVBKACT
  LOCKACT
  INTERIM LOCKACT
  LPAR
  INTERIM LPAR
  LPARCONF
  LSHARACT
  INTERIM LSHARACT
  MONDATA
  INTERIM MONDATA
  MONSET
  NSS
  INTERIM NSS
```

```
   PRIVOP
   INTERIM PRIVOP
   SPOOL
   SPOOL     USER
   STORAGE
   SYSCONF
   SYSSET
   SYSTEM
   SYSTRANS
   INTERIM SYSTRANS
   TOPOLOG
   VDISKS
   INTERIM VDISKS

*-Single System Image Data--------------------------------------*
   SSICONF
   SSISCHLG
   SSISMILG
   ISFECONF
   ISFEACT
   INTERIM ISFEACT
   ISFLCONF
   ISFLACT
   INTERIM ISFLACT
   ISFLALOG
   LGRDATA
   LGRELOG

*-I/O Device Data-----------------------------------------------*
   CACHDBSE
   INTERIM CACHDBSE
   CACHEXT
   INTERIM CACHEXT
   CHANNEL
   INTERIM CHANNEL
   CTLUNIT
   INTERIM CTLUNIT
   DEVCONF
   DEVICE
   INTERIM DEVICE
   DEVICE CPOWN
   INTERIM DEVICE CPOWN
   DEVICE HPF
   INTERIM DEVICE HPF
   FCHANNEL
   INTERIM FCHANNEL
   HIPSOCK
   INTERIM HIPSOCK
   IOASSIST
   INTERIM IOASSIST
   IOCHANGE
   LCHANNEL
   INTERIM LCHANNEL
   QDIO
   INTERIM QDIO
   QEBSM
   SCSI
   INTERIM SCSI
*  SEEKDIST
*  INTERIM SEEKDIST
*  SEEKLOC
*  INTERIM SEEKLOC
HPALIAS
INTERIM HPALIAS
HPSHARE
INTERIM HPSHARE
VOLUME
INTERIM VOLUME
* VOLLOG devnum

*-Virtual Network Data------------------------------------------*
   VSWITCH
   EVSWITCH
   GVSWITCH
   GLONACT
   VNIC
   EVNIC
   GVNIC

*-PCI Function Data---------------------------------------------*
   PCICONF
   PCIACT
```

```
   INTERIM PCIACT
   PCIACT  ROCE
   INTERIM PCIACT ROCE
   PCIACT  ZEDC
   INTERIM PCIACT ZEDC
   PCIACT  ISM
   INTERIM PCIACT ISM


*-User Data-------------------------------------------------------*
  IOPROCLG
  IOPQUEUE
  MTUSERS
  INTERIM MTUSERS
  SFS
  INTERIM SFS
  UCOMM      (100
  INTERIM UCOMM      (100
  UCONF      (100
  UPAGE      (100
  INTERIM UPAGE      (100   PAGESIZE 29
  UPGACT     (100
  INTERIM UPGACT     (100
  UPGUTL     (100
  INTERIM UPGUTL     (100
  UQDIO
  URESP      (100
  INTERIM URESP      (100
  USER       (100   SORT %CPU
  INTERIM USER       (100   SORT %CPU
  USTAT      (100
  INTERIM USTAT      (100
  UTRANS     (100
  INTERIM UTRANS     (100
* LINUX      userid
* LXCPU      userid
* LXFILSYS   userid
* LXMEM      userid
* LXNETWRK   userid


*-CPU Pool data--------------------------------------------------*
  CPLCONF
  CPLACT
  INTERIM CPLACT


*-System Load by Time--------------------------------------------*
  AGELLOG
  AUXLOG
  AVAILLOG
  AVLB2GLG
  AVLA2GLG
  BFSFILES
  BFSLOCKS
  BFSPIPES
  CORREL   ACT
  CPSALL
  CPSTOCP
  CPSTOVM
  CPSFAILS
  DASDLOG
  DEMNDLOG
  DSVCLOG
  DSVSLOG
* LOCKLOG
  LPARLOG
  MDCACHE
  MDCSTOR
  PINLOG
  PRCLOG
  PROCCONF
  PROCLOG
  PROCSUM
  PAGELOG
  PHYSLOG
  PUCFGLOG
  PUORGLOG
  REDISP
  RESPALL
  RESPUP
* RSKDET    userid
* RSKDISK   userid
* RSKLOG    userid
* RSKSTOR   userid
```

```
      SCHDATA
      SCHEDLOG
      SFSIOLOG
      SFSLOG
      SFSREQ
      STEALLOG
      STORLOG
      SUBPLOG
      SXSAVAIL
      SXSDEFER
      SXSPAGE
      SXSUTIL
      SYSLOG
      SYSSUM
      SYSSUMLG
* TCPACTLG userid
* TCPBPLOG userid
* TCPCONF  userid
* TCPDATLG userid
* TCPICMP  userid
* TCPIOLOG userid
* TCPLINKS userid
* TCPSESS  userid
* TCPUSERS userid
* TFTPLOG  userid
      UCOMMLOG
      USTLOG

*-Benchmarking Logs for Specific Objects------------------------*
* CACHELOG devnum
* CACHDLOG devnum
* CPOWNLOG devnum
* DEVLOG   devnum
* HPFLOG   devnum
* VOLLOG devnum
* SCSILOG  devnum
* SEEKDLOG devnum
* SEEKLLOG devnum
* MTUSRLOG userid
* QDIOLOG  userid
* QEBSMLOG userid
* UCOMMLOG userid
* UPAGELOG userid
* UPGACTLG userid
* UPGUTLLG userid
* UQDIOLOG userid
* URESPLOG userid
* USERLOG  userid
* USRMPLOG userid
* USTATLOG userid
* USTMPLOG userid
* UTRANLOG userid
* ISFELOG  endpoint
* ISFLLOG  loglink
* PCILOG   pcifid
```
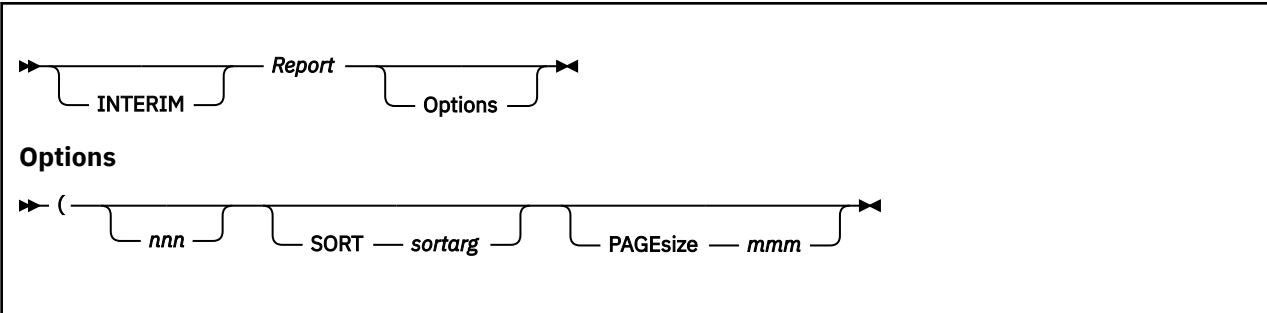
The initial default page size is set to OFF. That is, although a new page will be started for each new report, no intermediate headings will be inserted even if the output extends over several pages, unless you have performed one of the following actions:

- Set another default page size using the FC SETTINGS PAGESIZE command.
- Set a page size for a specific report by including the PAGESIZE option with the PRINT command or in the report selection record of the FCONX REPORTS file.

## FCONX REPORTS Statement

The following is the format of the FCONX REPORTS file statement:

```
►►─┬───────────┬─ Report ─┬───────────┬─►◄
   └─ INTERIM ─┘          └─ Options ─┘

Options

►►─ ( ─┬─────────┬─┬──────────────────┬─┬──────────────────────┬─►◄
       └─ nnn ─┘   └─ SORT ── sortarg ─┘   └─ PAGEsize ── mmm ─┘
```

The parameters are:

**INTERIM**
Specifies that a report is to be printed at the end of an INTERIM interval. The INTERIM parameter is valid only for reports that also support interim data; that is, averages calculated for the current interim period. A note at the end of each performance screen description will tell you whether INTERIM averages are supported.

***Report***
The name of any report that is to be printed.

The options are:

***nnn***
Defines the maximum number of lines to be printed. This option is useful in reducing the output size of the USER and DEVICE reports on large systems. (Make sure the lines are SORTed so that the meaningful information is contained in the first part of the report.)

**SORT** *sortarg*
Indicates that the report is to be sorted as defined by the *sortarg* argument. The SORT option is valid only for reports that also support sorting in monitor mode; that is, reports that are supported by the SORT subcommand. The same sort arguments can be used here as with the SORT subcommand.

**PAGEsize**
Defines the number of lines to be printed consecutively before a page break is forced, and the header lines are printed again at the top of the page.

## Printing Average Data Prior to RESET

You can cause a subset of the important performance data to be printed automatically before the counters are reset at the end of a period with high system load, using the FC MONCOLL RESET subcommand with a 'P' suffix appended to the reset time. This option is intended to provide an overview of system performance during a certain period and *average data* will be printed.

**Note:** This command also causes counters to be reset.

## Printing Interim Data

Similar to the printing of average data for the overall reporting period you can also activate automatic printing of 'interim' reports at the end of each INTERIM period. The INTERIM string prefixing a report is used to indicate that this report is to be printed at the end of each INTERIM period. The INTERIM period can be set in two ways:

1. Using the 'R_I' and 'I' suffixes of the FC MONCOLL RESET command, which gives you full freedom in setting the intervals
2. Using the 'INTERIM' argument of the FC SETTINGS command to request the use of fixed interim periods

**Note:** Automatic INTERIM report printing is active only during reporting periods where a general overall PRINT has been scheduled. The INTERIM reports will be stacked until the reports for the overall reporting period are printed. They are then appended to those reports with their own table of contents for each interim interval.

INTERIM data can be printed only for reports that support the INTERIM period.

# Writing Performance Reports to Disk

It is possible to have performance information available on disk for additional processing before it is to be printed (the disk files still include the necessary carriage control characters).

## Writing to Disk Concurrent with Display

The command PRINT ON DISK can be used to write performance information to disk concurrently with the display. Operation is similar to concurrent printing described above, except that the data will be written to disk file 'FCONMON LISTING A'. When activated, you can also use the PRINT command for writing *all* data for selected displays to disk. All performance monitor 'print' data will be written to disk until concurrent writing to disk is stopped using the command PRINT OFF, or if the output device is changed.

## Permanent Writing of Key Performance Variables to Disk

See "Data Collection in History Files on Disk" on page 110.

## Directing General Print Output to a Disk File

There are two ways to direct print output to a disk file.

- *Defining a default output file:* The REPFILID argument of the FC SETTINGS subcommand allows defining a disk file identification where all print output is to be directed by default. See the "FCONTROL SETTINGS" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.
- *Defining a disk file for one specific report:* The TOFILE argument of the PRINT command allows defining a disk file identification where the output for that specific print command is to be directed. See the "PRINT" subcommand in the *z/VM: Performance Toolkit Reference*.

# Resetting Performance Data/Defining Periods

Average performance data for longer periods (such as the whole prime shift) may be required for capacity planning purposes. Analyzing system performance over shorter periods of high load can give you a better indication of problem areas on the system.

Performance Toolkit keeps different sets of count fields for:

- 'Current' data, the last monitor sample interval, used when viewing performance displays with the 'current' setting
- 'Bytime' data, for averages to be used on 'by time' log detail lines
- 'Interim' data, for averages within an INTERIM period, used when viewing performance displays with the 'interim' setting or when requesting automatic printing of data for interim periods
- 'Average' data, for averages within the overall reporting period, used when viewing performance displays with the 'average' setting or when requesting automatic printing of data for the overall reporting period

Two additional sets of counters allow you to set independent periods for:

- 'Summary' record creation, with totals/averages for the 'summary' period
- 'Trend' record creation, with totals/averages for the 'trend' period.

The following figure illustrates the use of these different periods.

# CP Monitor Samples

```
      08:00  09:00  10:00  11:00  12:00  13:00  14:00  15:00  16:00  17:00
```

BYTIME

CURRENT

INTERIM Periods

R_I

AVERAGE Overall Reporting Period

P

R_P

SUMMARY
Period

SUMMARY
Period

R_S                              S      R_S                              S

TREND Period

R_T                                                                      T
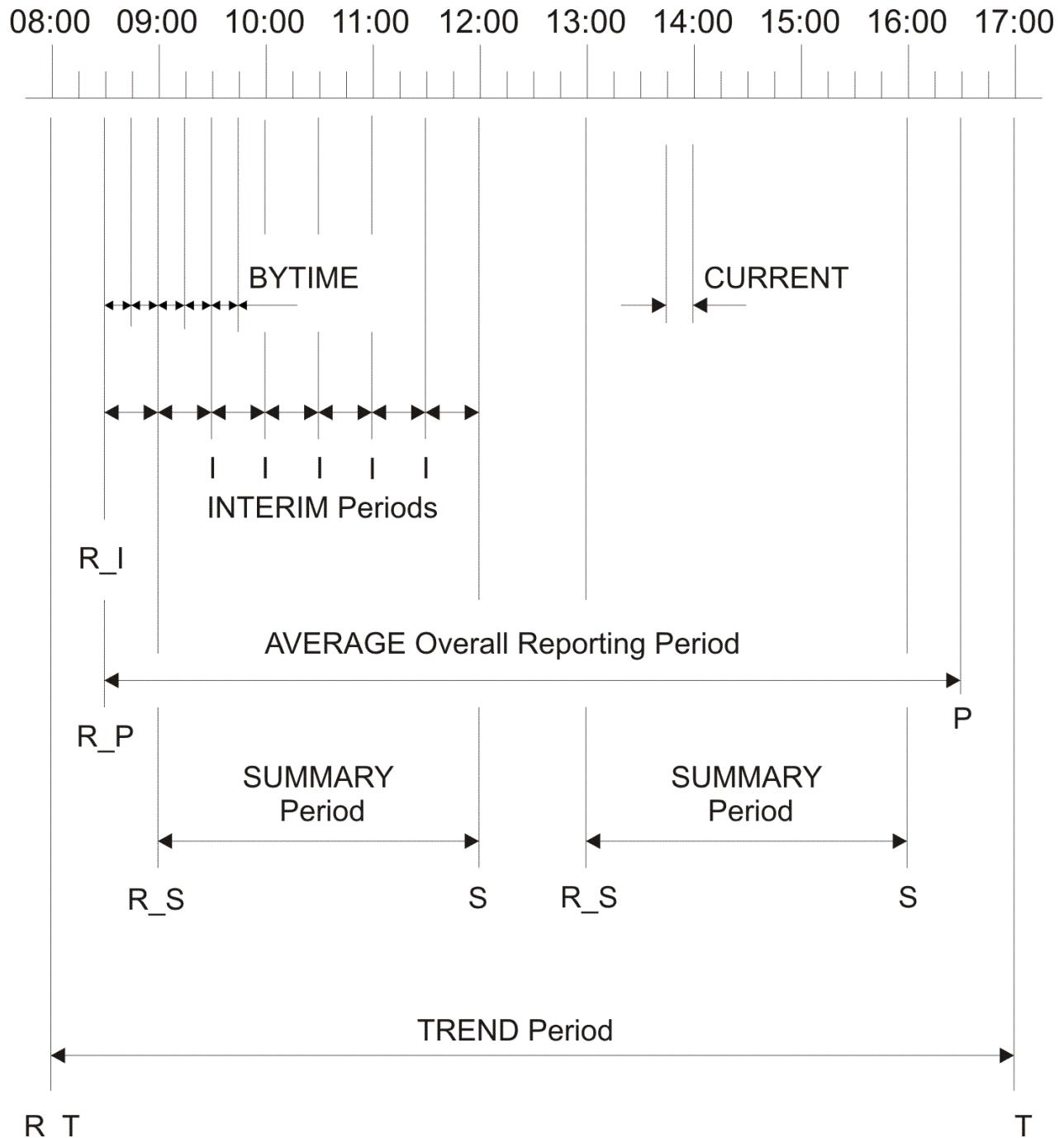
*Figure 33. Periods Used in Calculating Averages*

All periods are defined by the times at which the corresponding count fields are reset. For the major 'average', 'summary', and 'trend' periods this is achieved specifying time stamps with the FC MONCOLL RESET command, and by appending the corresponding 'reset qualifiers' to the time stamps. These 'reset qualifiers' have also been included in the above figure to help better understand the command logic. The following set of commands could be used to set periods similar to the ones shown above:

```
FC MONCOLL RESET 08:30R_P 16:30P
FC MONCOLL RESET 09:00R_S 12:00S 13:00R_S 16:00S (MERGE
FC MONCOLL RESET 08:00R_T 17:00T                 (MERGE
FC MONCOLL RESET 08:30R_I 09:00I 09:30I 10:00I 10:30I ... (MERGE
```

In this example we defined:

- A reporting period from 08:30 to 16:30. At 08:30 the counters are only reset ('R_P' qualifier) and at 16:30 we request automatic printing of all reports defined in the active REPORTS file ('P' qualifier, see also "Printing Average Data Prior to RESET" on page 120).
- Two summary periods from 09:00 to 12:00 and from 13:00 to 16:00, with the 'S' qualifiers at 12:00 and 16:00 indicating that summary files should then be created.
- A trend period from 08:00 to 17:00, with the 'T' qualifier at 17:00 indicating trend records should be built and appended to a trend file (note that resetting the summary and trend counters does not make much sense unless they are later used for actually building summary or trend records). See "Creating Extended Summary and Trend Files" on page 111 for more information on trend record creation.
- Plus a number of interim periods of 30 minutes each, requesting automatic printing of any INTERIM reports defined in the active REPORTS file.

**Note:** The second and following 'FC MONCOLL RESET ...' subcommands were entered with the ('MERGE' option: this is to indicate that later definitions (for example, the reset argument '08:30R_I') should be merged with previous ones and not replace them, so the previously entered '08:30R_P' request remains active.

See the "MONCOL RESET" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.

The 'bytime' and 'interim' periods serve to separate the 'average' overall reporting period into smaller pieces, to provide intermediate averages and thus finer granularity for that period. While 'interim' periods can also be set using the FC MONCOLL RESET command as shown in the above example, this is only necessary when setting interim periods of different lengths. Use the BYTIME and INTERIM arguments of the FC SETTINGS command to set 'bytime' and 'interim' periods of fixed length. The following commands could be used, for instance, to generate 'interim' reports equivalent to the ones defined above with the FC MONCOLL RESET command, and to also define a 'by time' period of 15 minutes:

```
FC SETTING INTERIM 30
FC SETTING BYTIME 15
```

Both the 'FC MONCOLL RESET ...' and the FC SETTINGS ...' commands should be included in your FCONX $PROFILE to reset the counters automatically.

# Remote Performance Monitoring

Performance Toolkit allows for remote access to the performance data collected in any specific machine, monitoring of many remote VM systems from one central point of control, and use of the Web browser to retrieve and analyze performance data. See "Remote Performance Monitoring Facility" on page 9, "Internet Interface" on page 45, and "APPC/VM Interface for Performance Data Retrieval" on page 32 for a detailed description of these facilities.

# Viewing History Data Files

## Simple History Files

The detailed and summary performance data files created by Performance Toolkit can easily be viewed using the REDISP and REDHIST commands, or used as input for the creation of graphics. Where many files are available (many generations of detailed performance log files kept on disk, and/or data from many different systems archived on a central system for analysis) you may want to use the 'History Data Menu' screen. It makes working with history data files easy. See "Viewing Performance History Data" on page 130 for details.

## Extended Summary and Trend Files

Extended summary and trend files can be processed using the TRNDSCAN command. You will be able to view all the performance displays for which trend or summary records were created. See section "Scanning Performance Toolkit Trend Data Files" on page 133 for a detailed description of trend file scan mode.

## Processing CP Monitor Data Files

CP monitor data can also be collected by the MONWRITE utility, and written to either a disk or a tape file for later processing. Performance Toolkit provides two different modes for working with such files:

- **Manual mode:** Allows you to browse through these files interactively. This can be very helpful if reports from users indicate a performance problem during a certain period of a specific day and an in-depth analysis in real-time was not possible. See "Scanning CP Monitor Data Files" on page 131 for a detailed description of monitor data scan mode.
- **Batch mode:** Allows you to process monitor data files in batch mode. See Chapter 6, "Operation in Batch Mode," on page 139 for a detailed description of batch processing mode.

## Exception, Threshold, and User Monitoring

The Performance Toolkit performance monitor will collect and display a lot of very detailed information on the current load of the CPU and I/O equipment, and on the users which have caused it. This is, of course, vital information for an experienced operator or a systems programmer who tries to *analyze* an existing bottleneck. The same information is, on the other hand, not really suited for *detecting* the beginning of a performance problem, one cannot seriously expect an operator or systems programmer to continuously watch a monitor screen and look for values that indicate performance degradation.

This problem is solved by special routines included in the monitor which check the collected data for indications of problems, and which generate alert messages if such indications are found:

- *Exception monitoring* informs the operator if unusual conditions are found which should be investigated.
- *Threshold monitoring* generates alert messages if the values of selected performance indicators for the whole system exceed pre-set limits.
- *User monitoring:* Alert messages can be generated for users who either:
  - Have exceeded certain limits in their CPU load or I/O rates
  - Appear to be in either a CPU loop, an I/O loop, in a 'constant WSS' loop, or which have been idle for a long time. Looping or idle users can optionally also be forced off the system.

The alert messages generated by these functions are displayed on the basic mode screen, thus informing the user there is a potential problem to be investigated. All these functions are operative only while permanent data collection is active (command FC MONCOLL ON entered). Their operation is explained in more detail in the following paragraphs.

### Exception Monitoring

Exception monitoring functions look for conditions which should not occur during normal operation of a correctly generated system. Current implementation includes checking in the following areas:

- *Missing interrupt detection* on real disks. Message number FCXxxx354 will be issued if a non-CP owned disk has been waiting for an interrupt for at least one measuring interval, and this message will be repeated until the condition has been cleared.
- *Check for pending RESERVE* on real disks. Message number FCXxxx356 will be issued for each disk where a RESERVE status was found to have remained pending for at least one measuring interval, without I/O activity to the device in the same interval. The message will be repeated until the condition has been cleared.

- *Real storage size checking* compares the real storage available on the system with the storage size generated. Message number FCXxxx323 will be issued if more than 1/16 of the generated storage is not available for use. This message is generated once during initialization, and it will be repeated whenever performance monitor mode is entered.

Messages generated by the above functions will be inserted into the basic mode screen.

The following messages are generated when unusual conditions are detected while looking at specific performance monitor screens:

- FCXxxx355M Interrupt(s) missing for device(s) of user xxxxxxxx indicates that at least one of the virtual devices of user *xxxxxxxx* is waiting for an interrupt. Monitored only while the detailed user display is selected for user *xxxxxxxx*, the device will be flagged accordingly on the screen.

With the exception of the missing interrupt checking, all of the above exception monitoring functions are automatically activated. They need no customizing.

# Threshold Monitoring

By setting thresholds for some key performance variables, you can cause alert messages to be generated when the set limits are exceeded, that is, when high values indicate a performance problem which should be investigated. However, the definition of simple thresholds does not always give satisfactory results:

- Some thresholds may be exceeded, and the corresponding messages generated, even though no marked performance degradation has been noticed (short peak loads, special jobs running on an otherwise empty system). A likely result might be that alert messages are not taken very seriously after too many false alarms.

- If limits are raised until false alarms are eliminated, some real bottlenecks may go undetected until users start to complain.

The original problem with simple thresholds is that large systems are so complex that their performance cannot be characterized by a single value.

An attempt has been made with Performance Toolkit to allow setting thresholds so they become more reliable indicators of actual bottlenecks. This is done with the command FCONTROL LIMIT xxxx nnn q1/qx t1/t2 (see the "FCONTROL LIMIT" subcommand in the *z/VM: Performance Toolkit Reference*). The performance values *xxxx* that can be monitored are:

- All performance variables included in the HISTLOG/HISTSUM performance history log files. See the "Record Layout of HISTLOG/HISTSUM Files" appendix in the *z/VM: Performance Toolkit Reference* for a complete listing. Good candidates are:

  **CPU**
  total CPU load

  **%VEC**
  total vector facility load

  **IO/S**
  total I/O rate

  **PG/S**
  total page rate (reads & writes)

  **ACT**
  the number of users active in an interval

  **C1ES**
  85% threshold of class 1 elapsed time

  **TR-Q**
  number of interactive users in queue

  **NT-Q**
  number of non-interactive users in queue

**%PQ**
    percentage of in-queue users found in page wait

**%IQ**
    percentage of in-queue users found in I/O wait

**%EL**
    percentage of in-queue users found in storage wait

**%ST**
    percentage of pageable DPA in use

**%PGSL**
    percentage of system PAGE area on DASD in use

**%SPSL**
    percentage of system SPOOL area in use

- Normalized CPU load (NORMCPU) and normalized logical CPU load (NORMLCPU).
- Device I/O request queue length (DVQUEUE).
- Device I/O response time (DVRESP).
- Missing interrupts for I/O devices.
- Channel busy percentage (%CHBUSY).

In addition to the actual threshold value *nnn* which can be set for each of these performance indicators, you can also define the following:

- The minimum number of users expected in-queue for class 1 (*q1*) and the total of in-queue users for classes 0, 2, and 3 (*qx*). An exceeded threshold is considered to be a problem only if both the Q1 and the Qx user count are at least as high as the specified minimum.
- The period, in minutes, that a problem has to exist before it is to be taken seriously (that is, before a message is to be generated), and the time interval after which the message is to be repeated if the problem persists.

This combination should help separate short peaks from real problems which need investigation.

## Setup for Threshold Monitoring

Setting reasonable thresholds can be very important. If an actual performance problem is to be analyzed, you will have to do so while it still is there, that is, you will have to react quickly if, as it is often the case when a bottleneck starts to show up, the problem lasts for a short period only. Use the following starting values until experience on your own system suggests different ones (these are the ones included in the sample initialization file 'FCONX $PROFILE', see also "FCONX $PROFILE" on page 92 and Appendix A, "A Sample Initialization Profile (FCONX $PROFILE)," on page 173):

```
FC MONCOLL ON
FC LIMIT NORMCPU  90 1/1 5/10
FC LIMIT C1ES    1.0 1/1 5/10
FC LIMIT %IQ      50 1/1 5/10
FC LIMIT %PQ      30 0/0 5/10
FC LIMIT %PGSL    80 0/0 1/10
FC LIMIT %SPSL    80 0/0 1/10
FC LIMIT %CHBUSY  40 1/1 5/10
FC LIMIT DVQUEUE .40 1/1 5/10
FC LIMIT DVRESP   30 1/1 5/10
FC LIMIT MISSINT   0 0/0 5/10
```

An alert message will be displayed, for instance, when the interactive system response time threshold C1ES has continuously exceeded 1.0 second for 5 minutes, with at least one user in-queue for classes 1 and 0/2/3, and the message will be repeated in 10 minute intervals if the problem persists. The FC MONCOLL ON command causes permanent performance data collection to be activated. It is a prerequisite for threshold monitoring, and it must precede any FC LIMIT statements.

When a few days' operation have shown some of the settings to be not quite adequate, you can adapt the thresholds to your particular environment by doing the following:

- Increase the number of in-queue users if you receive too many alert messages during off-hours, when thresholds may be exceeded due to a non-typical workload such as save jobs. Low numbers are set intentionally by the commands in the supplied sample file, so messages can be generated even on small systems. They will have to be increased on all large systems.
- Increase selected threshold values if you can tolerate higher ones on your system without performance degradation.
- Reduce threshold values or the number of in-queue users if the samples fail to generate alert messages during periods of high system load and bad response times.

The alert messages will be generated and displayed on the screen where the thresholds were entered. Alert messages will help only if they are seen, so you should adopt one of the following setups:

- Use a virtual machine which is always logged on, and which is more or less continuously monitored. The system operator machine usually is a good choice.
- In centers with more than one VM system, the recommended method is to implement a central data collector machine (see "Remote Performance Monitoring Facility" on page 9). Such a machine will then automatically receive all alert messages and will present a system load display that shows at one glance which systems have pending alerts.
- On single systems without a permanent system operator, or to avoid losing time when a threshold alert message eventually is displayed, these messages can also be routed on to another machine, or even to several machines, using the FC PROCESS command. For routing all these messages to userid MAINT, you would have to enter the following command: FC PROCESS ERRMSG 9 'A' REROUTE MAINT CPMSGN (Note that this requires CP privilege class 'B' for running the MSGNOH command. Use REROUTE MAINT CPMSG instead if you do not want to give the machine this privilege class). If the alert messages are to be routed to several machines, you need to use additional FC PROCESS commands. Each command will be run sequentially. For routing them to a virtual machine on another node, you will have to send the message using an EXEC procedure and the FILE argument of the FC PROCESS command as follows: FC PROCESS ERRMSG 9 'A' EXEC ALERT PASSARG This command will cause:
  - The EXEC procedure ALERT to be run
  - The complete message text to be passed as arguments to the ALERT procedure

The corresponding ALERT procedure would have to contain statements similar to the following:

```
/* Procedure for re-routing monitor 'A' messages to another node */
Parse arg msgdata
'SMSG RSCS MSG nodeid userid' msgdata
...
```

Include whatever commands you need in the initialization file FCONX $PROFILE so they are automatically run at each PERFKIT start.

## User Monitoring

Single users can impair the performance of all other users if:

- They consume more than their normal share of the system's resources for their otherwise legitimate work
- They enter a loop due to some programming error

The user monitoring functions have been designed to take care of both of the above cases, as described in the following paragraphs.

**User thresholds:** User threshold monitoring allows setting threshold values for a user's:

- CPU load
- I/O activity to non-spooled I/O devices
- UR I/O activity to spooled card readers, punches and to printers

Thresholds can be set on a system-wide basis for *all* virtual machines on a system, for a certain subset of users, and for individual users. General thresholds will be overridden by any individual thresholds set for a specific user. Alert messages will be generated when a virtual machine's load has exceeded one of the set limits for a certain time. You can have the machine's relative load share reduced automatically.

**User exception monitoring:** On systems with more than one processor, a single virtual machine could be looping, and use a large amount of CPU, without causing the threshold for total CPU utilization to be exceeded. If the system's CPU load is sufficiently high, not even the user threshold for CPU load may be exceeded.

By definition, a virtual machine is looping when it is permanently re-running the same instructions, that is, its working set should always be the same and it should be permanently in-queue. Performance Toolkit loop detection logic checks the following:

- *In-queue state:* A machine can be looping only if it is always found to be in queue at the end of monitor sample intervals, and if it is never found in test-idle or dormant state by high-frequency sampling.
- *Max. WSS delta:* A looping machine's working set size should not change between sample intervals by more than a specified (small) number of pages.
- *Max. WSS size:* Many typical loops run in a small piece of code and so result in a small working set. Large working sets can be used as indicators of the machine not being in a loop.

You can specify the page counts to be used for checking the users' WSS. Unfortunately, these criteria alone will often be insufficient;, that is, it would catch many non-looping machines. To narrow the search down, Performance Toolkit allows you to specify different criteria for the following subsets of loops:

- *CPU loops:* A virtual machine is assumed to be in a CPU loop when it has used a minimum amount of CPU (initially set to 10%), has had no IUCV activity, has not run a single non-spooled I/O, and not more than two I/O to spooled UR devices in a measuring interval.
- *I/O loops:* A virtual machine is assumed to be in an I/O loop when it has executed more than a specified number of virtual non-spooled I/O operations per second in a sample interval (initially set to 500/s).
- *WSS loops:* A virtual machine is assumed to be in a 'constant WSS loop' if it has used more than a specified amount of total CPU (same threshold as for CPU-loop) and has run any number of I/Os.

Additionally, you can activate a test for:

- *IDLE machines:* A virtual machine is treated as IDLE when it has used not more than 10ms of total CPU, no emulation CPU, has not executed a single non-spooled I/O, and has not run more than two I/O to spooled UR devices in a measuring interval.

IDLE machine monitoring allows installations with main storage restrictions to detect and FORCE users that have been idle for a long time: The CP control blocks of an average CMS machine with 4 MB of virtual storage occupy approximately nine pages of real storage in VM (for example, for VMDBK, page tables, virtual device control blocks), and this can considerably impact performance on systems with hundreds of users. Forcing idle users will free these control blocks and reduce paging activity for the remaining users. However, any data stored by a forced user onto temporary minidisks will be lost. When many of these forced users LOGON again later, the LOGON processing is going to cause some additional system load too. Force idle users only when a real storage constraint is the reason for unsatisfactory system performance. Use the 'NODISC' argument of the FC FORCEUSR command if only the logged on idle users are to be forced for security reasons.

## Setup for User Monitoring

**User thresholds:** User thresholds are set by the command FC USRLIMIT (see the "FCONTROL USRLIMIT" subcommand in the *z/VM: Performance Toolkit Reference*). The following commands have been included in the sample initialization file FCONX $PROFILE (see Appendix A, "A Sample Initialization Profile (FCONX $PROFILE)," on page 173 for a listing of the complete sample file):

```
    FC USRLIMIT * CPU  30 5/10
    FC USRLIMIT * IO/S 50 5/10
```

For all users which have used more than 30% CPU time, or generated more than 50 I/O requests per second, for a period of at least 5 minutes, one of the following messages is generated:

```
FCXxxx317A User xxxxxxxx %CPU nn.n exceeded threshold 30.0 for mm min.
```

```
FCXxxx317A User xxxxxxxx IO/S nn.n exceeded threshold 50.0 for mm min.
```

The message will be repeated in 10 minute intervals while the high load persists.

Note that this is just an example. You are expected to:

- Adapt these *general* user limits as required for your environment.
- Set additional *individual* thresholds for virtual machines which are expected to exceed the general thresholds during their normal work (for example, large guests, VTAM, TCP/IP, or RSCS service machines, ... ).

Failure to do so may result in a lot of alert messages being generated even while the system is operating normally.

When user thresholds have been set you will also find the relevant fields highlighted (red and reverse video) on the general user display whenever a threshold has been exceeded for more than the minimum time.

If desired, you can further extend the scope of user monitoring by providing a procedure with additional expert logic, which would then have to be called automatically whenever user threshold monitoring has detected an exception.

**Example:** The following FC PROCESS command will let the WARNING EXEC procedure be called, and the complete alert message be passed to the procedure as arguments: FC PROCESS ERRMSG 6 '317A' EXEC WARNING PASSARG. The following is a very simple example of the WARNING procedure to be called:

```
/* Procedure for informing user about exceeded CPU load and/or   */
/* I/O rate thresholds                          */
 Parse arg 'FCXPEU317A User' userid restofmsg
 'MSGNOH' userid restofmsg
 'MSGNOH' userid 'Please reduce load'
 ...
```

The above procedure informs the user about the problem. Further code could be added, for example, to lower the user's SHARE value, or even to force the virtual machine off the system.

You can also let the users' impact on system performance be reduced automatically by means of the HALVESHR argument of the 'FC USRLIMIT ...' command: the relative SHARE of users which have exceeded their thresholds for the specified period will then be set to half the preceding value, and this process will be repeated until their resource consumption has dropped below the set limit, or until their relative SHARE is down to '1'.

**User exception monitoring:** User exception monitoring specifications are defined by the command 'FC FORCEUSR' (see the "FCONTROL FORCEUSR" subcommand in the *z/VM: Performance Toolkit Reference*). The following commands have been included in the sample initialization file FCONX $PROFILE:

```
    FC FORCEUSR CPULOOP  ALL  5/0
    FC FORCEUSR IOLOOP   ALL  5/0
```

All users which have been in a CPU or I/O loop for more than 5 minutes will receive a warning message, and one of the following messages is generated:

```
FCXxxx338A USER xxxxxxxx in CPU loop during last 5 minutes
```

```
FCXxxx338A USER xxxxxxxx in I/O loop during last 5 minutes
```

However, the looping machines will *not* be forced. See the preceding section "Setup for Threshold Monitoring" on page 126 for guidelines on how to route such messages to another user.

Sending warning messages to **all** machines which fulfill our CPU or I/O loop criteria, and possibly forcing them, may not be desirable, especially if you have large guest systems which may have an I/O rate of more than 500/s during normal operation, or machines running long scientific calculations which might be completely CPU bound for several minutes. Use the EXCLUDE argument of the FC FORCEUSR command to tell Performance Toolkit which machines are to be excluded from user monitoring.

Specification of the correct 'exclude' list becomes even more important if you plan to activate *idle* user monitoring/forcing: better put **all** your service machines in the exclude list before activating it. (The sample initialization file also contains an example for the use of the EXCLUDE argument, but the userids specified are not likely to cover your requirements.)

The machine status will be shown as 'IDLE' or 'LOOP' and the status field highlighted (red and reverse video) on the general user display whenever the appropriate requirements are met.

# Viewing Performance History Data

To facilitate working with performance history data, especially where many different history data files are to be analyzed, a special 'History Data' display can be selected by means of the HISTDATA command. It will show, by default, all the history data files which still have the original default filetypes and which are located on any of your currently accessed disks. This also includes any benchmark log files if a user, I/O device, ISFC logical link, or ISFC end point is benchmarked, and if the log is being written to disk.

You can also restrict the file selection by providing file name, file type, and file mode specifications when entering the HISTDATA subcommand (see the "HISTDATA" subcommand in the *z/VM: Performance Toolkit Reference*).

The following is an example for the HISTDATA display:

```
FCX160       File Selection: All Files with Default Filetypes      History Data

        Move cursor to the history data file you are interested in and
        - enter 'G' to select the file for creating graphics
        - just press ENTER for viewing the data in REDISP format


 .        .        .   _____      .        .        .   _____
 S Filename Filetype FM Created      S Filename Filetype FM Created
 . CHVM4    RMONLOG  V  2003/07/13   . CHVM5    RMONLOG  V  2003/07/13
 . ACUM     HISTSUM  V  2003/07/12   . ACUM     PERFHIST V  2003/07/12
 . CHVM4    RMONLOG1 V  2003/07/12   . CHVM5    RMONLOG1 V  2003/07/12
 . CHVM4    RMONLOG2 V  2003/07/11   . CHVM5    RMONLOG2 V  2003/07/11
 . 071103   PERFLOG1 V  2003/07/11   . 19970711 HISTLOG1 V  2003/07/11
 . 071003   PERFLOG2 V  2003/07/10   . 19970710 HISTLOG2 V  2003/07/10
 . 070903   PERFLOG3 V  2003/07/09   . 19970709 HISTLOG3 V  2003/07/09
 . 070803   PERFLOG3 V  2003/07/08   . 19970708 HISTLOG3 V  2003/07/08
 . 070703   PERFLOG3 V  2003/07/07   . 19970707 HISTLOG3 V  2003/07/07
 . 070403   PERFLOG3 V  2003/07/04   . 19970704 HISTLOG3 V  2003/07/04




 Command ===> _
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 34. History Data Selection Menu*

The example is for a system where the old and the new extended history files are being created and where the PERFSVM machine also acts as a central collector machine (hence the RMONLOG files). No objects are benchmarked, so no benchmark logs are included in the display.

History files are sorted in descending order of creation date, as indicated by the underscore characters above the `Created` columns. This is also the initial default sequence. You can sort the files by file name, file type, or file mode sequence as indicated by the dots above the heading lines: just move the cursor to the respective part of the heading line and press the ENTER key, or enter the SORT command with the required argument (the column heading again).

Select a file for viewing in REDISP or REDHIST format by just moving the cursor in front of the file description and pressing ENTER, or insert a 'g' in front of the file description to select the GRAPHICS menu for that file if you want to display the data graphically.

This history data selection menu can be invoked by entering the HISTDATA command both in performance monitor mode and in basic mode. When invoked directly in basic mode, you will enter a special 'History Data' mode which will still allow you to view the history data files, even in a machine which has not been set up for performance monitoring and which does not have any special CP privilege classes, but you cannot select any other performance data in this case.

The field descriptions are:

**S**
> Cursor selection field. pressing the ENTER key without a selection character will let the data be displayed in REDISP or REDHIST format, and entering a 'G' will display the GRAPHICS menu for the selected file (not accepted in APPC/VM mode).

**Filename**
> File name of the history file

**Filetype**
> File type of the history file

**FM**
> File mode of the history file

**Created**
> Creation date, in format *yyyy/mm/dd*

## Scanning CP Monitor Data Files

While the detailed and summary performance data log files which Performance Toolkit writes to disk should usually allow you to find pointers to the bottleneck which caused a performance problem, it is a fact that only a very small subset of the performance data which are collected during each interval (and could be viewed interactively in real time mode) finds its way into these performance history files.

Where CP monitor records are collected by the MONWRITE utility and written to a disk file or to tape, such a file contains *all* the original monitor data. By providing a facility for scanning monitor data files interactively, Performance Toolkit allows a detailed analysis, based on all the original monitor data, in cases where a performance problem could not easily be understood based only on the relevant PERFLOG files.

The following figure illustrates the data flow:
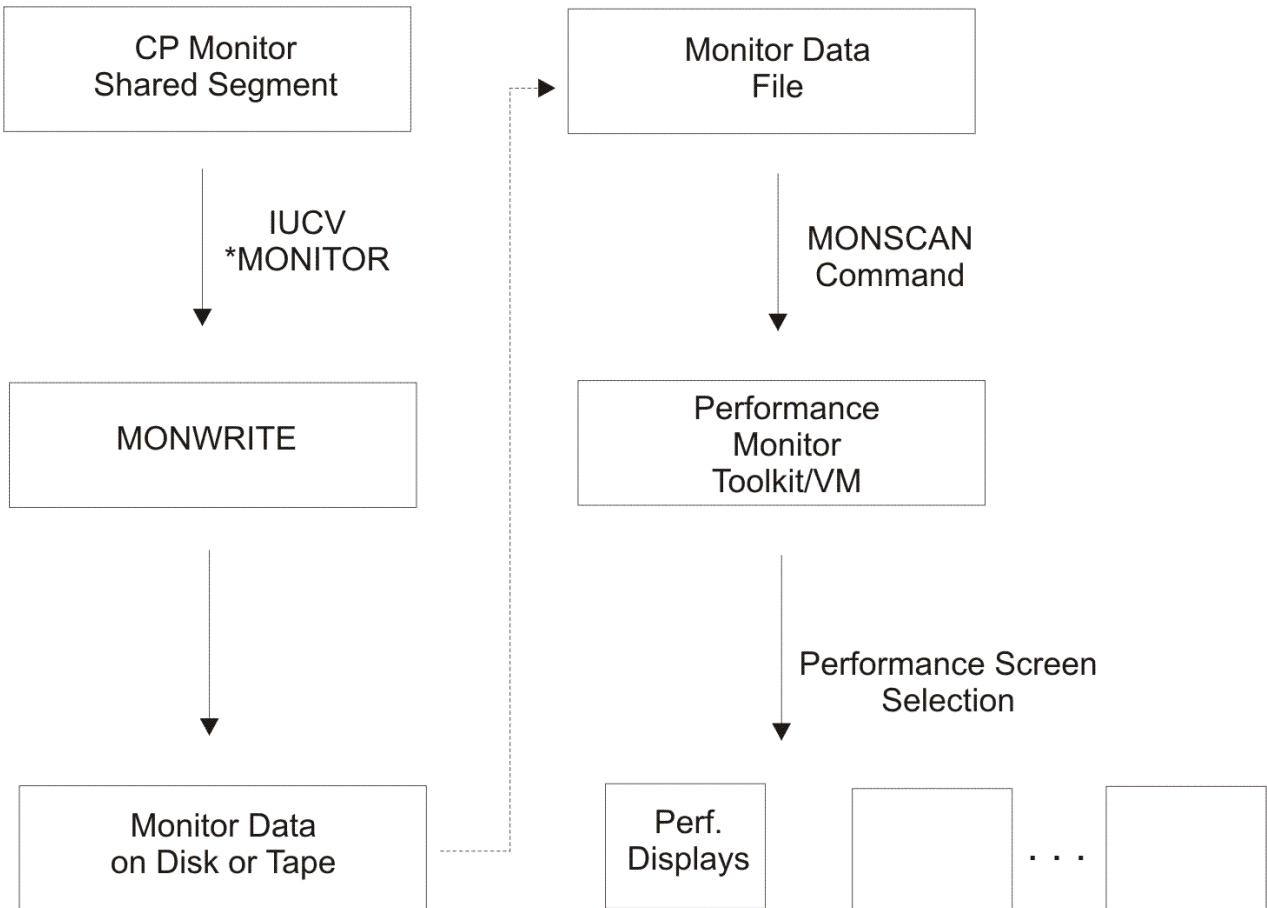
*Figure 35. Data Flow in Monitor Data Scan Mode*

**Differences from normal performance monitor mode:** Although monitor scan mode has been designed to look as similar as possible to real time performance monitor mode, the environment is not the same and you should be aware of the differences.

1. Selection of screens which are based mainly on data obtained directly from CP control blocks is inhibited (for example, the USER details screen or the I/O device details screen). The reason is that these screens could only show data for *current* time, obtained for *the system where Performance Toolkit is running*, while the monitor data file being viewed may be from a different system and certainly contains data from an entirely different period.

2. Some other screens, which although based on CP monitor data, can also provide information obtained from other sources when used in real time performance monitor mode, will contain *only* the CP monitor based data when invoked in monitor scan mode (for example, shared segments screen).

3. The header line of the performance data screens will show the full creation date and time of the monitor data instead of the serial number and processor model. See the system configuration display (SYSCONF) for the type of CPU used.

## Working in Monitor Data Scan Mode

- Invoke Monitor Data Scan Mode as follows

  - **For a disk file:** Enter the command 'MONSCAN DISK *fn ft fm*', specifying at least the file name and file type of the monitor data file which is to be scanned.

  - **For a tape file:** Enter the command 'MONSCAN TAPE TAP*n*', specifying the tape unit where the monitor data tape has been mounted and which has previously been attached to your virtual machine.

- **As a pipe stage:** See "Using CMS Pipelines as Monitor Data Source" on page 133 for more information.

  See the "MONSCAN" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.

  The same initial performance screen selection menu will be shown which you also see after entering normal performance monitor mode.

- Select performance data displays in one of the following ways, just as in normal performance monitor mode:

  - Select a screen with the cursor
  - Key in the number of the screen
  - Key in the fast path command for selecting the screen

  and then press the ENTER key.

  You can view the selected display and switch between different performance displays, just as in normal performance monitor mode. Use the "Performance Toolkit Subcommands and Screen Layouts" chapter in the *z/VM: Performance Toolkit Reference* for further information on the available commands, but be aware of the restrictions mentioned above (that is, some performance data displays cannot be selected because they are based on data obtained directly from CP control blocks, not from CP monitor records).

- Use the NEXTSAMP and SKIPSAMP commands for viewing data for another period. Both commands allow specifying a target time so you can skip directly to a period you are interested in. They differ in the actual processing performed during the skip operation:

  - NEXTSAMP will let all the intermediate data be read and processed too, up to and including the first monitor sample with a time stamp which exceeds, or is at least equal to, the target time you specified. This means you will see all the intermediate measurements in the 'by time' log files and that valid averages can be shown for all screens.

  - SKIPSAMP will just skip any intermediate data records up to the first monitor sample with a time stamp which exceeds, or is at least equal to, the target time you specified. This means that none of the intermediate measurements will be shown in the 'by time' log files and you must expect incorrect averages (there are fields where averages are built by adding values from each sample and then dividing by the number of samples, and these will always show incorrect averages after a SKIPSAMP command).

## Using CMS Pipelines as Monitor Data Source

Performance Toolkit also supports using CMS Pipelines as monitor data input for MONSCAN and BATCH operation. Whereas previously only one monitor data file per session could be processed, now multiple files can all be processed with one step. (Multiple files are those created by the MONWRITE utility with the CLOSE option in one session.) Furthermore, these files may remain in packed format. (Previously, they needed to be unpacked and merged.) With these restrictions eliminated, Performance Toolkit can now process multiple packed monitor data files at once, with no additional disk or memory resources required.

Performance Toolkit provides an FCXPIP MODULE that contains the interface to CMS Pipelines. When the FCXPIP MODULE is invoked, it installs itself as a filter package that allows a PERFKIT stage to invoke Performance Toolkit to process monitor data provided by CMS Pipelines.

To assist with the multiple monitor data files processing in MONSCAN and BATCH modes, the PTKSCAN EXEC procedure has been created. See Appendix C, "Using the PTKSCAN EXEC," on page 187 for the command syntax.

# Scanning Performance Toolkit Trend Data Files

Where performance information has been saved to disk in the form of extended trend data files, such files can be browsed in a special Trend File Scan Mode of Performance Toolkit. It will display the trend data in a manner very similar to the normal realtime monitoring mode. Because it also uses the same data areas, Trend File Scan Mode and realtime monitoring are mutually exclusive.

The **differences from normal performance monitor mode** are:

1. Performance screens can only be selected if trend records with the prerequisite data are part of the file being processed.

2. Only 'average' data can be shown, covering the period between the two reset times at the beginning and the end of the interval for which a specific set of trend records was collected. This means that the '>>Mean>>' lines of all 'by time' logs will always contain the same information as the last set of interval data, that is, no totals/averages over multiple trend data intervals are built.

3. The header line of the performance data screens will show the full creation date and time of the monitor data instead of the serial number and processor model. See the system configuration display (SYSCONF) for the type of CPU used.

## Working in Trend File Scan Mode

- Invoke Trend File Scan Mode by entering the command TRNDSCAN *fn ft fm* in Performance Toolkit basic mode and in a machine where permanent data collection is *not* active. (See the "TRNDSCAN" subcommand in the *z/VM: Performance Toolkit Reference* for a detailed description.)

  The same initial performance screen selection menu will be shown which you also see after entering normal performance monitor mode.

- Select performance data displays in one of the following ways, just as in normal performance monitor mode, for example,

  - Select a screen with the cursor

  - Key in the number of the screen

  - Key in the fast path command for selecting the screen

  and then press the ENTER key.

  You can view the selected display and switch between different performance displays, just as in normal performance monitor mode. Use the the "Performance Toolkit Subcommands and Screen Layouts" chapter in the *z/VM: Performance Toolkit Reference* for further information on the available commands, but be aware of the restrictions mentioned above (that is, some performance data displays cannot be selected because they are based on data not available from the current trend file and displays that usually show data for all users on the system, or for all I/O devices, will show only data for the users or I/O devices for which trend records were found).

- Use the NEXTSAMP and SKIPSAMP commands for viewing data for another period. Both commands allow specifying a target date so you can skip directly to a period you are interested in. They differ in the actual processing performed during the skip operation:

  - NEXTSAMP will let all the intermediate data be read and processed too, up to and including the first set of data with a time stamp which exceeds, or is at least equal to, the target date you specified.

    This means you will see all the intermediate measurements in the 'by time' log files.

  - SKIPSAMP will just skip any intermediate data records up to the first set of records with a time stamp which exceeds, or is at least equal to, the target date you specified.

    This means none of the intermediate measurements will be shown in the 'by time' log files.

## Benchmarking Objects

When the performance of a subset of such Performance Toolkit objects as users, I/O devices, ISFC logical links, ISFC end points, or PCI functions should be investigated more closely, a log-type display with performance data for a specific object on a "per sample" basis can provide help with analyzing the behavior of such an object.

To keep storage requirements and data collection overhead low, such log files are not created automatically. You can instruct Performance Toolkit to build log files for selected objects by using the FC BENCHMRK command. See the "FCONTROL BENCHMRK" subcommand in the *z/VM: Performance Toolkit Reference* for more information.

Log files will then be built for each selected object, with one entry per sample interval.

After benchmarking has been activated, you can select a benchmark overview menu that shows the logs currently being built. The following is an example of such a display.

```
FCX173       CPU nnnn  SER nnnnn        BENCHMRK Log Data      Perf. Monitor

------     .
  Object    Log File
S ID        Name       Description
. CFT2NDA   ISFLLOG    ISFC Logical Link activity log
. LXA00001  UCOMMLOG   User IUCV and VMCF communications log
. LXA00001  UPAGELOG   User paging load log
. LXA00001  URESPLOG   User response time log
. LXA00001  USERLOG    User resource consumption log
. LXA00001  USTATLOG   User wait state log
. LXA00001  UTRANLOG   User resources per transaction log
. RSTL3     UCOMMLOG   User IUCV and VMCF communications log
. RSTL3     UPAGELOG   User paging load log
. RSTL3     URESPLOG   User response time log
. RSTL3     USERLOG    User resource consumption log
. RSTL3     USTATLOG   User wait state log
. RSTL3     UTRANLOG   User resources per transaction log
. TCPIP     MTUSRLOG   User multitasking data log
. TCPIP     UCOMMLOG   User IUCV and VMCF communications log
. TCPIP     UPAGELOG   User paging load log
. TCPIP     UQDIOLOG   User QDIO activity log
. TCPIP     URESPLOG   User response time log
. TCPIP     USERLOG    User resource consumption log
. TCPIP     USTATLOG   User wait state log
. TCPIP     UTRANLOG   User resources per transaction log
. 00000001  ISFELOG    ISFC End Point activity log
. 2A60      CACHDLOG   I/O device CU cache data log
. 2A60      CACHELOG   Extended function cache data log
. 2A60      CPOWNLOG   CPOWNed I/O device log
. 2A60      DEVLOG     General I/O device performance log
. 2B20      DEVLOG     General I/O device performance log
. 2B20      HPFLOG     HPF I/O device activity log
. 4000      DEVLOG     General I/O device performance log
. 4000      HPFLOG     HPF I/O device activity log
. 4000      VOLLOG     DASD Volume performance log


 Select an object log with cursor and hit ENTER
 Command ===> _
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 36. Benchmark Log Selection Menu*

This example shows the types of logs that might be generated when benchmarking has been activated for users LXA00001, RSTL3, and TCPIP, for I/O devices 2A60, 2B20, and 4000, for ISFC endpoint 1, and for ISFC logical link CFT2NDA. Log file creation depends on the existence of the following:

• Monitor domains that are activated for the user, I/O device, logical link, or endpoint

• The user, I/O device, ISFC logical link, or ISFC endpoint existence itself.

Note the following in Figure 36 on page 135:

• The absence of seeks data for all I/O devices, because SEEKS event data collection was probably not enabled.

• The device 2A60 has cache data logs, because the disk was connected to a cache control unit.

• CPOWNLOG has been built, because that disk contains system areas.

• Other devices have general I/O logs only, because these devices are not disks.

• The benchmarked device 2B20 is a HyperPAV alias that generates multiple benchmark logs, including DEVLOG and HPFLOG.

• The benchmarked device 4000 is a HyperPAV base that generates multiple benchmark logs, including DEVLOG, HPFLOG, and VOLLOG.

• The benchmarked user LXA00001 is running on a virtual IFL processor and multithreading is enabled for the system that provided the data.

The logs are initially sorted by object ID. You can change the sequence to have them sorted by the log name instead.

**Field Descriptions:**

**S**
> The cursor selection field.

**Object ID**
> The object for which performance is logged. An object can be a user ID, a device number of I/O device, an ISFC logical link name, an ISFC end point ID, or a PCI function ID, for example.

**Log File Name**
> The command with which the log data can be selected for display.

**Description**
> The log file description.

# Monitoring Linux Internal Performance Data

## Concept

Any kind of Linux internal performance data must be collected inside the Linux system, that is, some kind of data gathering mechanism must be installed before such data can be retrieved by a VM performance monitor. The Performance Toolkit generates Linux internal data reports from two different sources:

- The APPLDATA call class that allows the CP monitor to cut application records containing *summary* Linux performance data similar to the APPLDATA provided by TCP/IP and the Shared File System. This information is used to prepare summary Linux performance displays. The data can also be saved in history files on VM.

- RMF PM Linux data gatherer that allows retrieval of *detailed* Linux performance data via TCP/IP over its DDS server interface. The Performance Toolkit only uses this information to prepare detailed performance displays for selected Linux systems on request. History data is kept in the Linux filesystem, but none of it is saved in history files on VM. z/VM Performance Toolkit support for rmfpms has been stabilized and may cease to function as the underlying Linux system evolves. Support for the Linux rmfpms agent has already been withdrawn and is available on an as-is basis.

In each case installation and activation of the corresponding data gathering code is a prerequisite for display of such data by the Performance Toolkit. Both methods are described in more detail in the following sections.

## Summary Linux Performance Data

Summary data is collected from the CP MONITOR data. On the z/VM system you need to start collection of APPLDATA from the monitor using:

```
CP MONITOR SAMPLE ENABLE APPLDATA ALL
```

The LINUX Kernel needs the "Linux-z/VM monitor stream" patch applied. LINUX must be running on the same system as PERFKIT. When you start LINUX you need to enter the following commands to start data collection:

```
echo 1 > /proc/sys/appldata/timer
echo 1 > /proc/sys/appldata/mem
echo 1 > /proc/sys/appldata/os
echo 1 > /proc/sys/appldata/
net_sum
```

and the following to set the interval time:

```
echo 1 > /proc/sys/appldata/interval
```

For this command the '1' represents the sample interval, expressed in seconds of virtual CPU time to be used by the Linux system before refreshing the data collected by the CP monitor – in this case, 1 second.

# Detailed Linux Performance Data

Data retrieval and display of Linux detailed performance data is based on the Linux DDS server interface, originally written for use with RMF PM. This means that:

- The DDS interface must be installed and active on all Linux systems that are to be monitored.
- Performance data is permanently collected only in the Linux systems themselves, and performance history data is available only from the filesystems of the Linux systems. PERFKIT does **not** collect and save any Linux internal performance data in a permanent way.
- Performance data retrieval is based on XML requests, sent to the Linux systems via TCP/IP. Only the data actually needed for building a specific Linux performance report are retrieved.

The following diagram illustrates the components involved.
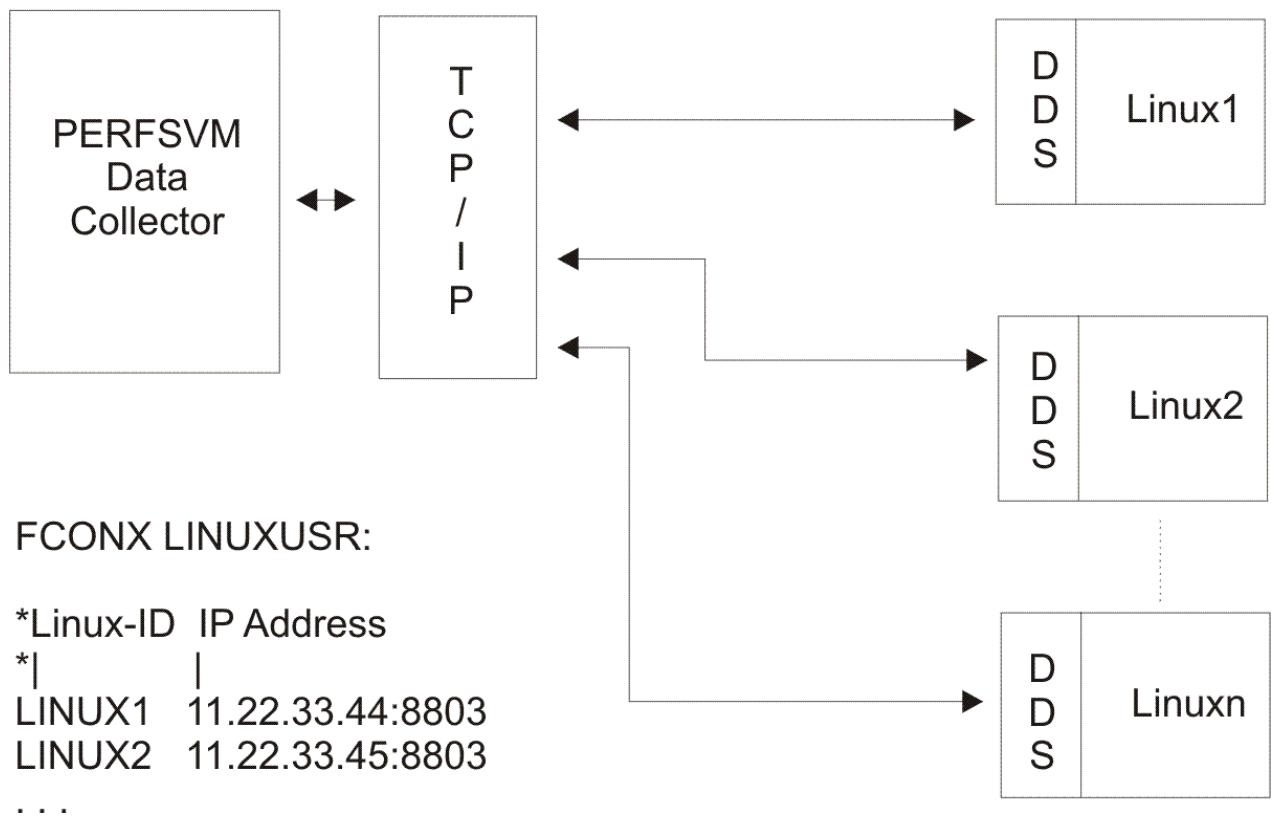


*Figure 37. Linux Performance Data Retrieval*

## Implementation

The following steps are required for implementing Linux performance data retrieval by Performance Toolkit:

1. Implement the most recent level of the RMF PM DDS code on each of the Linux systems that are to be monitored.
2. Prepare an FCONX LINUXUSR control file. This control file must be set up to contain one record for each Linux system, with the Linux system's identification and the IP address of its DDS interface, and it must be located on one of the minidisks accessed by the PERFSVM machine.

   All of the Linux names specified in this file will appear on the Linux Systems Selection Screen (FCX223). To avoid unsuccessful data retrieval attempts, you should include only the RMF PM interface

IP addresses of Linux guests whose interfaces are always up and running while the guests are logged on.

The following is an example for such a control file.

```
****************************************************************
**     Initialization file with IP address definitions     **
**     for Linux systems that may have to be monitored.     **
****************************************************************
*
*Linux-ID   IP Address for DDS interface
*|          |
GFREE18     9.164.137.116:8803
LNXBENKE    9.164.155.106:8803
LNXBENK1    9.164.155.190:8803
...
```

3. Uncomment the command:

```
FC MONCOLL LINUXUSR ON TCPIP TCPIP
```

in the FCONX $PROFILE initialization file, to let the TCP/IP interface needed for communicating with the Linux DDS interfaces be activated automatically during initialization of Performance Toolkit.

After these steps are completed you are ready for retrieving Linux performance data.

## Usage Notes for Linux Data Retrieval

As mentioned previously, Linux performance data collection is done asynchronously in each Linux system. This has many operational advantages, but in some cases it also leads to slightly different behavior from 'normal' VM performance data displays:

- The data collect cycles of CP monitor and Linux DDS data gatherer are not synchronized. While Performance Toolkit will try to retrieve Linux internal performance data for a period similar to the monitoring interval currently used for displaying VM data, in order to allow comparing the VM and Linux values, the start and end times will not usually be exactly the same. You must be aware of this when comparing the two sets of performance data.

- Because all Linux performance data have to be retrieved using TCP/IP, Performance Toolkit only retrieves what is currently needed to avoid unnecessary overhead. Even so, there may be a noticeable delay from the moment a specific display is requested until the complete data can be shown, especially if the TCP/IP connection to the Linux system is over a real telephone line and not just a virtual CTCA on the same VM.

- For Linux metrics where a whole series of values can be returned for a 'value' or metric, for example, in the case of metrics for 'by process', 'by network device' or 'by filesystem' values, only a maximum of the 40 highest values is currently retrieved.

  Up to 20 processes, network devices, or file systems will actually be displayed in the sorting sequence selected. Because the different values pertaining to a process, for instance, were all collected using the 'highest 40 values' logic, it is possible that a particular metric may not have returned a value for a process that has a relatively high value for another metric. Dots will be inserted in such cases instead of a normal value. While this may not look complete, realize that *the value was not returned because it was too small*, that is, it was not within the 40 largest ones.

- Availability for all of the retrieved metrics depends on the appropriate DDS interface level, and in some cases it also depends on the version of the Linux kernel. A Linux DDS error message, such as 'GPM0507I DDS could not retrieve data for the specified date and time' will be displayed, and dots inserted for some of the data fields, when either the DDS or Linux kernel level is too low, or if no data are available for the selected period for some other reason.

- The Linux data retrieval interface can only process one request at a time. To prevent interference by other requests, and to guarantee a complete set of data for each request, all requests are serialized. This may lead to additional delays if many users ask for different Linux reports simultaneously.

# Chapter 6. Operation in Batch Mode
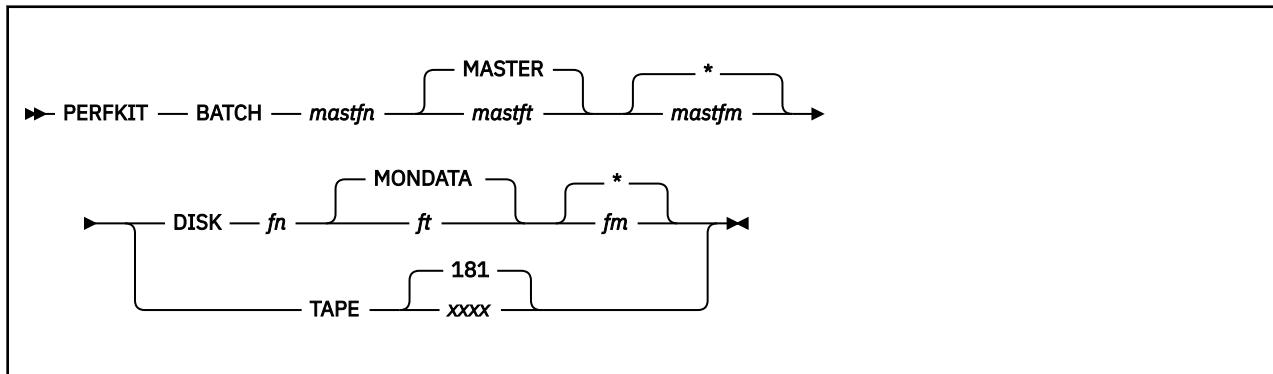
## General Guidelines

In the past, printed reports and performance history files had traditionally been created by programs that processed monitor data saved to disk or tape in batch mode. Performance Toolkit has been designed to process CP monitor data directly from the monitor saved segment (MONDCSS), allowing you to analyze the system's performance in real time. Printed reports and performance history files can be produced by Performance Toolkit as a by-product of normal real time operation without the overhead of first saving the monitor data in a disk or tape file for later batch processing. See "The FCONX REPORTS File" on page 116 for guidelines on how to set up the program to automatically print selected reports and "Creating Extended Summary and Trend Files" on page 111 for instructions on how to save performance data in 'trend' and 'summary' history files.

If, however, you need to process a monitor data file saved on disk or tape, either for detailed analysis of a problem on your system or for analysis of data obtained from another z/VM system, Performance Toolkit provides two methods of processing saved monitor data. The first method, interactive monitor data scan mode using the MONSCAN command (see the "MONSCAN" subcommand in the *z/VM: Performance Toolkit Reference*) allows you to view the saved monitor data as if you were using Performance Toolkit in real time. The menus and screens appear the same, however, only reports derived from the monitor data are available. The second method, operation in batch mode, allows you to process the saved monitor data using a set of control input and output files to create reports and performance history files. This chapter explains how to use Performance Toolkit in batch mode.

## Batch Mode Invocation

Batch mode is invoked by entering the 'BATCH' parameter after the PERFKIT command. The command format is:

```
>>-- PERFKIT -- BATCH -- mastfn --+-- MASTER --+--+------------+-->
                                  |            |  |    *       |
                                  +-- mastft --+  +-- mastfm --+

>--+-- DISK -- fn --+-- MONDATA --+--+------+---><
   |                |             |  |  *   |
   |                +-- ft -------+  +- fm -+
   |
   +-- TAPE --+-- 181 ---+
              |          |
              +-- xxxx --+
```

**BATCH**
   indicates PERFKIT batch mode. All specifications are made in the SETTINGS file which is expected to contain standard FCONTROL commands. All report names specified in the REPORTS file must follow the rules for Performance Toolkit report files as described in "The FCONX REPORTS File" on page 116.

**mastfn**
   is required and specifies the file name of the MASTER file.

**mastft**
   specifies the file type of the MASTER file. The default file type is MASTER.

**mastfm**
   specifies the file mode where the MASTER file is located. If the file mode is not specified, all accessed disks and directories are searched for the specified file name and file type.

**DISK**
   indicates that a monitor data file that has been saved to disk will be processed.

**fn**
is required and specifies the file name of the monitor data file.

**ft**
is the file type of the monitor data file. The default file type is MONDATA.

**fm**
is the file mode of the monitor data file. If the file mode is not specified, all accessed disks and directories are searched.
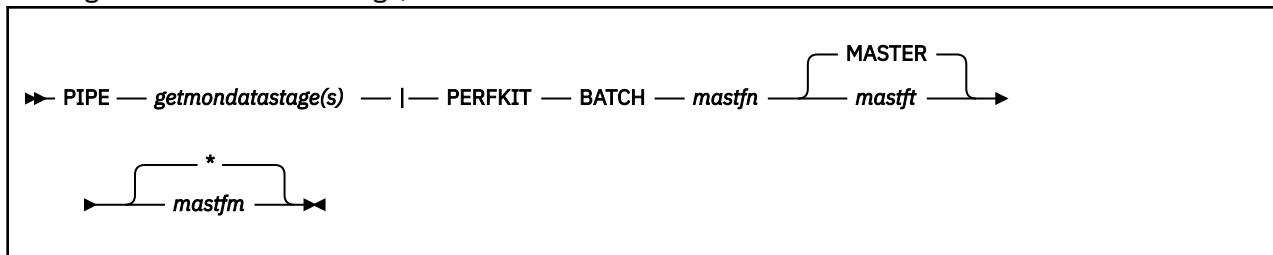
**TAPE**
indicates a monitor data file in MONWRITE format on an unlabeled tape is to be processed.

**xxxx**
is the device number of the tape unit where the tape with the monitor data file has been mounted. If the device number is not specified, a device number of 181 is used.

**Note:** If the monitor data file is on a tape device, the tape drive must be attached to your virtual machine and the tape volume mounted prior to running the PERFKIT command for batch mode.

If using PERFKIT as a PIPE stage, the command format will instead be as follows:

```
►►─ PIPE ── getmondatastage(s) ── |── PERFKIT ── BATCH ── mastfn ──┬── MASTER ──┬──►
                                                                   └── mastft ──┘

      ┌──── * ────┐
   ►──┴── mastfm ─┴──►◄
```
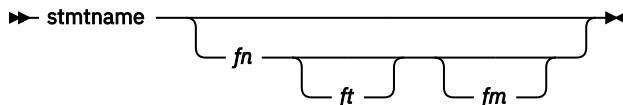
*getmondatastage(s)*
The PIPE stage(s) used to get the monitor data and pass it to the primary input of the PERFKIT stage.

See "Using CMS Pipelines as Monitor Data Source" on page 133 for more information.

## MASTER File

All files used, except for the monitor data file, are pointed to by the MASTER file. Note that these files must be either fixed or variable format, with a record length less than or equal to 80. Both input (parameter files) and output files are specified in the MASTER file.

The format for MASTER file statements is:

```
►►─ stmtname ──┬──────────────────────────────────┬──►◄
               └── fn ──┬──────────┬──┬──────────┬─┘
                        └── ft ────┘  └── fm ────┘
```

**stmtname**
is the name of an input or output file. The stmtname is used to associate a file ID with a specific input or output file. The following list contains the valid names:

- LISTING
- LOG
- REPORTS
- RUNFILE
- SETTINGS
- SUMMARY
- SUMREC
- TREND
- TRENDREC

See and for full descriptions.

**fn, ft, fm**

describes the file ID of the associated input or output file.

- **Input file** default file IDs:

  – File name 'FCONX',

  – File type of 'stmtname'. For example, 'FCONX REPORTS' to indicate the input file which contains the reports to be created.

  – File mode '*'. All accessed disks and directories are searched for the specified file name and file type.

- **Output file** default file IDs (except SUMMARY and TREND):

  – File name 'FCONX',

  – File type of 'stmtname'. For example, 'FCONX LISTING' to indicate the output file which contains the created reports.

  – File mode 'A1'.

- **SUMMARY and TREND** default file IDs:

  – File name is the z/VM system's system identification.

  – File type of 'FCXSUM' for SUMMARY files and 'FCXTREND' for TREND files. If an existing summary file with the same file identifier already exists, consecutive numbers from '01' to '99' will be appended to the file type if it is up to 6 characters long, or to the first 6 characters of the file type if it is longer, to create unique file identifiers.

  – File mode 'A1'.

The following diagram illustrates the files involved.
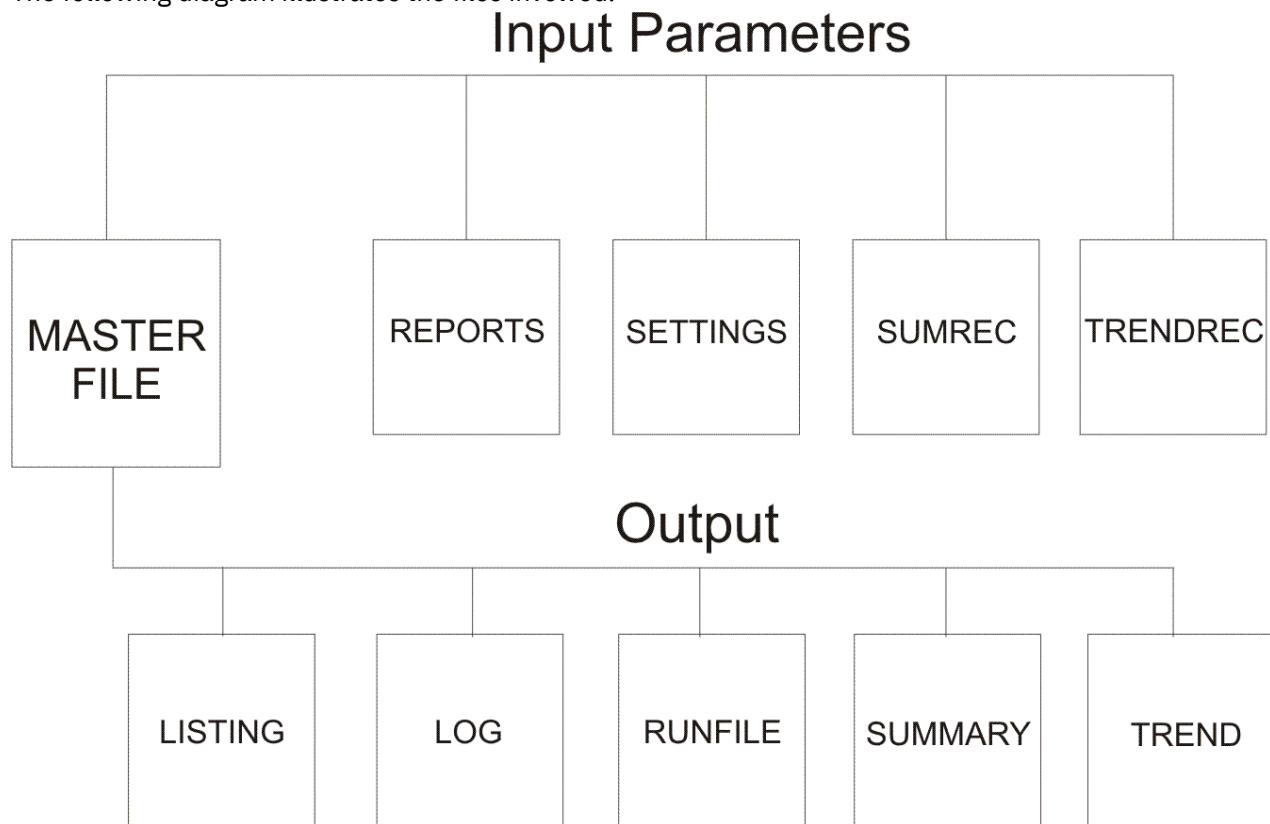


*Figure 38. MASTER file definitions*

## Parameter (input) files:

**REPORTS**

Determines which reports are to be produced and contains any report-specific parameters.

**SETTINGS**

Controls the running of Performance Toolkit and contains report and SUMMARY and TREND file start and end times.

**SUMREC**

Controls the type and number of SUMMARY file records to be created for each period.

**TRENDREC**

Controls the type and number of TREND file records to be created.

**Usage Notes:**

- The REPORTS and SETTINGS files are prerequisites for batch processing. If they are not explicitly defined in the MASTER file, the defaults are used. If the REPORTS statement is missing from the MASTER file, the default MONDATA report is generated. In case of a missing SETTINGS statement, all Performance Toolkit settings are set to their default values.

## Output files:

**LISTING**

Contains all performance reports as a single LISTING file.

**LOG**

Logs Performance Toolkit use in batch mode. This file is appended to each time log information is added.

**RUNFILE**

Contains detailed information on the most recent batch run. This file is used primarily for problem determination.

**SUMMARY**

Defines the file ID of the SUMMARY output file. See "Creating Extended Summary and Trend Files" on page 111 for more information on summary files and their definition.

**TREND**

Defines the file ID of the TREND output file. See "Creating Extended Summary and Trend Files" on page 111 for more information on trend files and their definition.

**Usage Notes:**

- The LOG and RUNFILE output files should always be created. The LISTING file should be created if at least one report is successfully generated.

- SUMMARY and TREND files are only created when they are specified.

- Each batch mode run recognizes only one MASTER file which is specified on the PERFKIT command. You may, however, find it convenient to establish multiple MASTER files to be used at different times. For example, you might want one MASTER file for processing data from one z/VM system and another MASTER file (adapted to the different requirements for user and I/O data) for processing data from another system. These MASTER files may point to common input or output files.

- If SUMREC and/or TRENDREC files are not specified and SUMMARY and/or TREND files are specified, all summary and trend records are created. SUMREC and/or TRENDREC files can be used to restrict record creation to prevent building large and detailed files when they are not needed.

## REPORTS File

The REPORTS file is an input file that lets you select the performance reports to be generated in the LISTING file.

When you specify the 'BATCH' option on the PERFKIT command, the REPORTS file must contain Performance Toolkit report specifications. See "The FCONX REPORTS File" on page 116 for a detailed

description of the file and its uses in Performance Toolkit. The MONDATA report file is the default report used for processing if no REPORTS file is specified in the MASTER file.

## SETTINGS File

The SETTINGS file is the main input file for controlling PERFKIT BATCH operation.

When you specify the 'BATCH' option on the PERFKIT command, the SETTINGS file contains FCONTROL subcommands which control how the reports are generated and formatted. The following table lists the most commonly used commands for batch processing.

| Command | Argument | Description |
|---------|----------|-------------|
| FC MONCOLL | RESET | Define the start and end times for the different periods to be used for report, trend, and summary files. See the "FCONTROL MONCOLL" subcommand in the *z/VM: Performance Toolkit Reference*.<br><br>Note that the FC MONCOLL RESET statements must be provided in the SETTINGS file for summary and/or trend reports when the 'BATCH' option is specified. If the corresponding FC MONCOLL RESET statements are not provided, summary and/or trend reports will not be generated. |
| FC SETTINGS | BYTIME | Sets the interval to be used to calculate averages for the detail lines of all of the general system 'by time' logs. |
| | INTERIM | Sets the 'interim' interval to be used for the printing of INTERIM reports. |
| | MAXDEVS | Sets the maximum number of I/O devices to be included in the I/O device reports. |
| | MAXUSERS | Sets the maximum number of users to be included in the user reports. |
| | PAGESIZE | Sets the number of print lines per printed page. |
| | SYSTEM | Allows you to set a character string used to identify the batch run in the report headers. |
| | SYSTEMID | Allows you to set the system identification or node name in the reports. |
| FC UCLASS | | Defines groups of users for which averages are to be included on user reports. If you have many users on your system, you may also want to specify FC SETTINGS MAXUSERS to limit the amount of user data reported. See the "FCONTROL UCLASS" subcommand in the *z/VM: Performance Toolkit Reference*. |

**Usage Notes:**

- Any command can be included in the SETTINGS file, however, not all the commands are appropriate for this file.
- An asterisk (*) in column 1 of the SETTINGS file indicates a comment line and will be ignored during processing.

- Though it is possible during batch operation to enter any command from the Performance Toolkit fullscreen console, such input is ignored in order to avoid any interference with batch processing. Any command used to control Performance Toolkit batch mode must be included in the SETTINGS file.

**Example:**

```
*
* Settings for GDLVMK4 test data processing
*
FC SET SYSTEM   "Testing GDLVMK4 test data"
*C SET SYSTEMID xxxxxxxx
FC SET BYTIME   15 minutes
FC SET INTERIM  30 minutes
FC SET MAXDEVS  100
FC SET MAXUSERS 100
FC SET PAGESIZE 65
* Report start/end times
FC MONCOLL RESET 08:00:00r_p 12:00:00p
* Summary file start/end times
FC MONCOLL RESET 08:00:00r_s 09:00s 10:00s 11:00s 12:00s (merge
* Trend file start/end times
FC MONCOLL RESET 08:00:00r_t 12:00t (merge
```

*Figure 39. Sample SETTINGS File*

# SUMREC and TRENDREC Files

SUMREC and TRENDREC files are input files that control summary and trend file record creation. See the "Layout of Extended Trend File Records" and "Layout of Extended Summary File Records" appendices in the *z/VM: Performance Toolkit Reference* for a description of their record specification format. If SUMREC and/or TRENDREC files are not specified and SUMMARY and/or TREND files are specified, all summary and trend records are created. SUMREC and TRENDREC files can be used to restrict record creation to prevent building large and detailed files when they are not needed.

# LISTING File

The LISTING file is an output file. It contains all reports specified in the REPORTS file produced by a single invocation of the PERFKIT BATCH command. The LISTING file contains carriage control characters in column 1.

The first page of each LISTING file contains a table of contents which shows the reports listed in the file. The actual performance reports are appended to the LISTING file in the order in which they are defined in the REPORTS file, except for INTERIM reports, which are appended to the end of the reports for the overall reporting period. The INTERIM reports are listed together with their own table of contents for each INTERIM interval.

If a LISTING statement is included in the MASTER file, then a new LISTING file will be created for each batch run, replacing any existing one of the same name. This also occurs even if no explicit output file ID is defined and instead the default listing file name and file type are used.

If report output is generated but **no** LISTING statement is found in the MASTER file, then the output will be created using the default LISTING file name and file type. Note, however, that if a file with the same name already exists, then the new records will just be appended to it, in order to prevent unintentional loss of existing data.

# LOG File

The LOG file is an output file that keeps a running log of PERFKIT BATCH invocations. Each invocation of the PERFKIT command in batch mode will add entries to the LOG file. The entries include the start and end time for each run, the CPU consumption, and the return code. If the file already exists, new entries from each invocation are appended to the end of the file. If the file does not exist, a new file is created.

**Example**

```
17 Sep 2020 10:36:45 Performance Toolkit; Version FL720 at Fix Level 00
17 Sep 2020 10:36:45 Performance Toolkit; Arguments passed for BATCH mode processing:
17 Sep 2020 10:36:45 Performance Toolkit; BATCH SYSVMK4 DISK SYSVMK4
17 Sep 2020 10:36:54 Performance Toolkit; Reduction Started
17 Sep 2020 10:39:36 Performance Toolkit; Vtime=3.63, Ttime=4.33, Connect time 171 sec.
17 Sep 2020 10:39:36 Performance Toolkit; Reduction Completed, Rc= 0
17 Sep 2020 13:41:06 Performance Toolkit; Version FL720 at Fix Level 00
17 Sep 2020 13:41:06 Performance Toolkit; Arguments passed for BATCH mode processing:
17 Sep 2020 13:41:06 Performance Toolkit; BATCH SYSVMK4 DISK SYSVMK4
17 Sep 2020 13:41:13 Performance Toolkit; Reduction Started
17 Sep 2020 13:43:58 Performance Toolkit; Vtime=3.55, Ttime=4.09, Connect time 172 sec.
17 Sep 2020 13:43:58 Performance Toolkit; Reduction Completed, Rc= 0
```

*Figure 40. Sample LOG File*

# RUNFILE File

The RUNFILE file is an output file that contains information on the latest PERFKIT BATCH invocation. This information includes:

- Informational, error, and warning messages
- Information on the monitor data used
- Notice of which files were used during processing
- Listing of input and output file specifications
- FCONTROL subcommands used to control the report output

The RUNFILE is always produced using the file ID specified by the MASTER file RUNFILE statement. If a file with the specified file ID already exists, the existing file will be replaced.

## Example

Here is a sample RUNFILE:

```
 17 Sep 2020 13:41:06 Performance Toolkit; Version FL720 at Fix Level 00
 17 Sep 2020 13:41:06 Performance Toolkit; Arguments passed for BATCH mode processing:
 17 Sep 2020 13:41:06 Performance Toolkit; BATCH SYSVMK4 DISK SYSVMK4
   13:41:06 * Processing MASTER   file 'SYSVMK4  MASTER   *'
   SYSVMK4  MASTER   *: ***********************************************************
   SYSVMK4  MASTER   *: *  MASTER File for SYSVMK4 Tests                          *
   SYSVMK4  MASTER   *: ***********************************************************
   SYSVMK4  MASTER   *: *** Input Parameter Files ***
   SYSVMK4  MASTER   *: SETTINGS      SYSVMK4 SETTINGS *
   SYSVMK4  MASTER   *: REPORTS       SYSVMK4 REPORTS  *
   SYSVMK4  MASTER   *: SUMREC        SYSVMK4 SUMREC   *
   SYSVMK4  MASTER   *: TRENDREC      SYSVMK4 TRENDREC *
   SYSVMK4  MASTER   *: *** Output Files ***
   SYSVMK4  MASTER   *: LISTING       SYSVMK4 LISTING  A
   SYSVMK4  MASTER   *: LOG           BATCH   LOG      A
   SYSVMK4  MASTER   *: RUNFILE       BATCH   RUNFILE  A
   SYSVMK4  MASTER   *: SUMMARY       SYSVMK4 FCXSUM   B
   SYSVMK4  MASTER   *: TREND         SYSVMK4 FCXTREND B
   13:41:06 =Basic=> FC SET SUMREC   SYSVMK4  SUMREC   *
   13:41:07 =Basic=> FC SET TRENDREC SYSVMK4  TRENDREC *
   13:41:07 =Basic=> FC SET REPFILID SYSVMK4  LISTING  A
   13:41:08 =Basic=> FC SET SUMFILID SYSVMK4  FCXSUM   B
   13:41:08 =Basic=> FC SET TRDFILID SYSVMK4  FCXTREND B
   13:41:09 =Basic=> FC TRACE RESET ON
 FCXCMK603I RESET trace activated
1  13:41:09 * Processing SETTINGS file 'SYSVMK4  SETTINGS *'
0  SYSVMK4  SETTINGS *: * Settings for SYSVMK4 data tests
   SYSVMK4  SETTINGS *: *
   SYSVMK4  SETTINGS *: FC SET system  "Testing SYSVMK4 data"
   13:41:09 =Basic=> FC SET system  "Testing SYSVMK4 data"
   SYSVMK4  SETTINGS *: *C SET systemid xxxxxxxx
   SYSVMK4  SETTINGS *: FC SET bytime    5 minutes
   13:41:09 =Basic=> FC SET bytime    5 minutes
   SYSVMK4  SETTINGS *: FC SET interim  15 minutes
   13:41:10 =Basic=> FC SET interim  15 minutes
   SYSVMK4  SETTINGS *: FC SET maxusers 100
   13:41:10 =Basic=> FC SET maxusers 100
```

```
    SYSVMK4  SETTINGS *: FC SET maxdevs  100
    13:41:11 =Basic=> FC SET maxdevs  100
    SYSVMK4  SETTINGS *: FC SET pagesize 65
    13:41:11 =Basic=> FC SET pagesize 65
    SYSVMK4  SETTINGS *: * Report start/end times
    SYSVMK4  SETTINGS *: FC MONCOLL RESET 08:50:00r_p 09:50:00p
    13:41:11 =Basic=> FC MONCOLL RESET 08:50:00r_p 09:50:00p
    SYSVMK4  SETTINGS *: * Summary file start/end times
    SYSVMK4  SETTINGS *: FC MONCOLL RESET 09:10:00r_s 09:40s (merge
    13:41:12 =Basic=> FC MONCOLL RESET 09:10:00r_s 09:40s (merge
    SYSVMK4  SETTINGS *: * Trend file start/end times
    SYSVMK4  SETTINGS *: FC MONCOLL RESET 09:20:00r_t 09:50t (merge
    13:41:12 =Basic=> FC MONCOLL RESET 09:20:00r_t 09:50t (merge
    13:41:13 * SETTINGS file processing completed with RC  0
1   13:41:13 * Processing REPORTS  file 'SYSVMK4  REPORTS  *'
0   SYSVMK4  REPORTS  *: * Reports for SYSVMK4 data tests
    SYSVMK4  REPORTS  *: AUXLOG
    SYSVMK4  REPORTS  *: CPU
    SYSVMK4  REPORTS  *: PAGELOG
    SYSVMK4  REPORTS  *: PROCLOG
    SYSVMK4  REPORTS  *: SCHEDLOG
    SYSVMK4  REPORTS  *: SYSSUM
    SYSVMK4  REPORTS  *: USER
    13:41:13 =Basic=> FC SET REPORTS SYSVMK4  REPORTS  *
    13:41:13 * REPORTS file processing completed with RC  0
117 Sep 2020 13:41:13 Performance Toolkit; Reduction Started
0   13:41:14 =Basic=> Monscan SYSVMK4  MONDATA  *
    13:41:14 * Monitor data start at 08:40:03 on 2007/06/24
 FCXPMC811T Proc/Reset init call at 08:40:03
    13:41:48 =Monscan=> Nextsamp EOF
 FCXPMC811T Proc/Reset process call for 08:50:00 .... PI..
 FCXPMC811T Proc/Reset process call for 09:10:00 .... ...S
 FCXPMC811T Proc/Reset process call for 09:20:00 .I.. .IT.
 FCXPMC811T Proc/Reset process call for 09:40:00 ...S ...S
    SYSVMK4  SUMREC    *: *---------------------------------------------------------------*
    SYSVMK4  SUMREC    *: * Performance Toolkit; Summary File Definition. This file  *
    SYSVMK4  SUMREC    *: * defines the kind of records to be generated when a summary     *
    SYSVMK4  SUMREC    *: * file build request is processed ('S'-suffix appended to FC   *
    SYSVMK4  SUMREC    *: * MONCOLL RESET times).                                        *
    SYSVMK4  SUMREC    *: * Always built:                                               *
    SYSVMK4  SUMREC    *: * Number              Data                                    *
    SYSVMK4  SUMREC    *: * X'FC00'             System configuration data               *
    ...
    ...
    SYSVMK4  SUMREC    *: * For the user related record types (DSPACES, USER, SFS, MTUSER, *
    SYSVMK4  SUMREC    *: * TCPIP, RSK) record creation can be further restricted with the *
    SYSVMK4  SUMREC    *: * USERID keyword, by defining that records should be build only  *
    SYSVMK4  SUMREC    *: * for the specified user IDs. Similarly, the DEVNO keyword      *
    SYSVMK4  SUMREC    *: * allows restricting summary record creation for DASD and SEEKS  *
    SYSVMK4  SUMREC    *: * records to the disks whose device numbers are explicitly      *
    SYSVMK4  SUMREC    *: * specified.                                                    *
    SYSVMK4  SUMREC    *: *                                                               *
    SYSVMK4  SUMREC    *: * Multiple USERID and DEVNO selection record sare accepted.     *
    SYSVMK4  SUMREC    *: *---------------------------------------------------------------*
    SYSVMK4  SUMREC    *: * General Summary Record Selection
    SYSVMK4  SUMREC    *: RECORDS   CHANNEL NSS DSPACES USER DASD SEEKS VSWITCH SFS MTUSER TCP...
    SYSVMK4  SUMREC    *: * Specific User Selection (for USER, SFS, MTUSER and TCPIP Records)
    SYSVMK4  SUMREC    *: * USERID    userid1 userid2 userid3  userid4  ...
    SYSVMK4  SUMREC    *: * USERID    OPERATOR RSCS     TCPIP     VTAM      RSKSERV
    SYSVMK4  SUMREC    *: * Specific DASD Selection (for DASD and SEEKS Records)
    SYSVMK4  SUMREC    *: * DEVNO     devno1 devno2 devno3  devno4  ...
    SYSVMK4  SUMREC    *: * DEVNO     4381     4382     4384     4386     4388
    SYSVMK4  SUMREC    *: * DEVNO     4398     4399
 FCXPMC811T Proc/Reset process call for 09:50:00 PIT. PIT.
    SYSVMK4  TRENDREC *: *---------------------------------------------------------------*
    SYSVMK4  TRENDREC *: * Performance Toolkit; Trend File Definition. This file *
    SYSVMK4  TRENDREC *: * defines the kind of records to be generated when a trend file  *
    SYSVMK4  TRENDREC *: * build request is processed ('T'-suffix appended to FC MONCOLL  *
    ...              *: * RESET times).                                                  *
    ...
    SYSVMK4  TRENDREC *: * General Trend Record Selection
    SYSVMK4  TRENDREC *: RECORDS   CHANNEL NSS DSPACES USER DASD SEEKS VSWITCH SFS MTUSER TCP...
    SYSVMK4  TRENDREC *: * Specific User Selection (for USER, SFS, MTUSER and TCPIP Records)
    SYSVMK4  TRENDREC *: * USERID    userid1 userid2 userid3  userid4  ...
    SYSVMK4  TRENDREC *: USERID    OPERATOR RSCS     TCPIP     VTAM      RSKSERV
    SYSVMK4  TRENDREC *: * Specific DASD Selection (for DASD and SEEKS Records)
    SYSVMK4  TRENDREC *: * DEVNO     devno1 devno2 devno3  devno4  ...
    SYSVMK4  TRENDREC *: DEVNO     0781 0782 0784 0786 0788
    SYSVMK4  TRENDREC *: * DEVNO     0798 0799
    13:43:57 * Last monitor sample processed for 09:50:04 on 2020/06/24
    13:43:57 =Monscan=> Return
```

```
      17 Sep 2020 13:43:58 Performance Toolkit; Vtime=3.55, Ttime=4.09, Connect time 172 sec.
      17 Sep 2020 13:43:58 Performance Toolkit; Reduction Completed, Rc= 0
```

# SUMMARY and TREND Files

SUMMARY and TREND files are output files that contain data derived from monitor data. See the following:

- "Creating Extended Summary and Trend Files" on page 111 for general information about the creation of summary and trend files
- The "Layout of Extended Trend File Records" appendix in the *z/VM: Performance Toolkit Reference* for a description of the specific record layouts

The actual contents of the summary and trend files are controlled by the record specifications made in the corresponding SUMREC and TRENDREC input files. If SUMREC and/or TRENDREC files are not specified, all summary and/or trend records are created

Note that the FC MONCOLL RESET statements must be provided in the SETTINGS file for summary and/or trend reports when the 'BATCH' option is specified. If the corresponding FC MONCOLL RESET statements are not provided, summary and/or trend reports will not be generated.

# Chapter 7. Analyzing Performance Problems

When a performance bottleneck is suspected, that is, when interactive system response time is degrading or when job turnaround times become too long, the reason for this unsatisfactory system performance has to be determined so actions can be defined for improving performance.

Information in the following publications will give you some general guidance on how to determine the causes of unsatisfactory system performance. These publications also contain operating-system-specific tuning hints:

- *z/VM: CP Planning and Administration*
- *z/VM: Running Guest Operating Systems*
- *z/VM: Performance*

While in-depth analysis of some performance problems could require a high degree of expertise and detailed knowledge of VM internals, the majority of bottlenecks can be analyzed with some general VM knowledge. The purpose of this chapter is to provide guidelines for locating such bottlenecks with a minimum of effort, and to allow doing so without extensive VM knowledge.

Because the potential for performance improvement is, in general, much bigger in such peripheral areas as I/O and the paging environment, where the correct placement of a minidisk or a paging area is more important than any tuning parameters, this approach is usually quite safe. Nevertheless, always refer to the information provided in the publications listed in this section. Very little of the system-specific information in them is repeated in this guide.

## General Considerations

After you have decided (or been told) that your system has a performance problem, there are three main areas which will have to be checked:

- CPU
- Storage and paging
- General I/O

Where the paging and general I/O are usually closely linked together, performance analysis should:

- Show the contribution of each of these areas to the problem on hand
- Suggest changes which promise to eliminate, or at least lessen, the problem

When analyzing a performance bottleneck, always remember that:

- Bottlenecks usually show up, at least at the beginning, only during some peak load periods. You will have to analyze the system while the bottleneck is still causing problems, because the Performance Toolkit monitor does not retain all the detailed information which you may need. You will receive alert messages to start looking at the monitor displays if you have set various thresholds as suggested in section "Setup for Threshold Monitoring" on page 126.
- The performance indicators described later on may seem to behave erratically, especially on small systems. Take into account the fact that:
  - A single user's workload can be responsible for a large percentage of total system load and thus can cause wide variations in these performance figures.
  - Some of the values are just samples taken at the end of the measuring interval (for example, in-queue users). Many of these values should show a similar trend before they are accepted as true indication of a performance problem.

  You may have to watch these indicators for several days until a clear pattern emerges.

In the next sections, detailed guidelines on how to detect a performance problem and how to use the Performance Toolkit performance monitor for tracking the problem to its origin.

# Problem Detection/Preliminary Analysis

You will usually not have to do much detecting to find out whether your system has a severe performance problem: end-users' reactions are a very sensitive indicator of system behavior, and they will inform you soon enough if performance is really bad. Basically, there are two possibilities:

- The problem concerns several or even all virtual machines, that is, there is a general performance problem for the whole system
- The problem is limited to a specific virtual machine

See the section "Analyzing a Single Virtual Machine" on page 167 if a single virtual machine is having problems. Here we will first try to handle the more general case where performance of the whole system is unsatisfactory.

## Response Time

While you can rely on end-users to tell you that there *is* a performance problem, you will need some figures which allow you to quantify it, so that the effect of changes made can be accurately measured.

Degrading response time for simple interactive transactions like scrolling in an XEDIT session is usually the main reason for user complaints. Actual interactive (or trivial) response time would, therefore, be an ideal performance indicator. Unfortunately, no information is available in system control blocks which would allow the calculation of this end-user response time, which is why it cannot be provided by the Performance Toolkit either. In the absence of real response time, it displays the next best internal response time indicators available in VM:

- The average internal response time for trivial transactions (transactions requiring just a single class 1 slice). The values are shown under the heading 'TR-T'.
- The average internal response time for non-trivial transactions. The value is shown under the heading 'NT-T'.
- The class 1 elapsed time slice. The value is shown under the heading 'C1ES'. The VM scheduler adjusts this value dynamically to keep the appropriate distribution of transactions in the various scheduler user classes. The class 1 elapsed time slice is a good indicator of bottlenecks as wait times, CPU, I/O, and paging are factors in its computation.

Neither the TR-T value nor C1ES are really equivalent to the actual interactive response time experienced by end users (neither include, for example, the time for data transfer between the user's terminal and the CPU), but you can expect both of them to be roughly proportional to that part of the response time which is directly influenced by the system. Both can be used, therefore, as indicators of general system responsiveness to trivial user commands. We will mainly refer to the C1ES value in the following examples.

Look at the corresponding 'non-trivial response time' values under heading NT-T if you need an indicator of system performance for non-trivial commands or batch jobs.

For VM systems made up of mostly guest operating systems, response time values are not indicative of the response times of commands with the guest. From VM's view, most guests appear to issue a few very large transactions. These large transactions may represent many smaller commands running inside the guest.

## Using the Performance Toolkit to Monitor

Analysis of a performance problem requires the ability to distinguish excessive values of specific performance indicators from healthy ones. Unfortunately there are very few generally accepted simple thresholds upon which to base this decision, because many of these thresholds depend on the size (storage, CPU power) and I/O configuration of the system being analyzed. In other words, do not rely too much on any rules of thumb (they may not be applicable to your system or the current problem) but

compare some important performance indicators on *your* system and check them for significant changes at different response time levels.

To do this, you will need the history data saved for the REDISPLAY screen, or, even better, the performance data saved in performance history files on disk. The necessary data will be available only if you have previously entered the command FC MONCOLL PERFLOG ON (for history files on disk), or at least the command FC MONCOLL ON (to activate permanent data collection), and if the performance monitor had already been running for some time. The data available in the redisplay buffer and in the performance history files include the internal system response time values C1ES needed to tell you when performance was bad, and also many other performance values which usually give some indication as to the reason for the bad performance. The data can be viewed using either of the following displays:

- REDISP screen (command 'REDISP' or PF 2). It allows you to view a maximum of the current day's history data simultaneously, but you will have to detect any correlations between different variables yourself.
- REDHIST screen (command 'REDHIST fn ft fm'), for viewing previous days' HISTLOG and HISTSUM files. It contains values for all the performance variables in the REDISP display, and also from most of the other 'by time' logs available for realtime monitoring.
- Variable Correlation Coefficients screen (see the "CORREL" subcommand in the *z/VM: Performance Toolkit Reference*). It shows the correlation coefficients for all the performance variables in a HISTLOG or HISTSUM file, for a selected base variable. Correlation coefficients indicate how closely a variable's values follow changes in the base variable's values. Values close to 1 indicate good correlation, that is, there appears to be a close relationship between the variables, and this can provide pointers to the probable origin of performance problems.
- Graphical CPU History Data Displays (commands PLOTSUM, GRAPHSUM, PLOTDET, and GRAPHDET) which show how the selected variables changed with time.
- Performance Variable Correlation Displays (commands PLOTVAR and GRAPHVAR). They allow you to correlate any of the performance variables with each other, and the resulting graphics are usually an excellent base for analyzing performance problems.

Any of these displays can help you in detecting the reason for the problem on hand so you can select the one you feel most familiar with. You will find examples for variable correlation displays later on in this chapter. Here we will start by using the REDISPLAY screen which will show you all the original raw data.

Either switch to the REDISPLAY screen (command REDISP or PF 2) or print the contents of the REDISPLAY buffer using the command PRINT REDISP, followed by PRINT OFF.

```
FCX101      CPU nnnn  SER nnnnn  Interval 09:12:19 - 09:54:27    Perf. Monitor

TIME >LOGN  ACT TR-T NT-T C1ES TR-Q NT-Q TR/S NT/S %PQ %IQ %LD %EL  Q1  Qx Q1L
09:15 1295  213 0.07 1.47 1.17   .8 14.3 11.2  9.7  18  22   6   0  13  17   2
09:16 1294  218 0.06 1.67 1.10   .8 16.4 12.0  9.7   9  31  23   0  11  16   6
09:17 1296  206 0.06 1.69 1.16   .6 15.8 11.0  9.3  17  27  13   0  11  27   3
09:18 1291  212 0.07 1.71 1.16   .7 14.4 10.3  8.4   4  50   7   0  14  13   2
09:20 1288  200 0.06 1.42 1.10   .7 13.9 11.2  9.7   5  52  18   0   5  12   2
09:21 1292  205 0.07 2.97 1.04   .7 24.4 10.0  8.2  20  50   6   0   4  13   1
..... ....  ... .... .... ....  ... .... ....  ...  ..  ..  ..  ..  ..  ..  ..
..... ....  ... .... .... ....  ... .... ....  ...  ..  ..  ..  ..  ..  ..  ..
09:46 1060  310 0.10 2.96 1.78  1.4 33.6 13.7 11.3  19  17   8   0  22  29   3
09:47 1067  270 0.11 5.22 1.83  1.1 47.1 10.6  9.0  12  17   6   0  10  35   1
09:49 1079  290 0.09 4.03 1.76  1.1 38.8 11.4  9.6  18  13   5   0  18  34   3
09:50 1083  289 0.10 5.39 1.75  1.0 50.1 10.6  9.2  30  12  11   0  20  32   5
09:52 1093  282 0.11 4.86 1.96  1.3 47.0 11.0  9.6  11  28   4   0  15  35   2
09:54 1101  256 0.10 2.88 1.83  1.1 26.4 10.7  9.1  32  20   7   0  14  25   3
```

Then, on the output received:

- Check the class 1 elapsed time slice shown under heading C1ES. Look for periods of good response times and of bad response times. In the example above, good response times should be below 1.2 seconds for the system shown. So, the good, or at least acceptable, response times are shown in the 9:15 - 9:21 time span and the bad, or unsatisfactory response times are shown in the 9:46 - 9:54 time span.

Note: you will have to establish your own thresholds for good or bad response times (the value depends very much on the 'job mix' which is active on the system).

- Try to locate the problem area by looking for other columns where high and low values are closely related to high and low response times.

  - Shift the display window to the right (enter the command TRANSACT or press PF11 until the transaction-related fields are shown) and check for a probable *paging problem* by looking at column %PQ which shows the percentage of in-queue users found in page wait at the end of each interval. It must be related to C1ES if paging really is the problem, but do not forget that all wait state values (%PQ, %IQ, and %EL) are just samples and *not* averages for the whole interval, so it would not be reasonable to expect perfect correlation. Look at column PG/S (page rate) if you suspect a paging problem. (It should also correlate to C1ES.) Skip to section "Analyzing a Paging Bottleneck" on page 154 if paging seems to be the problem.

```
FCX101       CPU nnnn   SER nnnnn   Interval 09:12:19 - 09:54:27    Perf. Monitor

TIME    CPU  %CP  %EM  %WT  %SY -  -  IO/S VIO/S PG/S XPG/S DIAG PRIV LOGN ACT
09:15   382  135  247   18   28 .  .   397   230  549   ... 1468 1295  213 226
09:16   386  125  261   14   25 .  .   341   189  483   ... 1227 1294  218 260
09:17   393  125  268    7   27 .  .   408   206  540   ... 1219 1296  206 257
09:18   395  117  278    5   24 .  .   329   194  478   ... 1199 1291  212 214
09:20   338  122  216   62   24 .  .   318   179  452   ... 1430 1288  200 228
09:21   377  143  234   23   28 .  .   327   188  411   ... 1822 1292  205 232
.....   ...  ...  ...   ..    . .  .    ..    ..   ..   ... .... ....  ... 234
.....   ...  ...  ...   ..    . .  .    ..    ..   ..   ... .... ....  ... 236
09:46   399  161  238    1   37 .  .   509   294  824   ... 1328 1060  310 229
09:47   396  142  254    4   32 .  .   483   284  693   ... 1298 1067  270 228
09:49   400  141  259    0   32 .  .   555   295  665   ... 1233 1079  290 230
09:50   397  151  246    3   34 .  .   510   280  738   ... 1365 1083  289 218
09:52   399  154  245    1   37 .  .   493   296  779   ... 1297 1093  282 210
09:54   398  141  257    2   31 .  .   441   257  641   ... 1292 1101  256 213
```

  - Check for a *storage problem*. A severe storage problem could be indicated by users ending up in the eligible list, resulting in any number other than zero in column %EL. This column shows the percentage of in-queue users found in resource wait, and if your CPU is not being overloaded there is a good chance the resource being waited for is storage. Skip to section "Analyzing a Storage Problem" on page 158 if you suspect some of your users end up in storage wait.

  - Look for an *I/O problem* by checking the values in column %IQ which show the percentage of in-queue users found in I/O wait at the end of each interval. %IQ should be related to C1ES if you really have a general I/O bottleneck. However, there can be variations here. The I/O problem may be restricted to a group of users which depend on the same overloaded disk volume, while all others do not experience it. %IQ might then not indicate this problem very clearly. Look also at column IO/S (total I/O rate) which should correlate with C1ES and %IQ too. Skip to section "Analyzing an I/O Bottleneck" on page 160 if general I/O seems to be at fault.

    Note that a certain number of users in I/O wait is usually much less of a problem than the same number of users in page wait: I/O wait is part of normal processing and will not disappear even on very lightly loaded systems. Page wait, on the other hand, will *always* increase response times. What %IQ value is 'normal' depends mainly on the job mix being run on any given system. Use the values as a standard which you find on *your* system while it is fairly busy with acceptable response times.

  - Finally, check for a *CPU problem* by verifying the values in column CPU (total CPU busy percentage). If they are at the system's maximum much of the time, and if the sum of Q1 + Qx is high while the values in columns %PQ and %IQ are low, then skip to section "Analyzing an Overloaded CPU" on page 164. This seems to have been the bottleneck in the example shown: The %IQ value is definitely lower for the 'bad' measurements, and, although we find higher values also for the %PQ values, CPU load is already at its maximum, so we do not yet see much of the effects of a developing paging bottleneck.

You will also find all of these system load fields on the general CPU screen, but the REDISPLAY screen is better suited for a general overview because the many measurements shown on the same screen allow you to see how other key system load values correlate with interactive response time, and this should give you the necessary pointers for proceeding.

In the above guidelines, we have always taken the wait states of in-queue users as indicators for system bottlenecks. The REDISPLAY screen shows these values together with the internal system response time C1ES and other system load figures, which is why we have used it. The %PQ and %IQ values shown there are not averages but represent just single samples, and they will usually vary quite a bit. Have a look at the user wait state log screen (command 'USTLOG') for a more reliable and more detailed overview over user wait states, based this time on the CP monitor high-frequency sampling technique.

```
FCX135     CPU nnnn  SER nnnnn  Interval 07:26:00 - 11:06:00   Perf. Monitor
                                                      <-SVM and->
End Time  %ACT  %RUN %CPU %LDG %PGW %IOW %SIM %TIW %CFW %TI %EL %DM %IOA %OTH
>>Mean>>    4     3    4    0   16    0    8   22   24    4   0  16    3    0
09:01:00    4     2    3    0   23    0    9   24   25    4   0  10    4    0
09:06:00    5     2    3    0   28    0   10   22   21    3   0  11    4    0
09:11:00    4     2    3    0   26    0    9   24   20    3   0  13    3    0
09:16:00    3     2    2    0   34    0    8   23   15    4   0  11    3    0
09:21:00    4     3    4    0   27    0    8   24   18    3   0  13    3    0
09:26:00    3     3    4    0   32    0   10   23   16    3   0   9    3    0
09:31:00    3     2    2    0   38    0    9   22   15    3   0   9    3    0
09:36:00    2     4    3    0   35    0    8   21   16    3   0  10    3    0
09:41:00    2     4    3    0   31    0   10   18   18    4   0  12    4    0
09:46:00    3     3    4    0   36    0    9   17   19    3   0   9    3    0
09:51:00    3     4    3    0   34    0    8   20   20    3   0   8    2    0
09:56:00    2     3    1    0   39    0    7   19   21    2   0   8    2    0
........   ..    ..   ..   ..   ..   ..   ..   ..   ..   ..  ..  ..   ..   ..
```

The display shows user wait states by time, similar to the REDISPLAY screen. If you want to see that same information by user, select the user wait state screen (command USTAT):

```
FCX114     CPU nnnn  SER nnnnn  Interval 11:16:00 - 11:21:00   Perf. Monitor
.          ____   .    .    .    .    .    .    .    .    .    .    .    .    .
                                                      <-SVM and->
Userid    %ACT  %RUN %CPU %LDG %PGW %IOW %SIM %TIW %CFW %TI %EL %DM %IOA %OTH
>System<    3     4    4    0   31    0   10   23   14    3   0   9    2    0
VSCS      100     8    3    0    7    0    2   42    0   37   0   0    0    0
VTAM      100     1    4    0    5    0    2   14    0   60   0   0   14    0
FTPQH     100     0    0    0   24    0    0    0    0    5   0  71    0    0
RSCS       99     1    3    0   15    0   32    0    8    0   0   0   42    0
SDTRACK    88     8    6    0   36    0   42    4    0    0   0   0    5    0
TOOLS      78     0    1    0   32    0   60    0    7    0   0   0    0    0
RACFVM     68     7    2    0    8    0    6   64    2    0   0   0   10    0
CHCAL      43     2   13    0   14    0   18   51    2    0   0   0    0    0
FTPDS2     43     0    0    0   33    0    0    4    0    6   0  57    0    0
........   ..    ..   ..   ..   ..   ..   ..   ..   ..   ..  ..  ..   ..   ..
```
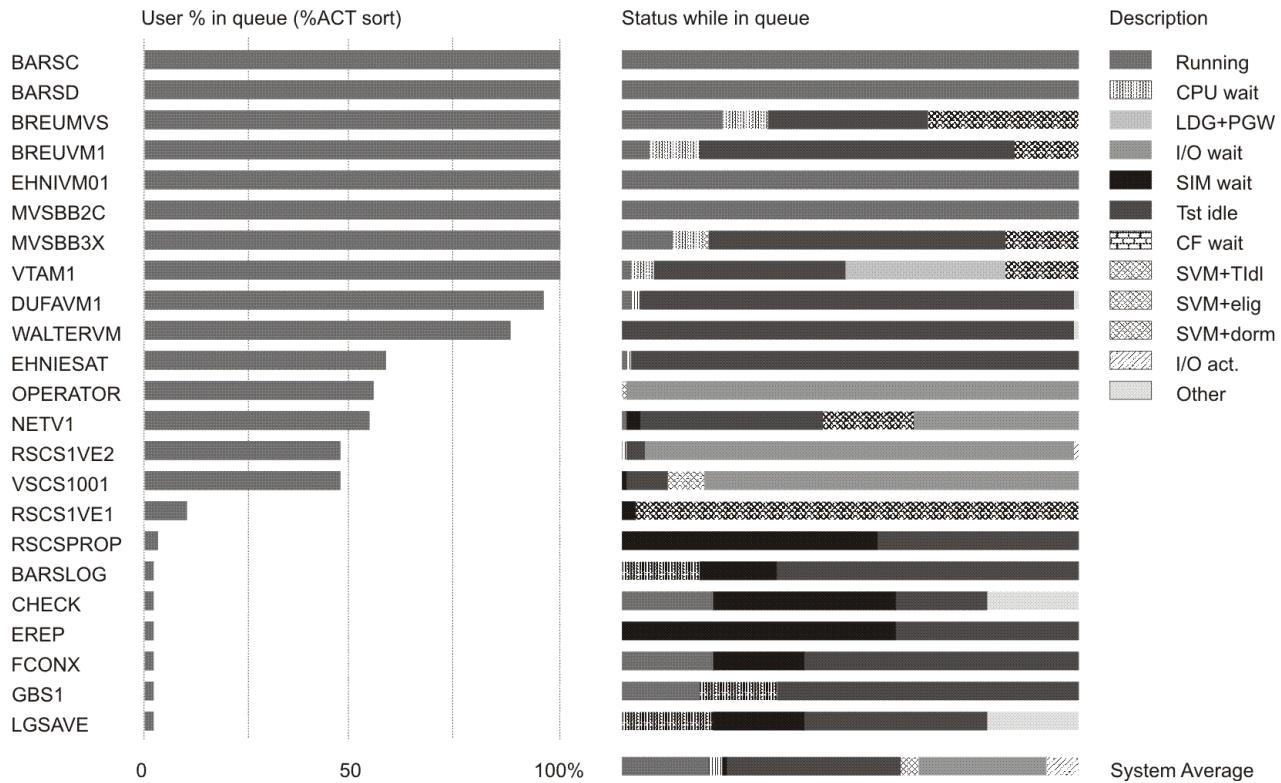
Evaluation of the wait state figures in these screens is similar to the one made for the REDISPLAY screen. Look for frequent:

- CPU wait states in column %CPU as an indicator for a CPU bottleneck
- Page wait states in column %PGW as an indicator for a paging bottleneck
- I/O wait states in columns %IOW **and** in column %SIM as indicators for an I/O bottleneck

Because the values in both these screens are based on 'high-frequency' sampling data from the CP monitor, they are much more reliable than the information in the REDISP output (which is based on single samples). However, you must be aware that I/O bottlenecks will usually not show up as high %IOW values: All diagnose I/O activity by CMS machines will be shown as 'instruction simulation wait' in the %SIM column, and **not** as %IOW.

The above example is from a system with a paging problem, as indicated by the high page wait percentages.

You can also use the graphic user status display (command USTATG) if you have GDDM and a display terminal with graphics capability:

| | User % in queue (%ACT sort) | Status while in queue | Description |
| --- | --- | --- | --- |

Concentrate on the last bar marked `System Average`. Like the `>System<` values of the USTAT display it shows the average non-dormant wait states for all users, and it should give you reliable pointers to potential problem areas (the example is not from the same machine as the previous USTAT display).

# Analyzing a Paging Bottleneck

Some paging activity is part of normal system behavior. Whether a certain page rate constitutes a problem depends on the specific system and its layout, such as:

- CPU power and size of real storage
- Availability and size of expanded storage
- Number and speed of paging devices
- Total I/O load on these devices and their I/O paths

No simple rule of thumb can be given for maximum healthy page rates.

Instead of concentrating on page rates only, we will look for other indicators which could point to paging problems. Any paging bottleneck must lead, by definition, to a high percentage of your in-queue users being found in 'page wait' state. This will be the first indicator to check. If you really have a paging problem, you will have to analyze the I/O performance of the disks used for paging in order to determine the reason for the problem.

## Using the Performance Toolkit to Monitor

Switch to the REDISPLAY screen.

```
FCX101        CPU nnnn   SER nnnnn   Interval 08:02:19 - 10:41:27    Perf. Monitor

TIME   CPU   %CP  %EM  %WT   %SY -  -   IO/S VIO/S PG/S XPG/S DIAG PRIV LOGN ACT
08:05  300   119  181  100    23 .  .    375   232  384   ... 1074 2131  592 158
08:06  291   123  168  109    23 .  .    363   250  367   ... 1191 2159  606 175
08:07  343   130  213   57    24 .  .    437   276  467   ... 1012 2186  624 191
08:08  372   138  234   28    26 .  .    416   276  513   ... 1390 2466  640 211
08:09  370   117  253   30    24 .  .    402   276  446   ... 1002 1949  652 182
08:10  381   127  254   19    25 .  .    406   239  476   ... 1118 2310  673 188
.....  ...   ...  ...  ...    .. .  .    ...   ...  ...   ... .... ....  ... ...
.....  ...   ...  ...  ...    .. .  .    ...   ...  ...   ... .... ....  ... ...
.....  ...   ...  ...  ...    .. .  .    ...   ...  ...   ... .... ....  ... ...
10:35  299   112  187  101    33 .  .    432   201  878   ...  811 1497 1394 252
10:36  325   110  215   75    33 .  .    462   219  796   ...  801 1458 1396 252
10:37  279   116  163  121    36 .  .    486   215  921   ...  996 1590 1397 281
10:39  319   129  190   81    38 .  .    496   248  857   ... 1125 1805 1397 292
10:40  288   117  171  112    33 .  .    484   259  770   ...  907 1576 1395 262
10:41  327   120  207   73    36       452   232  857        831 1516 1394 250
Use "PLOTxxx" or "GRAPHxxx" commands for history data graphics
Command ===> _
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F10=Left  F11=Right  F12=Return
```

To make sure you really have a paging problem, look at the following:

- Column PG/S (page rate). It should usually be higher when response times increase (you will have to shift the display window to the right by pressing PF11 to look at the internal class 1 'response time' C1ES).

- Column IO/S (total I/O rate). Keep in mind that paging performance may have degraded because there was too much other I/O on the same devices or on their I/O paths. A high total I/O rate could, therefore, be the reason for slow paging.

Shift the display window to the right by entering the command TRANSACT, or by pressing PF11, so the transaction-related data can be viewed.

```
FCX101        CPU nnnn   SER nnnnn   Interval 04:02:19 - 10:41:27    Perf. Monitor

TIME >LOGN  ACT TR-T NT-T C1ES TR-Q NT-Q TR/S NT/S %PQ %IQ %LD %EL  Q1  QX Q1L
08:05  592  158 0.05 4.37  .84   .5 37.1  9.8  8.5   0  45   0   0   6  17   0
08:06  606  175 0.04 1.34  .85   .4 12.2  9.9  9.1   7  40   4   0  12  12   1
08:07  624  191 0.04 2.57  .91   .5 27.7 12.0 10.7   3  28   0   0  12  22   0
08:08  640  211 0.06 2.43 1.01   .7 24.5 11.9 10.0   3  42   8   0  12  14   2
08:09  652  182 0.06 2.61  .98   .7 22.5 10.5  8.5   4  40   8   0   8  17   2
08:10  673  188 0.05 3.06 1.03   .6 31.5 12.0 10.3  12  56   4   0   7  16   1
..... ....  ... .... .... ....  ... .... .... ....  ..  ..  ..  ..  ..  ..  ..
..:.:. ....  ... .... .... ....  ... .... .... ....  ..  ..  ..  ..  ..  ..  ..
10:35 1394  252 0.08 2.19 1.59  1.0 21.9 12.1 10.0  43  43  15   0   9  12   2
10:36 1396  252 0.09 2.43 1.76  1.0 22.6 11.4  9.2  41  31  26   0   2  18   1
10:37 1397  281 0.10 4.36 2.10  1.1 38.6 11.4  8.8  40  31   9   0  13  19   3
10:39 1397  292 0.11 3.14 2.22  1.4 30.4 12.1  9.6  19  50  14   0  19  10   4
10:40 1395  262 0.11 4.77 2.06  1.5 50.0 13.2 10.4  51  19  10   0  19  21   4
10:41 1394  250 0.09 4.70 2.08  1.2 51.3 13.0 10.8  20  36  14   0  12  17   2
```

On this screen look at column %PQ which shows the percentage of in-queue users found in page wait at the end of each measuring interval. The values shown should be markedly higher during periods of bad response time. The example shows values which are higher both for the paging activity and the percentage of in-queue users in page wait during the period with bad reponse times (10:35 - 10:41), while CPU load and the percentage of in-queue users in I/O wait are not (you will notice that CPU load in our example was even lower than during the good period from 08:05 to 08:10). This points to a paging problem.
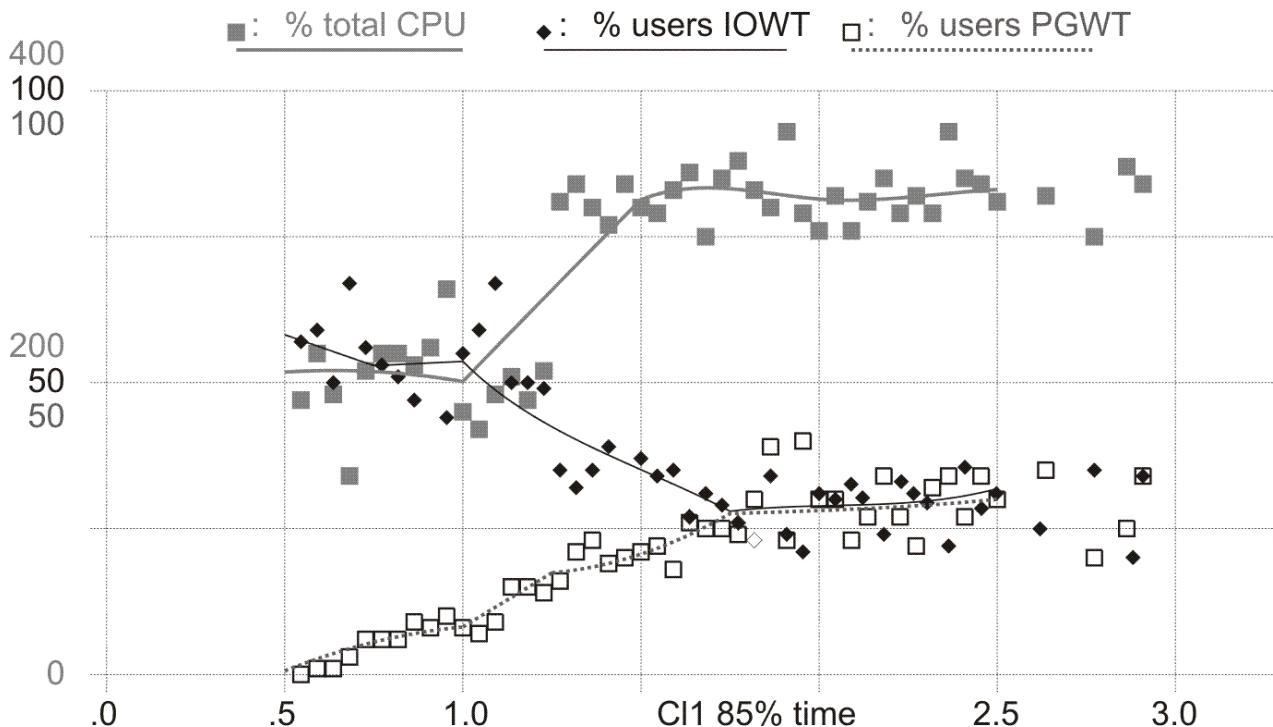
If you have GDDM and a display terminal with graphics capability, there is another easy way to check the correlation between the internal class 1 'response time' C1ES and other performance variables. Produce a 'variable correlation' plot with the variables we are interested in by entering the command 'GRAPHVAR C1ES CPU %IQ %PQ'.

The resulting display shows total CPU load and user I/O and page wait percentages on the vertical y-scale, based on the corresponding values of C1ES. The advantage of this type of display, compared to the checking of performance values on the redisplay screen, is its reliability. Rather than comparing just a few measurements for selected performance variables, you can see the correlation between these

variables based on *all* the available measurements. It also eliminates the danger of the performance analyst's unconscious search for values which might fit certain pre-conceived ideas.

On this display:

- Check whether the percentage of in-queue users which are in page wait really increases as response times degrade.
- See whether the percentage of users in I/O wait decreases. A decrease would indicate a paging problem only, while an increase both of the page wait and I/O wait percentages might indicate a general I/O problem which also influenced paging performance.
- Check whether CPU load could also have been a problem. If CPU load only increases up to a certain point well below maximum system capacity.



Again, all the values shown point to a probable paging problem, so we will now have a look at the paging devices used. Switch to the display of CP owned devices (command DEV CPOWN).

```
    FCX109        CPU nnnn   SER nnnnn   Interval 13:55:25 - 13:56:57    Perf. Monitor

    Page / SPOOL Allocation Summary
    PAGE slots available       855,000       SPOOL slots available       530,400
    PAGE slot utilization          37%       SPOOL slot utilization          66%
    T-Disk space avail. (MB)       600       DUMP slots available             0
    T-Disk space utilization       64%       DUMP slot utilization           0%

    Addr  Label   Type   Extent    Used  RD/s WR/s MLOAD    IO/s %Busy %Qued  Resp
    0140  XARES1  TDSK   0650-0699  98%   ...  ...   50.0    6.0    16     0  27.8
    0143  VMPG1B  PAGE   0300-0599  28%  21.1 14.5  10.6   10.2    48     0  46.8
    0146  XASER1  DIRECT 0001-0019  21%   ...  ...   50.0    2.5    12     0  50.8
    0150  VMSP1B  SPOOL  0001-0884  42%   3.1  1.4  21.5    4.5    10     0  22.3
    0151  VMPG1D  PAGE   0300-0599  37%  21.8 13.6   8.6   12.6    42     0  33.3
    0152  VMPG1E  PAGE   0300-0599  37%  20.5 18.2  11.2   11.6    41     0  35.4
    0153  VMPG1F  PAGE   0300-0599  35%  18.2 15.0   9.5   11.1    43     0  38.5
    0154  VMPG1G  PAGE   0300-0599  38%  23.3 21.4   7.2   12.7    41     0  32.5
    0156  VMPG1H  PAGE   0300-0599  36%  24.6 16.2  10.8   11.7    40     0  34.0
    0265  VMPG1I  PAGE   0300-0599  50%  25.7 14.8   8.1   16.1    35     0  22.0
    0266  VMSP1C  SPOOL  0001-0884  53%   6.2  3.1  10.6    9.4    11     0  11.9
    0270  VMSP1D  SPOOL  0001-0884  84%    .2   .1  35.6     .3     1     0  37.9
    0271  VMPG1J  PAGE   0300-0599  29%  17.3 15.7  17.6    9.1    52     0  57.8
    0272  VMPG1K  PAGE   0300-0599  28%  16.6 16.2   108     9.4    53     0  55.7
    0273  VMPG1L  PAGE   0300-0599  34%  20.1 17.3  15.3    8.8    52     0  59.3
    0276  VMPG1M  PAGE   0300-0599  35%  18.9 14.8  26.6    9.0    50     0  56.1
    Select device with cursor and press ENTER for more details
    Command ===> _
    F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F10=Left  F11=Right  F12=Return
```

The above screen shows the size and location of your system's page and spool areas, and the I/O load on all CP owned disks. These CP owned devices are not necessarily all paging disks, so you will have to check, in the column headed `Type`, whether a disk really contains a paging area.

If your system has a paging problem because of overloaded paging devices, this screen should show you which of the paging disks caused it. Look at columns:

- `%Busy, %Qued`, and `Resp`: They show the load and performance of the whole disk during the last interval. Overloaded devices should be characterized by high `%Busy` values (>35%), `%Qued` values (indicate I/O contention for the device), and `Resp` values (average device response time during the last interval, should be <30 msec for good performance and indicates a problem if it is much higher). There seem to be many overloaded paging disks on this system.

- `RD/s, WR/s`, and `MLOAD`: These values give the read and write rates, and the smoothed I/O response time for I/O to system areas on the disk. Paging performance is bad on disks with high `MLOAD` values.

In our example we find very high response times for the paging disks with the device numbers 271, 272, 273, and 276 which will need some explaining.

Remember that *all* the values shown are usually just the last measuring interval's average. Watch this screen for several intervals until you feel sure you know which devices are having I/O problems, or display *average* load figures. Make sure, when looking at averages, that the selected period really includes only measurements with peak load.

If you find that your paging disks have bad response times, you should check whether some of the I/O load has been caused by minidisks with a high activity on the same disk pack. Switch to the detailed device display for the disk by entering the command 'DEV cuu' (where 'cuu' is the disk's real device number as shown under column `Addr` of the CP owned device display).

If no minidisks are shown, all I/O activity on the disk from this system must be generated by CP. This is not necessarily all paging and spooling I/O. Check for other areas like checkpoint cylinders. (Consult your HCPSYS source to find their location. They are not shown on the display). I/O to directory cylinders should be no problem under VM, because CP works with a copy which is kept in storage. See the detailed device display for more detailed information on the paging and spooling activity of specific disks.

Also, be sure none of the minidisks are shared with another system. If they are, you will also have to check the I/O activity from the other system. It is *not* included in the figures shown on this system.

In our example so many disks are overloaded that we must consider the possibility of a more general I/O problem. Enter the command I/O to display the general I/O device screen, then sort the I/O devices according to their `%Busy` values by entering the command 'SORT %BUSY'.

```
FCX108        CPU nnnn   SER nnnnn   Interval 13:55:25 - 13:56:57    Perf. Monitor
  .       .       .        .     .     .    .    .    .       .      .    .     .
Addr Type   Label/ID    Links   IO/S  %Res Pend Disc Conn   Serv %Busy %Qued Resp
0726 3380-K VMPR26         97   21.2     0  3.5 18.0  9.1   30.6    65    32 40.4
0272 3380-A VMPG1K CP       0    9.4     0  4.1 44.6  7.0   55.7    53     0 55.7
0271 3380-A VMPG1J CP       0    9.1     0  4.5 45.1  8.2   57.8    52     0 57.8
0273 3380-A VMPG1L CP       0    8.8     0  4.1 47.5  7.7   59.3    52     0 59.3
0276 3380-A VMPG1M CP       0    9.0     0  4.3 43.6  8.2   56.1    50     0 56.1
0734 3380-K VMPG1A CP      55   13.7     0  4.0 25.8  7.0   36.8    50     0 36.8
0143 3380-A VMPG1B CP      74   10.2     0  4.3 33.9  8.6   46.8    48     0 46.8
0735 3380-K VMPG1C CP      43   11.9     0  5.0 28.1  7.0   40.1    48     0 40.1
0277 3380-A XAP003       3989   46.8     0  2.7  3.9  3.4   10.0    47     8 10.8
072C 3380-K VMPG1P CP       0   11.9     0  3.3 26.0  7.7   37.0    44     0 37.0
072E 3380-K VMPG1R CP       0   11.8     0  3.0 26.9  7.3   37.2    44     0 37.2
072F 3380-K VMPG1S CP       0   12.0     0  3.0 25.8  7.5   36.3    43     0 36.3
072D 3380-K VMPG1Q CP       0   12.5     0  3.5 24.0  7.2   34.7    43     0 34.7
```

Most of the devices shown are paging disks (all the disks with labels VMPG1x) and they show the high load which we already noticed on the previous display with the CP owned devices. But we can see something else, the paging disks with the numbers 272, 271, 273, and 276 have a %Busy value which is considerably higher than that of the other paging disks, in spite of a *lower* I/O rate. This points to an I/O problem. Because there is no other I/O activity on these packs (the values in column Links show that they do not contain any active minidisks), the problem should not be the disks themselves. See "Analyzing an I/O Bottleneck" on page 160 for a detailed explanation, where this example has been taken up again as a case of an overloaded channel.

## Action

Remove the cause for the high I/O contention on the paging volume(s). You can reduce your system's page rate on disks by increasing real storage size. Reducing the paging activity by increasing storage size will always give the best results because less I/O is required. However, it is relatively expensive, or you may already have installed the maximum real storage size supported by your CPU. In any case you should also try to optimize the performance of your paging disks:

- Remove minidisks with high I/O load from paging disk packs. It is all right to use the spare space on paging volumes for minidisks with very low I/O load. However, be sure to check, from time to time, whether the original assumptions as to a minidisk's I/O activity are still correct.

- Increase the number of paging volumes, or use faster devices for paging.

- Make sure the I/O paths used for paging (channel paths and control units) are not being overloaded by other I/O. Place paging volumes on separate paths with low load if possible.

  In our example we eventually found that only one of the two paths to two of the paging DASD strings was operational due to hardware problems in the string controllers. When that was fixed, the high paging load could be handled again with satisfactory response times.

- Consider installation of a better control unit.

This will improve paging on your system by allowing higher paging activity without I/O problems.

## Analyzing a Storage Problem
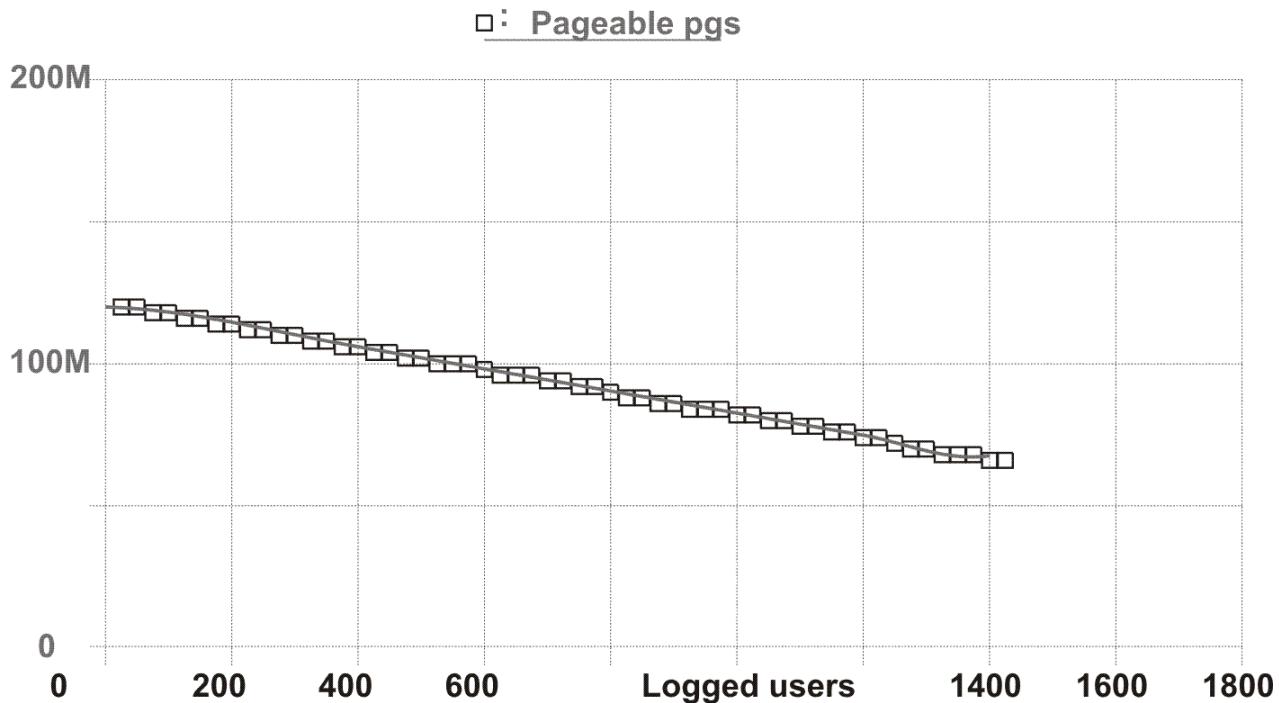
Insufficient real storage size and a dynamic page area which is too small can result in the following:

- Additional wait time in an 'eligible list' before users can be moved to a 'dispatch list' and start consuming resources. Depending on the severity of the real storage constraint, this additional wait time can become so long that response times are badly affected.

- Page stealing, that is, in order to satisfy one user's storage requests the system has to 'steal' some pages from another still *active* user. There is a good chance this other user will need those pages again before being dropped from the queue, which can cause a lot of unproductive paging activity (thrashing) and should be avoided.

You will usually also find the signs of a paging bottleneck when analyzing a storage problem. Improving the paging environment alone will not solve the problem in such a case, so it is quite important to realize that a real storage constraint is at the bottom of the problem.

# Using the Performance Toolkit to Monitor

Determine how real storage is used on the system. You will find information on the size of the pageable part of the system's DPA (the 'dynamic paging area') on the 'storage' display (enter the command STORAGE) and also on the 'redisplay' screen. The redisplay screen shows better what we are looking for: the number of pages in the dynamic paging area which is available to the system for paging is at a maximum directly after IPL. It then seems to get smaller and smaller as users start logging on. The best way for showing this relationship is, as usual, the variable correlation display. Enter the command 'GRAPHVAR LOGN PPAG'.



The resulting graph shows the available DPA space decreases from approximately 120MB (real storage size of the 3084 system shown is 128MB) as users connect to the system, until there are about 70MB left when 1400 users have logged on. This is because CP takes space out of DPA for the different control blocks it needs to run the system. It is part of normal CP operation, and nothing we can influence.

So far we have seen that the pageable part of the system's DPA will be considerably reduced on systems with many users. Let us now check whether this might be a problem in a specific case. Switch to the REDISPLAY screen (PF2) and shift the display window to the right (PF11, or command TRANSACT).

```
FCX101      CPU nnnn   SER nnnnn   Interval 10:40:19 - 10:41:27    Perf. Monitor

TIME >LOGN  ACT TR-T NT-T C1ES TR-Q NT-Q TR/S NT/S %PQ %IQ %LD %EL  Q1  QX Q1L
10:31 1396  248 0.08 3.49 1.62  1.0 30.3 11.8  8.6  37  33  12   0   9  17   2
10:32 1393  267 0.10 4.09 1.68  1.2 40.2 12.2  9.8  28  37   9   0  11  22   1
10:33 1395  271 0.09 3.51 1.72  1.1 35.8 11.9 10.2  41  32  15   0  14  19   4
10:35 1394  252 0.08 2.19 1.59  1.0 21.9 12.1 10.0  43  26  15   0   9  12   2
10:36 1396  252 0.09 2.43 1.76  1.0 22.6 11.4  9.2  41  31  26   0   2  18   1
10:37 1397  281 0.10 4.36 2.10  1.1 38.6 11.4  8.8  40  31   9   0  13  19   3
10:39 1397  292 0.11 3.14 2.22  1.4 30.4 12.1  9.6  19  50  14   0  19  10   4
10:40 1395  262 0.11 4.77 2.06  1.5 50.0 13.2 10.4  51  19  10   0  19  21   4
10:41 1394  250 0.09 4.70 2.08  1.2 51.3 13.0 10.8  20  36  14   0  12  17   2
```

On this screen:

- Check the values in column %EL. This is the percentage of in-queue users which were found in resource wait, that is, they could not be added to a dispatch list because some system resource would otherwise have been overcommitted. If the real storage constraint is severe enough, this could lead to users being placed in an eligible list until a sufficient number of free page frames can be made available. There

should be *no* users in eligible lists. If there are, they are likely to experience very bad response times. The storage constraint obviously was not yet bad enough in this example to lead to that result.

- Check also the number of pages read per queue drop This value is not directly available on any of the performance screens. You will have to calculate it by dividing the total page read rate per second by the sum of trivial and non-trivial transactions per second (`Page read rate` to be found in the 'General storage utilization display', trivial, and non-trivial transaction rate in fields `Trans. per sec.` of the 'General CPU screen', or in fields TR/S and NT/S of the redisplay screen).

  A high number of pages read per transaction can also be an indication of a storage bottleneck, even if %EL is still zero. In the above example, the page read rate was around 500-600 pages/sec. Even though the block paging mechanism of VM has reduced the number of physical I/O operations required to bring pages back into main storage, the approximately 25-30 pages which had to be read per transaction were near the limit which the system could handle.

## Action

Eliminate machines waiting in eligible lists for storage by *increasing the size of the dynamic page pool*. This will usually require an extension of real storage. ,

Wait time in eligible lists can also be eliminated for a specific virtual machine using the 'SET QUICKDSP' command, but this can only improve one user's performance at the expense of all others. It should be used, therefore, only as a temporary bypass for high priority machines until the basic problem can be solved by a storage upgrade.

*Try reducing your CMS users' working set sizes* by using shared segments for your applications where possible. Especially on large systems *use shared minidisk directories* for minidisks which contain a large number of files, and which are concurrently accessed by many users.

See the CP command 'SET SRM STORBUF ... ...' which allows you to set some parameters which control the algorithm used by the scheduler when deciding whether or not to place a user in an eligible list when main storage is constrained. You can use it to bias this decision so fewer users are kept waiting for storage in an eligible list.

A very small amount of storage can be saved by forcing off users. This will only be effective on systems with very little storage.

As a last resort, you can also FORCE idle users off the system. Assuming that several hundred of the logged-on users are idle, forcing them would free many CP control blocks and so increase the number of pageable pages, and reduce the current bottleneck. This can be achieved by activating idle user monitoring. See "User Monitoring" on page 127 and the "FCONTROL FORCEUSR" subcommand in the *z/VM: Performance Toolkit Reference*, which allows you to activate it.

## Analyzing an I/O Bottleneck

An I/O bottleneck means your system's throughput cannot be increased because I/O requests cannot be handled fast enough. If your users do not have this I/O problem on a lightly loaded system, then it must be caused by some form of I/O contention, that is, there must be too many simultaneous requests from different users for service on a device, control unit, or channel path level. Many of your in-queue users will then be found in 'I/O wait' state, and especially their I/O-bound work will take much longer during peak hours. Please realize, however, that the I/O problem might also be restricted to a specific disk. It could then concern only a certain group of users, and the %IQ value would not necessarily indicate the problem.

If, on the other hand, a user feels that an I/O bound job takes too long even when there is no other heavy I/O activity on the system, the problem must be caused by too many I/O requests being made by *this* user, and this may not lead to I/O contention but just to some I/O devices running at very high rates. You should then try to improve this specific user's I/O environment (see "Analyzing a Single Virtual Machine" on page 167).

# Using the Performance Toolkit to Monitor

For the general form of an I/O bottleneck where parts of the I/O environment are overloaded by too many simultaneous I/O requests, you will have to look for I/O contention on the:

- Channel path
- Control unit
- Device level

The PERFKIT monitor can display detailed performance information on the device level, and provided the CP MONITOR data extraction interface has been activated, you can also display channel load. Information on control unit load is not directly available, but we can analyze I/O performance carefully on the device level. Based on this information we can then draw certain conclusions as to the probable influence of channel path and control unit load.

**Contention on the Device Level:**

This is the most likely source of I/O problems. Switch to the general I/O device screen (command DEV), then:

- Sort the I/O devices according to their I/O load. Suitable sort keys are:

  - IO/S, the I/O rate for the devices
  - %Busy, the percentage of time during which the device was busy handling I/O requests. 30% is a commonly accepted threshold for disks which are interactively accessed by many different users.
  - %Qued, which indicates the ratio between queued I/O requests (queued because the device was already busy handling another I/O) and active I/O requests. Any value greater than zero indicates contention for the device.
  - Resp, the I/O response time for the device, that is, the average elapsed time, in milliseconds, it took to handle an I/O request. This includes time spent waiting in queue until the I/O could be started.

  Start by sorting the devices according to the %Busy value (command 'SORT %BUSY'). I/O devices will then be sorted in descending %Busy sequence at the end of the current measurement interval, as shown in the following example:

```
 FCX108     CPU nnnn   SER nnnn   Interval 10:42:25 - 10:43:31     Perf. Monitor
 .      .          .        .      .    .         .         .    ____   .    .
 Addr Type   Label/ID   Links   IO/S  %Res Pend Disc Conn   Serv %Busy %Qued Resp
 0277 3380-A XAP003      3754   27.3     0  1.3 23.7  5.8   30.8    71    94 59.8
 072B 3380-K OVVM02      4772   46.4     0  1.2  5.0  3.4    9.6    44    80 17.3
 0731 3380-K VMPR31        98   11.8     0  1.4 24.2  4.2   29.8    35     0 29.8
 0612 3380-K VMSP1A CP      0   13.3     0   .4 20.9  2.9   24.2    32     0 24.2
 0619 3380-K VMBA01 CP     14    8.6     0   .4 17.6 14.1   32.1    27     8 34.7
 0270 3380-A VMSP1D CP      0   13.4     0   .4 17.0  2.8   20.2    27     0 20.2
 0266 3380-A VMSP1C CP      0   13.9     0   .3 15.9  2.8   19.0    26     0 19.0
 0721 3380-K VMPR21       103    7.5     0  1.4 26.1  4.2   31.7    23     2 32.6
 072A 3380-K VMPR6A       141    7.4     0  1.8 25.4  4.1   31.3    23     0 31.3
 0607 3380-K VMPRA7       440    6.1     0   .4 32.2  4.0   36.6    22     0 36.6
 0606 3380-K VMPRA6       250   10.2     0   .8 17.1  2.7   20.6    21     7 22.2
 0150 3380-A VMSP1B CP      0   10.9     0   .4 14.8  2.8   18.0    19     0 18.0
 04CC CTCA   >CHRSCS     >04CC   2.5     0   .1 70.3  1.2   71.6    18     0 71.6
```

- Note the devices with the highest %Busy values shown on the first few lines of the display. All the values shown have been calculated based on the devices' activity during the last measuring interval. It is important that you analyze I/O bottlenecks while the problem still persists. Pay attention to devices with a high %Qued value, which indicates I/O contention for the device. Check column Resp with the I/O response time and look for disks with abnormally high response times.

  Then, for each of the disks found with a high %Busy, %Qued, or Resp value, let more details be shown by calling the 'detailed device display' for the disk (command 'DEV cuu' where 'cuu' is the disk's real device number). In the example above volume XAP003 with device number 277 has a very high %Busy value, and it also shows high I/O contention %Qued of 94%, so we will first have a look at this disk (see the figure on the next page). The detailed analysis of I/O activity on the disk pack will tell you which

users / minidisks are mainly responsible for the high I/O load, and you can then take the necessary steps to eliminate the problem.

Check also whether the disk is CP owned (indicated in the header line). If so, it may contain system areas like paging and spooling areas or checkpoint cylinders with a high I/O activity, but this activity will not be shown in the minidisk map of the detailed device display.

- If you have a shared DASD environment check to see if there is any I/O activity from another CPU on disk packs with bad performance. If this is the case, the figures shown for this system may be misleading.

- Look at the different components of the device response time which are shown under the headings `Pend`, `Disc`, and `Conn` on the 'general I/O device display', and under `Function pend.`, `Disconnected`, and `Connected` on the detailed I/O device display. They can give you pointers to I/O bottlenecks on the I/O path level. See "Contention on Control Unit and Channel Path Level" on page 163 for more information.

```
FCX110      CPU nnnn  SER nnnnn  Interval 10:51:19 - 10:52:32     Perf. Monitor

Detailed Analysis for Device 0277 ( SYSTEM )
Device type :  3380-A     Function pend.:     .6ms      Device busy   :   95%
VOLSER      : XAP003       Disconnected  :   27.2ms     I/O contention:  154%
Nr. of LINKs:    3749      Connected     :    4.0ms     Reserved      :    0%
Last SEEK   :     256      Service time  :   31.8ms     SENSE SSCH    :    0
SSCH rate/s :    29.8      Response time :   80.8ms     Recovery SSCH :    0
Status: SHARABLE

Path(s) to device 0277:    02    42
Channel path status    :    ON    ON

        MDISK Extent       Userid   Addr IO/s VSEEK Status    LINK MDIO/s
   +-----------------------------------------------------------------------+
   |     0201 -  0400      USER1    0334 6.3     0 RR                      |
   |                       USER2    0334 3.2     0 RR,BUSY                 |
   |                       USER3    0334 2.4     0 RR                      |
   |                       USER4    0334 1.6     0 RR                      |
   |                       USER5    0334  .8     0 RR          1248   16.2 |
   |     0656 -  0680      USER6    0399 1.2     0 RR                      |
   |                       USER7    0399  .9     0 RR                      |
   |                       USER3    0399  .8     0 RR,BUSY                 |
   |                       USER8    0399  .8     0 RR                      |
   |                       USER9    0399  .7     0 RR          1251   10.7 |
   |     0681 -  0700      SPBARS   0191  .0     0 WR             1     .0 |
   |     0866 -  0874      USER10   0335  .2     0 RR                      |
   |                       USER11   0335  .1     0 RR          1247     .3 |
   +-----------------------------------------------------------------------+

Command ===> _
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

## Action (Device Level)

Try to reduce device contention, and therefore device response time. Let high-use minidisks use caching, either in minidisk cache or in a caching control unit. Use of the minidisk cache is preferable because it eliminates not only the 'seek' and 'search' time associated with an ordinary disk I/O operation, but also the data transfer time over the channel path.

If you have little caching storage, you can also prevent other, less important minidisks from filling the cache by adding MINIOPT statements with NOMDC and NOCACHE arguments after their definitions in the directory to make sure only the data most important for your system's performance will remain in cache.

If caching is not possible:

- Migrate some high-use minidisks to another disk pack with low activity

- Duplicate, if necessary, common shared minidisks with extremely high I/O activity (for example, CMS S-disk, Y-disk)

In the above example, the display shows the volume contains three minidisks which are accessed by many users, two of which have a very high I/O activity. These would normally have to be distributed over different disk packs. However, because the disk pack is connected to a cached 3880-23 control unit, all three minidisks were intended to use the cache, and this would have taken care of the I/O load found. The problem was, in this case, that caching had been deactivated following a power failure which also needed intervention on the control unit by the customer engineer. After re-activating the cache, the average response time for the disk dropped to about 5 msec. and the `%Busy` value to less than 20%.

## Contention on Control Unit and Channel Path Level

Performance Toolkit contains an interface which allows it to extract CP MONITOR data. When activated, we can then display information on 'channel busy' states. No information is available, however, on control unit load; we can only obtain pointers to probable bottlenecks in this area by carefully analyzing the I/O elapsed time values provided on the general device screen.

First, let us look again at the general DEVICE display from the example shown in the section dealing with paging bottlenecks.

```
FCX108      CPU nnnn  SER nnnnn  Interval 13:55:25 - 13:56:57    Perf. Monitor
.   .                  .     .                       .    _____   .     .
Addr Type  Label/ID   Links  IO/S  %Res Pend Disc Conn   Serv  %Busy %Qued Resp
0726 3380-K VMPR26       97  21.2     0  3.5 18.0  9.1   30.6    65    32 40.4
0272 3380-A VMPG1K CP     0   9.4     0  4.1 44.6  7.0   55.7    53     0 55.7
0271 3380-A VMPG1J CP     0   9.1     0  4.5 45.1  8.2   57.8    52     0 57.8
0273 3380-A VMPG1L CP     0   8.8     0  4.1 47.5  7.7   59.3    52     0 59.3
0276 3380-A VMPG1M CP     0   9.0     0  4.3 43.6  8.2   56.1    50     0 56.1
0734 3380-K VMPG1A CP    55  13.7     0  4.0 25.8  7.0   36.8    50     0 36.8
0143 3380-A VMPG1B CP    74  10.2     0  4.3 33.9  8.6   46.8    48     0 46.8
0735 3380-K VMPG1C CP    43  11.9     0  5.0 28.1  7.0   40.1    48     0 40.1
0277 3380-A XAP003     3989  46.8     0  2.7  3.9  3.4   10.0    47     8 10.8
072C 3380-K VMPG1P CP     0  11.9     0  3.3 26.0  7.7   37.0    44     0 37.0
072E 3380-K VMPG1R CP     0  11.8     0  3.0 26.9  7.3   37.2    44     0 37.2
072F 3380-K VMPG1S CP     0  12.0     0  3.0 25.8  7.5   36.3    43     0 36.3
072D 3380-K VMPG1Q CP     0  12.5     0  3.5 24.0  7.2   34.7    43     0 34.7
0153 3380-A VMPG1F CP     1  11.1     0  4.2 26.0  8.3   38.5    43     0 38.5
0151 3380-A VMPG1D CP     0  12.6     0  3.2 22.5  7.6   33.3    42     0 33.3
0154 3380-A VMPG1G CP     0  12.7     0  2.9 22.0  7.6   32.5    41     0 32.5
0152 3380-A VMPG1E CP     0  11.6     0  3.9 23.9  7.6   35.4    41     0 35.4
0156 3380-A VMPG1H CP     0  11.7     0  3.4 22.9  7.7   34.0    40     0 34.0
060D 3380-K VMPG1O CP     0  15.6     0  2.4 14.3  6.7   23.4    36     0 23.4
0265 3380-A VMPG1I CP     0  16.1     0  2.9 12.5  6.6   22.0    35     0 22.0
060C 3380-K VMPG1N CP     2  15.1     0  2.0 13.1  7.1   22.2    33     0 22.2
```

The meaning of the different components of the I/O service time which are shown on this screen are as follows:

**Pend**
*Operation pending time:* This is the time an I/O request remained pending in the I/O subsystem before it could be passed on to the device. This value should normally be around 2 msec. for disks. If it is much longer, we must have some contention in the path to the device which caused the additional delay. The additional delay is usually very small (a few milliseconds).

**Disc**
*Device disconnected time:* This is the average time during which the device remained disconnected when handling an I/O request. This normally points to overhead in the control unit for cache management or high seek time on much older devices.

**Conn**
*Connect time:* This is the average time during which the device was connected to the channel path for running the SEARCH order and for actually transferring data. It is a function mainly of the amount of data transferred per I/O operation, that is, of the block size and the number of blocks read or written per I/O. As such, it is more an indicator of how the I/O was programmed than of any form of I/O contention.

What do these values tell us for our example?

The display shows that the devices with numbers 27x have extremely high busy percentages and, for most of them, also very high response times, *even though the I/O rate to these packs was lower than that*

*of similar disks on other strings*. All disks with labels 'VMPG1x' are paging disks, normally without other activity on the same pack (note that the `Links` count is zero for most of them). They should, therefore, be very evenly loaded, and any large discrepancies between load figures are probably caused by the I/O subsystem. (All of these disks are connected to a cached 3880-23 control unit, but caching has not been enabled for the paging disks. Therefore, there is a much lower response time for disk XAP003, device number 277, which uses the cache.)

The detailed response time split shows that this bad response time is due almost exclusively to very long 'disconnected' times, which are probably caused by a high percentage of reconnect misses. This is an indication of overloaded channel paths.

By using the CHANNEL, We can see that several channels are highly loaded, but that the channel with CHPID 42 is much worse than the others. This is one of the two channels serving the string 27x, the other is 02 with a much lighter load.

```
FCX107        CPU nnnn   SER nnnnn   Interval 14:46:00 - 14:51:00     Perf. Monitor

CHPID    %Busy                Channel %Busy Distribution 07:31:00-14:51:00
 Nr.   Cur Ave   0-10 11-20 21-30 31-40 41-50 51-60 61-70 71-80 81-90 91-100
 42    66  57      3    1     2     8    17    24    28    12     4      0
 01    45  39      4    6    14    26    34    14     2     0     0      0
 02    41  33      5   10    17    38    22     7     0     0     0      0
 41    41  38      5    7    16    22    30    17     3     0     0      0
 56    37  32      8   12    22    35    19     3     0     0     0      0
 07    37  28      9   12    28    34    14     2     0     0     0      0
 06    33  30      8   14    24    33    18     2     0     0     0      0
 57    29  29      7   15    29    30    16     5     0     0     0      0
 04     8   9     73   24     3     0     0     0     0     0     0      0
 55     8   7     82   17     0     0     0     0     0     0     0      0
 26     4   3     92    7     0     0     0     0     0     0     0      0
```

Further investigation (using the command 'Q PATHS 271') revealed that one of the two paths to this string was offline due to a hardware problem. When that problem was fixed, and the path varied online, the 'disconnect' times and the total device response times were back to normal again.

## Action (Control Unit and Channel Path Level)

*Reduce channel path I/O contention* by re-distributing disks to channel paths with less load or using more paths for handling the total I/O.

*Reduce control unit contention* for shared DASD strings by re-distributing disks with high I/O load to strings which are mainly used by *this* system.

In the example above all DASD strings were shared, but mostly for occasional read-only accesses to minidisks on the other system. The I/O performance data provided by the PERFKIT performance monitor does not allow you to distinguish between problems caused by an overloaded channel path or an overloaded control unit, unless different control units are connected to the same channel path(s). In such a case you may detect an overloaded control unit if just the disks connected to one control unit show abnormally long disconnect times.

Something very similar actually happened in the example shown. Disk VMPG1I with number 265 is part of a different string, but it is connected to the same channel paths and did *not* show the excessive disconnect times of the paging disks on the 27x string because only a channel path to the second string was offline. The problem was not the channel path itself, but a bug in the string controller (head of string).

## Analyzing an Overloaded CPU

A CPU is overloaded when a high percentage of active users is found in CPU wait, that is, does not get as much processor time as they want to. This situation is likely to develop on systems which run near to their maximum capacity for much of the time. The things to look for are:

• High CPU wait percentages for the users, and a high number of active users in the dispatch list

• High CPU load figures

However, the problem can be a bit more complicated:

*CPU Wait:* The information is obtained by the high-frequency sampling technique of the CP monitor.

*CPU Load:* The CPU load percentages are based on real elapsed time. For VM systems running in an LPAR or second level under another VM, this may mean that processors may run at maximum capacity even though their utilization is still well below 100%. On the other hand, a high CPU load alone does not necessarily indicate a bottleneck. CP's dispatching algorithm usually takes care of a single CPU-intensive machine, and, as long as the CPU-bound user does not have a high dispatching priority, CP will distribute CPU time so well that only very little degradation is felt even if one machine is in a permanent CPU loop.
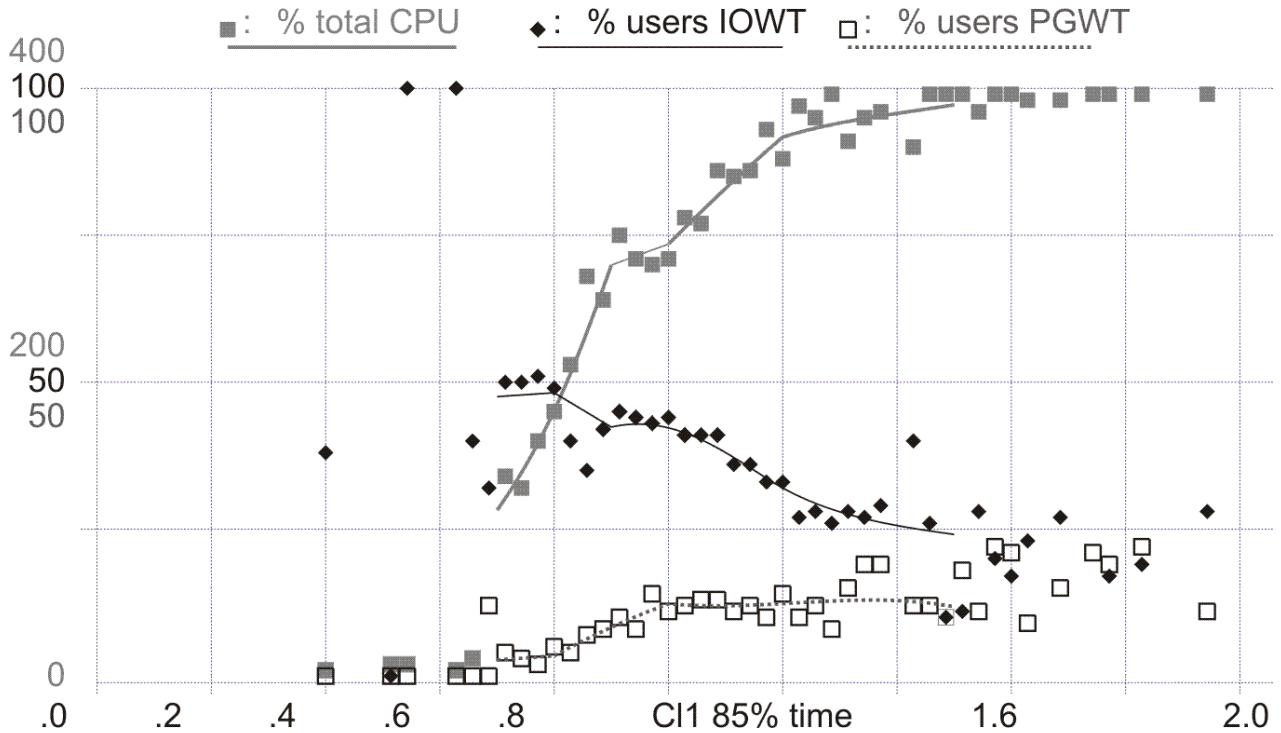
## Using the Performance Toolkit to Monitor

We will use the 'variable correlation display' to check how response times are related to CPU load and user wait states: enter the command 'GRAPHVAR C1ES CPU %IQ %PQ'.
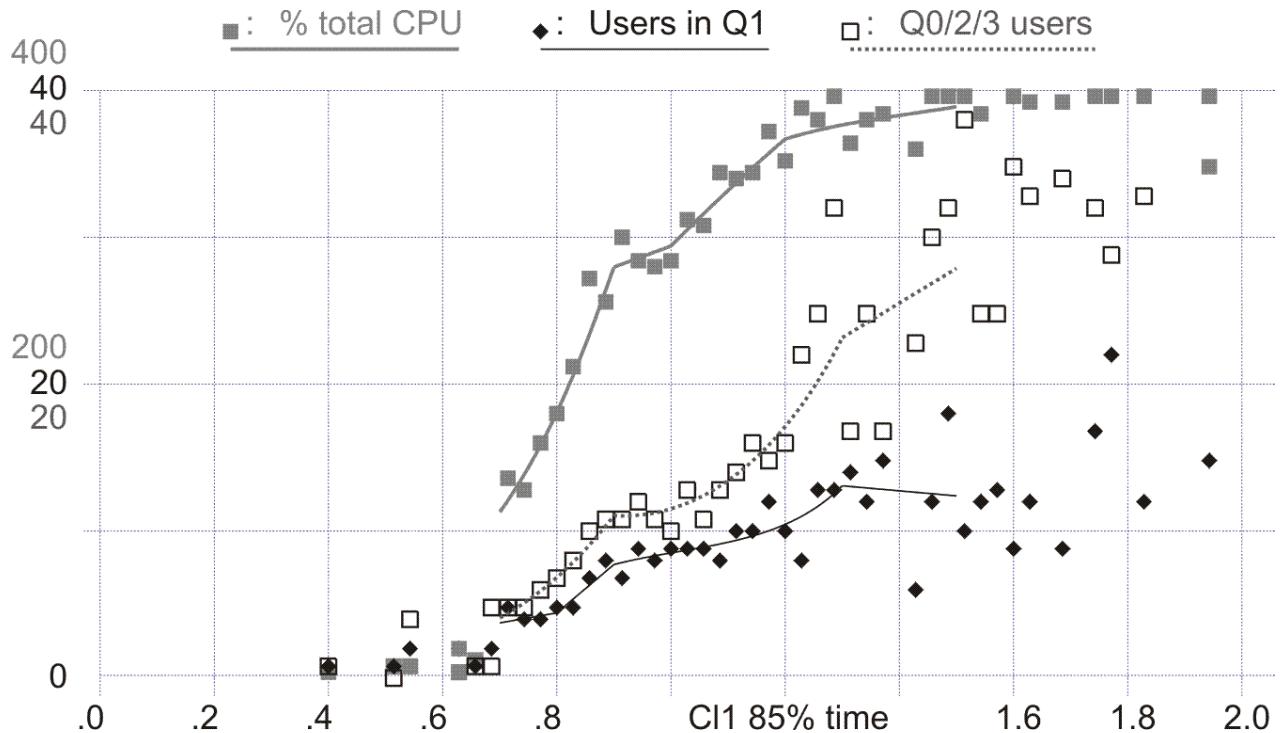
The resulting display (see the following figure) shows total CPU load and user I/O and page wait percentages on the vertical y-scale, based on the corresponding values of C1ES, the class 1 elapsed time slice within which 85% of all class 1 transactions could be completed (our internal interactive 'response time') which is plotted on the horizontal x-scale.

On this display:

1. Check whether CPU load is at, or near, system capacity for high response times.

2. If the CPU load really is high (as in the example), check the plots for variables %IQ and %PQ. They indicate what percentage of the in-queue users were found in I/O wait and pagewait, and they should all be low even for high response times. High %IQ or %PQ values indicate your problem is not (only) CPU load, but that other factors at least contribute to it.



Plot also the values for Q1 and Qx (command 'GRAPHVAR C1ES CPU Q1 QX').

| : % total CPU | : Users in Q1 | : Q0/2/3 users |

They give the average number of users in queues 1 and 0/2/3, and they should show higher values during periods of high response time and CPU utilization (note the increase of users waiting in queues 0, 2, and 3 in the example). If they do not, there probably is no real contention for the CPU, but only one machine using up whatever CPU power is left by the others.

To verify whether a single user might have caused most of the high CPU load, switch to the REDISPLAY screen by entering the command 'RED MAXUSER' and then look at the column headed MAX. CPU.
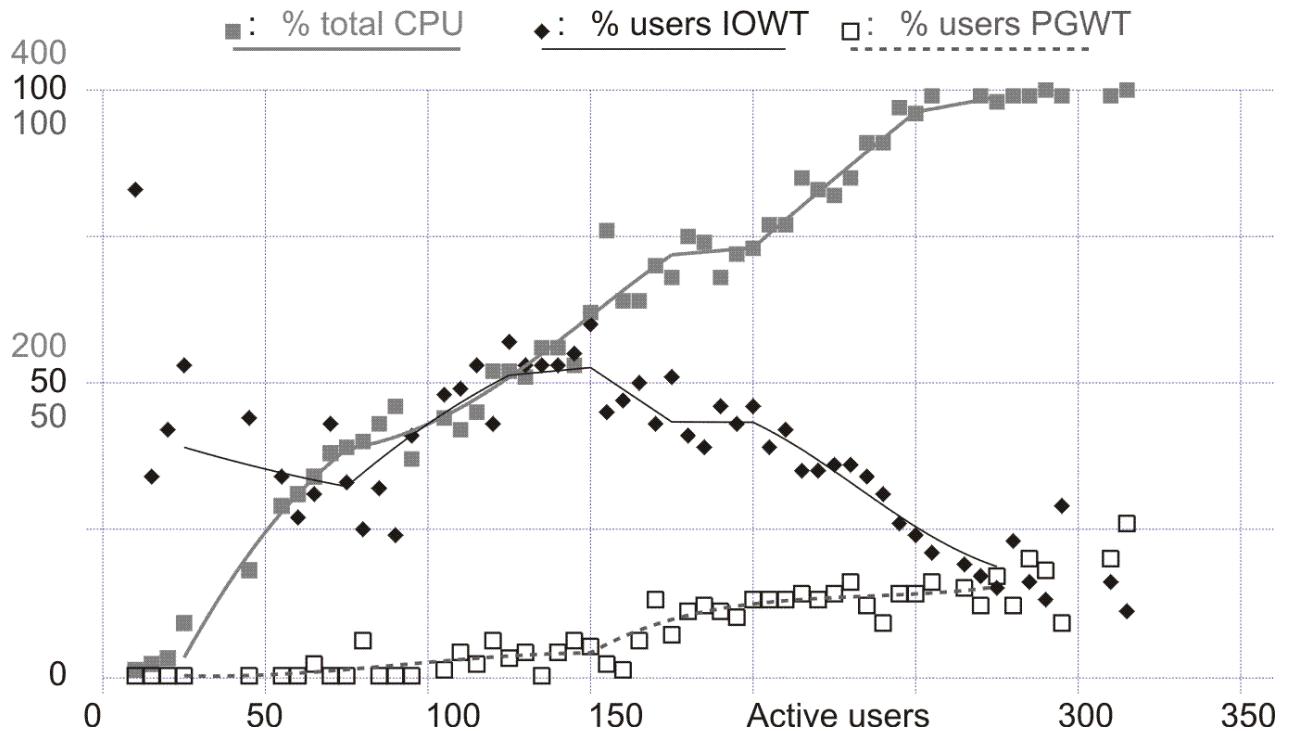
```
FCX101      CPU nnnn  SER nnnnn  Interval 09:52:51 - 09:54:23    Perf. Monitor

TIME >MAX. CPU       MAX. VECTOR    MAX. I/O       MAX. PAGING    MAX. RESIDEN
09:46 CHVTAM    30.4 ........    ... CHVTAM    38.6 CHCAL    16.2 CHCAL     4
09:47 CHVTAM    30.1 ........    ... CHVTAM    39.3 CHCAL    20.4 CHRSCS    3
09:49 CHVTAM    29.4 ........    ... CHVTAM    38.3 CHCAL    17.7 FIS       4
09:50 CHVTAM    32.7 ........    ... CHVTAM    41.7 CHCAL    23.4 CHVTAM    3
09:52 CHVTAM    30.4 ........    ... CHVTAM    38.3 CHCAL    20.8 CHVTAM    3
09:54 CHVTAM    27.6 ........    ... CHVTAM    35.0 CHCAL    17.4 MPP       3
..... ......    .... ........    ... .......    .... .....    .... .....    ..
```

This column shows the ID and CPU usage of the virtual machine with the highest CPU load during each of the measuring intervals, and we can see that, although there *was* a single virtual machine responsible for the highest CPU load for all of the measurements, it was:

- The VTAM service machine, which *has* to run, regardless of its load, and

- The VTAM machine's load, even though high, still represents only about 8% of total system load (the user CPU load percentages shown are based on *one* processor only).

In other words, there is no single end-user to blame for the heavy CPU load.

So far, our example has shown all the signs of an overloaded CPU. To better explain what is happening in the system, we will create yet another plot: enter the command 'GRAPHVAR ACT CPU %IQ %PQ'.

400
100
100

200
50
50

0

0   50   100   150   Active users   300   350

The graphs show how CPU load increases more or less linearly as more users become active on the system. With about 250 active users the CPU is fully loaded and any additional users will slow down the rate at which others receive service.

This is further illustrated by the percentage of in-queue users which are found in I/O wait as long as the system is moderately loaded. This percentage is between 40% and 60% which apparently is normal for the job mix being run. As CPU load exceeds 300% — the load for all four processors on a 3084 — we find that the percentage of users in I/O wait decreases. The reason for this is that the in-queue users now spend a larger fraction of their time waiting for CPU. The system can easily handle up to 220 active users, but it cannot give adequate service to more than 250 active users for the current job mix.

## Action

If, as in the example shown, there is no single machine to blame, or if its priority must remain high, you need a more powerful CPU. To bridge the gap until the additional CPU power is available, you could try to:

- Shift some CPU-intensive work to off-hours
- Submit this work to a CMS Batch machine, thereby making sure only one CPU-bound job can run at a time
- Give your important machine(s) a high share of CPU, and lower the share of others where possible (for example, CMS batch)

If the data in column MAX. CPU of the redisplay screen tells you a specific user is reponsible for a large share of total CPU load, you should analyze the work performed in that machine. If this user's work is not really very urgent, you can overcome your CPU problem simply by giving it a lower priority (that is, define a value lower than the default of 100 for the relative share value in the SHARE statement of its directory entry, or specify a lower number temporarily using the SET SHARE CP command).

## Analyzing a Single Virtual Machine

If any single virtual machine has a performance problem, it can be due to one of the following:

- Specific requirements made on system resources (CPU, paging, general I/O)
- Problems within the application(s) running in the machine (for example, data base organization, I/O blocking factors)

The Performance Toolkit monitor is a tool which can only help you to analyze the virtual machine's behavior from the outside, that is, you can use it to make sure the machine gets all the system resources it can reasonably expect. Quite often this approach gives you all the information needed to eliminate the problem. Even if the problem turns out to be caused by application design or implementation, your 'external' analysis will, in most cases, provide pointers which allow a more efficient 'internal' analysis of these applications.

## Using the Performance Toolkit to Monitor

Switch to the detailed user display (USER *userid* command).

```
FCX115      CPU nnnn  SER nnnnn  Interval 16:19:03 - 16:20:17    Perf. Monitor

Detailed data for user CHCAL

Total CPU   :   3.7%    Storage def. :     4MB    Page fault rate:   8.9/s
Superv. CPU :   2.3%    Resident <2GB:      34    Page read rate :  18.5/s
Emulat. CPU :   1.4%    Resident >2GB:     235    Page write rate:  13.4/s
Reserved                Proj. WSET   :     266    Pgs moved >2GB>:  ..../s
Reserved                Reserved pgs :       0    Main > XSTORE  :  ..../s
Reserved                Locked pages :       0    XSTORE > main  :  ..../s
Reserved                XSTORE dedic.:     0MB    XSTORE > DASD  :  ..../s
I/O rate    :  10.6/s   XSTORE pages :       0    SPOOL pg reads :    .0/s
DASD IO rate:  10.6/s   DASD slots   :     519    SPOOL pg writes:    .0/s
UR I/O rate :    .0/s   IUCV X-fer/s :    .0/s    MDC insert rate:  ..../s
Diag. X'98' :    .0/s   Share        :       0    MDC I/O avoided:    .0/s
*BLOCKIO    :    .0/s   Max. share   :     ...

#I/O active :      0    Active   : 36%    PSW wait : 35%    I/O act. :  0%
Stacked blk :     ..    Page wait : 37%   CF wait  :  0%    Eligible :  0%
Stat.:  BCM,DSC,PGWT    I/O wait  : 26%   Sim. wait:  0%    Runnable :  2%

Proc.   %CPU   %CP   %EM     IO/S    Status
00       .1    .1    .0     4.5    EME,P06,PSWT
01       .0    .0    .0     2.1    EME,P06,PSWT
02       .0    .0    .0      .9    EME,P06,RNBL
03       .0    .0    .0      .5    EME,P06,PSWT
04       .0    .0    .0      .0    EME,P06,PSWT
05       .0    .0    .0      .0    EME,P06,PSWT

Data Space Name             Size Mode  PgRd/s PgWr/s XRd/s XWr/s Migr/s Steal/s
BASE                      4096KB Priv  ...    ...    ...    ...    ...      ...

Device activity (for dedicated devices) and status:

0009 3215    .0                        000C 254R     CL *, EOF    NOH NCNT
000D 254P      CL M, CO 01, NOH NCNT   000E 1403     CL A, CO 01, NOH NCNT
0190 3380    .. 0147,RR,  72CYL,--->0  0191 3380  1.1 0607,WR,   5CYL,--->0
019D 3380    .. 0146,RR,  34CYL,--->0  019E 3380    .. 0140,RR,  44CYL,--->0
0314 3380    .. 0157,RR, 100CYL,--->0  0315 3380    .. 0157,RR, 130CYL,--->0
0319 3380    .. 0157,RR, 100CYL,--->0  031A 3380    .. 0157,RR,  10CYL,--->0
0333 3380    .. 0147,RR,  10CYL,--->0  0391 3380    .. 060B,WR,   5CYL,--->0
0395 3380    .. 072A,RR,  31CYL,--->0  0398 3380    .. 060B,RR,  18CYL,--->0
0399 3380    .. 0277,RR,  25CYL,--->0  04FB 3380    .. 0604,WR,  20CYL,--->0
04FC 3380    .. 0607,WR,  20CYL,--->0  04FD 3380    .. 0604,WR,  20CYL,--->0
04FE 3380    .. 0603,WR,  20CYL,--->0  04FF 3380    .. 0603,WR,  20CYL,--->0
05FB 3380  4.8 0724,WR,  50CYL,BUSY    05FC 3380    .. 0727,WR,  50CYL,--->0
05FD 3380  1.3 0726,WR,  50CYL,--->0   05FE 3380  1.7 0604,WR,  50CYL,--->0
05FF 3380   .6 060F,WR,  50CYL,--->0
Enter 'STOrage Display' for storage details

Command ===> _
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 41. A User Resource Details report (FCX115)*

The example is from a PROFS™ calendar service machine on a system with several hundred PROFS users.

Check the machine status fields first (second group of values at center of display). These fields are copies from the user status display and the values shown are calculated based on the machine's status flags as they were sampled at the end of each measuring interval. They will usually provide pointers to the current problem.

- Check for a paging problem by looking at fields `PG wait` (percent page wait state), `Eligible` (percent waiting in an eligible list, possibly for storage), and `Page fault rate`. A high page wait percentage may be caused by a general paging problem or by a high page fault rate, and a high page fault rate is usually caused by a large working set (many pages required in storage to allow the application to run) and/or a storage constrained environment which causes the machine's pages to be removed from real storage before they are referenced again. The result is a correspondingly long response time because CMS machines will be in a wait state after a page fault. Although this may not be the case for other operating systems running under VM (for example, a VSE machine running with PAGEX set ON), excessive page faults will always cause this machine's response times to degrade.

  Any value other than zero for `Eligible` indicates some severe resource shortage, possibly a storage constrained environment which forces the machine to wait in an eligible list until sufficient main storage is available for accommodating the machine's working set.

  **Action:** You can try either one of the following:

  - Improving the system's paging environment in general (see actions defined under "Analyzing a Paging Bottleneck" on page 154)
  - Reducing paging for this specific machine, either by dedicating a certain number of pages to it, or by preventing premature removal of its pages from real storage

    Use the SET RESERVE command for dedicating a number of pages. Use the SET QUICKDSP command to prevent the machine from entering eligible lists. Both the SET RESERVE and SET QUICKDSP commands will allow better performance for this machine but at the expense of all others, so these commands should be used with caution.

    The high page wait percentage and the high page fault rate in our example both indicate a paging problem. Performance for this machine could be significantly improved by dedicating 1MB (256 pages) of storage to it. This made sense because the PROFS calendar machine in our example has to provide service for several hundred users on the system, and the impact on the other users due to less pageable storage could be justified.

- Check for an I/O problem by looking at field `I/O wait` (percent I/O wait state). Look at the load of the machine's virtual I/O devices, especially minidisks (listed under the heading `Device activity and status`). Problems with a specific minidisk could be indicated by a frequent busy status being shown.

  If you find minidisks with a very high I/O rate, the device is probably operating at its maximum speed, and it is being overloaded by *this* machine. The remedy would be to use caching if that is possible (control unit cache or minidisk cache), or to split the I/O over more real disks (for example, allocate highly used data sets on different real disks, and maybe split them if required) *if the application and operating system are capable of taking advantage of such a split.*

  If all minidisks show moderate I/O rates only and you still suspect an I/O problem, look at the real disks where the minidisks have been allocated. You can find the real device number of these disks immediately to the right of the I/O rate. Switch to the detailed device display by entering the DEV *cuu* command, where *cuu* is the real device number mentioned above.

```
FCX110       CPU nnnn   SER nnnnn   Interval 16:31:22 - 16:32:31      Perf. Monitor

Detailed Analysis for Device 0724 ( SYSTEM )
Device type :  3380-K       Function pend.:    4.5ms       Device busy   :    49%
VOLSER      :  VMPR64       Disconnected  :   14.4ms       I/O contention:    22%
Nr. of LINKs:     149       Connected     :    8.9ms       Reserved      :     0%
Last SEEK   :     508       Service time  :   27.8ms       SENSE SSCH    :     0
SSCH rate/s :    18.3       Response time :   33.9ms       Recovery SSCH :     0
Status: SHARABLE

Path(s) to device 0724:    07    4F
Channel path status   :    ON    ON

      MDISK Extent        Userid   Addr IO/s VSEEK Status     LINK MDIO/s
    +--------------------------------------------------------------------+
    |     0001 -  0015    MHO      0191  .0     0 WR            1     .0  |
    |     0031 -  0035    MALA     0191  .0     0 WR            1     .0  |
    |     0036 -  0050    JAK      0191 2.3     0 RR            2    2.3  |
    |     0068 -  0072    BBA      0291  .0     0 RR            3     .0  |
    |     .... -  ....                                              ...  |
    |     0498 -  0547    CHCAL    05FB 8.1     0 WR            1    8.1  |
    |     .... -  ....                                              ...  |
    |     2406 -  2410    NIL      0191  .8     0 WR            1     .8  |
    |     2541 -  2545    SWID     0191  .0     0 WR            1     .0  |
    |     2561 -  2570    RSIG     0191  .0     0 WR            1     .0  |
    +--------------------------------------------------------------------+

Command ===> _
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

*Figure 42. An I/O Device Details report (FCX110)*

This display gives you a lot of detailed information on the disk, and you should do one of the following:

– Check the total I/O rate on the disk (field SSCH rate/s), the smoothed I/O contention for the device (field I/O contention), the device busy value (field Device busy), and its response time (field Response time). A disk is usually considered 'overloaded' if it has to handle too many *simultaneous* I/O requests, so that some of them have to be queued while another I/O is already active. This queueing activity is indicated by the I/O contention values. A high I/O rate and/or high device busy values can also be indications of an overloaded disk.

– Check the activity of each minidisk located on this volume. If you find several minidisks with high load, and if the load is generated by different users, then there probably is I/O contention on the disk, that is, there is an I/O bottleneck on the device due to too many simultaneous I/O requests. Because this I/O activity could be sporadic, you may have to view the screen for several minutes until you really know what is happening.

– Look for possible channel path and control unit contention by checking the Disconnected time for the disk. Abnormally long Disconnected times are usually caused by overloaded control units or channel paths (see "Contention on Control Unit and Channel Path Level" on page 163 for details).

**Action:** If the volume contains several minidisks with high I/O activity from different users, you should either:

– Let them use caching (if possible)

– Move them to separate volumes

– See that some of the I/O intensive work is shifted to off-hours

If an overloaded path seems to be the problem, you can either move the whole volume, or at least the minidisk(s) with high load to another disk string on a different path.

In the case of the PROFS calendar machine from our example, I/O activity on the calendar minidisks soon became the next bottleneck once the paging problem had been solved. By placing the calender minidisks on disk packs connected to a control unit with cache, the virtual machine's performance could be dramatically improved.

- If neither paging nor I/O seems to be the problem, look at field `Total CPU` which shows the total CPU usage of the machine. If CPU usage is often high, and if your CPU is highly loaded anyway, this may be the bottleneck.

  **Action:** If the machine's applications are important enough, give it a higher dispatching priority (that is, define a higher relative or absolute share in the machine's directory entry) or use the SET SHARE CP command to temporarily allocate the machine a bigger share of CPU.

# Appendix A. A Sample Initialization Profile (FCONX $PROFILE)

```
*************************************************************************
*************************************************************************
*                                                                       *
**        Initialization profile for 'Performance Toolkit for VM'.    **
*         This file is intended for customizing operation by including  *
*         the necessary 'FCONTROL ....' commands:                       *
*                                                                       *
*         FCONTROL ...                                                  *
*                                                                       *
*         .. ACTMSG                                                     *
*         .. AUTOREFR                                                   *
*         .. BENCHMRK                                                   *
*         .. COLOR                                                      *
*         .. DEFLOG                                                     *
*         .. DEFSCRN                                                    *
*         .. DEVGROUP                                                   *
*         .. DUMPID                                                     *
*         .. EMERGENC                                                   *
*         .. FORCEUSR                                                   *
*         .. GDDMSPEC                                                   *
*         .. HMA                                                        *
*         .. LIMIT                                                      *
*         .. MAINTID                                                    *
*         .. MAXREC                                                     *
*         .. MINPATHS                                                   *
*         .. MODEL                                                      *
*         .. MONCACHE                                                   *
*         .. MONCOLL                                                    *
*         .. MSGCLEAR                                                   *
*         .. MSGWAIT                                                    *
*         .. NUMBERS                                                    *
*         .. PFKEY                                                      *
*         .. PROCESS                                                    *
*         .. RMTLINES                                                   *
*         .. SCROLL                                                     *
*         .. SEARCH                                                     *
*         .. SECUSER                                                    *
*         .. SETEVENT                                                   *
*         .. SETTINGS                                                   *
*         .. TIMSTAMP                                                   *
*         .. UCLASS                                                     *
*         .. UPDTCMS                                                    *
*         .. UPDTSCRN                                                   *
*         .. USERBOTL                                                   *
*         .. USERHDR                                                    *
*         .. USERVAR                                                    *
*         .. USRLIMIT                                                   *
*                                                                       *
*         See User's Guide or 'HELP' text for command syntax and        *
*         meaning of arguments.                                         *
*         Other commands can also be included, but they will be         *
*         executed only once PERFKIT is fully initialized.              *
*                                                                       *
*************************************************************************
*************************************************************************
*
*-----------------------------------------------------------------------*
*    Define number of CP action messages to be left pending             *
*    at top of screen (default is 10 messages)                          *
*-----------------------------------------------------------------------*
FC ACTMSG  10
*
*-----------------------------------------------------------------------*
*    Define an object for which detailed by-time logs                   *
*    are to be built (requires additional space in your virtual         *
*    storage; activate only when needed).                               *
*                                                                       *
*    Allowed objects are: I/O device number, user/identity ID,          *
*    Logical Link Partner name, End Point ID, PCI Function ID.           *
*-----------------------------------------------------------------------*
```

```
*C BENCHMRK DEVICE 550      FILE 08:00 TO 17:00
*C BENCHMRK USER   PERFSVM
*C BENCHMRK LLINK  SSIMEMB2
*C BENCHMRK EPOINT 00000001
*C BENCHMRK PCIFUNC 00000040
*
*-----------------------------------------------------------------------*
*    Define extended highlighting and colors to be used                 *
*-----------------------------------------------------------------------*
FC COLOR    TOPDAT    BLUE
FC COLOR    TOPSCRL   TURQUOIS
FC COLOR    TOPSECU   TURQUOIS
FC COLOR    TOPSTAT   YELLOW
FC COLOR    NUMBERS   BLUE
FC COLOR    ERRMSG    RED
FC COLOR    ARROW     WHITE
FC COLOR    INAREA    WHITE     UNDER
FC COLOR    BOTLINE   TURQUOIS
FC COLOR    TOFEOF    WHITE
FC COLOR    INRED     WHITE
FC COLOR    CPMSG     GREEN     REV
FC COLOR    CPWNG     YELLOW    REV
FC COLOR    CPMSGN    WHITE
FC COLOR    CPOUT     BLUE
FC COLOR    CPAMSG    YELLOW
FC COLOR    CPEMSG    RED
FC COLOR    CPIMSG    YELLOW
FC COLOR    CPSCIF    GREEN
FC COLOR    VMOUT     GREEN
*
*-----------------------------------------------------------------------*
*    Define destination ID for PERFKIT dumps, in case of an abend       *
*-----------------------------------------------------------------------*
*C DUMPID MAINT
*
*-----------------------------------------------------------------------*
*    Define colors and shading patterns for GDDM graphics               *
*-----------------------------------------------------------------------*
FC GDDMSPEC  VAR1   COL   YELLOW    PAT   5
FC GDDMSPEC  VAR2   COL   TURQUOIS  PAT   14
FC GDDMSPEC  VAR3   COL   ORANGE    PAT   12
FC GDDMSPEC  VAR4   COL   GREEN     PAT   9
*
*-----------------------------------------------------------------------*
*    Indicate whether perf. data are to be collected continuously,      *
*    even while not in perf. monitor mode, and other performance        *
*    monitor related information (valid for privilege class 'E' only)   *
*    The 'FC MONCOLL ON' statement MUST precede any 'FC EMERGENC',      *
*    'FC FORCEUSR', 'FC LIMIT' and 'FC USRLIMIT' commands.              *
*-----------------------------------------------------------------------*
FC MONCOLL REDISP 720
*    Following command for MONITOR data extraction from disk file
*C MONCOLL CPMON DISK ON MONWRITE 191 B MONFILE DATA
*    Following command for MONITOR data extraction from MONDCSS segment
FC MONCOLL CPMON DCSS ON MONDCSS
*    Following command to start populating formatted output collector
*C MONCOLL SEGOUT ON PERFOUT
*-----------------------------------------------------------------------*
* The following FC MONCOLL RESET commands are used to set the times     *
* for PRINT, TREND and SUMMARY files                                    *
*-----------------------------------------------------------------------*
* Setup for daily reports
FC MONCOLL RESET 00:00:00R_P 23:59:59P
* Setup for interim reset at midnight
FC MONCOLL RESET 00:00:00R_I 23:59:59I (MERGE
* Setup for trend file by shift
FC MONCOLL RESET 00:00:00R_T 08:00:00T 16:00:00T (MERGE
* Setup for hourly summary files
*C MONCOLL RESET 00:00:00R_S 01:00:00S 02:00:00S 03:00:00S (MERGE
*C MONCOLL RESET 04:00:00S 05:00:00S 06:00:00S 07:00:00S (MERGE
*C MONCOLL RESET 08:00:00S 09:00:00S 10:00:00S 11:00:00S (MERGE
*C MONCOLL RESET 12:00:00S 13:00:00S 14:00:00S 15:00:00S (MERGE
*C MONCOLL RESET 16:00:00S 17:00:00S 18:00:00S 19:00:00S (MERGE
*C MONCOLL RESET 20:00:00S 21:00:00S 22:00:00S 23:00:00S (MERGE
*    Following command controls the collection of general performance
*    data on disk during the specified time period
FC MONCOLL PERFLOG ON 06:00 19:59 MO-FR
*    Following command activates VMCF data retrieval interface
*C MONCOLL VMCF ON
*    Define the maximum allowed number of Internet connections
*C MONCOLL WEBSERV MAXCONN 10
*    Define the timeout of inactive Internet connections in minutes
```

```
*C MONCOLL WEBSERV TIMEOUT 30
*    Following command activates Internet interface
*C MONCOLL WEBSERV ON TCPIP TCPIP 81
*    Following command activates Internet interface with SSL
*C MONCOLL WEBSERV ON SSL TCPIP TCPIP 81
*    Following command activates TCP/IP interface for data retrieval
*    from LINUX RMF DDS interface
*C MONCOLL LINUXUSR ON TCPIP TCPIP
*
*-------------------------------------------------------------------------*
*    Define a new performance log screen, using parts of other       *
*    performance screens.                                            *
*-------------------------------------------------------------------------*
*C DEFLOG MYLOG H1 Sample log with some fields copied from the SYSTEM screen
*C DEFLOG MYLOG DESC Sample DEFLOG - fields from the SYSTEM screen
*C DEFLOG MYLOG COL 12 LEN 5 COPY SYSTEM LINE 14 COL 34 NAME SieEx
*C DEFLOG MYLOG COL 18 LEN 5 COPY SYSTEM LINE 14 COL 75 NAME SieInter
*C DEFLOG MYLOG COL 25 LEN 13 COPY SYSTEM LINE 19 COL 26 NAME Subp-free-<2G
*C DEFLOG MYLOG COL 39 LEN 13 COPY SYSTEM LINE 20 COL 26 NAME Subp-free->2G
*C DEFLOG MYLOG COL 53 LEN 13 COPY SYSTEM LINE 19 COL 67 NAME Totl-free-<2G
*C DEFLOG MYLOG COL 67 LEN 13 COPY SYSTEM LINE 20 COL 67 NAME Totl-free->2G
*
*C DEFLOG MYLOG2 H1 Sample log with fields from STORAGE / RESPALL / USER / DEV
*C DEFLOG MYLOG2 DESC Sample DEFLOG - fields from STORAGE/RESPALL/USER/DEV
*C DEFLOG MYLOG2 COL 12 LEN 7 COPY STORAGE LINE 28 COL 73 NAME MDC-Rd
*C DEFLOG MYLOG2 COL 21 LEN 7 COPY STORAGE LINE 29 COL 73 NAME MDC-Wr
*C DEFLOG MYLOG2 COL 29 LEN 7 COPY STORAGE LINE 30 COL 73 NAME MDC-Ht
*C DEFLOG MYLOG2 COL 37 LEN 5 COPY STORAGE LINE 31 COL 74 NAME MDC-%
*C DEFLOG MYLOG2 COL 44 LEN 5 COPY RESPALL LINE LAST COL 12 NAME Trv-s
*C DEFLOG MYLOG2 COL 51 LEN 5 COPY RESPALL LINE LAST COL 24 NAME NTr-s
*C DEFLOG MYLOG2 COL 58 LEN 8 COPY USE LOC 'PERF' ATC 3 COL 2 NAME Userid
*C DEFLOG MYLOG2 COL 68 LEN 6 COPY DEV LOC '0020' ATC 3 COL 7 NAME D-Type
*C DEFLOG MYLOG2 FILE 08:30 TO 16:30
*
*
*-------------------------------------------------------------------------*
*    Define a new performance screen, using parts of other perfor-   *
*    mance screens.                                                  *
*-------------------------------------------------------------------------*
*C DEFSCRN SYSTSUM DESC Sample DEFSCRN - lines from CPU/DEVICE/CHANNEL/USER
*C DEFSCRN SYSTSUM LINE  2 TO  9 COPY CPU      FROM 1
*C DEFSCRN SYSTSUM LINE 11 TO 12 COPY DEVICE   FROM 2
*C DEFSCRN SYSTSUM LINE 13 TO 16 COPY DEVICE   FROM 5
*C DEFSCRN SYSTSUM LINE 18 TO 21 COPY CHANNEL FROM 1
*C DEFSCRN SYSTSUM LINE 23 TO 23 COPY USER     FROM 4
*C DEFSCRN SYSTSUM LINE 24 TO 27 COPY USER     FROM 6
*
*-------------------------------------------------------------------------*
*    Define a GENERAL screen                                         *
*-------------------------------------------------------------------------*
*C DEFSCRN GENERAL DESC Sample DEFSCRN - lines from USER/DEVICE/CPU
*C DEFSCRN GENERAL LINE  2 TO  2 COPY USER     FROM 4
*C DEFSCRN GENERAL LINE  3 TO  6 COPY USER     FROM 6
*C DEFSCRN GENERAL LINE  7 TO  7 COPY DEVICE   FROM 2
*C DEFSCRN GENERAL LINE  8 TO 11 COPY DEVICE   FROM 5
*C DEFSCRN GENERAL LINE 12 TO 19 COPY CPU      FROM 1
*
*-------------------------------------------------------------------------*
*    Activate the Emergency Safeguard Feature (ESF) of PERFKIT       *
*-------------------------------------------------------------------------*
*    Define the actions to be performed at the emergency event
*C EMERGENC EXECPROC MYPROC
*C EMERGENC ORDER EXECPROC PRINT INTERIM TREND SUMMARY REMPRINT
*    Enable ESF
*C EMERGENC ON
*
*-------------------------------------------------------------------------*
*    Define thresholds when idle users, and users in a CPU loop      *
*    or an I/O loop are to receive warnings and are to be forced     *
*    off the system.                                                 *
*    CAUTION: use 'TEST' mode first, or specify FORCE limit of '0'   *
*    to make sure that the exclude list is complete, and no service  *
*    machines are going to be forced!                                *
*-------------------------------------------------------------------------*
*    Set size of EXCLUDE table
FC FORCEUSR EXCLMAX 200
*    Define users which are NOT to be forced
FC FORCEUSR EXCLUDE OP       VTAM*    VSCS*    RACF     RSCS
FC FORCEUSR EXCLUDE MVS*     VMUTIL
*    Define loop detection criteria
*C FORCEUSR SETLIM  CPULOOP  MINCPU 10  WSSDELTA 1  MAXWSS 10000
*C FORCEUSR SETLIM  IOLOOP   MINIO 500  WSSDELTA 4  MAXWSS 10000
```

```
*C FORCEUSR SETLIM  WSSLOOP              WSSDELTA 15 MAXWSS 10000
*   Define warning and FORCE thresholds (in minutes)
*C FORCEUSR IDLE    NODISC TEST  LIMIT    30/40
*C FORCEUSR IDLE    DISC         LIMIT    17/20
FC FORCEUSR IDLE    OFF
*C FORCEUSR CPULOOP ALL          LIMIT     5/0
FC FORCEUSR CPULOOP OFF
*C FORCEUSR IOLOOP  ALL          LIMIT     5/0
FC FORCEUSR IOLOOP OFF
*C FORCEUSR WSSLOOP ALL          LIMIT    15/20
FC FORCEUSR WSSLOOP OFF
*
*---------------------------------------------------------------------*
*    Set thresholds for some key performance indicators               *
*    (can be set only if permanent perf. data collection has          *
*    previously been activated)                                       *
*---------------------------------------------------------------------*
FC LIMIT    NORMCPU  90  1/1  5/10  WEIGHT  1
*C LIMIT    C1ES   1.00  1/1  5/10  WEIGHT 20
FC LIMIT    %IQ      50  1/1  5/10  WEIGHT  1
FC LIMIT    %PQ      30  0/0  5/10  WEIGHT  1
FC LIMIT    %SPSL    80  0/0  1/10  WEIGHT 40
FC LIMIT    %PGSL    80  0/0  1/10  WEIGHT 40
FC LIMIT    %CHBUSY  40  1/1  5/10  WEIGHT  5
FC LIMIT    DVQUEUE .40  1/1  5/10  WEIGHT  1
FC LIMIT    DVRESP   30  1/1  5/10  WEIGHT  1
FC LIMIT    MISSINT   0  0/0  5/10  WEIGHT 10
*   The following three thresholds are set to control the virtual
*   storage shortage condition:
*   HMAPAGE  - HMA storage >2G allocated to the page pool
*   HMAPRINT - HMA storage >2G allocated to the print buffer
*   STORB2G  - total conventional storage <2G allocated
FC LIMIT    HMAPAGE  80  0/0  1/10  WEIGHT 50
FC LIMIT    HMAPRINT 80  0/0  1/10  WEIGHT 50
FC LIMIT    STORB2G  80  0/0  1/10  WEIGHT 50
*
*---------------------------------------------------------------------*
*    Define destination ID for high priority alert messages           *
*---------------------------------------------------------------------*
*C MAINTID MAINT AT mynode
*
*---------------------------------------------------------------------*
*    Define number of records to be written before CLOSE              *
*    (default is 10 records)                                          *
*---------------------------------------------------------------------*
FC MAXREC  12
*
*---------------------------------------------------------------------*
*    Set thresholds for automatic channel path monitoring             *
*---------------------------------------------------------------------*
*C MINPATHS 2 TO 600-61F LIMIT 0/60 WEIGHT 50
*C MINPATHS 2 TO 720-73F LIMIT 0/60 WEIGHT 50
*C MINPATHS 4 TO 840-85F LIMIT 0/60 WEIGHT 20
*
*---------------------------------------------------------------------*
*    Activate monitoring of cache status for selected disks           *
*---------------------------------------------------------------------*
*C MONCACHE 720-724 AD*  LIMIT 0/60 WEIGHT 10
*C MONCACHE 725-72F AA*  LIMIT 0/60 WEIGHT 10
*C MONCACHE 730-73F A**  LIMIT 0/60 WEIGHT  5
*
*---------------------------------------------------------------------*
*    Specify action to be taken when CLEAR command is executed,       *
*    or PA2 or CLEAR key hit (CP messages and/or PERFKIT action       *
*    messages to be cleared too?)                                     *
*---------------------------------------------------------------------*
FC MSGCLEAR ALL
*
*---------------------------------------------------------------------*
*    Define delay in minutes for messages to be left on screen        *
*    until warning message is displayed (initial value: OFF)          *
*---------------------------------------------------------------------*
*C MSGWAIT  5
*
*---------------------------------------------------------------------*
*    Specify whether line numbers are to be inserted on left side     *
*---------------------------------------------------------------------*
FC NUMBERS  OFF
*
*---------------------------------------------------------------------*
*    Define program function keys                                     *
*    Note that the 'FUNCTION' definitions just repeat the initial     *
```

```
*    definitions, they have been included only to illustrate the      *
*    use of the 'FC PFKEY' command.                                    *
*----------------------------------------------------------------------*
*  Definitions for common functions (basic, redisplay and monitor)
FC PFKEY SET  1  FUNCTION HELP
FC PFKEY SET  2  FUNCTION REDISP
FC PFKEY SET  3  FUNCTION QUIT
FC PFKEY SET  4  FUNCTION TOP
FC PFKEY SET  5  FUNCTION BOTTOM
FC PFKEY SET  6  FUNCTION SAVE
FC PFKEY SET  7  FUNCTION BACKWARD
FC PFKEY SET  8  FUNCTION FORWARD
FC PFKEY SET  9  FUNCTION PRINT
FC PFKEY SET 10  FUNCTION LEFT
FC PFKEY SET 11  FUNCTION RIGHT
FC PFKEY SET 12  FUNCTION RETURN
*
*  Definitions for commands to be executed in basic mode
FC PFKEY SET  4  BASMODE   MONITOR
*C PFKEY SET  5  BASMODE   ...
FC PFKEY SET 11  BASMODE   ?
*
*  Definitions for commands to be executed in performance monitor mode
*C PFKEY SET  3  MONMODE   ...
*
*----------------------------------------------------------------------*
*    Allow special processing for some kinds of output lines          *
*----------------------------------------------------------------------*
FC PROCESS CPMSG  * 'FROM RSCS' DISP CPMSGN
*C PROCESS CPO     * 'LOGON' NODISP
*C PROCESS CPO     * 'LOGOFF' NODISP
*C PROCESS CPO     * 'RECONNECT' NODISP
*C PROCESS CPO     * 'DSCONNECT' NODISP
*C PROCESS CPO     * 'DISCONNECT' NODISP
*C PROCESS CPO    51 'FORCED' DISPLAY CPO
*C PROCESS CPMSG  * 'BAND' | * 'TAPE' REROUTE TAPEOP    CPMSGN
*C PROCESS ERRMSG 9 'A'                 REROUTE OPERATOR CPMSGN
*
*----------------------------------------------------------------------*
*    Set default number of output lines to be retrieved in remote     *
*    performance data retrieval mode                                  *
*----------------------------------------------------------------------*
*C RMTLINES 50
*
*----------------------------------------------------------------------*
*    Define scroll mode                                               *
*----------------------------------------------------------------------*
FC SCROLL   AUTO  12
FC SCROLL   MSGS  ON
*
*----------------------------------------------------------------------*
*    Define command search order                                     *
*----------------------------------------------------------------------*
FC SEARCH   CPCMS
*
*----------------------------------------------------------------------*
*    Define USERID for which we are acting as secondary console       *
*----------------------------------------------------------------------*
FC SECUSER  OFF
*
*----------------------------------------------------------------------*
*    Define timer events                                             *
*----------------------------------------------------------------------*
FC SETEVENT M-F 12:00 CP MSG * It is now 12:00h
*
*----------------------------------------------------------------------*
*    General definitions for controlling history file creation        *
*    and print output format                                         *
*----------------------------------------------------------------------*
FC SETTINGS HISTFILE NEW
*C SETTINGS MAXDEVS  50
*C SETTINGS MAXUSERS 50
FC SETTINGS PAGESIZE 60
FC SETTINGS INTERIM  60
FC SETTINGS SYSTEM   This is a performance report for system XYZ
*C SETTINGS SYSTEMID XYZ
*
*----------------------------------------------------------------------*
*    Control insertion of time-stamp in front of output lines         *
*----------------------------------------------------------------------*
FC TIMSTAMP ON
*
```

```
*-----------------------------------------------------------------------*
*    Define device groups for which group averages are to be calculated*
*-----------------------------------------------------------------------*
*C DEVGROUP ADD SYSVOLS 1A20-1A27 3C00 5328-532F
*C DEVGROUP ADD HILOAD  26B0-26FF 5101 520A
*
*-----------------------------------------------------------------------*
*    Define user classes for which group averages are to be calculated*
*-----------------------------------------------------------------------*
*C UCLASS   *    General
*C UCLASS   VSE* Guests
*C UCLASS   MVS* Guests
*C UCLASS   RSCS Service
*C UCLASS   VSCS Service
*C UCLASS   VTAM Service
*
*-----------------------------------------------------------------------*
*    Define screen update mode while CMS command is active          *
*-----------------------------------------------------------------------*
FC UPDTCMS  DELAYED
*
*-----------------------------------------------------------------------*
*    Define screen update mode: ANYMODE will let messages be inserted *
*    into the BASIC mode screen even while the program is operating  *
*    in another mode.                                                *
*-----------------------------------------------------------------------*
FC UPDTSCRN ANYMODE
*
*-----------------------------------------------------------------------*
*    Define user data to be inserted into bottom line               *
*-----------------------------------------------------------------------*
*C USERBOTL BASIC  PF4: MON      PF12: ?
*
*-----------------------------------------------------------------------*
*    Define user data to be used as default header info             *
*-----------------------------------------------------------------------*
FC USERHDR  Performance Toolkit for VM
*
*-----------------------------------------------------------------------*
*    Define user variables which you often need as input for graphics *
*    (GRAPHxxx or PLOTxxx commands). The following are just a few    *
*    simple examples to show how it works:                          *
*-----------------------------------------------------------------------*
* USERCP  = CP 'overhead' which can be attributed to users
FC USERVAR SET USERCP   = %CP - %SY
FC USERVAR SET USERCP   DESCR USER %CP
* USERINQ = the total number of in-queue users (dispatch+eligible lists)
FC USERVAR SET USERINQ  = Q1 + Qx + E1 + Ex
FC USERVAR SET USERINQ  DESCR USERS IN-Q
* CPU/TR  = amount of CPU used per transaction (trivial + non-trivial),
*           in milliseconds
FC USERVAR SET 'CPU/TR' = CPU*100 / ('TR/S'+'NT/S')
FC USERVAR SET 'CPU/TR' DESCR CPU MSEC/TR
*
*-----------------------------------------------------------------------*
*    Set user load thresholds (can be set only when permanent       *
*    perf. data collection has previously been activated)           *
*-----------------------------------------------------------------------*
FC USRLIMIT  *  %CPU  30  5/10  WEIGHT 10
FC USRLIMIT  *  IO/S  50  5/10  WEIGHT 10
*
*-----------------------------------------------------------------------*
*    Configure the High Memory Area (HMA) storage above the 2G line  *
*    and display the current settings                               *
*-----------------------------------------------------------------------*
*  Allocate 10% of HMA residing between 2G and 4G to the page pool
*  (the rest will be allocated to the print buffer)
FC HMA PAGE 10
*  Display HMA settings and statistics
FC HMA QUERY
*
*-----------------------------------------------------------------------*
*    Execute other commands for customizing PERFKIT                 *
*-----------------------------------------------------------------------*
*    The example sorts users in %CPU load sequence, and I/O devices
*    in I/O rate sequence. Remove the asterisks to activate it (works
*    only with priv. class 'E', causes exit from PERFKIT otherwise)
*MONITOR
*   SORT USER %CPU
*   SORT DEV  I/O
*   SORT CACH IO/S
*   SORT UPAG READS
```

```
*QUIT
*    End of example for sorting
*
*    End of Performance Toolkit for VM profile
```

# Appendix B. Building a DCSS for a Formatted Output Collector

To use Performance Toolkit to work with a formatted output collector (such as OMEGAMON XE on z/VM and Linux), you must set up Performance Toolkit to collect z/VM data and then write it to a discontiguous shared segment (DCSS). This DCSS must be defined to z/VM prior to use by Performance Toolkit or by the monitoring agent.

Use the FCONTROL MONCOLL subcommand to control data collection for performance monitoring. Use the SEGOUT option on that subcommand to specify the name of the DCSS that will be used to hold the data. Note that the default name of the DCSS for the SEGOUT option is PERFOUT – so if you decide to use the default name of the DCSS, specify PERFOUT. See the *z/VM: Performance Toolkit Reference* for more information on the FCONTROL MONCOLL subcommand.

## Estimating the Size of the DCSS

Before you can create the DCSS, you must determine the approximate size of the DCSS, based upon the workload demands on your system. That entails finding out how many devices are in use, how many users are accessing the system, and how many items are being monitored. For a small to medium system, the DCSS must be able to contain at least 6 MB of storage. In number of pages, this translates to 600 hexadecimal pages. In decimal, this translates to 1536 pages.

A REXX executable file, FCXSEGSZ EXEC, is packaged with Performance Toolkit. Use this exec to calculate the size of the DCSS. (It runs on any virtual machine where the CMS is available.) FCXSEGSZ EXEC does not delete any segments and it does not define any segments. It is a size calculator only. You need to provide input each time you invoke the tool, as the input is not saved across invocations.

Note that your responses to the prompts when running this exec must take into account normal fluctuations in system utilization in your installation. Your responses must also take into account expected increases in defined resources or increases in numbers of users. So when in doubt, it is best to provide higher counts. This will save you from having to increase counts in the future. Note, also, that the contents of a DCSS are not saved in the spool file – only the definitions for the DCSS are saved, thus the DCSS itself occupies a minor amount of spool space.

The size of the DCSS is based on a number of distinct record types, as listed below. The data records written to the DCSS will vary in length based on the type of record with which they are associated. The frequency of data collection is determined by the monitor SAMPLE interval setting.

The amount of data written to the DCSS is determined by the CP monitor domains that are enabled. If a domain is disabled, no data is written to the DCSS for that particular domain. You must enable the CP monitor domains for the types of data to be collected.

To run the size calculator, enter the following command at the CMS virtual machine that has access to the disk where Performance Toolkit is installed:

```
FCXSEGSZ
```

When you run FCXSEGSZ EXEC, you are prompted to provide counts for the following items in your installation:

1. **Defined LPARs (1-255)**. Enter the number of LPAR partitions that are currently defined or that will be defined in the processor complex where the monitored z/VM system resides. Provide a count of all partitions defined, regardless of the operating system. Include the monitored z/VM partition in the count.

2. **Monitored disk devices (1-65535)**. Enter the number of disk devices that are attached to the monitored z/VM system. If you use the MONITOR SAMPLE commands to limit the number of I/O

devices monitored, enter the maximum number of devices that are enabled to MONITOR SAMPLE. The number of SCSI devices must be excluded from this count.

3. **Paging and SPOOLing areas (1-255)**. Enter the maximum number of paging and spooling areas that are expected to be online to the monitored z/VM system. An area is defined as a single contiguous allocation of cylinders (CKD devices) or blocks (FBA devices) defined to CP as paging or spooling space. If a device has multiple paging or spooling areas, each separate area must be counted.

4. **Logged-on users (1-999999)**. Enter the maximum number of users logged on concurrently to the monitored z/VM system.

5. **HiperSockets devices (1-65535)**. Enter the maximum number of HiperSockets channel paths online to the monitored z/VM system. If you use the MONITOR SAMPLE commands to limit the number of I/O devices monitored, enter the maximum number of HiperSockets devices that are enabled to MONITOR SAMPLE.

6. **Network virtual switch devices (1-65535)**. Enter the maximum number of virtual switch devices defined to the monitored z/VM system. If you use the MONITOR SAMPLE commands to limit the number of I/O devices monitored, enter the maximum number of virtual switch devices that are enabled to MONITOR SAMPLE.

7. **Network server virtual machines (1-255)**. Enter the maximum number of network server virtual machines logged on to the monitored z/VM system. Each network server virtual machine must be configured to handle network traffic (for example, TCP/IP), and must have application monitor data collection enabled.

8. **Network users (1-999999)**. Enter the maximum number of users of the network server virtual machines specified in Step 7, that are logged on to the monitored z/VM system. The count of logged on network users includes the number of users logged on to z/VM using CMS. The count also includes the number of users logged on to network applications, such as FTP.

9. **Linux virtual machines (1-999999)**. Enter the maximum number of virtual machines running as Linux on zSeries guest systems on the monitored z/VM system.

   **Note:** The collection of Linux data requires that you enable the MONITOR SAMPLE APPLDATA domain. Also, the Linux Monitor Stream Support must be installed and activated within Linux.

10. **Engines defined to LPARs (1-65535)**. Enter the total number of processors of all types (CP, IFL, zIIP, and ICF) defined to the LPAR partitions in the processor complex. Count all processors, either shared or dedicated, regardless of the operating system running in the owning LPAR partition. Include the processors used by the monitored z/VM LPAR partition in the count.

11. **Channels defined to z/VM (1-1024)**. Enter the total number of channels defined to the z/VM system to be monitored. Do *not* include FICON channels in the count as they will be counted separately.

12. **LPAR Channels defined to z/VM (1-1024)**. Enter the total number of channels defined to the LPAR partition hosting the z/VM system that will be monitored. Do *not* include FICON channels in the count.

13. **FICON Channels defined to z/VM (1-1024)**. Enter the total number of FICON channels defined to the LPAR partition hosting the z/VM system that will be monitored.

14. **Disk cache devices defined to z/VM (1-65535)**. Enter the number of disk devices with cache capability that are typically attached to the monitored z/VM system. If you use the CP MONITOR SAMPLE command to limit the number of I/O devices monitored, then enter the maximum number of disk cache devices that are typically enabled to MONITOR SAMPLE.

15. **Cached control units defined to z/VM (1-65535)**. Enter the number of control units with cache capability that are typically attached to the monitored z/VM system. If you use the CP MONITOR SAMPLE command to limit the number of I/O devices monitored, then enter the maximum number of cached control units that are typically enabled to MONITOR SAMPLE.

16. **VDISKs defined to z/VM (1-999999)**. Enter the number of virtual disks (VDISKs) in storage that are typically attached to users logged on to the monitored z/VM system.

To skip a count because you do not monitor a type of item, specify 1 at the prompt for that item.

After entering values for each of the above listed items, the FCXSEGSZ EXEC file runs. The following lines are sample output:

```
     Minimum SEGOUT segment allocation needed (bytes, in decimal): 208704
     Minimum SEGOUT segment allocation needed (pages, in decimal): 52
     Minimum SEGOUT segment allocation needed (bytes, in hexadecimal): 32F40
     Minimum SEGOUT segment allocation needed (pages, in hexadecimal): 34
```

The sample output displays the minimum size of the DCSS. From here you can proceed to the next step of determining the start and end address of the DCSS.

**Note:**

1. FCXSEGSZ EXEC calculates enough segment space to hold two data samples from Performance Toolkit. This allows the monitoring agent sufficient time to process the most recent sample before the space is reused. During Performance Toolkit initialization, the segment is divided roughly in half, with each half holding one entire sample.

2. The areas used to hold sample data are allocated only when a sample is collected. There are no fixed allocations by record type. For example, during one sample, the USER data may need 14 KB of the segment, and DEVICE data will need 24 KB. In the next sample, changes to device configuration and additional users may result in 20 KB for USER data and 18 KB for DEVICE data. In either case, 38 KB of the segment will be consumed for USER data and for DEVICE data combined.

3. This design provides the most effective use of the allocated segment. It is still possible to allocate a segment that is not large enough to contain all the data available and enabled for collection. If Performance Toolkit detects that a segment is not large enough, it issues the following message at the Performance Toolkit console:

```
     FCXXSM774E Insufficient space in SEGOUT segment. Larger Segment needed.
```

   This error causes SEGOUT data collection to stop. You must then manually restart SEGOUT data collection after you expand the size of the DCSS.

4. When expanding the size of the DCSS, verify that the Linux definitions align with the segment definitions.

# Defining and Building the DCSS

In this topic, you will use VMSES/E to define and build the DCSS. Specifically, the VMFSGMAP segment mapping tool will be used to define the segment, and the VMFBLD command will be used to build it. If necessary, review the "Using VMSES/E to Define, Build, and Manage Saved Segments" chapter in *z/VM: Saved Segments Planning and Administration*.

Note that the defining and building of the DCSS should be performed from the installation user ID. If you move any segment that is currently defined on your system, you must make sure it is rebuilt from the user ID that maintains them.

To define and build the DCSS, follow these steps:

1. Log on to the installation user ID.

2. Establish write access to the VMSES/E and software inventory disks:

```
link MAINTvrm 51d 51d mr
access 51d d
```

3. Add the segment object definition to the SEGBLIST EXC00000 build list by invoking the vmfsgmap command:

```
vmfsgmap
```

   This command displays a panel for making segment updates, as shown in this example:

```
                    VMFSGMAP - Segment Map
Meg           400-MB          401-MB          402-MB          403-MB
St Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
   MONDCSS   DCS CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC>

Meg           404-MB          405-MB          406-MB          407-MB
St Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
   MONDCSS   DCS >CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC>

Meg           408-MB          409-MB          40A-MB          40B-MB
St Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
   MONDCSS   DCS >CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC>

     ⋮

Meg           458-MB          459-MB          45A-MB          45B-MB
St Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
   MONDCSS   DCS >CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC>

Meg           45C-MB          45D-MB          45E-MB          45F-MB
St Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
   MONDCSS   DCS >CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

Meg           460-MB          461-MB          462-MB          463-MB
St Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
   PERFOUT   DCS NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN>

Meg           464-MB          465-MB          466-MB          467-MB
St Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
   PERFOUT   DCS >NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN6..............7...............
 F1=Help      F2=Chk Obj   F3=Exit      F4=Chg Obj   F5=File      F6=Save
 F7=Bkwd      F8=Fwd       F9=Retrieve  F10=Add Obj  F11=Del Obj  F12=Class
====> _
```

*Figure 43. A Sample Segment Map Panel*

4. Position the cursor to the left of the PERFOUT segment name on the screen.

5. Go to the `Change Segment Definition` panel by pressing F4. That panel will look like this example:

```
                    Change Segment Definition
                                              Lines 1 to 12

   OBJNAME....:  PERFOUT
   DEFPARMS...:  46000-465FF SN
   SPACE......:
   TYPE.......:  SEG
   OBJDESC....:  PERFORMANCE TOOKLIT DATA COLLECTOR SEGMENT
   OBJINFO....:
   GT_16MB....:  YES
   DISKS......:
   SEGREQ.....:
   PRODID.....:  7VMPTK30 PERFTK
   BLDPARMS...:  PROD(SAVEONLY)




 F1=Help     F2=Get Obj   F3=Exit      F4=Add Line  F5=Map       F6=Chk
 F7=Bkwd     F8=Fwd       F9=Retrieve  F10=Seginfo  F11=Adj MEM  F12=Can
====>
```

*Figure 44. A Sample Change Segment Definition Panel*

6. On the `Change Segment Definition` panel, change the DEFPARMS field to reflect your new size as determined by running the FCXSEGSZ EXEC. You can change the addresses of the segment by typing over the information in the DEFPARMS field.

**Note:** Make sure the PERFOUT segment and the MONDCSS segment do not overlap or conflict in any way. Both segments are used by Performance Toolkit and both must be able to be loaded simultaneously.

7. Save your changes and press F5 to go back to the `VMFSGMAP - Segment Map` panel.

8. Save the new information and exit from the `VMFSGMAP - Segment Map` panel.

9. Prepare to build the segment.

    a. IPL CMS to clear the virtual storage:

    ```
    ipl cms parm clear nosprof instseg no
    ```

    **Important:** Do *not* press Enter while in VM READ mode.

    Note that this command bypasses the execution of the system profile (SYSPROF EXEC) and without loading the installation saved segment (CMSINST).

    b. Bypass the execution of the PROFILE EXEC:

    ```
    access (noprof
    ```

    c. Access the VMSES/E code:

    ```
    access 5e5 b
    ```

    d. Establish write access the software inventory disk:

    ```
    link MAINTvrm 51d 51d mr
       access 51d d
    ```

10. Issue the VMFBLD command to build the segment:

    ```
    vmfbld ppf segbld esasegs segblist perfout (serviced
    ```

    **Note:** If you receive this message:

    ```
    VMFBDS2003W The SYSTEM SEGID D(51D) file has been changed and must
                be moved to the S disk.
    ```
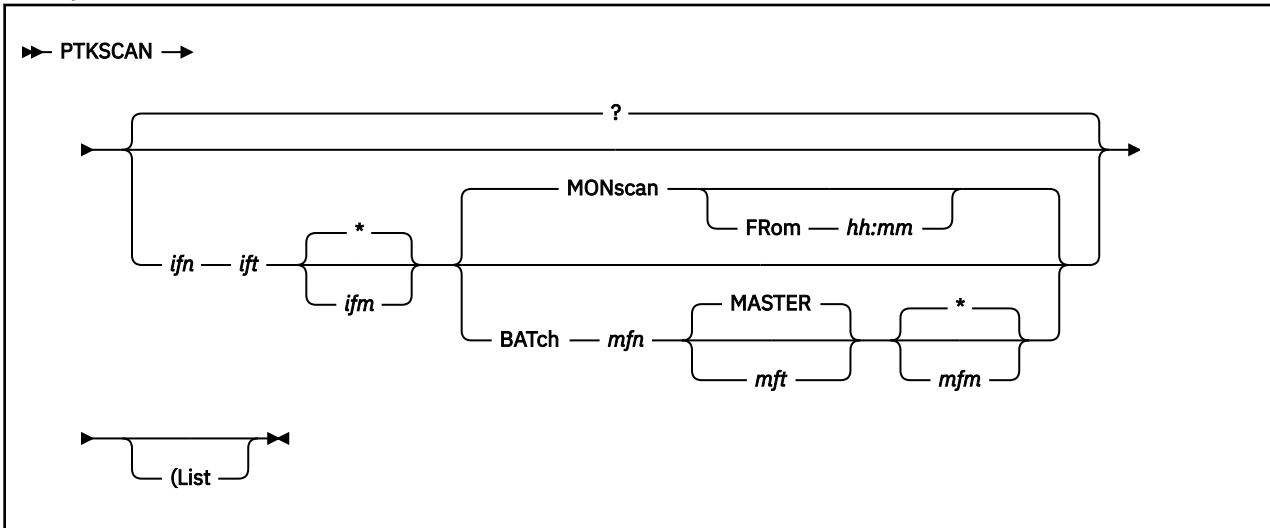
    the SYSTEM SEGID file on the CMS system disk (MAINT 190) and CMS test system disk (MAINT*vrm* 490) must be updated. You need to log on to your MAINT user ID and copy the SYSTEM SEGID file from the MAINT*vrm* 51D disk to the MAINT 190 and MAINT*vrm* 490 disks. (The SYSTEM SEGID must have a filemode of **2**.)

# Appendix C. Using the PTKSCAN EXEC

Use the PTKSCAN EXEC to process one or more monitor data files to produce a single set of reports. The monitor files can be packed using the COPYFILE command. The input file for PTKSCAN is a single monitor data file or a file that lists the individual monitor data files in the correct order.

The syntax is as follows:



where:

**?**
Displays the procedure's help information. This is the default if no other parameters are specified.

*ifn*
Is the file name of the input monitor data file (default), or the monitor data files list (if the **(List** option is specified).

*ift*
Is the file type of the input monitor data file or monitor data files list.

*ifm*
Is the file mode of the input monitor data file or monitor data files list. The default file mode is '*', meaning that all accessed disks will be searched for the specified file if the file mode is omitted.

**FROM** *hh:mm*
Allows specifying a start time which is to be searched for, where the time must be specified in exactly that format, i.e. 2 digits each for hours and minutes with a separating colon.

*mfn*
Is the file name of the Performance Toolkit batch master file.

*mft*
Is the file type of the Performance Toolkit batch master file. The default is MASTER.
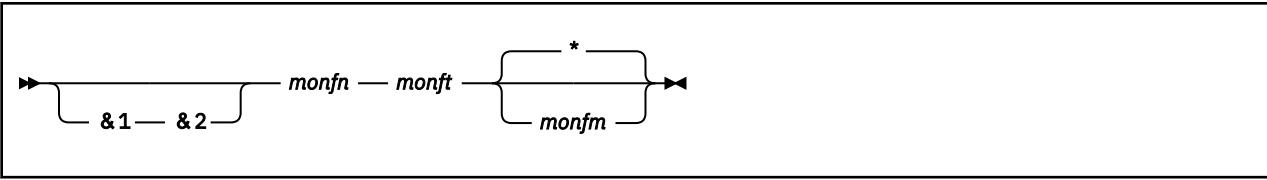
*mfm*
Is the file mode of the Performance Toolkit batch master file. The default file mode is *, meaning that all accessed disks will be searched for the specified file if the file mode is omitted.

**(List**
Specifies that the input monitor data file *ifn ift ifm* is a list of multiple monitor data files.

If the **(List** option is specified, the monitor data files list must follow this format:

```
                                                            *
     ►►──┬─────────────┬──── monfn ── monft ──┬────────────┬──►◄
         └─ &1 ── &2 ──┘                        └── monfm ──┘
```

where:

**monfn**
> Is the file name of the individual input monitor data file.

**monft**
> Is the file type of the individual input monitor data file.

**monfm**
> Is the file mode of the individual input monitor data file. The default file mode is '*', meaning that all accessed disks will be searched for the specified file if the file mode is omitted.

**Note:** An asterisk (*) in the first column of any line in this monitor data files list denotes a comment (and the entire line is ignored).

# Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*
*IBM Corporation*
*North Castle Drive, MD-NC119*
*Armonk, NY 10504-1785*
*US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing*
*Legal and Intellectual Property Law*
*IBM Japan Ltd.*
*19-21, Nihonbashi-Hakozakicho, Chuo-ku*
*Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing*
*IBM Corporation*
*North Castle Drive, MD-NC119*
*Armonk, NY 10504-1785*
*US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

# IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see:

- The section entitled **IBM Websites** at IBM Privacy Statement (https://www.ibm.com/privacy)
- Cookies and Similar Technologies (https://www.ibm.com/privacy#Cookies_and_Similar_Technologies)

# Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., in the United States and/or other countries. Other product and service names might be

trademarks of IBM or other companies. A current list of IBM trademarks is available on IBM Copyright and trademark information (https://www.ibm.com/legal/copytrade).

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

# Bibliography

This topic lists the publications in the z/VM library. For abstracts of the z/VM publications, see *z/VM: General Information*.

## Where to Get z/VM Information

The current z/VM product documentation is available in IBM Documentation - z/VM (https://www.ibm.com/docs/en/zvm).

## z/VM Base Library

### Overview

- *z/VM: License Information*, GI13-4377
- *z/VM: General Information*, GC24-6286

### Installation, Migration, and Service

- *z/VM: Installation Guide*, GC24-6292
- *z/VM: Migration Guide*, GC24-6294
- *z/VM: Service Guide*, GC24-6325
- *z/VM: VMSES/E Introduction and Reference*, GC24-6336

### Planning and Administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6261
- *z/VM: CMS Planning and Administration*, SC24-6264
- *z/VM: Connectivity*, SC24-6267
- *z/VM: CP Planning and Administration*, SC24-6271
- *z/VM: Getting Started with Linux on IBM Z*, SC24-6287
- *z/VM: Group Control System*, SC24-6289
- *z/VM: I/O Configuration*, SC24-6291
- *z/VM: Running Guest Operating Systems*, SC24-6321
- *z/VM: Saved Segments Planning and Administration*, SC24-6322
- *z/VM: Secure Configuration Guide*, SC24-6323

### Customization and Tuning

- *z/VM: CP Exit Customization*, SC24-6269
- *z/VM: Performance*, SC24-6301

### Operation and Use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6260
- *z/VM: CMS Primer*, SC24-6265
- *z/VM: CMS User's Guide*, SC24-6266
- *z/VM: CP Commands and Utilities Reference*, SC24-6268

- *z/VM: System Operation*, SC24-6326
- *z/VM: Virtual Machine Operation*, SC24-6334
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6337
- *z/VM: XEDIT User's Guide*, SC24-6338

### Application Programming

- *z/VM: CMS Application Development Guide*, SC24-6256
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6257
- *z/VM: CMS Application Multitasking*, SC24-6258
- *z/VM: CMS Callable Services Reference*, SC24-6259
- *z/VM: CMS Macros and Functions Reference*, SC24-6262
- *z/VM: CMS Pipelines User's Guide and Reference*, SC24-6252
- *z/VM: CP Programming Services*, SC24-6272
- *z/VM: CPI Communications User's Guide*, SC24-6273
- *z/VM: ESA/XC Principles of Operation*, SC24-6285
- *z/VM: Language Environment User's Guide*, SC24-6293
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-6295
- *z/VM: OpenExtensions Callable Services Reference*, SC24-6296
- *z/VM: OpenExtensions Commands Reference*, SC24-6297
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-6298
- *z/VM: OpenExtensions User's Guide*, SC24-6299
- *z/VM: Program Management Binder for CMS*, SC24-6304
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-6313
- *z/VM: REXX/VM Reference*, SC24-6314
- *z/VM: REXX/VM User's Guide*, SC24-6315
- *z/VM: Systems Management Application Programming*, SC24-6327
- *z/VM: z/Architecture Extended Configuration (z/XC) Principles of Operation*, SC27-4940

### Diagnosis

- *z/VM: CMS and REXX/VM Messages and Codes*, GC24-6255
- *z/VM: CP Messages and Codes*, GC24-6270
- *z/VM: Diagnosis Guide*, GC24-6280
- *z/VM: Dump Viewing Facility*, GC24-6284
- *z/VM: Other Components Messages and Codes*, GC24-6300
- *z/VM: VM Dump Tool*, GC24-6335

## z/VM Facilities and Features

### Data Facility Storage Management Subsystem for z/VM

- *z/VM: DFSMS/VM Customization*, SC24-6274
- *z/VM: DFSMS/VM Diagnosis Guide*, GC24-6275
- *z/VM: DFSMS/VM Messages and Codes*, GC24-6276
- *z/VM: DFSMS/VM Planning Guide*, SC24-6277

- *z/VM: DFSMS/VM Removable Media Services*, SC24-6278
- *z/VM: DFSMS/VM Storage Administration*, SC24-6279

## Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6281
- *z/VM: Directory Maintenance Facility Messages*, GC24-6282
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283

## Open Systems Adapter

- Open Systems Adapter/Support Facility on the Hardware Management Console (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC14-7580-02.pdf), SC14-7580
- Open Systems Adapter-Express ICC 3215 Support (https://www.ibm.com/docs/en/zos/2.3.0?topic=osa-icc-3215-support), SA23-2247
- Open Systems Adapter Integrated Console Controller User's Guide (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC27-9003-02.pdf), SC27-9003
- Open Systems Adapter-Express Customer's Guide and Reference (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/ioa2z1f0.pdf), SA22-7935

## Performance Toolkit for z/VM

- *z/VM: Performance Toolkit Guide*, SC24-6302
- *z/VM: Performance Toolkit Reference*, SC24-6303

The following publications contain sections that provide information about z/VM Performance Data Pump, which is licensed with Performance Toolkit for z/VM.

- *z/VM: Performance*, SC24-6301. See z/VM Performance Data Pump.
- *z/VM: Other Components Messages and Codes*, GC24-6300. See Data Pump Messages.

## RACF® Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6305
- *z/VM: RACF Security Server Command Language Reference*, SC24-6306
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6307
- *z/VM: RACF Security Server General User's Guide*, SC24-6308
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6309
- *z/VM: RACF Security Server Messages and Codes*, GC24-6310
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6311
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6312
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6324

## Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6316
- *z/VM: RSCS Networking Exit Customization*, SC24-6317
- *z/VM: RSCS Networking Messages and Codes*, GC24-6318
- *z/VM: RSCS Networking Operation and Use*, SC24-6319
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6320

### TCP/IP for z/VM

- *z/VM: TCP/IP Diagnosis Guide*, GC24-6328
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6329
- *z/VM: TCP/IP Messages and Codes*, GC24-6330
- *z/VM: TCP/IP Planning and Customization*, SC24-6331
- *z/VM: TCP/IP Programmer's Reference*, SC24-6332
- *z/VM: TCP/IP User's Guide*, SC24-6333

# Prerequisite Products

### Device Support Facilities

- Device Support Facilities (ICKDSF): User's Guide and Reference (https://www.ibm.com/docs/en/SSLTBW_3.1.0/pdf/ickug00_v3r1.pdf), GC35-0033

# Related Products

### XL C®++ for z/VM

- *XL C/C++ for z/VM: Runtime Library Reference*, SC09-7624
- *XL C/C++ for z/VM: User's Guide,* SC09-7625

### z/OS®

IBM Documentation - z/OS (https://www.ibm.com/docs/en/zos)

# Index

FCONX $PROFILE *(continued)*
    commands for threshold monitoring 126
    customizing FCON/ESA 7
    operation during initialization 92
FCONX LINUXUSR file
    defining Linux systems 137
FCONX REPORTS file 116
FCONX TRENDREC file 113, 114
files
    FCONX REPORTS 116
function pending time 163

## G

GDDM 4, 6
general
    I/O device screen
    161
general considerations for performance analysis 149
generations
    console logs 94
    exception log files (EXCPLOG) 10
    perf. monitor data on disk 110
    remote perf. log files (RMONLOG) 10
graphical
    displays in remote perf. monitoring mode 23
    history data display 151
    variable correlation plots 151, 165, 166
graphical history data display 151
guided tour 63

## H

handling
    CP messages and warnings 3, 78, 88
    Performance Toolkit errors 96
HELP facility
    access via general HELP menu 83
    context sensitive selection 72, 85
    direct access to specific help text 69, 84
highlighting and color 87
history data display 131, 151
history data display, extended 131, 151
history files
    ext. trend file creation 111
    performance graphics based on 103
    writing perf. history files on disk 110

## I

I/O bottlenecks, analysis of 160
I/O contention
    channel path level 164
    control unit level 164
    device level 157, 161, 162
I/O device screen 161
I/O rate
    minidisks 162, 170
    real devices 157, 161, 162, 170
    system areas 156
ignoring received lines 89
imbedding control files 94
immediate CMS commands 77

Implementing the Linux data retrieval interface 136
initialization profile, sample 173
installation
    customizing Performance Toolkit 7
    implementing remote perf. monitoring 10
    performance considerations 7
interim data
    printing 120
Internet interface 45
introduction
    general system operating 75
    performance monitoring 99
    performance problem analysis 149

## L

Linux data retrieval interface implementation 136
logical escape character 77
logical line end character 77

## M

message area 76, 79, 82
messages
    error messages 79
    handling of CP messages 78, 88
    Performance Toolkit 'action' messages 79, 88
minidisk
    I/O rate 162, 168, 170
    load analysis 162, 168, 170
    status 162, 168, 170
MLOAD, smoothed CP I/O response time 157
Monitoring Linux internal performance data 136

## O

objects
    benchmarking 134
operation
    in basic mode 76, 81
    in performance monitor mode 99, 130
    in redisplay mode 81
    remote performance monitoring operation and data flow
    25
output
    to disk 121
    to printer 115
overloaded CPU, analysis of 164

## P

page rate
    on system area 157
    system page rate 152, 155
paging bottlenecks, analysis of 154
performance data
    sorting 108
performance data, printing
    FCONX REPORTS file 116
performance monitor displays
    channel load screen 164
    CP owned device screen 156
    extended history data log screen 151

## U

user monitoring
    description 127
    loop detection (exception monitoring) 124, 128
    setup for 128
    user thresholds 124, 127
user status screen 153
user wait state log screen 153
user-generated exception information 18

## V

variable correlation plot 151, 165, 166
variable correlation screen 151
virtual machine, analysis of 167
virtual machine, detailed analysis 167
VM/ESA 6
VMC compatibility mode 28
VMCF interface
    enabling 28
    using for performance data retrieval 27
VMCX interface 30

## W

wait states
    user 153
Web server 45
weight of exceptions
    use of 'weight' for calculating exception server code 16
working set 160
writing performance data to disk 121

**IBM**®

Product Number:   5741-A09

Printed in USA